MAIA

ERASMUS MUNDUS

JOINT MASTER
IN MEDICAL IMAGING
AND APPLICATIONS

Joint Master in Medical Imaging and Applications

Master Thesis Proceedings

Promotion 2018-20

www.maiamaster.org

MAIA

An international programme by the University of Girona (Spain), the University of Bourgogne (France) and the University of Cassino (Italy) funded by Erasmus + Programme.

Erasmus+ Universitat de Girona uB

Published by the MAIA Master

www.maiamaster.org

This document is a compendium of the master thesis works developed by the students of the Joint Master Degree in Medical Imaging and Applications. Therefore, each work is independent on the other, and you should cite it individually as the final master degree report of the first author of each paper (Student name; title of the report; MAIA MSc Thesis; 2020).

Editorial

Computer aided applications for early detection and diagnosis, histopathological image analysis, treatment planning and monitoring, as well as robotised and guided surgery will positively impact health care during the new few years. The scientific community needs of prepared entrepreneurships with a proper ground to tackle these topics. The Joint Master Degree in Medical Imaging and Applications (MAIA) was born with the aim to fill this gap, offering highly skilled professionals with a depth knowledge on computer science, artificial intelligence, computer vision, medical robotics, and transversal topics.

The MAIA master is a two-years joint master degree (120 ECTS) between the Université de Bourgogne (uB, France), the Università degli studi di Cassino e del Lazio Meridionale (UNICLAM, Italy), and the Universitat de Girona (UdG, Spain), being the latter the coordinating institution. The program is supported by associate partners, that help in the sustainability of the program, not necessarily in economical terms, but in contributing in the design of the master, offering master thesis or internships, and expanding the visibility of the master. Moreover, the program is recognised by the European Commission for its academic excellence and is included in the list of Erasmus Mundus Joint Master Degrees under the Erasmus+ programme.

This document shows the outcome of the master tesis research developed by the MAIA students during the last semester, where they put their learnt knowledge in practice for solving different problems related with medical imaging. This include fully automatic anatomical structures segmentation, abnormality detection algorithms in different imaging modalities, biomechanical modelling, development of applications to be clinically usable, or practical components for integration into clinical workflows. We sincerely think that this document aims at further enhancing the dissemination of information about the quality of the master and may be of interest to the scientific community and foster networking opportunities amongst MAIA partners.

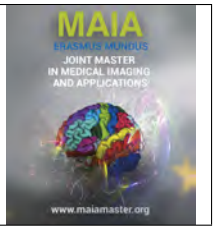
We finally want to thank and congratulate all the students for their effort done during this last semester of the Joint Master Degree in Medical Imaging and Applications.

MAIA Master Academic and Administrative Board

Contents

Medical Imaging and Applications

Master Thesis, August 2020



Hemorrhagic stroke lesion segmentation using a 3D U-Net with squeeze-and-excitation blocks

Valeriia Abramova, Albert Clèrigues, Arnau Oliver, Xavier Lladó

Computer Vision and Robotics Group, University of Girona, Catalonia, Spain

Abstract

Hemorrhagic stroke is the condition involving the rupture of the vessel inside the brain. It is characterized with high mortality rates, so it is important to act fast to prevent irreversible consequences. In this master thesis, a deep learning-based approach to segment hemorrhagic stroke lesions in CT scans is proposed. Our proposal is based on a 3D U-Net architecture which incorporates the recently proposed squeeze-and-excitation blocks. Restrictive patch sampling is proposed to alleviate the class imbalance problem and also the issue of intraventricular hemorrhage, which is a specific subtype of stroke located inside brain ventricles that has not been included as a lesion in our study. Moreover, we also studied the effect of patch size, the use of different modalities, data augmentation and the incorporation of different loss functions on the segmentation results. All analysis have been performed using a five fold crossvalidation strategy on a private dataset composed of 76 cases, which was provided by the collaborating Hospital Dr. Josep Trueta, Girona, Spain. Obtained results demonstrate that the introduction of squeeze-and-excitation blocks, together with the restrictive patch sampling and symmetric modality augmentation provide the highest mean DSC of 0.862±0.074, showing promising automated segmentation results.

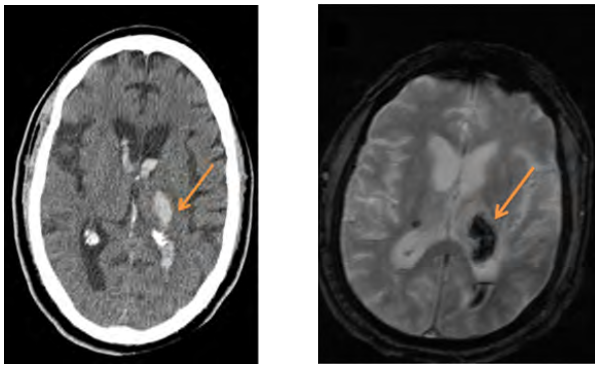
Keywords: Hemorrhagic stroke, Squeeze-and-Excitation, Intraventricular hemorrhage, U-Net, Balanced sampling

1. Introduction

Nowadays stroke is one of the most common causes of death, holding the third position after an ischemic heart disease and neonatal disorders (Roth et al. (2018)). It is a medical condition in which the brain tissues lose the ability to get oxygen due to reduced or fully cut blood flow. This rapidly leads to the death of brain cells. There are two types of stroke: ischemic and hemorrhagic. Ischemic stroke is the most common type of stroke (around 87% of all strokes (Mozaffarian et al. (2016))) and it is caused by reduction of blood supply to the brain tissues; the rest of strokes are hemorrhagic ones and they involve the rupture of a vessel inside the brain. In this case, brain cells get damaged because of the pressure of the leaked blood. Even though hemorrhagic stroke is a less common condition, it is characterized with high mortality rates (Kidwell and Wintermark (2008)).

The stroke lesion consists of two parts: the core, which is basically the irreversibly injured tissue or the hematoma in the case of hemorrhagic stroke, and the edema around the core, which is caused by the brain tissues swelling. Time is the key factor of successful treatment of stroke, since early stroke diagnosis and treatment are related to positive patient outcome (Matsuo et al. (2017)). Therefore, fast clinical actions are required in order to give the patient the most appropriate treatment.

Stroke can be diagnosed through different techniques, including imaging, which is a key assistance to define the type of stroke. The most common imaging modalities for stroke diagnosis are Magnetic Resonance Imaging (MRI) and Computed Tomography (CT). It was shown that Gradient Echo (GRE) MRI sequences are as accurate as CT in detecting hematoma (Kidwell and Wintermark (2008)). Moreover, they can be better than CT in detecting chronic hemorrhages and sometimes can detect lesions which were missed in CT. However, CT is the dominant modality for diagnosing hemorrhagic stroke, as it is clearly seen there. In addition,



(a) Hemorrhagic stroke in non-contrast CT

(b) Hemorrhagic stroke in Gradient Echo (GRE) MRI

(c) Ischemic stroke in non-contrast CT

(d) Ischemic stroke in Diffusion-Weighted Imaging (DWI) MRI

Figure 1: Appearance of hemorrhagic and ischemic stroke (orange arrows) on CT and MRI sequences (Muir and Santosh (2005), Siddiqui et al. (2011)).

CT imaging is widely available, it is rapid, which is a key factor in stroke, inexpensive, and suitable for all the patients, unlike MRI. An example of hemorrhagic and ischemic stroke imaging in CT and MRI is shown in Figure 1.

Image segmentation is an important part in diagnosis and management of the disease. In clinical practice the standard approach nowadays is manual delineation of the stroke lesion. However, this approach has disadvantages; it is both time-consuming and operator-dependent, which leads to subjective and not reproducible results. To address these issues, automated segmentation algorithms have been suggested during the past years (e.g. Forbes et al. (2010), Shahangian and Pourghassem (2015)). Research on automated approaches can also help to better understand pathologies of the brain and construct statistical patterns to generalize the results across the population. Moreover, they can offer objective, accurate and reproducible methods to quantitatively assess stroke lesions and help doctors to evaluate all the risks.

The initial attempts to segment hemorrhagic stroke lesions mostly relied on processing of CT images. They involved approaches based on different image analysis techniques, their variations and hybrid strategies (Pérez et al. (2008)). For example, such methods involved clustering (Lončarić et al. (1995), Cosić and Lončarić (1997)), morphological operations (Lončarić et al. (1995), Perez et al. (2007)), region growing or level sets (Bardera et al. (2009)).

The recent breakthrough and popularity of deep learning techniques increased the research interest and the number of proposed algorithms to segment brain lesions and, specifically, stroke lesions. In particular, Convolutional Neural Networks (CNNs) (Lecun et al. (1998)) have shown big potential in different biomedical imaging tasks including medical image segmentation. One of the CNNs, which has shown great performance in biomedical segmentation tasks is the U-Net architecture (Ronneberger et al. (2015)). This architecture inspired and served as a base for a big number of different segmentation approaches (Piantadosi et al. (2018), Li et al. (2019), Alom et al. (2019)), including the ones in the field of neuroimaging (Guerrero et al. (2017), Dong et al. (2017)).

In this work, a deep learning approach for segmentation of hemorrhagic stroke lesions in non-contrast CT is proposed. Similarly to Woo et al. (2019), we introduce squeeze-and-excitation blocks to a U-Net architecture, but in our case we do it in a 3D U-Net implementation. A balanced sampling technique is introduced for patch extraction, restricting this patch sampling spatially and quantitatively to address the problem of intraventricular hemorrhage. In addition to standard data augmentation techniques, symmetric modality augmentation is also performed to benefit from the brain hemispheres symmetry property to find more robust image features, as done in Clèrigues et al. (2019). The segmentation algorithm proposed is evaluated with a crossvalidation strategy over a dataset of 76 cases, analyzing the impact of the different contributions introduced in our approach.

2. State of the art

Segmentation of stroke lesions is important in the field of diagnosis and treatment. It is the first step to do the research related to lesion outcome prediction, and to establish correlations to use later in the clinical practice. Hematoma segmentation helps to evaluate if the lesion will grow, which is important clinically.

Intracranial hemorrhage (ICH) is not such a common type of stroke as an ischemic one, therefore its automated segmentation has not been the main research focus in the community. Traditional semi-automated approaches had been developed, but, during the last two years, automated approaches for hematoma segmentation have started to appear.

2.1. Segmentation of ischemic stroke

The early works in segmentation of ischemic stroke core were focused on CT images and consisted of applying traditional approaches. For instance, the work of

Matesin et al. (2001) was composed of techniques, such as region growing and also different image features, e.g. related to symmetry, brightness or area. In the works of Usinskas et al. (2004) and Tang et al. (2011) textural features were utilized to deal with the segmentation problem.

Regarding the ischemic stroke segmentation in MRI, different approaches have been proposed. For example, the work of Wanida Charoensuk and Likitjaroen (2015) was based on applying an active contour algorithm to Diffusion Weighted Imaging (DWI) MRI modality. The approaches of Hevia et al. (2007) and Li et al. (2009) suggested to apply mean shift algorithm for the purpose of stroke segmentation.

The launch of the Ischemic Stroke Lesion Segmentation (ISLES) challenge (http://www.isles-challenge.org/) in 2015 increased the interest to the topic and initiated a burst of automated ischemic stroke segmentation methods. The challenge was held until 2018. The first three years it was focused on lesion segmentation in different MRI modalities, while in the 2018 the dataset provided to the participants for development of their algorithms consisted of CT perfusion scans. A lot of participants of the challenge took advantage of deep learning techniques, because of their recent advance. For instance, in 2015 the winning method was an 11-layer 3D CNN from Kamnitsas et al. (2015) and was later extended to create the well-known *DeepMedic* architecture (Kamnitsas et al. (2016)). In the work of Choi et al. (2016), 3D multi-scale residual U-Net was utilized for stroke core segmentation. In 2017, Lucas et al. (2018) proposed a fully-convolutional network based on 2D U-Net with additional short skip connections. In 2018, four out of five presented algorithms were based on a U-Net architecture. Despite the fact that the challenge finished in 2018, new approaches based and evaluated on the ISLES dataset are currently being developed. As an example, the work of Clèrigues et al. (2019) proposed a 2D patch-based residual encoder–decoder architecture to segment stroke core lesions from CT perfusion scans.

2.2. Segmentation of hemorrhagic stroke on MRI images

In terms of hemorrhagic stroke segmentation, nowadays the number of works developed on MR sequences is increasing, because of its superior sensitivity for detecting brain pathologies compared to CT. For instance, in the work of Roy et al. (2015) gamma transformation together with an expectation-maximization algorithm was used to segment hemorrhages. The work of Pszczolkowski et al. (2019) proposed a method based on shape and intensity analysis of both T2* GRE and FLAIR sequences for hematoma segmentation. The properties of lesion intensities in these MRI sequences were used together with masks of brain tissues to detect hemorrhage voxels.

2.3. Segmentation of hemorrhagic stroke on CT images

In general, research on hemorrhagic stroke segmentation started from segmenting CT images. One of the earliest works by Lončarić et al. (1995) presented a semi-automated method based on k-means histogram-based clustering. Cosić and Lončarić (1997) proposed an approach consisting of unsupervised fuzzy clustering and expert system-based labeling. In the paper of Majcenić and Lončarić (1998) the implementation of simulated annealing, which is a function optimization method, was applied to perform the segmentation. The images used for these works were digitized CT films and the presented methods were computationally complex.

Later, Perez et al. (2007) presented a set of three methods for hematoma segmentation, where two methods were carried out in a semi-automatic way and the remaining one was performed in a manual way. They were using 3D mathematical morphology operations and live wire technique for object contour extraction. In the work of Bardera et al. (2009) the semi-automated method for hematoma segmentation was developed based on a region growing algorithm.

Automatic methods for hemorrhage segmentation had also been proposed recently. Sharma and Venugopalan (2012) proposed a method based on k-means clustering with automatic initialization of cluster centers. The approach of Bhadauria and Dewal (2014) combined fuzzy c-means clustering and region-based active contour method. It was evaluated on 2D CT scans of 20 patients. The work of Shahangian and Pourghassem (2013) suggested using thresholding for the segmentation step. This approach demonstrated good results with evaluation on 2D 128×128 CT scans. The method of Gillebert et al. (2014) was developed for segmenting images with both ischemic and hemorrhagic stroke and consisted of normalizing CT scans to a template space and applying subsequent voxelwise comparison with a group of control CT scans in order to define areas with hypo- or hyper-intense signals. The method was evaluated on scans with both simulated and real stroke lesions.

With the recent rise of deep learning techniques, they started to be used for the hemorrhagic stroke segmentation problem, even though there are yet not so many proposed approaches. For example, Wang et al. (2018) used a 3D U-Net in their work for brain hemorrhage segmentation. As the groundtruth in this study was presented in .csv files, the masks were obtained using this information and using morphological operations. The size of image patch used was $64 \times 64 \times 64$ and data augmentation was also applied. Chang et al. (2018) proposed a mask R-CNN algorithm, which used a custom hybrid 3D/2D variant of the feature pyramid network as a backbone to generate a shared set of image features for segmentation of different types of hemorrhagic stroke. Singh et al. (2019) presented a 3D CNN to segment several hematoma types and also they use a novel thresh-

olding method, which improved their results. Hssayeni et al. (2020) used 2D U-Net for segmenting intracranial hemorrhage. In their work, CT scans from 82 patients were used without any image preprocessing. Furthermore, there are two works that have recently been proposed. One of them by Kuang et al. (2020) used a novel network named ψ-Net, where two attention blocks were used to suppress the irrelevant information, and capture the spatial contextual information to refine the border areas of the stroke lesion. The dataset used in this paper consisted of 150 CT scans and the method was evaluated on 2D slices with different hematoma types. The other work proposed by Yao et al. (2020) proposed to use multi-view CNN with a mixed loss function. The architecture share some similarities with U-Net, and the mixed loss function was proposed to make the system robust to CT scans acquired in different medical centers using different protocols.

3. Materials and methods

3.1. Dataset

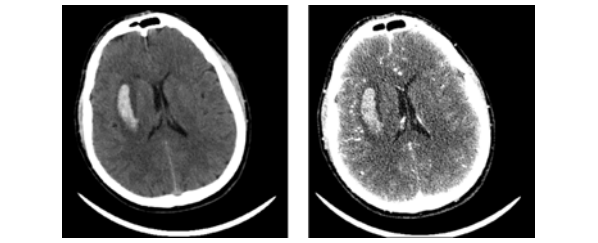
The dataset used in this project was acquired in the Hospital Dr. Josep Trueta, Girona, Spain. It consists of 76 cases, each of them with non-contrast head CT together with CT angiography images (provided not for all the cases) and CT perfusion images. Some examples of the Trueta dataset can be seen in Figure 2.

The image acquisition protocol was the following. All the examinations were performed on 128-slice CT scanner (Ingenuity; Philips Healthcare). Some image characteristics for each modality were different, as presented in Table 1: for non-contrast CT (CT NC) the slice thickness was 3 mm and the gap was of 1.5 mm, whereas for CT angiography (CT Angio) the slice thickness was 0.9 mm with the gap of 0.45 mm. The CT perfusion (CTP) images consisted of 4 slices of 10 mm (Puig et al. (2017)).

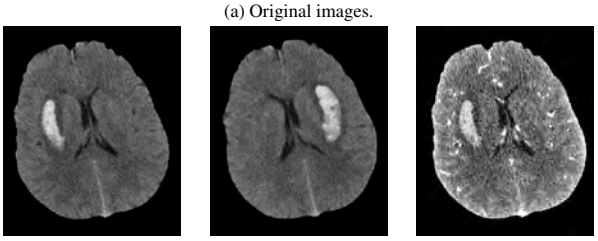
The scans were acquired for research, which goal was to investigate the relationship between perfusion characteristics of the lesion and its evolution. The gold standard, manual stroke lesion segmentations, were delineated by expert radiologists on non-contrast CTs.

Table 1: Image characteristics of all presented in dataset modalities: non-contrast CT (CT NC), CT angiography (CT Angio), CT Perfusion (CTP).

Image modality	Matrix	Slice Thickness, mm	Gap, mm
CT NC	512×512	3	1.5
CT Angio	512×512	10	10
CTP	512×512	0.9	0.45



(a) Original images.



(b) Non-contrast CT (c) Symmetric version of non-contrast CT (d) CT angiography

Figure 2: Top row: original image modalities. Bottom row: Preprocessed modalities used in the approach as input.

3.1.1. Intraventricular hemorrhage

Intraventricular hemorrhage (IVH) or intraventricular bleeding is an extension of hemorrhage, which occurs within brain parenchyma, inside brain ventricles, where the cerebrospinal fluid is produced. One of its sources can be the hemorrhagic stroke lesion adjacent to the ventricles. Such pathology is a bad prognosis sign, as expected mortality from it is between 50% and 80% (Hinson et al. (2010)). Usually, IVH can be clearly confirmed from CT imaging from the presence of the blood inside the ventricles. Therefore, the problem of segmenting intraparenchymal hemorrhage (IPH) from IVH arises (Figure 3).

Almost 15% of the cases in the Trueta dataset have IVH together with the stroke lesion. As the research objective of this dataset was stroke, which is located within brain tissues only, the intraventricular hemorrhage was not delineated as a stroke class and, therefore, it was not segmented on the provided groundtruth. However, as the source of signal in both pathologies is the same (blood leakage), the intensities of both regions on non-contrast CTs are also similar (Figure 3). Thus, the developed algorithm should also learn to differentiate between IPH and IVH, as we will see in the following sections.

3.1.2. Data preparation

The initial preparation of non-contrast CT head scans require removal of the coil and skull stripping (see Figure 2) as those regions can confuse the algorithm and lead to undesired results. To remove the coil, the original image was binarized and the biggest connected component, which is the head, was kept. The skull removal process is similar, it utilizes morphological operations to remove the borders of the skull and the final

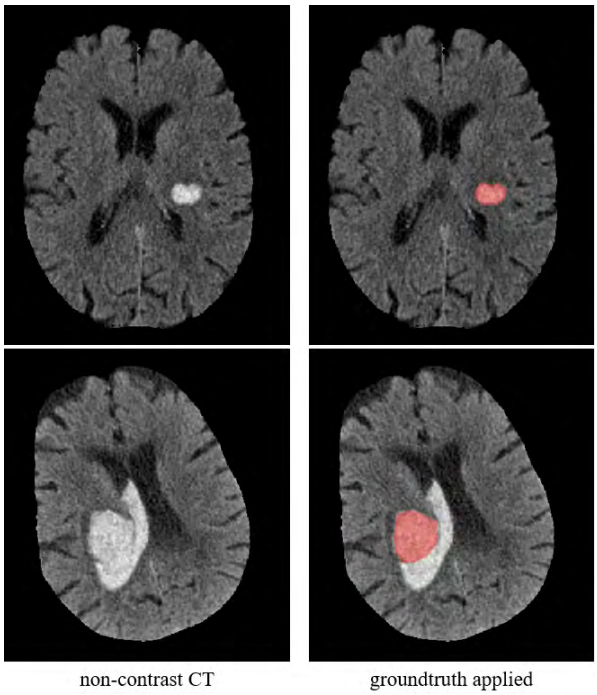


Figure 3: Dataset examples. The top row corresponds to IPH case. The bottom row is an example of intraventricular hemorrhage. Both IPH and IVH have similar intensities, even though the groundtruth provided only has IPH. Moreover, IVH deforms brain ventricles.

brain extraction is also based on extracting the biggest connected component.

As stroke can appear in one of the hemispheres of the brain, it could be useful to utilize features based on mid-sagittal symmetry of those hemispheres, as proposed in Clèrigues et al. (2019). Therefore, a symmetric image of the brain was created (Figure 2c). We firstly flipped the brain CT and secondly the flipped image was registered to the initial one. For the registration purposes the FLIRT tool from FSL was used (Jenkinson and Smith (2001), Jenkinson et al. (2002)). This tool provided fast and accurate registrations and is well-known in the research community.

Furthermore, CT angiography can be used additionally as another input channel for the segmentation. However, they have to be preprocessed before applying an automated segmentation algorithm. The provided dataset was acquired within everyday clinical practice so it was not prepared to be analysed automatically. Hence, additional preparation of the angiography scans was necessary. From the whole dataset, CT angiography scans of 18 patients were containing not only the head, but also the upper body, so these scans had to be cropped according to the corresponding non-contrast CTs. Moreover, as the voxel spacing of non-contrast and angiography CTs was different, the cropped angiography image had to be registered to the space of non-contrast CT (Figure 2d).

3.2. Proposed Method

The proposed approach is a 3D patch based deep learning method based on a U-Net architecture for segmentation of the hemorrhagic stroke lesion core from non-contrast CT scans. As the lesion mostly occupies only a small volume inside the brain, class imbalance is a problem that has to be taken into account. To train the network, this should be necessarily done in order to avoid overfitting to the negative class, which will influence the segmentation results. Moreover, the dataset used has another type of hemorrhage presented - the intraventricular one, which is not assigned to lesion class, and this also has to be considered in the algorithm developed. In this work the main contributions are presented to address both issues: balanced sampling technique to ensure equally distribution of both classes in the training set and restrictions in regions to extract patches to distinguish between intraparenchymal and intraventricular hemorrhages. At the training stage, to tackle these problems, regularization techniques were also applied, such as: (a) dropout, (b) data augmentation, and (c) early stopping. During testing, high overlap between extracted patches was used to improve the segmentation results.

3.2.1. Patch sampling

The proposed approach is a patch based CNN architecture, which prevents from computationally heavy load of large input images and also offers reduced training time (Long et al. (2015)).

For patch based methods, class imbalance can be an issue. So, it is important to control the process of patches extraction, otherwise only a minor number of patches will be taken from the lesion class. In such a case, such data imbalance can lead to poor performance of the network and misclassification of lesion voxels.

To address this problem, researchers proposed different methods. For instance, in the work of Guerrero et al. (2017) training patches were sampled so that they always contain lesion voxels. Moreover, they were randomly shifted so that the center of the patch does not necessarily include the lesion voxel. Another example can be the work of Clèrigues et al. (2019), which used a balanced sampling strategy so that the equal number of patches representing both classes were extracted from each image.

In our work, the balanced sampling technique was used together with other ways of controlling the patch extraction, like in the work of Kushibar et al. (2018) and Clèrigues et al. (2019). The same number of patches were extracted from background and lesion. At the initial step, the region for extracting negative patches was restricted. To avoid extracting a lot of patches from background and take more advantage of the dataset, the area to extract non-lesion patches was limited with the brain mask; this way the negative patches were uniformly extracted only from the region inside the brain.

To improve the segmentation results on the lesion borders, we selected the region around hematoma's borders in order to extract additional negative samples from the area just around the lesion. Thus, we make the network learn more from the boundaries of the hematoma, like done in work of Kushibar et al. (2018) for MRI segmentation of brain subcortical structures.

Data augmentation is another popular technique to artificially expand the dataset. In our approach, the reasonable transformations we applied in terms of brain anatomy were: horizontal and vertical flip, patch rotations at 90°, 180°, 270°. Therefore, in the training set, one patch was used six times, each time transformed with a different augmentation type.

The patches were extracted from non-contrast CT scans, which in some experiments, as we will see in the Section 4, were supplemented with their symmetric versions or angiography CT as additional input channels to the network.

3.2.2. Intraventricular hemorrhage problem

As mentioned before, some images of the dataset contain intraventricular hemorrhage, which is not delineated as the groundtruth. As this type of stroke has the same appearance as intraparenchymal one, we should find a way to segment IPH from IVH.

The initial steps to resolve this issue can be taken at the data preparation step, while sampling training patches. This can be done, as the voxels of IVH belong to the negative class, whereas the voxels of IPH belong to the lesion class. To make the network learn this dependence, more patches from the IVH area should be represented in the training set.

This hypothesis leads us to make additional restrictions to negative patch sampling. As this abnormality occurs inside the brain ventricles, it would be essential to use the spatial information related to brain anatomy and extract more patches from the area of CSF ventricles. Unfortunately, methods for brain tissue segmentation cannot be applied in this case as in the ventricles we can have both hypointense regions related to normal ventricles and hyperintense regions related to IVH. One way to overcome this obstacle could be defining a region of interest around the center of the brain volume, as shown in Figure 4. In our approach, the coordinates of brain center were calculated for each image separately. For each dimension we defined the first and last slice where the brain appears and took the coordinate of the middle slice. Such choice of ROI comes from the observation that the brain ventricles are located in the medial area of the brain.

Another way to solve the IVH issue we studied was to put more attention on the IVH voxels in the training stage. In our case, we considered intraventricular hemorrhage as hyperintense volume inside the brain, which does not belong to IPH. Therefore, we could extract more patches from this area. To define it, we tried thresholding the whole image using an empirically found threshold that represented intensity of the blood signal and then we excluded those pixels, which belong to the lesion, using the provided groundtruth. Moreover, some other hyperintense structures as brain borders and midline were excluded by utilizing the fact that IVH has bigger volume.

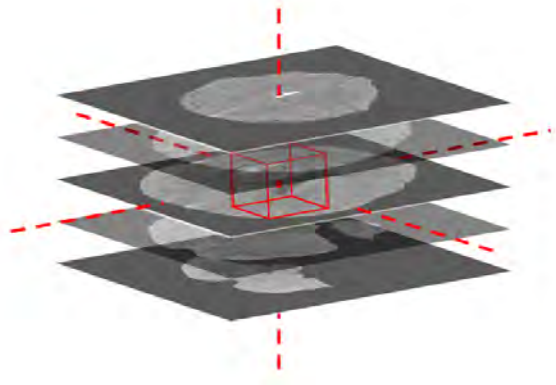


Figure 4: The region of interest (red cube) defined around the point of the brain center

When the desired ROI was defined, we could restrict the training patch sampling by assigning the fraction of negative patches which can be forcibly extracted from this region. As the fraction of the scans with IVH is not that huge, we can extract even more training patches from this region by extracting the background patches only from the defined ROI, but only if the image has the ROI.

In the overall patch extraction pipeline, firstly, the target number of patches was set for each patient. 50% of patches were extracted uniformly from the whole volume of the brain, yet some predefined fraction of them was forcibly extracted from the area around lesion boundary and the area related to the brain ventricles. Here, uniform sampling was done in order to make sure that all the parts of the brain are equally represented. In addition, 50% of patches were extracted from the lesion voxels. If the lesion is small and the number of its voxels is smaller than the desired number of patches, the voxels were repeated until that number is reached. But, if the number of lesion voxels is bigger, then they were regularly resampled, so that all the parts of the lesion were presented homogeneously. Then, those voxels serve as centers for patch extraction. Furthermore, one or several types of data augmentation can be applied to the extracted patches, as presented in Section 4.5, increasing the size of the patch dataset proportionally to the number of patches specified in the beginning.

3.3. Deep learning approach

U-Net architecture was firstly proposed by Ronneberger et al. (2015) for the task of cell segmentation. Rapidly, it earned the recognition in the community and

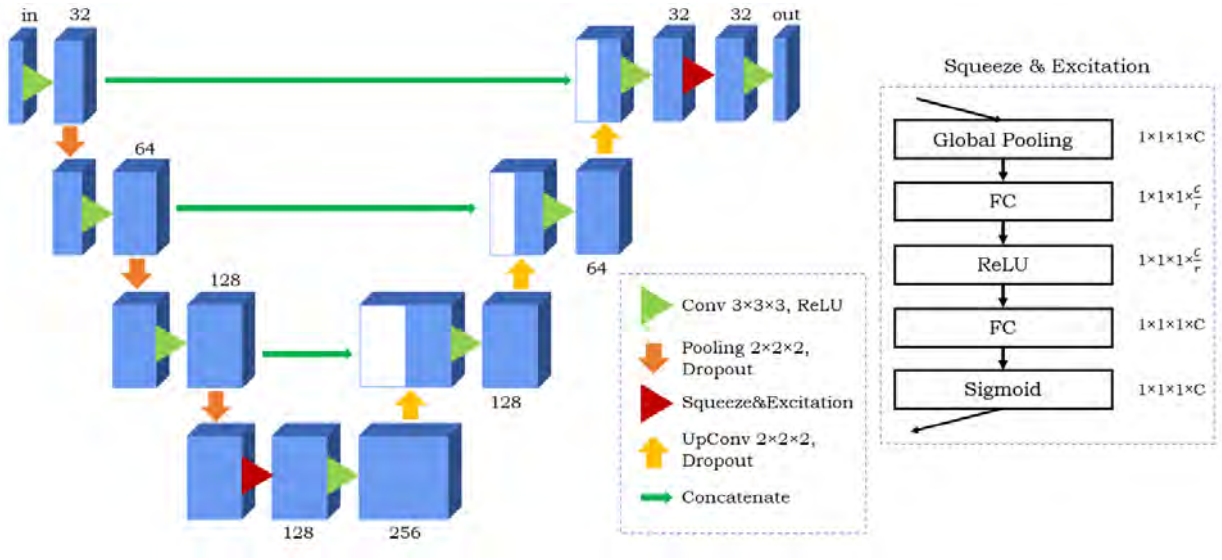


Figure 5: The architecture used in the proposed approach. The network is inspired by 3D U-Net with incorporation of squeeze-and-excitation blocks.

inspired a lot of image segmentation approaches, especially in the field of biomedical imaging, as mentioned in Section 1. Also, it was extended for 3D segmentation in the work of Çiçek et al. (2016) and in this way it was also used for different image segmentation tasks, particularly for stroke lesion segmentation, as we saw in Section 2. This master thesis work is also based on 3D U-Net, considering its prevalence and the fact that 3D patches may provide more information of surroundings for the voxel that is being classified. In addition, U-Net has proven itself to work well with small datasets. Similarly to what was done in the work of Woo et al. (2019) for ischemic stroke segmentation problem, we propose to incorporate squeeze-and-excitation blocks to the 3D U-Net architecture, as it showed improved performance for the segmentation task.

3.3.1. Squeeze-and-Excitation

The squeeze-and-excitation (SE) blocks were firstly introduced by Hu et al. (2018). They can be used as building blocks for existing CNNs at slight additional computational cost and their goal is to improve the quality of representations produced by a network by explicitly modelling the interdependencies between the channels. Structurally, this computational unit consist of: (a) a *squeeze* operator, which produces a channel descriptor by aggregating feature maps across their spatial dimensions so that each learned filter can exploit contextual information from the global receptive field of the network; and (b) an *excitation* operator, which aims to fully capture channel-wise dependencies. All this is done to boost informative features and suppress the weak ones.

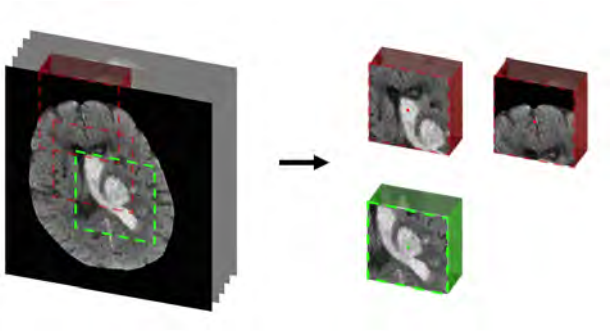
In the original paper, these blocks were evaluated on different image classification tasks and have been shown to improve the network's performance. In medical imaging tasks this technique has also been incorporated. Namely, the work of Rundo et al. (2019) incorporates squeeze-and-excitation blocks into U-Net to tackle the prostate zonal segmentation task in MR images. The approach showed high performance in comparison to other state-of-the-art methods. In the field of neuroimaging, the work of Woo et al. (2019) proposes the combination of squeeze-and-excitation blocks with U-Net and DenseNet for segmentation of acute ischemic lesions on Diffusion-Weighted Imaging. Their approach showed results superior to conventional algorithms and they concluded that squeeze-and-excitation operations may help improve segmentation for discontinuous or small lesions.

Our work utilizes an architecture, inspired by 3D U-Net with incorporation of squeeze-and-excitation blocks, as shown in Figure 5. In the contracting path, each 3×3×3 convolution is followed by a rectifed linear unit, dropout and a 2×2×2 max pooling with stride 2 for downsampling. In the expansive path, each layer consists of an upconvolution of 2×2×2 with stride of two, followed by dropout and 3×3×3 convolution followed by a rectifed linear unit. Squeeze-and-excitation operations are introduced after encoder and decoder. The effect of including these blocks will be evaluated in the experimental results section.

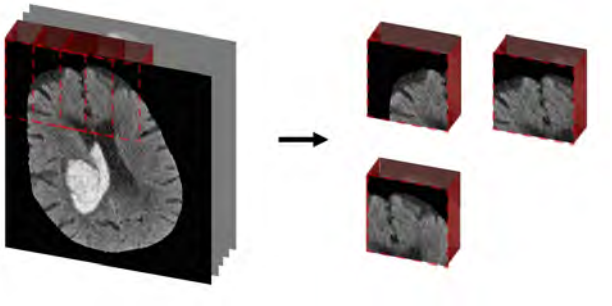
3.3.2. Training/testing pipeline

In this section we describe how training and testing was done in order to obtain and evaluate the segmentation results.

For the training stage, we composed the training and validation set from the provided scans to train the weights of the network. The effect of different number of patches and different patch sizes was analyzed. At the



(a) Training patch sampling. Negative patches (red contour) are extracted uniformly from the whole volume of the brain, and some fraction of them is extracted from the area around lesion boundary and area related to the brain ventricles. Positive patches (green contour) are centered on lesion voxels.



(b) Testing patch sampling. The patches are extracted uniformly from the whole volume, preserving some degree of overlap.

Figure 6: Implemented patch sampling for training and testing stages.

end, from each image in the training set 3000 patches of size $32 \times 32 \times 16$ were extracted following the patch sampling technique introduced before, as presented in Figure 6a. Notice that the positive patches, shown with green contour, are centered on lesion voxels, while negative patches with red contour are centered on the voxels, not related to IPH.

To take advantage of symmetry features, the symmetric image was added as another channel to each training image, therefore building each training image as Channels, Height, Width, Depth. Data augmentation was performed to increase the number of patches per image to 18000, and its effect will be also analyzed in Section 4.

Considering the approaches of ISLES participants, the two most common loss functions were tried for hematoma segmentation task: Focal loss and combined Dice loss and Crossentropy loss. Focal loss was introduced in the work of Lin et al. (2017) as an extension of the crossentropy loss. It gives less weight to easily classified examples and more weight to hard to classify examples, therefore it can be useful for the tasks with class imbalance. On the other hand, combination of Dice loss and Crossentropy loss was also popular for ischemic stroke segmentation tasks (e.g. the work of Clèrigues et al. (2019)). While the crossentropy loss is minimized with correct confident predictions, the DL is minimized when the relative overlap between prediction and ground truth is maximizing. Adadelta was used as an optimizer since it did not require manual tuning of the learning rate (Zeiler (2012)). To prevent overfitting, early stopping technique with patience of 15 was utilized when reaching the minimal loss on validation set. In our approach, the maximum number of epochs to train was set to 100 or the training was performed until it meets the early stopping condition, therefore the model with the best validation metrics was saved. In practice, the number of epochs to train was between 20-50.

For the testing stage, firstly we selected the scans for the test sets. In order to evaluate the algorithm on all the available scans, a 5-fold crossvalidation was done. Therefore, for each fold 20% of the data was separated for testing. In this stage, for each voxel we predict its probability to belong to particular class using the trained model. Given an image to segment, patches of the same size as in the training step were extracted uniformly from the whole image volume, as shown in Figure 6b. We can also notice that the extracted patches preserve predefined degree of overlap, which in our experiments was more than 50%, which was done to improve segmentation results. Every patch was passed through the network, resulting in a predicted probability for each voxel. The output binary segmentation was produced by assigning the class label according to the maximum probability for each voxel.

3.3.3. Implementation details

The proposed approach was implemented in Python using the Pytorch machine learning framework (Paszke et al. (2017)). Our work was developed using mainly the baseline niclib[1] library, which was developed within the VICOROB research group in the University of Girona. This library offers variety of utilities for developing neuroimaging pipelines with deep learning. All experiments were running on Ubuntu with 256GB RAM, the network training was done on TITAN V GPU with 12 GB memory.

4. Results

Different experiments were performed to show the improvement of the pipeline through the development process. We analyzed the influence of different steps in the pipeline on the segmentation results: (a) incorporation of squeeze-and-excitation blocks to the U-Net architecture, (b) loss functions, (c) restrictive patch extraction to improve segmentations and to solve the problem of intraventricular hemorrhage, (d) usage of different modalities and (e) data augmentation.

All the experiments were performed with the 5-fold crossvalidation across all 76 cases of the provided

[1] https://nic.udg.edu/niclib/

Table 2: DSC obtained in the dataset in crossvalidation experiment with and without incorporation of squeeze-and-excitation blocks (SE) into 3D U-Net. Introduction of these blocks significantly improved the segmentation result.

	DSC	DSC of IVH samples	DSC of samples with no IVH
3D U-Net	0.765±0.217	0.640±0.217	0.787±0.212
3D SE U-Net	**0.828±0.127**	**0.683±0.118**	**0.852±0.112**

dataset, having 61 image in the training set and 15 image in the testing set. In each fold, a network was trained on the training patches and then, at the testing stage, the voxels of testing scans were predicted. After finishing all the folds, we got one resulting segmentation for each image of the dataset.

Dice similarity coefficient (DSC) was used as evaluation metric, as it is widely used to assess segmentation tasks as a measure of overlap between output segmentation and groundtruth:

$$DSC = \frac{2TP}{2TP + FP + FN} \quad ,$$

where TP, FP and FN refer to true positive, false positive and false negative voxels respectively. To evaluate the statistical significance of differences between the obtained results, we consider dependent t-test for paired samples.

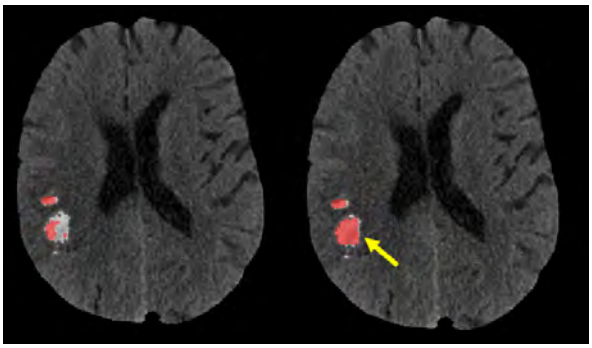
4.1. Squeeze-and-Excitation blocks

The first experiment was performed in order to understand if the changes, which were introduced to the standard 3D U-Net architecture, improved the performance of the network or not. For this experiment balanced patch sampling technique was used with target number of patches of 3000 per patient, which was found empirically and provided the best trade-off between performance and computational cost. The negative patches were extracted from the brain area only. From the results presented in Table 2 we can see that incorporating of squeeze-and-excitation blocks into standard U-Net architecture significantly improved the overall segmentation results of the dataset, increasing the average DSC and reducing standard deviation ($p<0.01$).

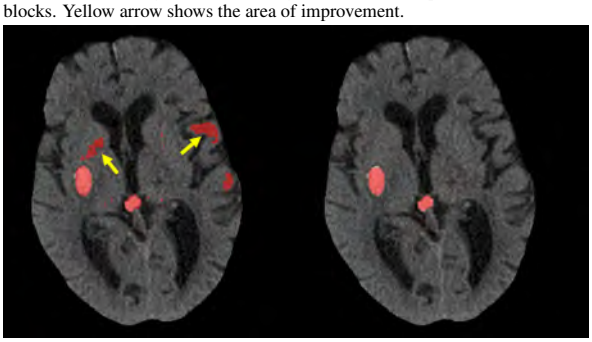
Qualitatively, the improvement can be observed in Figure 7. However, the introduction of squeeze-and-excitation blocks could not help to break through the maximum segmentation DSC, obtained for one case - it changed only from 0.967 to 0.968.

4.2. Loss functions

The loss functions we tried were Focal loss and combined Dice loss and Crossentropy loss as they are one of the most commonly used and majority of participants of ISLES challenge used them for the task of ischemic



(a) The example of better segmentation of irregular lesion while introducing squeeze-and-excitation operations. The image on the left is segmented using the model without squeeze-and-excitation blocks, while the improved segmentations on the right are made using the model with squeeze-and-excitation blocks. Yellow arrow shows the area of improvement.



(b) The example of better segmentation of small lesion while introducing squeeze-and-excitation operations. Yellow arrows show missegmentation done without squeeze-and-excitation blocks applied.

Figure 7: The qualitative evaluation of segmentation with the incorporation of squeeze-and-excitation blocks into baseline architecture.

stroke segmentation. The initial conditions for the experiment consisted of extracting 3000 patches of size (32, 32, 16) per image, as it was the best size observed empirically (Section 4.3), and using brain mask and ROI around hematoma as restrictive conditions. The resulting segmentations of the model trained with the Focal loss as a loss function showed the average DSC of 0.796±0.158. An experiment with the combination of Crossentropy and Dice loss significantly improved the segmentation result ($p<0.01$) with the average DSC of 0.841±0.108.

4.3. Restrictive patch extraction

To improve segmentation results, several considerations were taken into account. Firstly, the optimal patch size had to be chosen. The studies of Farabet et al. (2013), Li et al. (2014) showed that using bigger patches in CNNs may improve segmentation results, as the network can capture more contextual information, which in our case can be also beneficial to differentiate IPH and IVH. Therefore, an experiment was performed to study the effect of different patch sizes. Considering the architecture used (with squeeze-and-excitation blocks incorporated) and the computational load, three patch sizes were tested: (24, 24, 8), (32, 32, 16), (48, 48, 24). The results are presented in Figure 8. On the one

Table 3: The resulting DSC for the whole dataset and its parts with and without hemorrhage using different patch restriction steps

	DSC	DSC of samples with IVH	DSC of samples without IVH	max DSC	min DSC
no restrictions	0.599±0.284	0.546±0.239	0.608±0.291	0.946	0.028
brain mask	0.807±0.159	0.664±0.188	0.831±0.141	0.959	0.183
brain mask + ROI around hematoma	0.841±0.108	0.692±0.128	**0.866±0.081**	0.964	**0.530**
brain mask + ROI around hematoma + ROI around brain center	**0.842±0.115**	**0.699±0.126**	0.867±0.094	**0.968**	0.435
brain mask + ROI around hematoma + hyperintense ROI	0.823±0.136	0.687±0.124	0.846±0.125	0.963	0.306

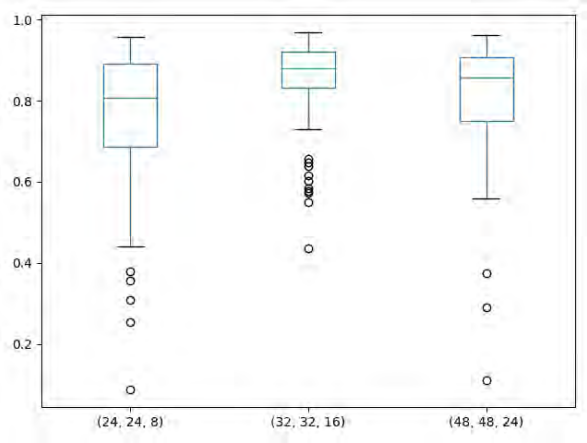


Figure 8: DSC values obtained within three experiments with different patch sizes: (24, 24, 8), (32, 32, 16), (48, 48, 24)

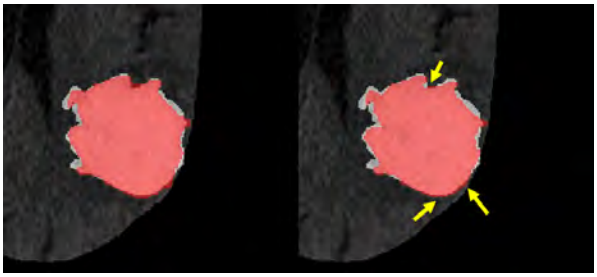


Figure 9: Example of producing more refined contours after the ROI around hematoma was fixed. Yellow arrows emphasize the particular parts of the lesion contour, that improved

hand, increasing patch size to (32, 32, 16) significantly improved mean DSC ($p < 0.001$) from 0.759±0.183 to 0.842±0.115, but, on the other hand, when the patch size was enlarged more to (48, 48, 24), the average DSC was significantly decreased ($p<0.01$) to 0.805±0.156. Moreover, none of the patch sizes helped to overcome the upper boundary of DSC (maximum DSC achieved are 0.958, 0.968 and 0.963, respectively). From the Figure 8 we noticed that the results obtained with patches of the size (32, 32, 16) were within the smaller range, which states lower dispersion. For the later experiments we opted the medium patch size of (32, 32, 16).

Next, the area to extract negative patches from was restricted by applying the brain mask, so that the patches were extracted only from the brain volume. This way we avoided extracting completely black patches from the image background. Table 3 shows that this constraint helped to significantly improve segmentation results by 11.7% ($p < 0.001$).

To refine the lesion contours, the region of interest was defined around the hematoma and a fraction of patches, which we defined empirically to be 30%, was extracted from this region. This ROI was represented as cubic volume 25 pixels away from hematoma borders. Defining this region made it possible to focus the network more on the area near the lesion borders and it helped to significantly improve resulting segmentations ($p < 0.01$), as can be observed from the average DSC from Table 3 and qualitatively from Figure 9. In addition, the minimal DSC in this experiment is notably improved, going from 0.183 to 0.530.

To solve the problem of intraventricular hemorrhage segmentation, we wanted to make the network learn more from IVH areas to distinguish IVH from IPH and from CSF ventricles areas. Firstly, the cubic region of interest was defined around central point of brain. This way we could extract more patches from the area related to CSF ventricles. In the experiment we compared the influence of forcibly extracting patches from this ROI on the final results. The patch extraction pipeline consisted of restrictions on the negative patch sampling applied by the brain mask and the region of interest around hematoma. The results of this experiment, presented in Table 3, showed that introducing this condition could increase the resulting DSC only a little, but the standard deviation rises, which means that more variability is added. However, these changes were not considered as significant ($p>0.05$).

To improve the results of the previous experiment, the

Table 4: The evaluation metrics of the experiment with and without symmetric modality as an additional input channel. Different patch restriction cases are compared.

	Input modalities	DSC	DSC of samples with IVH	DSC of samples without IVH	max DSC	min DSC
ROI around hematoma	original image	0.841±0.108	0.692±0.128	0.866±0.081	0.964	0.530
	with symmetric modality	0.849±0.099	0.720±0.116	0.871±0.078	0.964	0.530
ROI around hematoma + brain center	original image	0.842±0.115	0.699±0.126	0.867±0.094	**0.968**	0.435
	with symmetric modality	0.856±0.088	0.728±0.112	0.878±0.061	0.956	0.553
ROI around hematoma + IVH area	original image	0.823±0.136	0.687±0.124	0.846±0.125	0.963	0.306
	with symmetric modality	0.857±0.085	0.728±0.107	**0.879±0.057**	0.964	0.581
ROI around hematoma OR IVH area	with symmetric modality	**0.862±0.074**	**0.777±0.101**	0.876±0.059	0.957	**0.632**

cubic ROI was replaced by a ROI, representing IVH - a hyperintense volume in the image excluding IPH. The results of new test, shown in Table 3, indicate that the resulting segmentation DSC significantly decreased ($p < 0.01$) for all the dataset and for groups with and without IPH.

4.4. Usage of different modalities

As stroke appears in one hemisphere of the brain, symmetric non-contrast CT image was added to the original one as another input channel to exploit the property of brain symmetry. The experiment was performed, holding the same training parameters and patch sampling strategy, defined in Section 3.2. The results are presented in Table 4. Adding the symmetric modality to a segmentation pipeline with the fixed ROI around the lesion made segmentations better, with mean DSC improvement from 0.841±0.108 to 0.849±0.099, being statistically significant ($p<0.01$). We can also see in this experiment the improvement in segmentation of images with IVH by 2.8%, and generally, symmetric modality augmentation reduced standard deviation. Qualitatively, it can be observed in Figure 10, where yellow arrows indicate the areas, which were correctly segmented as background after introducing the symmetric input.

Even though defining a cubic ROI around the brain center and using symmetric image as additional input channel improved segmentation results, these improvements were not significant ($p>0.05$). However, if symmetric modality augmentation is used when patches are forcibly extracted from IVH volumes, the DSC of resulting segmentations, especially for the images with IVH, as can be noticed from Table 4, significantly improved ($p<0.01$) by 4.9%.

Angiography CT was also used as an additional input channel together with original non-contrast CT. Not all the cases from the provided dataset included this modality, so for a fair comparison, these cases were excluded (totally 10 cases). We observed that when introducing angiography image segmentation DSC significantly decreased ($p<0.001$), leading average DSC from 0.834±0.120 with no additional modality augmentation to 0.795±0.156.

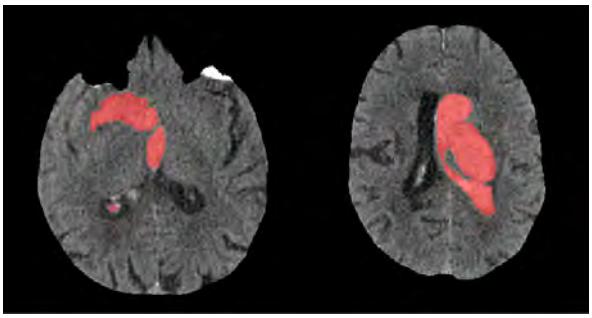
4.5. Data augmentation

The influence of data augmentation techniques was studied by applying the previously mentioned transformations to all the patches extracted within the experiment. Moreover, we also analyzed the effect of the number of patches used into the segmentation result.

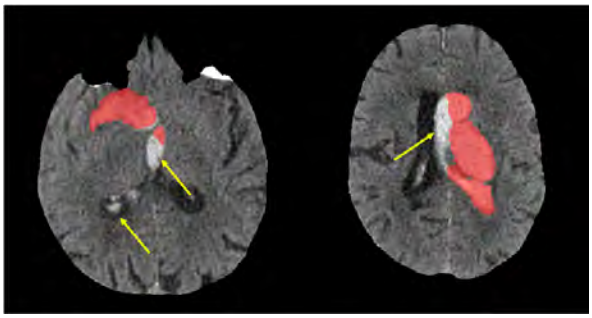
Firstly, the target number of patches was set to 500, so that they can be augmented up to 3000. Compared to the baseline approach (Table 5), the segmentation results with a new patch dataset composition significantly decreased ($p<0.05$). The following data augmentations implied to increase the number of patches. As we increased the initial number of extracted patches, the resulting DSC slightly improved, only decreasing when the initial number of patches was 1500. These changes from one number of patches to another were statistically significant ($p<0.01$, $p<0.01$ and $p<0.005$, respectively). However, the difference between the results obtained with 3000 patches and with 3000 patches augmented to 18000 was not statistically significant ($p>0.05$).

4.6. Final configuration

Taking into account all these previous experiments, the final configuration of the proposed method was chosen. The network architecture included squeeze-and-excitation blocks and was trained with 3000 patches of size (32, 32, 16) with combination of Dice and Crossentropy losses as a loss function. Even though results



(a) Examples of images segmented with only CT NC as an input



(b) Examples of images segmented with symmetric CT NC as an additional input

Figure 10: Results of hematoma segmentation with and without using symmetric CT NC as an additional input. Yellow arrows show the changes in the results achieved by incorporating symmetric modalities as another input channel.

Table 5: The evaluation metrics of experiments checking the influence of data augmentation and its size on the segmentation results

	number of patches	DSC	std
	3000	0.862	0.074
augmentation	500 $\longrightarrow$ 3000	0.845	0.092
	1000 $\longrightarrow$ 6000	0.866	0.078
	1500 $\longrightarrow$ 9000	0.846	0.092
	3000 $\longrightarrow$ 18000	0.868	0.076

with data augmentation improved the mean DSC, considering computational cost versus amount of improvement, data augmentation was not considered in the final method. Original image together with its symmetric modality were used as input, and patches were extracted from them restrictively, limiting the area to extract patches from with brain mask, ROI around hematoma and IVH volume. With this final design, we could achieve a DSC of 0.862 with processing segmentation time of 17.15 seconds per patient. The qualitative example of one of the best segmentation results is shown on Figure 11.

Comparison with state-of-the-art approaches was difficult, since all methods used different datasets, which may include different levels of severity. Our goal was to segment intraparenchymal hemorrhage only, excluding intraventricular hemorrhage regions. However, in other state-of-the-art approaches IVH cases were not presented, therefore these works could train specifically IPH regions, providing higher overall DSC values. Moreover, these methods used datasets with more scans than in our case, which could also influence the resulting segmentations. For instance, the work of Chang et al. (2018), reported an average DSC of 0.931 with a dataset of 10159 CT scans where 8.9% of scans had ICH, while the approaches of Singh et al. (2019) and Kuang et al. (2020) provided DSC values of 0.932 and 0.864, with datasets composed of 399 and 150 scans, respectively. The approach of Yao et al. (2020) reported an average DSC of 0.697 with a dataset of 120 CT scans from different centers. Nevertheless, the overall results obtained for IPH with our approach (DSC of 0.879) should be considered as very satisfactory, specially taking into account the challenges presented in our dataset, both the consideration of not including the intraventricular hemorrhage as a lesion, and the number of such cases in the whole dataset, being almost 15%.

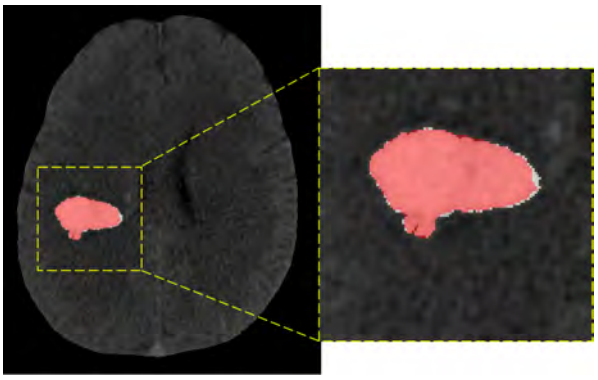


Figure 11: Example of a good segmentation result. The red overlay represents resulting segmentation, while the white one represents groundtruth.

5. Discussion

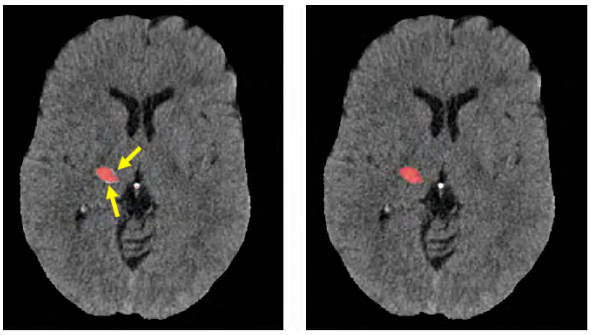
This study presented a deep learning approach for hemorrhagic stroke lesions segmentation. Despite the fact that two subtypes of hemorrhagic stroke lesions are presented in the dataset, only one of them, namely intraparenchymal hemorrhage, is considered as lesion to be delineated, as the other subtype, intraventricular hemorrhage, is mostly secondary, resulting from existing IPH. Hence, the issue of differentiating it from the remaining subtype of stroke, intraventricular one, has to be taken into account within the segmentation framework. In this task, the IVH problem was the main one to worsen the segmentation results, so mostly the steps we took were attempting to reduce the number of segmented IVH voxels. However, some options were tried to generally improve the obtained segmentations and refine the contours of the stroke lesion.

Hemorrhagic stroke appears hyperintense on CT images, therefore, as previous studies show, promising performance could be achieved. The simple patch-based 3D U-Net architecture with the only restriction of a brain mask for negative class already achieved a mean DSC of 0.765±0.217, as shown in Table 2. Notice that this result is better than the one reported by the approach of Yao et al. (2020) who achieved an average DSC of 0.697 for IPH segmentation against our result of 0.765. Their approach was also inspired with U-Net, yet their dataset is heterogeneous, acquired in different medical centers, unlike ours, which may be the reason of such increase in performance.

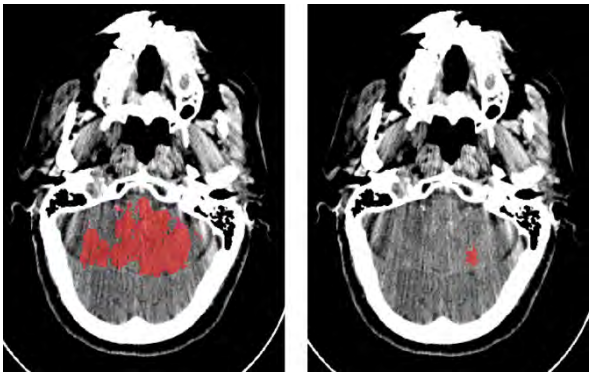
Incorporation of squeeze-and-excitation blocks were performed similarly to the approach of Woo et al. (2019) for segmentation of ischemic stroke lesions. Results showed that it helped to significantly improve the average segmentation DSC by 6.3%, as well as to reduce variability by decreasing the standard deviation from 0.217 to 0.127 (Table 2). Likewise the results of Woo et al. (2019), squeeze-and-excitation operations helped to better segment small and irregular lesions, as can be seen in Figure 7. While segmenting small lesions, missegmentations were reduced; for the discontinued lesions, which have more variability in intensities through single lesion, squeeze-and-excitation blocks helped to detect more hematoma voxels.

Regarding loss functions, we expected to observe comparatively similar performance of Focal loss and the combination of Dice loss and Crossentropy loss. However, the combination of DL and Crossentropy loss significantly outperformed the Focal loss, increasing the average DSC by 4.5% and reducing standard deviation by 5%. In most cases, the model trained with Focal loss undersegmented the borders of small lesions, as shown by yellow arrows on Figure 12a. Also, it wrongly segmented the voxels near the brain stem areas in some images, as can be seen in Figure 12b. It might be because the beam hardening effect artifact is possible in these areas, as they were located within big volume of dense tissues, like bone and teeth, which may contain some external materials. This artifact is due to scattering of the X-ray beam and to alteration of the average power of the X-ray beam as it passes through relatively dense structures, producing bright streaks on the image. Therefore, these streaks were misclassified by model trained with Focal loss.

The experiments with different patch sizes showed unexpected behavior. As mentioned before, enlargement of patch size should have helped the network to capture more contextual information. However, our experiments showed, that increasing of the patch size up to (48, 48, 24) worsened the mean DSC and introduces more variability to the samples, as shown in Figure 8. Moreover, the larger patch size could not help to capture dependencies between IPH and IVH. Individually, the most significant decrease of DSC was happening with



(a) The example of small lesions segmentation when the model is trained with Focal loss (on the left) and with Crossentropy+Dice loss (on the right). Yellow arrows show undersegmentation of lesion border.



(b) The example of missegmentation of areas affected by beam-hardening artifact. The result on the left image is obtained with the model trained with Focal loss, whereas the segmentation on the right is produced by model trained with Crossentropy+Dice loss.

Figure 12: The qualitative results obtained by models, trained with Focal loss and Crossentropy+Dice loss.

small lesions, which states that for this particular dataset the patch size of (48, 48, 24) was too big to capture the small hematoma volumes. A similar behavior was observed in the work of Bernal et al. (2019) for tissues segmentation in MRI in some of the studied approaches. Increasing patch size from (24, 24, 8) to (32, 32, 16) led to better segmentations, as expected initially. Therefore, with regard to the results of the experiments, the patch size of (32, 32, 16) was chosen as the optimal one.

The restrictive patch sampling was mostly performed to solve the problem of intraventricular hemorrhage. Nevertheless, the initial constraint of extracting patches only from the brain volume was performed in order to generally make the segmentations better. Obtained results proved that sampling patch centers only from brain voxels greatly increased the DSC over all cases of the dataset. Consequently, the training patch set never includes empty patches of the image background, so the network receives more valuable information as input.

The significant rise of the DSC in the case of fixing the region of interest around the lesion was expected, as it was also having a big impact on the work presented by Kushibar et al. (2018). In most cases introducing this condition helped to refine the lesion contours, as can be observed in qualitative examples provided in Figure 9.

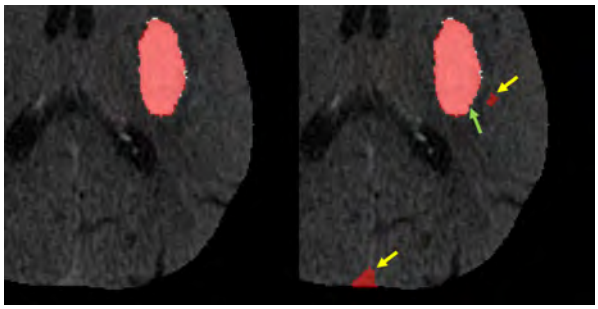


Figure 13: Example of missegmentation. Even though ROI helped to refine contours (green arrow), voxels not related to the lesion were also segmented (yellow arrows).

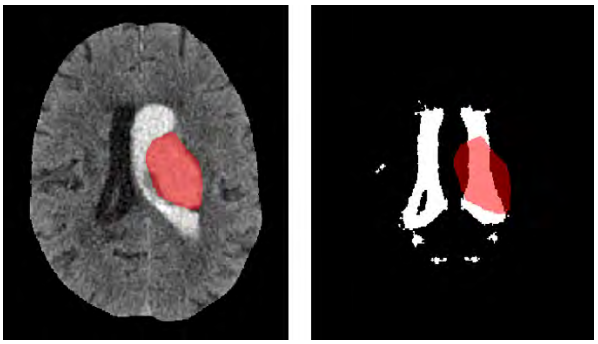


Figure 14: The left image shows the example of ventricle deformation in the presence of IVH. The image on the right shows the incorporation of symmetric ventricle to cut off ventricular voxels, though mostly it cuts the IPH voxels.

However, for some images improper outputs were produced. Notice that less patches are extracted from overall brain volume, therefore some voxels in unseen areas of the background were mistaken for lesion. See for instance the qualitative examples shown in Figure 13, where the part indicated by the brain middle line segmented as lesion.

The attempts to solve the issue of intraventricular hemorrhage by establishing another regions of interest around CSF ventricles did not significantly change hemorrhage segmentation, introducing more variability to the data (Table 3). This was possibly due to the fact that in this case the network received more patches with IVH and parts of CSF ventricles, and overall distribution of patches therefore was not enough to successfully distinguish IPH and IVH.

Symmetric modality augmentation showed an interesting behavior. Even without extra guidance to ventricle areas it could improve segmentations of both groups of scans with and without IVH, exploiting the fact, that hemorrhage occurs in one of the brain hemispheres.

Even though fixing the ROIs related to brain ventricles did not help to achieve better results, they were successfully used together with symmetric modality augmentation. We can observe from Table 4, that fixing these ROIs improved more the segmentations, compared to fixing only ROI around hematoma. This happened due to the fact that these two novelties together guide where the ventricles and normal tissues are, exploiting the information about ventricles shape and their deformation in the presence of intraventricular hemorrhage. Moreover, this answers the question, why the symmetry of a healthy ventricle cannot be utilized to reduce the affected area of the opposite ventricle - a ventricle with IVH deforms in a way, so that this assumption can cut the correctly segmented IPH.

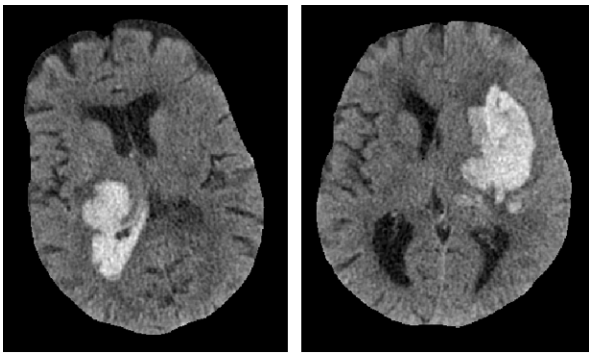
Qualitatively, Figure 10 shows that incorporation of symmetric modality as an additional input channel could reduce segmentation of voxels which are located in the other hemisphere of the brain than the one damaged by stroke. Moreover, segmentation of voxels of the same hemisphere, but related to intraventricular hemorrhage, was also reduced. However, brain ventricles can be malformed because of the intraventricular hemorrhage, as shown in Figure 14. If the symmetric version of the normal ventricle was applied as a mask, it would cut the large part of IPH.

Reduction in segmentation results after introducing CT angiography as additional input channel was expected, as these images are more noisy, than non-contrast CT scans. However, CT angiography is widely spread as a gold standard in the imaging of cerebral parenchymal hemorrhage to assess the spot sign - an indicator of ongoing bleeding. Therefore, it can be assumed that this modality can be helpful for stroke segmentation, adding additional information for the network to learn. However, it was shown by Koculym et al. (2013), that the spot sign detection on CT angiography images demonstrates low sensitivity of 44%, which can be the reason of poor performance of angiography modality augmentation.

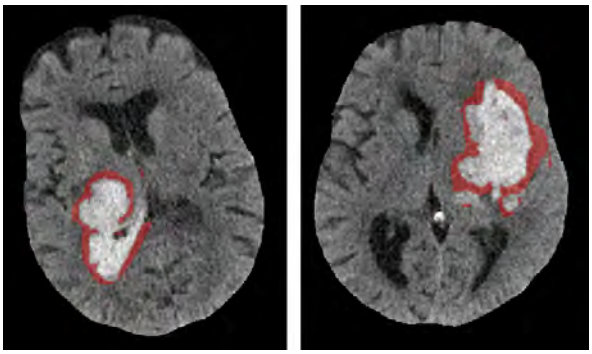
Artificially expanding the dataset using data augmentation techniques did not make a breakthrough in the segmentation results. Moreover, augmenting the patches from 3000 to 18000 did not introduce much improvement improvement compared to the results obtained with the patch dataset of only 3000 without any augmentation (Table 5). The overall difference in the results was not significant ($p>0.05$). Indeed, considering the mean DSC and standard deviation values from Table 5, one can see that the general behavior of the DSC distribution was similar.

5.1. Perihematomal edema segmentation

Apart from all the analyses presented in this work, we also studied the segmentation of the perihematomal edema, which is a brain swelling region around the hemorrhagic stroke lesion. It is an inflammatory response to the hematoma and it causes further damage to the brain tissues within some time period after stroke onset. Therefore, detecting and segmenting this region can help to predict clinical outcome of a stroke patient. In



(a) The original CT images. Perihematomal edema has poor contrast with surrounding healthy tissue.



(b) The examples of edema segmentation using deep learning and previously produced edema masks. The left image shows the segmented voxels around damaged brain ventricle.

Figure 15: Original images together with edema segmentation results.

our work, we analyzed the application of our approach to this problem.

In stroke imaging, T2-weighted MRI helps to effectively delineate edema from the surrounding parenchyma (Ironside et al. (2019)). However, in hemorrhagic stroke patients magnetic resonance imaging is not routinely used as a neuroimaging modality. In this case, CT may be a more suitable modality, but segmentation of edema from a CT image is a very challenging task due to its poor contrast with surrounding tissues (Urday et al. (2015), Ironside et al. (2019)). We can visually observe this from Figure 15a.

In literature, few attempts were taken to segment perihematomal edema from CT images. For instance, Cosić and Lončarić (1997) used expert-system based labeling for both hematoma and edema segmentation. Chen et al. (2013) proposed a method based on region growing with seeds obtained from expectation-maximisation algorithm. The method was evaluated on a dataset of 36 patients. Volbers et al. (2011) suggested a threshold based edema segmentation. Their proposed thresholds were identified to provide the best correlation between the resulting segmentations in CT and manual delineations in MRI.

As the dataset we used was acquired for the clinical study of Hospital Dr. Josep Trueta, the perihematomal edema segmentation was also proposed as a task. The main problem to overcome was the absence of the groundtruth segmentations of edema. At initial step, the edema masks based on distance transform were generated within the VICOROB group. The same deep learning approach, as for hematoma, was also tried for edema segmentation, with restrictive patch sampling, where patches were extracted from the brain volume and region around hematoma. The previously introduced masks were provided as groundtruth. Moreover, the previously obtained stroke core segmentations were provided as additional input channel to the network to guide the location of lesion borders. As no groundtruth was provided, we could not perform a quantitative analysis of these experiments. Qualitatively, from Figure 15 we can observe visually tolerable edema segmentations. However, the problem of intraventricular hemorrhages arises here as well, as edema can be segmented around brain ventricle, which can be noticed in the left example of Figure 15b.

The limitation for solving this task is that there are no datasets available with groundtruth of the edema. Having available groundtruth could help to adapt our current proposal for this problem or even to refine the results of previous approaches, which could be a good line to explore as a future work.

6. Conclusions

In this master thesis we proposed a deep learning method for hemorrhagic stroke lesions segmentation. The proposed approach, based on 3D U-Net with integration of squeeze-and-excitation blocks, was tested on the clinical dataset of 76 cases, provided by a local collaborating hospital (Hospital Dr. Josep Trueta). All the obtained results were qualitatively and quantitatively evaluated on the whole dataset using a 5-fold crossvalidation strategy. Our approach was inspired by the work of Woo et al. (2019), who incorporated squeeze-and-excitation blocks into different architectures to solve the ischemic stroke segmentation task. We could show that such architecture significantly improved segmentation results ($p<0.01$).

Moreover, we showed that data preparation step is very important to obtain a good segmentation method. By using restrictive balanced sampling technique, we could tackle the class imbalance problem as well as the problem of intraventricular hemorrhage. When applying different constraints on the patch extraction pipeline, we quantitatively showed the statistical significant improvements. In addition, we were able to study the influence of using different modalities as input to the network and we could show the improvements achieved by the introduction of symmetric modality as additional input channel. Different patch characteristics were also studied, allowing us to choose the optimal patch size (32, 32, 16) as well as show the influence of number of patches and data augmentation on the obtained final results.

Two loss functions, Focal loss and combined Crossentropy and Dice loss, which are commonly used for ischemic stroke segmentation problem, were also examined. Our results showed the superior performance of the model trained with the combination of Crossentropy and Dice loss.

Having DSC as main evaluation metric, we could achieve a mean segmentation result of 0.862 ± 0.074 for all cases and 0.879 ± 0.057 for cases without intraventricular hemorrhage. The task of perihematomal edema was also approached and qualitatively evaluated.

Finally, although our dataset and provided groundtruth annotations (without considering the IVH regions) did not allow a direct comparison with state-of-the-art approaches, our proposal showed, that incorporation of squeeze-and-excitation blocks to the 3D U-Net together with symmetric modality as additional input channel provides promising results with accurate automated segmentations of the hemorrhagic stroke lesions.

7. Acknowledgments

I would like to thank my supervisors Dr Xavier Lladó and Dr Arnau Oliver for their support, feedback and the opportunity to develop this master thesis within VICOROB group. Also my gratitude to Kaisar Kushibar and Albert Clèrigues for their help and recommendations. I would like to thank Hospital Dr. Josep Trueta, especially Dr. Salvador Pedraza and Dra. Yolanda Silva, for providing the dataset for research.

References

Alom, M.Z., Yakopcic, C., Hasan, M., Taha, T., Asari, V., 2019. Recurrent residual u-net for medical image segmentation. Journal of Medical Imaging 6.

Bardera, A., Boada, I., Feixas, M., et al, 2009. Semi-automated method for brain hematoma and edema quantification using computed tomography. Computerized medical imaging and graphics 33, 304–11.

Bernal, J., Kushibar, K., Cabezas, M., Valverde, S., Oliver, A., Lladó, X., 2019. Quantitative analysis of patch-based fully convolutional neural networks for tissue segmentation on brain magnetic resonance imaging. IEEE Access 7, 89986–90002.

Bhadauria, H., Dewal, M., 2014. Intracranial hemorrhage detection using spatial fuzzy c-mean and region-based active contour on brain ct imaging. Signal, Image and Video Processing 8.

Chang, P., Kuoy, E., Grinband, J., et al, 2018. Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head ct. American Journal of Neuroradiology .

Chen, M., Hu, Q., Liu, Z., Zhou, S., Li, X., 2013. Segmentation of cerebral edema around spontaneous intracerebral hemorrhage. Applied Mathematics & Information Sciences 7, 563–570.

Choi, Y., Kwon, Y., Lee, H., et al, 2016. Ensemble of deep convolutional neural networks for prognosis of ischemic stroke, pp. 231–243.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Springer International Publishing, Cham. pp. 424–432.

Clèrigues, A., Valverde, S., Bernal, J., et al, 2019. Acute ischemic stroke lesion core segmentation in ct perfusion images using fully convolutional neural networks. Computers in Biology and Medicine 115, 103487.

Cosić, D., Lončarić, S., 1997. Computer system for quantitative: analysis of ich from ct head images, in: Annual International Conference of the IEEE Engineering in Medicine and Biology Society., pp. 553–556 vol.2.

Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y., 2017. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks, pp. 506–517.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1915–1929.

Forbes, F., Doyle, S., Garcia-Lorenzo, D., Barillot, C., Dojat, M., 2010. Adaptive weighted fusion of multiple mr sequences for brain lesion segmentation, in: 2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 69–72.

Gillebert, C.R., Humphreys, G.W., Mantini, D., 2014. Automated delineation of stroke lesions using brain ct images. NeuroImage: Clinical 4, 540 – 548.

Guerrero, R., Qin, C., Oktay, O., et al, 2017. White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks. NeuroImage: Clinical 17.

Hevia, N., Jiménez-Alaniz, J., Medina, V., et al, 2007. Robust nonparametric segmentation of infarct lesion from diffusion-weighted mr images. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2007, 2102–5.

Hinson, H., Hanley, D., Ziai, W., 2010. Management of intraventricular hemorrhage. Current neurology and neuroscience reports 10, 73–82.

Hssayeni, M., Al-Janabi, M., Salman, A., et al, 2020. Intracranial hemorrhage segmentation using a deep convolutional model. Data 5, 14.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141.

Ironside, N., Chen, C.J., Ding, D., Mayer, S.A., Connolly, E.S., 2019. Perihematomal edema after spontaneous intracerebral hemorrhage. Stroke 50, 1626–1633.

Jenkinson, M., Bannister, P., Brady, M., Smith, S., 2002. Improved optimization for the robust and accurate linear registration and motion correction of brain images. NeuroImage 17, 825 – 841.

Jenkinson, M., Smith, S., 2001. A global optimisation method for robust affine registration of brain images. Medical Image Analysis 5, 143 – 156.

Kamnitsas, K., Chen, L., Ledig, C., Rueckert, D., Glocker, B., 2015. Multiscale 3d convolutional neural networks for lesion segmentation in brain mri. Proc. MICCAI Ischemic Stroke Lesion Segmentation Challenge .

Kamnitsas, K., Ferrante, E., Parisot, S., et al, 2016. Deepmedic for brain tumor segmentation, pp. 138–149.

Kidwell, C., Wintermark, M., 2008. Imaging of intracranial haemorrhage. The Lancet Neurology 7, 256–267.

Koculym, A., Huynh, T., Jakubovic, R., Zhang, L., Aviv, R., 2013. Ct perfusion spot sign improves sensitivity for prediction of outcome compared with cta and postcontrast ct. American Journal of Neuroradiology 34, 965–970.

Kuang, Z., Deng, X., Yu, L., et al, 2020. -net: Focusing on the border areas of intracerebral hemorrhage on ct images. Computer Methods and Programs in Biomedicine 194, 105546.

Kushibar, K., Valverde, S., González-Villà, S., et al, 2018. Automated sub-cortical brain structure segmentation combining spatial and deep convolutional features. Medical Image Analysis 48, 177 – 186.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278 – 2324.

Li, H., Zhao, R., Wang, X., 2014. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. ArXiv abs/1412.4526.

Li, M., Ai, L., He, H., et al, 2009. Segmentation of infarct in acute ischemic stroke from mr apparent diffusion coefficient and trace-weighted images. Proceedings of SPIE 7497, 74971U.

Li, S., Dong, M., Du, G., Mu, X., 2019. Attention dense-u-net for automatic breast mass segmentation in digital mammogram. IEEE Access 7, 59037–59047.

Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440.

Lončarić, S., Dhawan, A., Broderick, J., Brott, T., 1995. 3-d image analysis of intra-cerebral brain hemorrhage from digitized ct films. Computer methods and programs in biomedicine 46, 207—216.

Lucas, C., Kemmling, A., Madany Mamlouk, A., Heinrich, M., 2018. Multi-scale neural network for automatic segmentation of ischemic strokes on acute perfusion images.

Majcenić, Z., Lončarić, S., 1998. Ct image labeling using simulated annealing algorithm, in: 9th European Signal Processing Conference (EUSIPCO 1998), pp. 1–4.

Matesin, M., Lončarić, S., Petravić, D., 2001. A rule-based approach to stroke lesion analysis from ct brain images, pp. 219 – 223.

Matsuo, R., Yamaguchi, Y., Matsushita, T., et al, 2017. Association between onset-to-door time and clinical outcomes after ischemic stroke. Stroke 48.

Mozaffarian, D., Benjamin, E., Go, A., et al, 2016. Heart disease and stroke statistics-2016 update a report from the american heart association. Circulation 133, e38–e48.

Muir, K.W., Santosh, C., 2005. Imaging of acute stroke and transient ischaemic attack. Journal of Neurology, Neurosurgery & Psychiatry 76, iii19–iii28.

Paszke, A., Gross, S., Chintala, S., et al, 2017. Automatic differentiation in pytorch, in: NIPS-W.

Perez, N., Valdes, J., Guevara Lopez, M.A., Rodríguez, L., Molina, J., 2007. Set of methods for spontaneous ich segmentation and tracking from ct head images, pp. 212–220.

Piantadosi, G., Sansone, M., Sansone, C., 2018. Breast segmentation in mri via u-net deep convolutional neural networks, in: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3917–3922.

Pszczolkowski, S., Law, Z.k., Gallagher, R., et al, 2019. Automated segmentation of haematoma and perihaematoma oedema in mri of acute spontaneous intracerebral haemorrhage. Computers in Biology and Medicine 106.

Puig, J., Blasco, G., Daunis-i Estadella, P., et al, 2017. High-permeability region size on perfusion ct predicts hemorrhagic transformation after intravenous thrombolysis in stroke. PLOS ONE 12, e0188238.

Pérez, N., Valdés, J., Guevara Lopez, M.A., Silva, A., 2008. Spontaneous Intracerebral Hemorrhage Image Analysis Methods: A Survey. pp. 235–251.

Ronneberger, O., P.Fischer, Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 234–241. (available on arXiv:1505.04597 [cs.CV]).

Roth, G.A., Abate, D., Abate, K.H., et al, 2018. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017. The Lancet 392, 1736 – 1788.

Roy, S., Ghosh, P., Bandyopadhyay, S., 2015. Contour extraction and segmentation of cerebral hemorrhage from mri of brain by gamma transformation approach.

Rundo, L., Han, C., Nagano, Y., et al, 2019. Use-net: Incorporating squeeze-and-excitation blocks into u-net for prostate zonal segmentation of multi-institutional mri datasets. Neurocomputing 365, 31 – 43.

Shahangian, B., Pourghassem, H., 2013. Automatic brain hemorrhage segmentation and classification in ct scan images, in: 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP), pp. 467–471.

Shahangian, B., Pourghassem, H., 2015. Automatic brain hemorrhage segmentation and classification algorithm based on weighted grayscale histogram feature in a hierarchical classification structure. Biocybernetics and Biomedical Engineering 36.

Sharma, B., Venugopalan, K., 2012. Automatic segmentation of brain ct scan image to identify hemorrhages. International Journal of Computer Applications 40, 1–4.

Siddiqui, F., Bekker, S., Qureshi, A., 2011. Neuroimaging of hemorrhage and vascular defects. Neurotherapeutics 8, 28–38.

Singh, S., Ker, J., Bai, Y., et al, 2019. Image thresholding improves 3-dimensional convolutional neural network diagnosis of different acute brain hemorrhages on computed tomography scans. Sensors 19, 2167.

Tang, F.H., Ng, D., Chow, D., 2011. An image feature approach for computer-aided detection of ischemic stroke. Computers in Biology and Medicine .

Urday, S., Beslow, L.A., Goldstein, D.W., Vashkevich, A., Ayres, A.M., Battey, T.W., Selim, M.H., Kimberly, W.T., Rosand, J., Sheth, K.N., 2015. Measurement of perihematomal edema in intracerebral hemorrhage. Stroke 46, 1116–1119.

Usinskas, A., Dobrovolskis, R., Tomandl, B., 2004. Ischemic stroke segmentation on ct images using joint features. Informatica, Lith. Acad. Sci. 15, 283–290.

Volbers, B., Staykov, D., Wagner, I., Dörfler, A., Saake, M., Schwab, S., Bardutzky, J., 2011. Semi-automatic volumetric assessment of perihemorrhagic edema with computed tomography. European journal of neurology : the official journal of the European Federation of Neurological Societies 18, 1323–8.

Wang, Y., Liu, H., Liu, Y., Liu, W., 2018. Deep learning framework for hemorrhagic stroke segmentation and detection, in: BIBE 2018; International Conference on Biological Information and Biomedical Engineering, pp. 1–6.

Wanida Charoensuk, Nongluk Covavisaruch, S.L., Likitjaroen, Y., 2015. Acute stroke brain infarct segmentation in dwi images. International Journal of Pharma Medicine and Biological Sciences 04, 115–122.

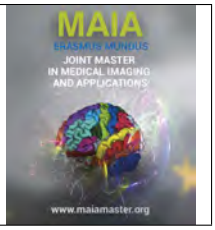
Woo, I., Lee, A., Jung, S., et al, 2019. Fully automatic segmentation of acute ischemic lesions on diffusion-weighted imaging using convolutional neural networks: Comparison with conventional algorithms. Korean Journal of Radiology 20, 1275.

Yao, H., Williamson, C., Gryak, J., Najarian, K., 2020. Automated hematoma segmentation and outcome prediction for patients with traumatic brain injury. Artificial Intelligence in Medicine 107, 101910.

Zeiler, M.D., 2012. Adadelta: An adaptive learning rate method. arXiv:1212.5701.

Medical Imaging and Applications

Master Thesis, August 2020



Automatic Myocardial Scar Segmentation from Multi-Sequence Cardiac MRI using modified FC-Densenet with Region Mutual Information Loss

Tewodros Weldebirhan Arega, Stéphanie Bricq

ImViA Laboratory, Université Bourgogne Franche-Comté , Dijon, France

Abstract

Myocardial infarction, commonly known as a heart attack, is the irreversible death of heart muscle (myocardium) due to lack of oxygen supply (ischemia). In a clinical routine, the infarcted myocardium is often segmented manually. Since manual segmentation is time consuming and suffers from intra- and inter-observer variability, it is of great interest to develop an automatic and accurate myocardial scar segmentation. However, the automatic method is also difficult due to the presence of motion artifact and low contrast between scar and its surrounding in the cardiac magnetic resonance (CMR) images. In this paper, we proposed a fully-automatic scar segmentation method using a cascaded segmentation networks of three Fully Convolutional Densenet (FC-Densenet) with Inception and Squeeze-Excitation module. It is called Cascaded FCDISE. The first FCDISE is used to extract the region of interest and the second FCDISE to segment myocardium and the last one to segment scar from the pre-segmented myocardial region. In the proposed segmentation network, the inception module is incorporated at the beginning of the network to extract multi-scale features from the input image, whereas the squeeze-excitation blocks are placed in the skip connections of the network to transfer recalibrated feature maps from the encoder to the decoder. To encourage higher order similarities between predicted image and ground truth, we adopted a dual loss function composed of logarithmic Dice loss and region mutual information (RMI) loss. Our method is evaluated on the Multi-sequence CMR based Myocardial Pathology Segmentation challenge (MyoPS 2020) dataset. On the test set, our fully-automatic approach achieved an average Dice score of 0.590 for scar and 0.686 for scar+edema. This is higher than the inter-observer variation of manual scar segmentation. The proposed method outperformed similar methods and showed that adding the two modules to FC-Densenet improves the segmentation result with little computational overhead.

Keywords: Multi-sequence cardiac MRI, MyoPS, Myocardial scar, Segmentation, Deep Learning, CNN, Fully-automatic

1. Introduction

Cardiovascular diseases (CVDs) are the number one cause of death globally. More people die annually from CVDs than from any other cause according to World Health Organization (WHO). An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% are due to heart attack and stroke (Organization, 2017). Cardiovascular disease (CVD) is a general term for conditions affecting the heart or blood vessels.

Myocardial infarction (MI), commonly known as a heart attack, is the irreversible death of heart muscle (myocardium) due to lack of oxygen supply (ischemia). This happens when there is plaque or blood clot in the coronary artery which is responsible for supply of blood and oxygen to heart muscles. As the cells are deprived of oxygen, cellular injury occurs which leads to the infarction or death of the cells (Belleza, 2017).

Edema is the excess accumulation of fluid in the myocardial interstitium which develops as a result of imbalance between filtration from the coronary microvasculature and removal of interstitial fluid via lymphatic vessels and epicardial transudation. This can create with acute conditions like myocardial infarction and with chronic conditions like pulmonary hypertension (Dongaonkar et al., 2012).

Cardiac magnetic resonance (CMR) is a set of magnetic resonance imaging (MRI) used to provide anatomical and functional information of the heart. There are many types of CMR sequences. To mention some: Late Gadolinium Enhancement (LGE), balanced Steady State Free Precession (bSSFP) cine sequence and T2-weighted MR. These sequences bring complimentary information to each other (Hammer-Hansen et al., 2016).

Late Gadolinium Enhancement (LGE), sometimes called delayed-enhancement MRI, is a gold standard for the quantification of myocardial infarction. It shows an optimum contrast between normal and infarcted myocardium (Fig. 1). This is done first by administering intravenously gadolinium-based contrast agents (GBCAs) and then performing a delayed imaging at least 10-15 minutes later. A T1-weighted inversion recovery (IR) sequence is used to null the signal from normal myocardium (Zhuang, 2018).

T2-weighted CMR is mostly used to visualize myocardial edema (Fig. 1). T2-weighted cardiac MRI of edema is acquired by combining different imaging techniques which are used to freeze the cardiac and respiratory motions giving high contrast among blood, fat, normal myocardium, and myocardial edema. T2-weighted sequence provides a complementary information to LGE (Amano et al., 2012).

The bSSFP cine sequence captures cardiac motion and has clear myocardial boundaries (Fig. 1). It is a modification of gradient echo imaging which consists of a train of rapidly acquired RF-pulses with echoes formed by balanced imaging gradients. It is relatively fast acquisition which produces bright blood images with excellent contrast between myocardium and blood pool. It has also high temporal resolution (Norton, 2013).

Cardiac image segmentation involves the delineation of left ventricular myocardium, blood pool, right ventricle in addition to scar, edema and no-reflow segmentation. Most of the researches conducted in cardiac images segmentation uses single modality input image. For left ventricular myocardium and blood pool segmentation the most common modalities are bSSFP CMR, LGE and T1-map (Fahmy et al., 2019; Isensee et al., 2017; Kurzendorfer et al., 2018). While for scar segmentation, majority of them use LGE (Amado et al., 2004; Dikici et al., 2004; Moccia et al., 2019; Positano et al., 2005; de la Rosa et al., 2019; Zabihollahy et al., 2018).

Myocardial scar is often segmented manually in a clinical routine. However, manual segmentation is very exhausting and suffers from intra- and inter-observer variability. This problem can be addressed by developing an automatic segmentation method. Having said that, automatic segmentation also comes with its own challenges. Large shape and size variation of the heart and infarcted myocardium, heterogeneous intensity distributions of myocardium, motion artifact, low contrast between scar and blood pool in LGE as well as low contrast between edema and healthy myocardium in T2 make developing automatic segmentation methods difficult.

In this paper, we proposed a fully-automatic scar segmentation method using a cascaded Fully Convolutional-Densenet (FC-Densenet) (Jégou et al., 2017) with Inception (Szegedy et al., 2016) and Squeeze-Excitation (SE) modules (Hu et al., 2018). The input to our method was a multi-modal image which consists of LGE, T2 and bSSFP CMR sequences.

Our work has the following main contributions: 1) We proposed three cascaded segmentation networks that extract the region of interest then segment myocardium and finally segment scar from pre-segmented myocardial region. This resulted in higher Dice score and lower false positives compared to the one that uses 2 cascaded networks. 2) We demonstrated that combining multiple cardiac MR images can improve the cardiac segmentation result. 3) We showed that incorporating SE blocks and inception module to FC-Densenet improves the segmentation performance with little computational overhead. SE blocks are incorporated in the skip connections of the network to transfer a recalibrated feature maps from encoder to decoder and inception module is added at the beginning of the network to extract multi-scale features from the input image. 4) We proposed a novel loss function that combines the conventional logarithmic Dice loss with region mutual information (RMI) loss (Zhao et al., 2019). This objective function can be useful to segment small structures and pixels with weak visual evidence such as myocardial scar and edema. 5) Our fully-automatic approach showed a promising result on Multi-sequence CMR based Myocardial Pathology Segmentation (MyoPS 2020) challenge dataset by achieving a higher Dice score for scar than the inter-observer variation of manual scar segmentation.

2. State of the art

2.1. Left Ventricular Blood Pool and Myocardium Segmentation

Segmenting blood pool and myocardium is important to accurately identify the extent of infarcted tissue and quantify it. Several deep learning methods have been proposed to segment ventricles and myocardium. Some of these studies used only cine MR sequence. For instance, Isensee et al. (2017) tackled the segmentation problem using an ensemble of 2D and 3D Unet trained on cine MR dataset called Automated Cardiac Diagnosis Challenge (ACDC) dataset [1]. Using the same dataset, Khened et al. (2019) used 2D Dense-Unet with inception module to aggregate the features extracted

[1] https://www.creatis.insa-lyon.fr/Challenge/acdc/

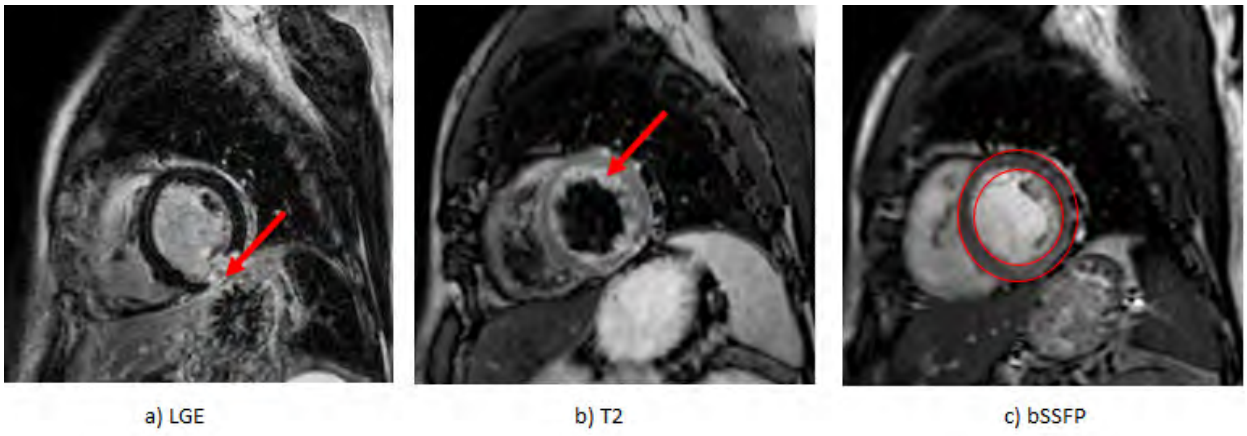


Figure 1: Illustrative example showing the three cardiac MR sequences. The red arrow in (a) LGE CMR and (b) T2 CMR indicate the presence of scar and edema respectively. The red circles in (c) bSSFP CMR shows myocardial boundaries.

by different kernel sizes from the input image for robust segmentation in images with variable sized heart shapes. To localize region of interest (ROI), Khened et al. (2019) used basic image processing techniques like Circular Hough Transform and Fourier analysis. While a method proposed by Li et al. (2019) used 2D FCN for ROI localization and another 2D FCN for segmentation which is a multi-stage network. These methods achieved very high segmentation accuracy on ventricles and myocardium. Despite the fact that bSSFP CMR has clear myocardial boundaries, it is difficult to get myocardial pathology information from it unlike LGE CMR.

Using LGE CMR, Kurzendorfer et al. (2019) implemented a multiscale fully convolutional neural network with residual units and weighted cross-entropy loss function to segment the left ventricles's endocardium and epicardium. As a post-processing technique, they selected the largest connected component and estimated a convex hull for the component in order to remove small wholes. The myocardium segmentation result, however, was relatively lower than the one that used cine CMR. To take advantage of cine MRI, Wei et al. (2011) and Tao et al. (2015) proposed to segment first cine CMR images and then propagated the obtained contours to LGE MRI through image registration. However, these methods require accurate registration between cine CMR and LGE CMR which can be challenging due to variation in image contrast and imaging field-of-view between them.

There are also other methods which incorporate shape prior and spatial prior. Oktay et al. (2018) proposed a method to segment cardiac MR images using anatomically constrained neural networks (ACNN). ACNN is a Unet based segmentation model which incorporates shape prior as regularization term. An autoencoder is trained first using ground truth images. Then loss is calculated as a distance between the latent space features generated from ground truth and predicted image. Similarly, Yue et al. (2019) proposed a deep learning method which incorporates shape and spatial priors. The main segmentation network was similar to Unet. The network included shape prior by adding a pretrained shape reconstruction neural network as a constraint to regularize a segmentation result into plausible shape. While the spatial constraint module is added to bottleneck of segmentation network to predict the position of LGE MRI slice. This spatial prior is added as a penalization of wrongly predicted spatial positions. Zotti et al. (2018)'s approach embedded cardiac shape prior by concatenating the shape prior probability map to the feature map located before the last convolution layer of the segmentation model. These approaches improved myocardium segmentation result. However, the addition of shape prior is computationally expensive because the approaches require separately training autoencoders and the segmentation networks.

2.2. *Infarcted Myocardium Segmentation*

Most scar segmentation studies can be categorized into two main groups: non-deep learning based and deep learning based methods. The non-deep learning based approaches are mainly focused on thresholding and clustering. The threshold based approaches exploit the enhanced intensity of the infarcted myocardium compared with the healthy myocardium. The threshodling method proposed by Amado et al. (2004) is called full width at half maximum (FHFW). As its name suggested, the threshold value is defined as the half value of the infarcted myocardium's maximum intensity. Kim et al. (1999)'s method defined the threshold as an intensity value n standard deviations higher than the mean intensity of the healthy myocardium (nSD) where n can be between 2 and 6. Both methods were simple, however, they required manual interaction of a user to determine region of interest that defines the

threshold values. In clustering based approach, Hennemuth et al. (2012) utilized a Gaussian mixture model analysis of the myocardial intensities and used the intensity threshold information for watershed segmentation. Baron et al. (2013) adopted Fuzzy C-means Clustering to segment scar by classifying the cluster probability of myocardial intensities using fuzzy inference.

Recently, few studies have been proposed to segment scar using semi-automatic and fully-automatic deep learning methods. Zabihollahy et al. (2018) used manual segmentation for myocardium and then 2D Fully Convolutional Network to segment scar from the myocardium. Moccia et al. (2019) proposed semi-automatic and fully-automatic scar segmentation method. Their semi-automatic approach, which manually segments the myocardial region, performed better than the one that uses automatic approach due to the mediocre segmentation performance of the network on myocardium. Another fully-automatic approach by de la Rosa et al. (2019) used a 2D Unet based myocardium segmentation followed by a top-hat transforms based coarse scar segmentation and finally a voxel classification of healthy and infarcted myocardium. However, using morphological operation to segment a scar can be unreliable particularly when the images have heterogeneous intensity distribution and motion artifact.

3. Material and methods

3.1. Dataset

The dataset used in this paper was Multi-sequence CMR based Myocardial Pathology Segmentation Challenge (MyoPS 2020)[2]. It is part of Statistical Atlases and Computational Modeling of the Heart (STACOM) 2020 workshop and Medical Image Computing and Computer Assisted Intervention (MICCAI) 2020. The dataset consists of three sequence CMR of 45 subjects diagnosed with myocardial infarction. From the 45 subjects, 25 of them are used for training and the rest for testing. The sequences are LGE CMR, T2-weighted CMR and bSSFP cine sequence. LGE CMR is a T1-weighted, inversion-recovery, gradient-echo sequence. The bSSFP CMR is a balanced steady-state, free precession cine sequence and T2 CMR is a T2-weighted, black blood Spectral Presaturation Attenuated Inversion-Recovery (SPAIR) sequence. The three sequences were breath-hold and scanned at end-diastolic phase. They were also acquired in the ventricular short-axis views. The typical parameters of the three sequences are summarized in Table 1.

According to the organizers of the challenge, the dataset was manually annotated by three independent observers and the final ground truth was achieved by averaging the three manual delineations using shape based approach (Zhuang, 2016, 2018). In addition, they registered the three CMR sequeces into a common space and an average spatial resolution of 0.75 x 0.75mm using multivariate mixture model (MvMM) method (Zhuang, 2018). MvMM is a method proposed by Zhuang (2018) for simultaneous registration and segmentation of multi-source images.

All images have annotation for right ventricle, left ventricle, myocardium, scar, edema and scar+edema. According to the organizers of the challenge, scar+edema is the infarcted myocardium which considers scar and edema as one class. For this task, we focused on segmentation of all except right ventricle because right ventricle does not have enough pathological information about scar compared to left ventricle and myocardium.

As a pre-processing step, the intensity of every patient image is normalized to have zero-mean and unit-variance. The dataset is already registered, as mentioned before. However, there are slight variations of spatial resolution among the patients (0.72 - 0.76 mm). To account for this, all patients were re-sliced to have the same spatial resolution of 1.0×1.0 mm. The z spacing of the voxel spacing is not changed.

3.2. Proposed Pipeline

The proposed pipeline consists of data pre-processing and deep learning based region of interest extraction, myocardium and scar segmentation (Fig. 2). In our approach, a cascaded segmentation network consisting of three FC-Densenet with Inception and Squeeze-Excitation module (Cascaded FCDISE) were used to extract the region of interest and then segment myocardium and finally segment scar from the pre-segmented myocardial region. The segmentation network architecture used for the three tasks are almost the same. The only differences are the number of pooling/upsampling layers and their weights as they are trained independently. The segmentation network is based on 2D convolution operations.

3.2.1. Network Architecture

The proposed method is based on FC-Densenet (Jégou et al., 2017). To enhance FC-Densenet's performance, we incorporated two important modules: SE blocks and inception module. We named the proposed segmentation network FCDISE.

One of the problems with very deep neural networks is vanishing gradient. To alleviate this problem, many novel architectures has been proposed. Densely Connected Convolutional Network (DenseNet), which is proposed by Huang et al. (2017), is one of them. This network was designed to address the problem of vanishing gradient by directly connecting each layer to every

[2] http://www.sdspeople.fudan.edu.cn/zhuangxiahai/0/myops20/

Table 1: MRI parameter setting for bSSFP, LGE and T2 CMR sequences

Parameter	**bSSFP**	**LGE**	**T2**
TR/TE	2.7/1.4 ms	3.6/1.8 ms	2000/90 ms
Slice Thickness	8-13 mm	5 mm	12 - 20 mm
In-plane resolution	1.25 x 1.25 mm	0.75 x 0.75 mm	1.35 x 1.35 mm

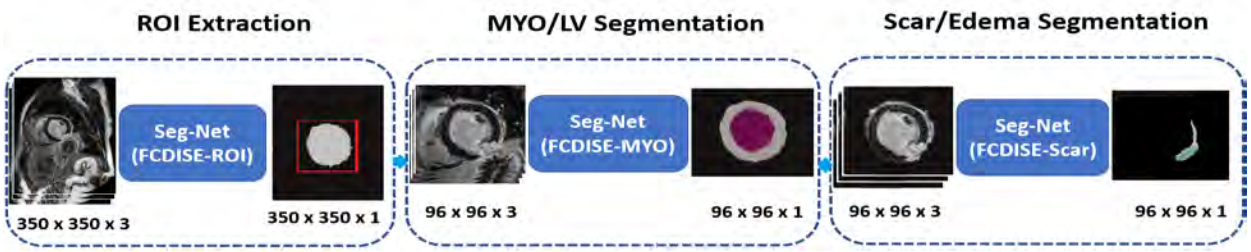


Figure 2: Proposed pipeline. **FCDISE-ROI**: segmentation network used for ROI extraction, **FCDISE-MYO**: segmentation network used for myocardium segmentation, **FCDISE-Scar**: segmentation network used for scar segmentation.

other layer in a feed-forward fashion. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. Unlike residual neural networks (ResNets), the feature maps received from previous layers are concatenated not summed. Densenet also has other advantages like strong feature propagation, feature reuse and reduced number of parameters (Huang et al., 2017).

Jégou et al. (2017) extended Densenets to deal with semantic segmentation task. Densenets are good fit for semantic segmentation because they have skip connections and multi-scale supervision by design. But directly extending Densenet as Fully Convolutional Network (FCN) will lead to feature map explosion in the decoder part. To mitigate this problem, only the features maps created by the preceding dense block are upsampled. Like FCN, skip connections are used to transfer the higher resolution information from encoder to decoder (Jegou et al., 2017).

Similar to FC-Densenet, our network architecture consists of downsampling path, upsampling path and skip connections. The downsampling path is composed of dense blocks and transition down layers as shown in Fig. 3. The upsampling path also has dense blocks and transition up layers. In the dense block, each layer receives feature maps from all preceding layers and forwards its feature map to all subsequent layers as shown in Fig. 4 (a). Each dense block layers are made up of Batch Normalization, rectified linear unit (ReLU) activation function, 3×3 convolution and drop out with probability 0.2 as shown in Fig. 4 (b). Transition down is composed of Batch Normalization, ReLU activation function, 1×1 convolution, drop out with probability 0.2 and 2×2 max-pooling layer with stride 2 to downsample the feature maps into latent space (Fig. 6 (a)). From the latent space, transition up recovers the input spatial resolution by upsampling the feature maps using 3×3 transposed convolution with stride 2 (Fig. 6 (b)). Skip connections are used to concatenate the feature maps from downsampling path to the corresponding feature maps in the upsampling path.

SE block is used to model channel relationships and channel inter-dependencies. It can also be regarded as self-attention on the channels. SE block consists of global average pooling and fully connected (FC) layers. It has "squeeze" and "excitation" step. In "squeeze" step, it squeezes global spatial information into channel descriptor (channel-wise statistics) using global average pooling across the spatial dimension. The "excitation" step is intended to fully capture channel-wise dependencies. It maps the output of "squeeze" step to a set of channel weights using two FC layers and channel-wise scaling (Hu et al., 2018).

There are different varieties of SE block. The most common ones are SE-Inception module and SE-ResNet module. In the latter one, the SE block's output is taken to be the non-identity branch of a residual module as depicted in Fig. 5 (a). For our model, SE-ResNet module was used.

SE block can be integrated into network architecture in several ways. We decided to incorporate SE blocks only in the skip connections after experimenting SE block usage at different positions of the network. As shown in Fig. 3, the SE module in our network receives the feature maps from encoder and then recalibrates the feature maps before concatenating them to the corresponding feature maps in the decoder. The module helps the decoder to receive refined feature maps.

The second module integrated to FC-densenet is inception module (Szegedy et al., 2016). Inspired by Khened et al. (2019), the inception module is incorpo-

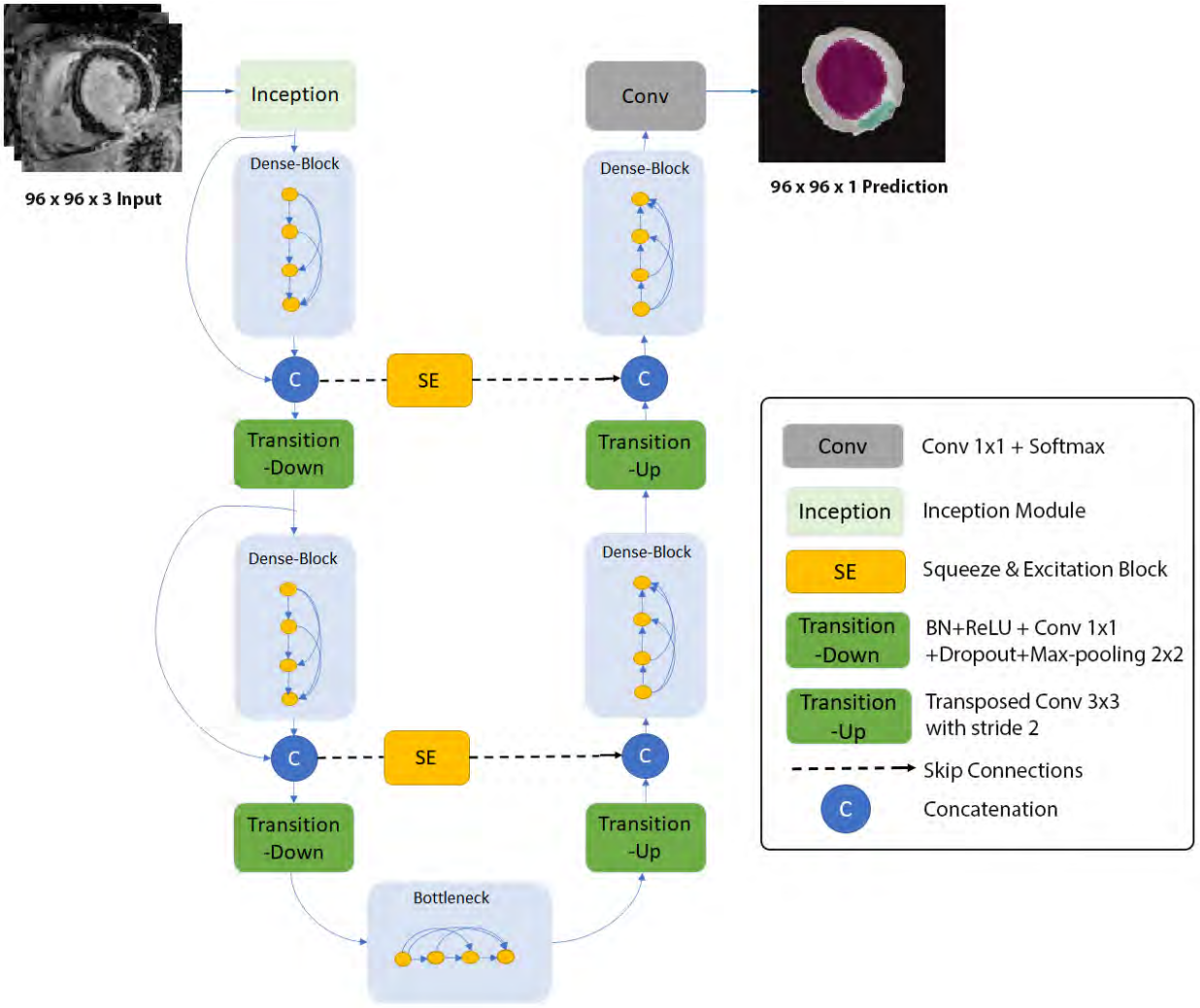


Figure 3: Proposed network architecture (FCDISE)

rated at the beginning of the network. Inception module similar to Densenets were introduced to mitigate the vanishing gradient problem of bigger and wider networks. Szegedy et al. (2016) proposed a new approach of creating deep networks which keeps the computational budget constant using a sparsely connected network architecture.

Inception module has two common versions. The naive version which is composed of multiple parallel layers such as 1x1 Convolutional layer, 3x3 Convolutional layer and 5x5 Convolutional layer with their output filter maps concatenated into a single output vector. The second version introduces a dimensionality reduction layer, 1x1 convolutional layer, before applying any other layer. As can be seen from Fig. 5 (b), the inception module in our network is a bit modified from the naive inception layer as it contains only three kernels (3x3, 5x5 and 7x7 kernels) and their output is summed instead of concatenated because summation yielded better results.

The reason we used inception module as first layer of the network is to extract multi-scale features simultaneously from the input image using different sized kernels and to send the aggregated features to the next layer of the network. This is helpful because heart size varies from one patient to another and even in one patient there is variation of size from apex to base. The different sized kernels in the inception help to capture the relevant features from the input image irrespective of the size of the heart.

The segmentation network used to detect ROI is called FCDISE-ROI. It has 5 pooling layers. For myocardium and scar segmentation, we employed FCDISE-MYO and FCDISE-Scar respectively. Both of them have 3 pooling layers.

3.2.2. Region of Interest (ROI) Detection

The first stage in the proposed pipeline is ROI extraction. In the full size cardiac MR images, the heart covers very small part of the image. Due to this, it is necessary

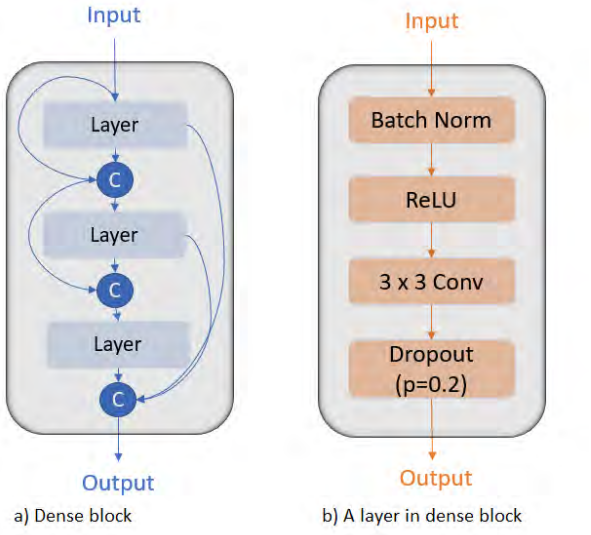


Figure 4: Diagram of (a) dense block and (b) a layer in dense block used in our proposed model

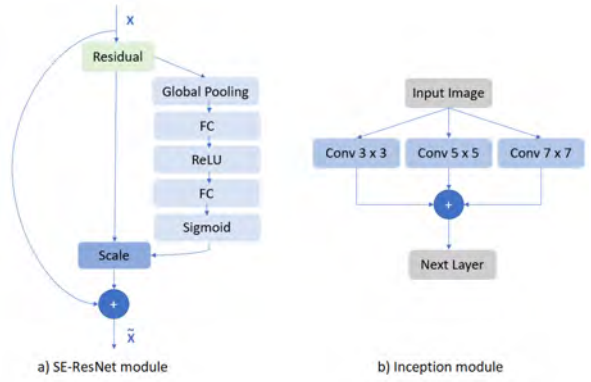


Figure 5: Diagram of (a) Squeeze-Excitation block and (b) Inception module employed in our model

to extract a region of interest around the ventricles before proceeding to the next stages in the pipeline. Our ROI extraction method is done by first segmenting the epicardial region from the full-size cardiac MR using FCDISE-ROI. Then the center of the segmented epicardial region is calculated. Finally, we applied center cropping from the computed center of epicardial region with a patch size of 96×96. This particular size is chosen after taking into consideration the largest diameter of epicardium from the training set images.

This method places the ventricles in the center of the cropped region. This has three advantages for the next stages in the pipeline. It reduces the false positives and alleviates the class imbalance between the background and the ventricles/scar classes. Furthermore, it decreases the computation time as the size of input images are decreasing.

3.2.3. Myocardium and Left Ventricle Segmentation

The second stage in the proposed pipeline is myocardium and left ventricular blood pool segmentation. The inputs to FCDISE-MYO are output of ROI detection stage which are 2D slices of size 96 x 96. When we used input size 96 x 96 with our segmentation network which has 5 pooling layers, the latent space feature map size becomes very small which makes reconstruction of the segmentation map difficult. To avoid this problem, we reduced the number of pooling layers in the network from 5 to 3. That is why FCDISE-MYO has 3 pooling layers.

3.2.4. Scar Segmentation

Scar segmentation stage is very similar to myocardium segmentation stage except for the input image. The input image here contains only the pre-segmented epicardial region, the region which includes left ventricular blood pool and myocardium. As myocardium segmentation may not be perfect, we also included the surrounding area near the epicardium border by applying dilation on the pre-segmented epicardial region with a rectangular structuring element of size 5 x 5. The input image size is 96 x 96 but contains only background pixels and the pre-segmented region. The segmentation network used in this stage is FCDISE-Scar, which is similar to the previous stage's segmentation network.

As a post-processing step, we applied 2D connected component analysis and morphological operations like dilation and erosion to the segmented image to further improve the segmentation result and reduce outliers.

3.2.5. Loss Function

As an objective function, we proposed a dual loss function which is a weighted combination of logarithmic Dice loss (Wong et al., 2018) and region mutual information (RMI) loss (Zhao et al., 2019).

Logarithmic Dice loss (log Dice loss) is known for its robust performance on small structures (Wong et al., 2018). Compared to linear Dice loss, it focuses more on less accurate classes. Log Dice loss is computed as the mean value of the natural logarithm of the Dice coefficient as stated in Eq. 1. It also introduces an exponent γ that controls the non-linearity of the loss function. When $\gamma > 1$, the log Dice loss focuses even more on the less accurate classes. If the non-linearity is $0 < \gamma < 1$, the loss works better because it supports improvement at both low and high accuracy. To improve the segmentation of small structures like scar and edema, we chose a logarithmic Dice loss.

The second loss function used is region mutual information loss. Unlike pixel-wise loss, RMI loss takes into account the dependencies among the pixels. Each pixel in an image is represented by the pixel itself and its neighbouring pixels. In other words, the pixel will be

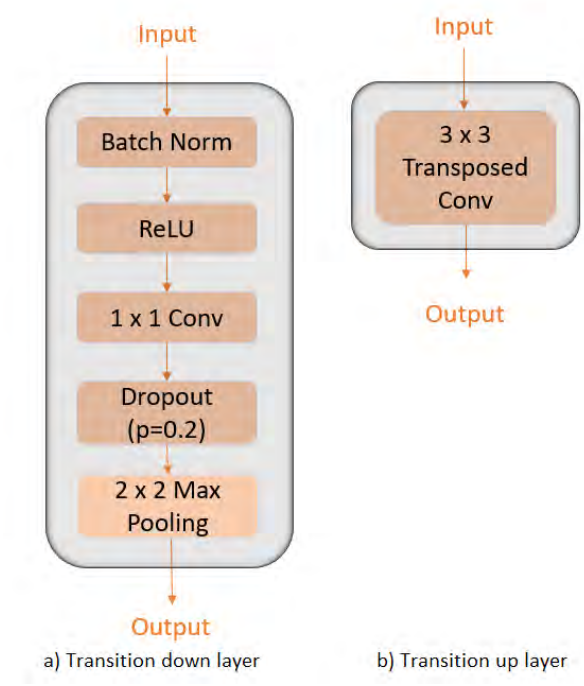


Figure 6: Diagram of (a) transition down layer and (b) transition up layer used in our proposed model

represented by multi-dimensional point and the image will be a multi-dimensional distribution of these points. Maximizing the mutual information between the multi-dimensional distributions of the ground truth and predicted image will result in high order consistency between these two images. One of the metrics used for image similarity is mutual information. However, analysis of multi-dimensional distribution of an image is difficult due to pixels dependence. This makes calculation of mutual information complex. Instead, the authors used a lower bound of MI because increasing this value results in increasing the real value of mutual information (Zhao et al., 2019).

This loss function captures the structural differences between the shapes of predictions and ground truth. It is also helpful in identifying pixel whose visual evidence is weak or when the pixel belongs to objects with small spatial structures (Zhao et al., 2019). This makes it ideal for myocardium and scar segmentation.

From Eq. 2, Y is multi-dimensional distribution of the ground truth and P is multi-dimensional distribution of the predicted image. $\Sigma_{Y|P}$ is the posterior covariance matrix of Y given P and $det()$ is determinant of the matrix. $I(Y;P)$ is a lower bound of the mutual information. Then the total RMI loss is computed as a combination of the pixel-wise cross entropy loss (L_{CE}) and lower bound MI as stated in Eq. 3. In this equation, B and C represent mini-batch size and number of classes respectively.

To take advantage of both log Dice loss and RMI loss, we used a weighted combination of these two losses as our objective function as stated in Eq. 4, where λ_{Dice} and λ_{RMI} are the weighting factors for log Dice loss (L_{Dice}) and RMI loss (L_{RMI}) respectively.

$$L_{Dice} = E[(-ln(Dice_i)^{\gamma}] \tag{1}$$

$$I(Y;P) = -\frac{1}{2}log((2\pi\epsilon)^{d}det(\Sigma_{Y|P})) \tag{2}$$

$$L_{RMI} = L_{CE} + \frac{1}{B}\sum_{b=1}^{B}\sum_{c=1}^{C}(-I(Y;P)) \tag{3}$$

$$L_{Total} = \lambda_{Dice}L_{Dice} + \lambda_{RMI}L_{RMI} \tag{4}$$

3.2.6. Training

The three segmentation networks in the pipeline are trained independently. The weights are initialized using *He normal* initialization method (He et al., 2015). The optimization of the weights are done using Adam optimizer with learning rate of 0.001. The mini-batch size was 16. The model was trained for 80 epochs. We empirically selected a weighting factor of 0.8 for log Dice loss and 0.2 for RMI loss after experimenting with different weighting factors. For log Dice loss, a non-linearity of 0.3 was used. The frameworks used to implement the model and the code are PyTorch and Python.

In order to avoid over-fitting, we have adopted three techniques: dropout, early stopping and weight regularization. Dropout is a regularization technique where randomly selected neurons are dropped during training. The ignored neurons will not have contribution during a forward and backward pass. Dropout reduces overfitting by preventing complex co-adaptations on training data. In our model, we used a dropout with probability of 0.2.

When training the network, usually in the beginning the training and validation loss decreases. After some epochs, the training loss will still decrease but the validation loss will eventually go up. This will result in overfitting. To monitor this, we used early stopping. So when the validation loss starts increasing the training will be stopped after particular number of epochs (patience). In our experiments, the patience for the early stopping was 10 epochs.

The third technique is weight regularization. This updates the cost function by adding a regularization term. In case of L2 regularization (weight decay), it is the sum of the square of the weights. This regularization forces the weights to decay towards zero but not zero. The regularization term has a hyper-parameter called *lambda* which controls the relative contribution of the regularization term to the cost function. We used L2 regularization with *lambda* hyper-parameter set to $1e-8$.

4. Results

To evaluate the segmentation results, we used Dice coefficient, Hausdorff distance (HD), sensitivity and

specificity metrics. Dice coefficient measures the similarity of two images. It is calculated as the size of the overlap between segmented image and ground truth divided by the total size of the two images as defined in Eq. 5. In this equation, Y and P represent the ground truth image and the predicted image respectively. This measures the overall quality of a segmentation. Sensitivity measures the percentage of pixels of pathology area that are correctly segmented as pathology (Eq. 6). While specificity measures the percentage of pixels of non-pathology area that are truly segmented as non-pathology (Eq. 7). In Eq. 6 and 7, TP is true positive pixels, FN is false negative pixels, TN is true negative pixels and FP is false positive pixels. Dice coefficient, sensitivity and specificity are used to evaluate both scar and myocardium segmentation results.

Hausdorff distance is the greatest of all distances from a point in one set to the closest point in the other set. This metrics focuses on outliers. Hausdorff distance metric (2D) is used to evaluate myocardium segmentation result. Calculating Hausdorff distance for scar and edema can be difficult because they are dispersed regions.

$$Dice = \frac{2|Y \cap P|}{|Y| + |P|} \tag{5}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

To evaluate our models, we employed a five fold cross-validation as well as train-validation-test evaluation methods. For the latter method, from a total of 25 subjects, 17 were used for training, 3 for validation and 5 for test.

4.1. Myocardium and Left Ventricle Segmentation

The proposed method yielded a Dice score of 0.872 and Hausdorff distance (2D) of 3.392 mm on myocardium (MYO) segmentation and a Dice score of 0.921 and Hausdorff distance (2D) of 2.577 mm on left ventricle (LV) segmentation (Table 2).

Table 2: Myocardium and left ventricle segmentation result of the proposed method

Metrics	**LV**	**MYO**
Dice	0.921 ± 0.041	0.872 ± 0.041
HD(mm)	2.577 ± 0.578	3.392. ± 0.514
Specificity	0.978 ± 0.016	0.949 ± 0.022
Sensitivity	0.939 ± 0.058	0.876 ± 0.058
Accuracy	0.971 ± 0.013	0.931 ± 0.019

The inter-observer variation of manual segmentation of MYO are Dice scores of 0.757, 0.824 and 0.812 for LGE, T2 and bSSFP respectively. Comparing to our model's performance on each CMR separately, our method yielded Dice scores of 0.771, 0.798 and 0.854 for MYO using LGE, T2 and bSSFP sequences respectively. This result was on average better than the inter-observer variation. Besides, the Dice score of MYO increased to 0.872 when we combined the three modalities as an input to our method.

To evaluate the effect ROI in our pipeline, we compared the results with and without ROI. When we directly segment heart from the full-sized cardiac MR, our method yielded Dice scores of 0.905 and 0.853 for LV and MYO respectively. However, when we employed ROI, our method achieved an improved Dice score of 0.921 for LV and 0.872 for MYO. Moreover, the obtained Hausdorff distance was on average 0.22 mm lower than the one that did not use ROI.

Table 3 quantitatively compares the proposed loss with the conventional loss functions such as cross-entropy loss, Dice loss, logarithmic Dice loss. The proposed loss outperformed the other loss functions by achieving the highest Dice score in both LV and MYO. To better investigate the qualitative performance of the loss functions, we selected a typically challenging image which has scar tissue, as depicted in Fig. 7. The proposed loss produced robust segmentation result.

Table 3: Quantitative comparison of loss functions using Dice score

Loss Function	**LV (Dice)**	**MYO (Dice)**
Cross-entropy	0.905 ± 0.067	0.849 ± 0.086
Dice Loss	0.903 ± 0.073	0.858 ± 0.067
Log Dice Loss	0.909 ± 0.056	0.865 ± 0.054
Proposed Loss	**0.921 ± 0.041**	**0.872 ± 0.041**

4.2. Scar Segmentation

The performance of the proposed method in scar, edema and scar+edema segmentation is presented in Table 4. Note that scar+edema considers scar and edema as one class. Having one class can be helpful to evaluate the model's performance on detecting the infarcted myocardium in general instead of dividing the infarcted region into scar and edema. Our method performed well on infarcted myocardium (scar+edema) segmentation. However, our model's performance decreased a little bit when separately segmenting scar and edema.

Similar to myocardium segmentation, we studied the effect of using single modal CMR and multi-modal CMR as shown in Fig. 8. Comparing the three modalities, using only LGE CMR achieved the best Dice score

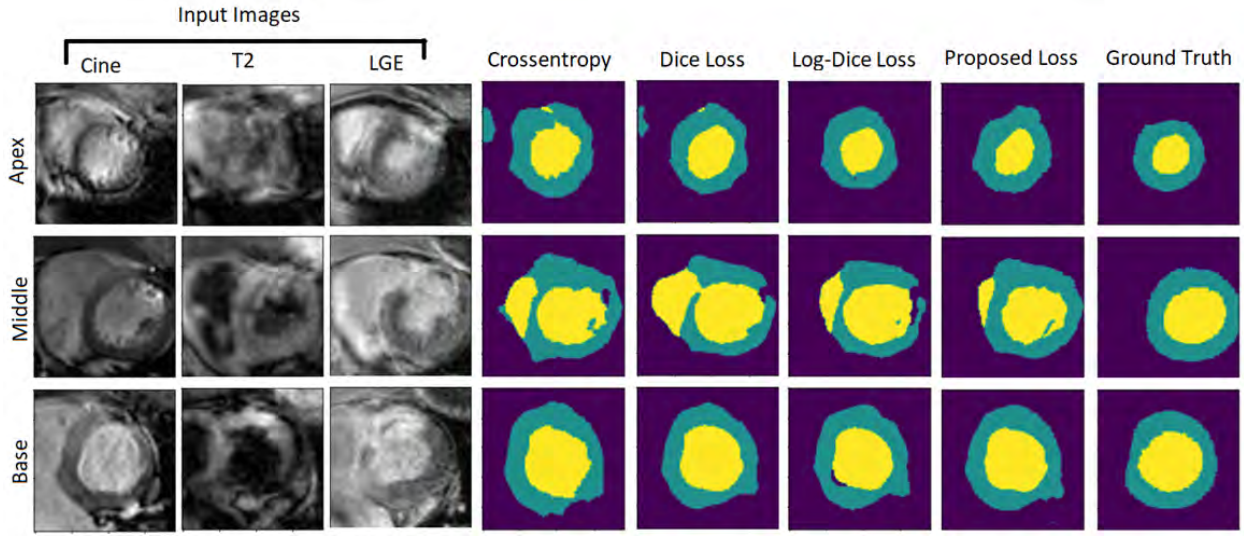


Figure 7: Qualitative comparison of loss functions on a typically challenging image. Note that the results are before post-processing. Myocardium (green) and left ventricle (yellow).

for scar (0.603) whereas using only T2 CMR yielded the best result for scar+edema (0.644). When we combined the three modalities, the Dice score of scar slightly increased to 0.604 while that of scar+edema significantly increased to 0.687.

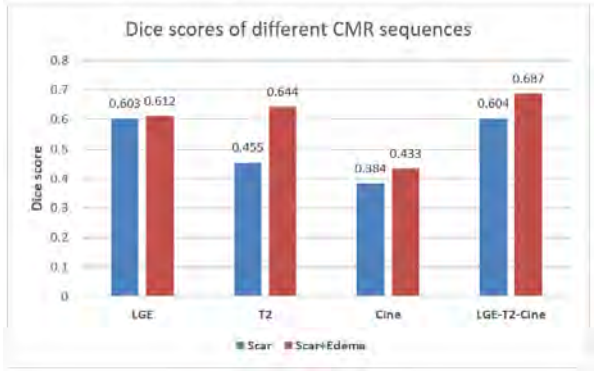


Figure 8: Comparison of different cardiac MR sequences performance on scar and scar+edema segmentation. Where LGE is late gadolinium enhancement cardiac MR and T2 is T2-weighted cardiac MR. Cine is bSSFP cine sequence and LGE-T2-Cine is multi-modal image consisting of LGE, T2 and bSSFP sequences.

Comparing the performance of the loss functions on the segmentation of scar and edema, the proposed loss outperformed the conventional loss functions. Crossentropy loss yielded Dice scores of 0.527 for scar and 0.567 for scar+edema whereas Dice loss achieved Dice scores of 0.543 for scar and 0.575 for scar+edema. Log Dice loss, compared to the first two losses, provided better result for both scar (0.588) and scar+edema (0.606). When we combined RMI loss with log Dice loss, the segmentation result of scar increased a little bit to 0.604 while the improvement for scar+edema was substantial as it enhanced the Dice score from 0.606 to 0.687.

4.2.1. Ablation Study

To evaluate the effect of addition of inception and SE module to FC-Densenet, we compared the proposed method with FC-Densenet and FC-Densenet with only SE module (FCDensenet_SE). As presented in Table 5, the baseline model (FC-Densenet) achieved comparable result in scar but failed in Edema. Adding SE blocks to the baseline substantially improved the segmentation accuracy (Dice score) for scar+edema by nearly 10%. While the proposed method, which adds both SE block and inception module to the baseline, improved the Dice value for scar+edema achieving a 14% increase compared to the baseline. The improvement is also demonstrated in the qualitative result as can be seen from Fig. 9. Comparing their distribution in Fig. 10, the proposed method has on average the lowest Dice variance among the three methods.

4.2.2. Comparison with Alternative Methods

We compared our proposed method with three other methods which employed the same pipeline that is a cascaded three networks. The segmentation networks used in place of FCDISE are Unet (Ronneberger et al., 2015), Attention-Unet (Oktay et al., 2018) and Res-Unet. Unet is one of the most commonly used segmentation networks for medical images. Attention-Unet is a standard Unet with attention gate which recalibrate feature maps spatially. Res-Unet is also a Unet with residual encoder and decoder.

The comparison was done both qualitatively and quantitatively, as shown in Fig. 9 and Table 5 respectively. Unet has good result on scar but its performance decreased on edema. While Res-Unet did not perform well on both scar and edema. Observing their distribution on Fig. 10, Unet and Attention-Unet have sim-

Table 4: Scar, edema and scar+edema segmentation result of the proposed method

Metrics	**Scar**	**Edema**	**Scar+Edema**
Dice	0.604 ± 0.167	0.488 ± 0.172	0.687 ± 0.072
Specificity	0.977 ± 0.092	0.967 ± 0.112	0.962 ± 0.081
Sensitivity	0.627 ± 0.128	0.457 ± 0.125	0.739 ± 0.094
Accuracy	0.959 ± 0.093	0.946 ± 0.113	0.941 ± 0.098

Table 5: Dice score comparison of various methods for scar and scar+edema segmentation

Methods	**Scar (Dice)**	**Scar+Edema (Dice)**	**No of Parameters**
Unet	0.577 ± 0.095	0.558 ± 0.131	0.84 million
Attention-Unet	0.566 ± 0.144	0.610 ± 0.118	2.6 million
Res-Unet	0.535 ± 0.176	0.560 ± 0.284	6.7 million
FCDensenet	0.579 ± 0.148	0.540 ± 0.229	0.65 million
FCDensenet_SE	0.584 ± 0.181	0.640 ± 0.134	0.68 million
Proposed method	**0.604 ± 0.167**	**0.687 ± 0.072**	0.69 million

ilar or lower variance in comparison with the proposed method, while Res-Unet has the highest Dice score variance.

As shown in Table 5, we also compared the number of trainable parameters. The ones that use dense blocks have the lowest number of parameters. FC-Densenet has the fewest number of parameters which is 0.65 million. The proposed method has 0.69 million parameters. While Res-Unet has the highest number of parameters which is 6.7 million.

5. Discussion

In this paper, we evaluated our proposed pipeline and segmentation network on multi-sequence cardiac MR dataset which has LGE, T2 and bSFFP CMR images. The MyoPS 2020 challenge dataset is very small and has poor quality (lots of motion artifact). This makes deep learning based segmentation difficult. To solve the problem, we proposed a method which cascaded three segmentation networks. The first one is aimed at a robust ROI detection that reduces the false positives as well as mitigates the class imbalance between the background and the ventricle classes. The second network is focused on accurate segmentation of myocardium which is important for the next stage in the pipeline. The third network segments scar from the pre-segmented myocardium. The segmentation network used is FC-Densenet with Inception and Squeeze-Excitation modules.

As mentioned in Section 2.2, one of the main problems for fully-automatic scar segmentation was the mediocre segmentation performance of the models on myocardium. There are many reasons for this. The main reason can be due to the fact that most of these studies used only LGE CMR. LGE CMR can visualize myocardial scar better than other CMR modalities but the intensity range of myocardium in LGE CMR overlaps with its surrounding resulting in blurry myocardial boundaries. This makes myocardium segmentation from LGE CMR a difficult task. The use of multi-sequence CMR which includes LGE, T2 and bSSFP CMR addresses this issue as bSSFP CMR has clear myocardial boundaries resulting in higher segmentation performance in myocardium.

Assessing the inter-observer variation of manual scar segmentation, there was low agreement between the observers for scar in terms of Dice score showing the difficulty of the task and the discrepancy to identify the infarcted myocardium. In spite of this, our method achieved a higher Dice score on scar than the inter-observer variation. There are many reasons for this. One of them can be the high segmentation accuracy of our method on myocardium and left ventricle which leads to better segmentation performance on scar.

Comparing the CMR modalities, the multi-modal input has better segmentation performance than single sequence inputs. The bSSFP CMR has accurate information about the ventricles and myocardium compared to the other two modalities. While LGE and T2 CMR

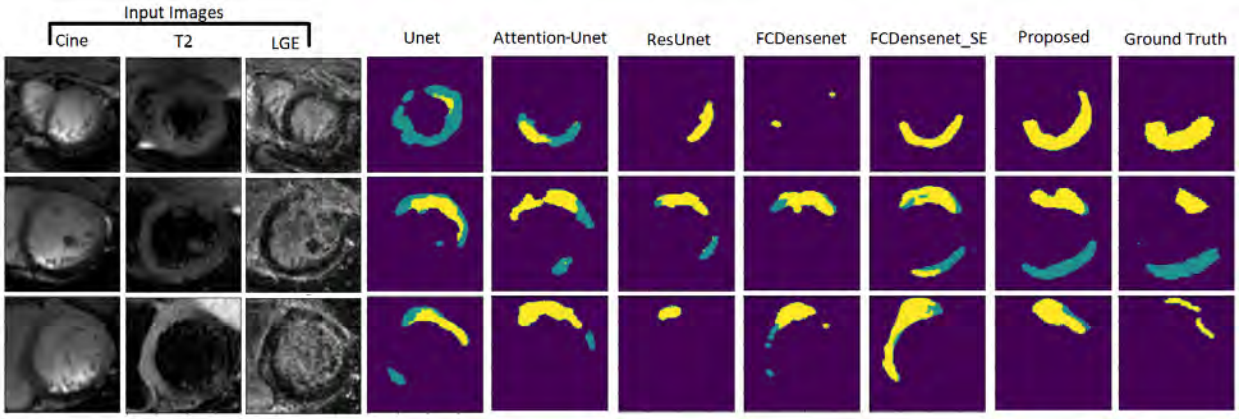


Figure 9: Qualitative comparison of different models on scar (yellow) and edema (green) segmentation.

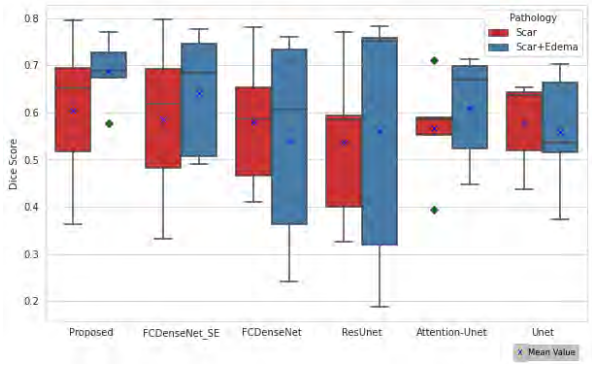


Figure 10: Dice score distributions of different models for scar and scar+edema segmentation.

have superior results on scar and edema respectively. The bSSFP sequence's segmentation performance on both scar and edema was inferior compared to the other two CMR sequences. This can be due to the fact that bSSFP CMR has less information about scar and edema. Combining the three CMR modalities improved the segmentation accuracy of the heart structures as well as scar/edema. This was expected as the three sequences (bSSFP, LGE and T2) have complementary information.

Regarding the use of ROI, the pipeline with ROI achieved better result in terms of Dice score and HD. This shows how extracting ROI can improve the result by reducing the false positives and mitigating the class imbalance problem between the background and ventricle classes. In addition, it decreased the training and inference time due to the significantly reduced input sizes.

Experiments comparing the proposed loss with the conventional loss functions indicated a superior segmentation performance of the proposed loss in both LV and MYO. The conventional loss functions particularly failed because they segmented the scar as blood pool instead of MYO (middle slice in Fig. 7). Here is when the addition of RMI loss becomes very handy. Because RMI loss takes into account the pixel dependencies unlike the pixel-wise losses. This helps the model to achieve high order consistency between the prediction and ground truth.

As in the case of MYO and LV segmentation, the proposed loss also outperformed the conventional loss functions in scar and edema segmentation. Log Dice loss's result on scar was good but its result on edema was mediocre. Similarly, the addition of RMI loss to log Dice loss improved the segmentation results particularly that of edema.

The proposed loss function's robust segmentation performance on scar/edema and myocardium verified the benefit of combining log dice loss with a loss function that considers the dependencies among the pixels. To the best of the authors' knowledge, this is the first time a region mutual information loss is being used in medical image segmentation. Due to its promising segmentation performance, it can be employed in the segmentation of medical organs whose pixels have weak visual evidence.

In the ablation study, the proposed model yielded higher Dice coefficient than FC-Densenet and FC-Densenet with only SE. From Fig. 9, it can be observed that the proposed method has comparatively better performance at detecting different sized scars. This showed the benefit of the extracted multi-scale features from the input image and confirmed the advantage of the incorporated SE block. Our method achieved this enhancement with minimal computational overhead.

When comparing FCDISE with other similar segmentation networks, Res-Unet had the worst segmentation performance (Table 5). This is because it overfitted on the small dataset (25 cases). Both Attention-Unet and FCDISE which use attention on feature maps achieved better result on scar+edema than the ones that did not use. This demonstrated the benefits of recalibrating feature maps spatially or channel-wise on helping the model to increase its focus on scar and edema. However, when Attention-Unet is compared to the pro-

posed method, our method achieved more accurate segmentation performance in both scar and scar+edema. Besides, the proposed method is robust at detecting scar at different heart positions and has less false positive cluster of scar compared to the other methods.

In Table 5, we also compared the number of trainable parameters. The ones that use dense blocks have the lowest number of parameters because Densenet encourages feature reuse which substantially reduces the number of parameters. Besides, our method is ideal on tasks with smaller training set sizes like MyoPS 2020 because the dense connections in the network have a regularizing effect which reduces overfitting.

The proposed framework has some limitations. Inaccuracies in the ROI stage or in the myocardium segmentation stage can adversely affect the segmentation accuracy of scar because our proposed pipeline uses a cascaded network to segment scar. Deep learning based ROI extraction increases the detection accuracy, however, it can slows down the segmentation speed compared to the one that employs conventional computer vision techniques like Circular Hough Transform.

6. Conclusions

In this paper, we proposed a deep learning based fully-automatic myocardial scar segmentation method from multi-sequence cardiac MR images. Our method employs three cascaded segmentation networks to first extract ROI then segment myocardium and finally use the pre-segmented myocardium to segment scar and edema. Each segmentation network used FC-Densenet with Inception and Squeeze-Excitation module (FCDISE). The SE blocks are incorporated in the skip connections and the inception module is added in the initial layer of the network to concatenate different field-of-views of image features. We demonstrated that adding these two modules to FC-Densenet substantially improves the segmentation result with little computational overhead. Compared to other similar networks, our method is better at locating different size scar and edema, and performs well on small training set. Furthermore, we showed that region mutual information loss combined with logarithmic Dice loss achieves high order consistency between the prediction and ground truth. It can also be of great interest for segmentation of medical organs whose pixels have weak visual evidence.

Despite having a very challenging dataset, our approach yielded very good result on the test set achieving an average Dice score of 0.590 for scar and 0.686 for scar+edema which is higher than the inter-observer variation of scar segmentation 0.524 (Dice score of scar). Future work will aim in using multi-planar network that will include sagittal, coronal and axial views to further improve the segmentation result.

Finally, this paper has been accepted for publication at Statistical Atlases and Computational Modeling of the Heart (STACOM) workshop which is part of Medical Image Computing and Computer Assisted Intervention (MICCAI) 2020.

7. Acknowledgments

T.W. Arega received an Erasmus+ scholarship from the Erasmus Mundus Joint Master Degree in Medical Imaging and Applications (MAIA), a program funded by the Erasmus+ program of the European Union. This work was also supported by the French National Research Agency (ANR), with reference ANR-19-CE45-0001-01-ACCECIT.

T.W. Arega would like to thank his supervisor Dr. Stéphanie Bricq for her support and guidance during the master thesis. He would like also to thank his friends Kibrom, Yeman, Abdullah and Zohaib as well as the whole MAIA family for sharing their knowledge and helping him to become a better person.

References

Amado, L.C., Gerber, B.L., Gupta, S.N., Rettmann, D.W., Szarf, G., Schock, R., Nasir, K., Kraitchman, D.L., Lima, J.A., 2004. Accurate and objective infarct sizing by contrast-enhanced magnetic resonance imaging in a canine myocardial infarction model. Journal of the American College of Cardiology 44, 2383–2389.

Amano, Y., Tachi, M., Tani, H., Mizuno, K., Kobayashi, Y., Kumita, S., 2012. T2-weighted cardiac magnetic resonance imaging of edema in myocardial diseases. The Scientific World Journal 2012.

Baron, N., Kachenoura, N., Cluzel, P., Frouin, F., Herment, A., Grenier, P., Montalescot, G., Beygui, F., 2013. Comparison of various methods for quantitative evaluation of myocardial infarct volume from magnetic resonance delayed enhancement data. International journal of cardiology 167, 739–744.

Belleza, M., 2017. Myocardial infarction: Nursing management and study guide. URL: `https://nurseslabs.com/myocardial-infarction/`.

Dikici, E., O'Donnell, T., Setser, R., White, R.D., 2004. Quantification of delayed enhancement mr images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 250–257.

Dongaonkar, R.M., Stewart, R.H., Quick, C.M., Uray, K.L., Cox Jr, C.S., Laine, G.A., 2012. Time course of myocardial interstitial edema resolution and associated left ventricular dysfunction. Microcirculation (New York, NY: 1994) 19, 714.

Fahmy, A.S., El-Rewaidy, H., Nezafat, M., Nakamori, S., Nezafat, R., 2019. Automated analysis of cardiovascular magnetic resonance myocardial native t 1 mapping images using fully convolutional neural networks. Journal of Cardiovascular Magnetic Resonance 21, 1–12.

Hammer-Hansen, S., Bandettini, W.P., Hsu, L.Y., Leung, S.W., Shanbhag, S., Mancini, C., Greve, A.M., Køber, L., Thune, J.J., Kellman, P., et al., 2016. Mechanisms for overestimating acute myocardial infarct size with gadolinium-enhanced cardiovascular magnetic resonance imaging in humans: a quantitative and kinetic study. European Heart Journal-Cardiovascular Imaging 17, 76–84.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, pp. 1026–1034.

Hennemuth, A., Friman, O., Huellebrand, M., Peitgen, H.O., 2012. Mixture-model-based segmentation of myocardial delayed enhancement mri, in: International Workshop on Statistical Atlases and Computational Models of the Heart, Springer. pp. 87–96.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.

Isensee, F., Jaeger, P.F., Full, P.M., Wolf, I., Engelhardt, S., Maier-Hein, K.H., 2017. Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features, in: International workshop on statistical atlases and computational models of the heart, Springer. pp. 120–129.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 11–19.

Khened, M., Kollerathu, V.A., Krishnamurthi, G., 2019. Fully convolutional multi-scale residual densenets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers. Medical image analysis 51, 21–45.

Kim, R.J., Fieno, D.S., Parrish, T.B., Harris, K., Chen, E.L., Simonetti, O., Bundy, J., Finn, J.P., Klocke, F.J., Judd, R.M., 1999. Relationship of mri delayed contrast enhancement to irreversible injury, infarct age, and contractile function. Circulation 100, 1992–2002.

Kurzendorfer, T., Breininger, K., Steidl, S., Brost, A., Forman, C., Maier, A., 2018. Myocardial scar segmentation in lge-mri using fractal analysis and random forest classification, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE. pp. 3168–3173.

Kurzendorfer, T., Breininger, K., Steidl, S., Maier, A., Fahrig, R., 2019. Left ventricle segmentation in lge-mri using multiclass learning, in: Medical Imaging 2019: Image Processing, International Society for Optics and Photonics. p. 1094929.

Li, C., Tong, Q., Liao, X., Si, W., Chen, S., Wang, Q., Yuan, Z., 2019. Apcp-net: aggregated parallel cross-scale pyramid network for cmr segmentation, in: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), IEEE. pp. 784–788.

Moccia, S., Banali, R., Martini, C., Muscogiuri, G., Pontone, G., Pepi, M., Caiani, E.G., 2019. Development and testing of a deep learning-based strategy for scar segmentation on cmr-lge images. Magnetic Resonance Materials in Physics, Biology and Medicine 32, 187–195.

Norton, P., 2013. Steady state free precession. URL: `https://www.med-ed.virginia.edu/courses/rad/cardiacmr/Techniques/SSFP.html`.

Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al., 2018. Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 .

Organization, W.H., 2017. Cardiovascular diseases (cvds). URL: `https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`.

Positano, V., Pingitore, A., Giorgetti, A., Favilli, B., Santarelli, M.F., Landini, L., Marzullo, P., Lombardi, M., 2005. A fast and effective method to assess myocardial necrosis by means of contrast magnetic resonance imaging. Journal of Cardiovascular Magnetic Resonance 7, 487–494.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.

de la Rosa, E., Sidibé, D., Decourselle, T., Leclercq, T., Cochet, A., Lalande, A., 2019. Myocardial infarction quantification from late gadolinium enhancement mri using top-hat transforms and neural networks. arXiv preprint arXiv:1901.02911 .

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826.

Tao, Q., Piers, S.R., Lamb, H.J., van der Geest, R.J., 2015. Automated left ventricle segmentation in late gadolinium-enhanced mri for objective myocardial scar assessment. Journal of Magnetic Resonance Imaging 42, 390–399.

Wei, D., Sun, Y., Chai, P., Low, A., Ong, S.H., 2011. Myocardial segmentation of late gadolinium enhanced mr images by propagation of contours from cine mr images, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 428–435.

Wong, K.C., Moradi, M., Tang, H., Syeda-Mahmood, T., 2018. 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 612–619.

Yue, Q., Luo, X., Ye, Q., Xu, L., Zhuang, X., 2019. Cardiac segmentation from lge mri using deep neural network incorporating shape and spatial priors, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 559–567.

Zabihollahy, F., White, J.A., Ukwatta, E., 2018. Myocardial scar segmentation from magnetic resonance images using convolutional neural network, in: Medical Imaging 2018: Computer-Aided Diagnosis, International Society for Optics and Photonics. p. 105752Z.

Zhao, S., Wang, Y., Yang, Z., Cai, D., 2019. Region mutual information loss for semantic segmentation, in: Advances in Neural Information Processing Systems, pp. 11117–11127.

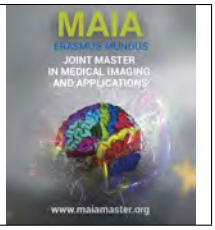
Zhuang, X., 2016. Multivariate mixture model for cardiac segmentation from multi-sequence mri, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 581–588.

Zhuang, X., 2018. Multivariate mixture model for myocardial segmentation combining multi-source images. IEEE transactions on pattern analysis and machine intelligence 41, 2933–2946.

Zotti, C., Luo, Z., Lalande, A., Jodoin, P.M., 2018. Convolutional neural network with shape prior applied to cardiac mri segmentation. IEEE journal of biomedical and health informatics 23, 1119–1128.

Medical Imaging and Applications

Master Thesis, August 2020



Point Tracker Network for Multimodal 2D-3D Registration

Patricia Cabanillas, John Hipwell

Canon Medical Research Europe, Edinburgh (UK)

Abstract

In medical imaging, image registration is a crucial step for many clinical tasks. 2D-3D registration consists of aligning a pre-operative 3D volume and live 2D X-ray images to the same coordinate frame. 2D-3D registration is used in guiding minimally invasive interventions, quantitative measures of relevant anatomical parts or pre-processing for segmentation. One of the main problems in image registration is the initial displacement position between the two images to register. This work attempts to solve this initial alignment problem employing a state of the art Point-Of-Interest network for tracking a set of matching points between a CT image and a X-ray image. The experiments show the potential of the proposed method in finding the correspondence between 2D and 3D points (X-ray and CT respectively), decreasing the target registration error from around 10 mm to 1.42 mm, when only a small dataset (one patient) is used for training and validation. Yet the decrease in performance for more diverse data indicate that larger training datasets are required for a more effective and robust registration approach.

Keywords: 2D/3D rigid registration, convolutional neural network, point tracker, image-guided intervention, CT, fluoroscopy

1. Introduction

Convolutional Neural Networks (CNN) have shown a huge success in different tasks such as medical image segmentation and classification problems (Tajbakhsh et al., 2020) (Litjens et al., 2017). However, comparatively less studies have been reported on their used for medical image registration tasks until recent years (Fu et al., 2020).

Image registration consists of aligning two or more sources of data to the same coordinate frame. In medical imaging, image registration is crucial for many clinical tasks, including guiding minimally invasive interventions, quantitative measures of relevant anatomical parts or pre-processing for segmentation. Depending on the dimension of the reference and target data, registration methods are divided into 3D to 3D, 2D to 2D and 2D to 3D (Fu et al., 2020). This work is focused on 2D to 3D registration.

Registration of 2D-3D data is one of the key technologies for image-guided radiation therapy, radiosurgery, minimally invasive surgery, endoscopy, and interventional radiology (Markelj et al., 2012).

Generally, 3D modalities such a Magnetic Resonance (MR) or Computed Tomography (CT) imaging are used in clinical diagnosis and treatment planning, but their use as intra-operative imaging modalities has been limited, meanwhile 2D data, such as ultrasound or X-ray fluoroscopy, is mostly used for guiding interventions (Penney et al., 1998). These 2D modalities are "real-time", but have limited spatial information and a number of important anatomical features can not be well visualized.

To solve this issue and take benefit of the spatial and anatomical information of CT images during interventional procedures, 2D/3D image registration methods are used. The objective is to bring the pre-operative 3D data and intra-operative 2D data into the same coordinate system through a transformation, i.e., rigid, affine, projective or non-rigid.

In 2D-3D registration, the use of CT images as pre-operative data has an advantage when X-ray or fluoroscopic images are the target 2D modality, due to the possibility of projecting them into a 2D image plane as digitally reconstructed radiographs (DRRs) (Sherouse et al., 1990). The DRRs are normally created by ray

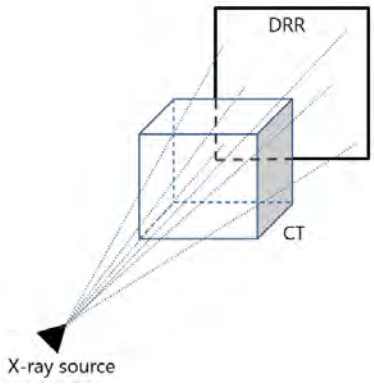


Figure 1: DRR generation by ray casting

casting, where the CT voxels intensities along each ray are summed and projected onto a 2D image (Figure 1). This projection has a similar appearance to X-ray images which facilitates solving the registration problem.

Before the appearance of deep neural networks, a large number of papers in conventional 2D-3D registration have been published. A review of these different methods can be found in Markelj et al. (2012). Regarding the nature of registration basis, the approaches can be divided along extrinsic or intrinsic methods. Extrinsic methods register artificial objects such as implants or markers, which makes the registration a simpler task, but is invasive and time-consuming to set-up. Intrinsic registration consists in the alignment of anatomical structures, which is a challenging task for multi-modal images due to the nonlinear pixel relation, the computational cost of generating the DRRs and the initial displacement position between the images (Akter et al., 2015).

Focusing on intrinsic registration, they can be classified in two main groups: intensity-based and feature-based approaches. Intensity-based methods compare the information contained in pixel and voxels, and feature-based methods try to minimize the distance between salient features like surfaces, contours or points, which have been previously selected (Penney et al., 1998). The feature based methods can be chosen to deal with the problem of high misalignment in a initial step as described in Akter et al. (2015). This initial registration is needed to avoid the method converging to a local minimum when the proximity between the correct position and the initial one is not enough (Liao et al., 2019).

This work will asses the use of Convolutional Neural Networks for the extraction of features providing correspondence between X-ray images and DRRs (CT projections). The corresponding features can be used for the estimation of rigid motions. This thesis is structured as follow: Section 2 provides a review of the State Of The Art in medical imaging registration; Section 3 describes the material and the method used; Section 4 shows the experiments proposed and their corresponding results; Section 5 contains the discussion of the work; And Section 6 and 7 provide the limitations and future works, and the conclusions of this work, respectively.

2. State of the art

The use of deep learning-based methods has recently gained importance in 2D-3D registration tasks. The learning-based approaches in 2D-3D registration have the objective of predicting 3D deformation from a pair of X-ray and DRR images. In Miao et al. (2016) is employed CNN regressors to directly estimate the transformation parameters.

Jaderberg et al. (2015) introduced a learnable module called spatial transformer network (STN), which can be inserted into existing convolutional networks, and performs a spatial transformation of the features in a network which computes a specific task. Inspired by STN, de Vos et al. (2017) presents a deformable image registration (DIRNet). This network is an unsupervised learning network that can perform the registration using the local deformation parameters predicted by a convolutional neural network. Although, they referenced method such us 2D/3D registration, it was only applied in 2D images and the authors only considered extending it to 3D images in the conclusion. Sheikhjafari et al. (2018) proposed a different DIR network, which contains a deep fully connected network to generate spatial transformations. This approach attempts to maximize the similarity metric between the 2D images based on latent inputs initialized by random vectors.

The deformable image registration proposed by de Vos et al. (2017), was extended later from 2D to 3D images in de Vos et al. (2019). They presented a Deep Learning Image Registration (DLIR) framework, which exploits image similarity between pairs of 3D images to train a convolutional network for hierarchical multi-resolution and multi-level image registration.

Alternatively, there are methods which use reinforcement learning to predict the best actions and estimate the transformation. In Miao et al. (2018), the 2D-3D registration has been formulated as a Markov Decision Process (MDP), where multi agents are trained with a dilated Fully Convolutional Network (FCN) and weighted by the corresponding confidence level, to take the final action and drive the registration. To tackle the image artifacts problem they used an auto-attention mechanism to observe the regions with trusted visual hints.

The goal in 2D-3D multi-modal registration is to determine the spatial correspondence between the different imaging domains. Thus, some works, such as Valmadre et al. (2017) or Tustison et al. (2019) have used Siamese networks, where both fixed and moving images are fed as a input of two identical branches, whose

weights are shared. Also, in visual object tracking, Siamese architecture has been utilized to calculate the similarity between two input images and it has the ability to learn their discriminant features (Fiaz et al., 2019).

An alternative to directly registering the two images is to establish a point-to-point correspondence between multi-view or bi-planar X-rays and their corresponding DRR images (Liao et al., 2019). This paper includes a triangularization layer which projects the 2D estimated points back into the CT image. Since the method requires at least two X-rays from different views, this paper use cone-beam CT (CBCT) images, which are reconstructed from more than 700 X-ray images. This has the advantage that pairs of images, with a known spatial correspondence to the reconstructed CBCT volume, can be selected from the acquired CBCT set prior to the reconstruction and used for training. However these images will not be representative of typical fluoroscopic images acquired in routine clinical practice.

Traditional CT is a common choice of pre-operative image for a wide range of image-guided interventions. So, this work is focused on Liao et al. (2019) approach to tracking a set of interest points but using traditional CT images and a single X-ray target image (only one view) instead of CBCT (multi-view X-rays). The Liao et al. (2019) paper has been selected in this work for the following reasons. First, it gives a lot of details about the approach, which makes its implementation straight forward. Second, this method has improved the robustness and speed of the state-of-the-art optimization-based approaches. Finally, as we commented before, it can be adapted to different images modalities, such as traditional CT, and applications such as image-guided interventions for spine, vessels or organs.

3. Material and methods

3.1. Data

The data used in this 2D/3D registration is a combination of cohorts obtained on three different occasions. All CT and fluoroscopy data was provided by Prof. Guiu, Centre Hospitalier Universitaire de Montpellier (L'Hôpital Saint Eloi) for research purposes and anonymised in accordance with a collaborative agreement with Canon Medical Systems Corporation. The data was acquired during a range of abdominal liver and spine interventions and the complete dataset is formed by 9 patients. Each patient normally have only one CT and a certain number of different X-ray images, where some of them are composed by multiple frames. In order not to have similar X-ray frames, the frames where a significant motion has occurred have been selected. So, the pair of images in a patient will be between the same CT and different X-rays or X-ray frames. In Figure 2 is possible to see some pair of samples from the dataset given. In Table 1 is possible to see a summary of the number of images content in each dataset.

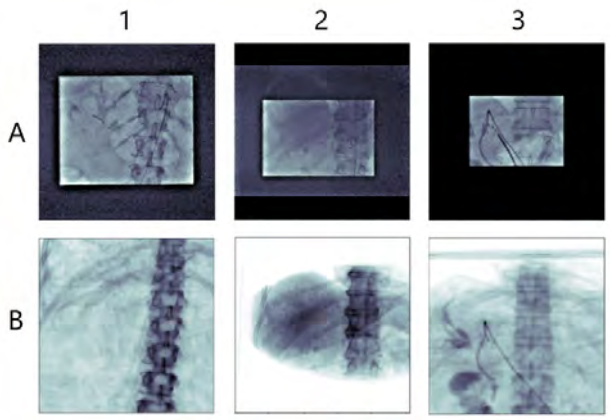


Figure 2: Samples of the dataset. The numerated columns represent different patients. A and B rows represent a pair of X-ray and DRR, respectively.

Cohort	Patients	CTs	X-rays	X-ray frames
1	3	3	7	35
2	2	2	2	4
3	4	4	9	18
Total	**9**	**9**	**18**	**57**

Table 1: Summary of the cohorts and their images used to train this network.

To validate the 2D-3D registration algorithm, it is needed to calculate the ground truth. A first approach was a manual labelling of a minimum of three corresponding landmark points on both sets of matching scans (CT and X-ray). For this procedure, it has been use a internal annotation tool, called TIARA, developed by Canon Medical Research Europe (CMRE), Edinburgh, UK (Figure 3.1a/1b). However, this method in some cases is not enough accurate due to the possible mistake made when selecting the points. So, an alternative approach is the use of 3D road-map tool developed by CMRE, which enables the user to translate, rotate and control the transparency of a 3D spine model, that comes from the CT image, over a X-ray image (Figure 3.2).

Due to the heterogeneity and the small size of the dataset, to reduce the images variability and therefore the complexity of the problem, it has been ruled out the patients which do not satisfy some requirements. First, the patients which do not present spine in their images have been discarded; second, the patients whose images are poor quality and finally, the patients whose ground truth is insufficient accurate, have been subtracted from the dataset. No other quality based selection criteria were used to avoid biasing the algorithm performance.

Also, to increment the dataset it has been generated an artificial data by applying perturbations to the DRR images of the patient. However, for each CT, there is a wide range (from 1 to 4) of associated number of matching temporal X-rays sequences. Furthermore, for

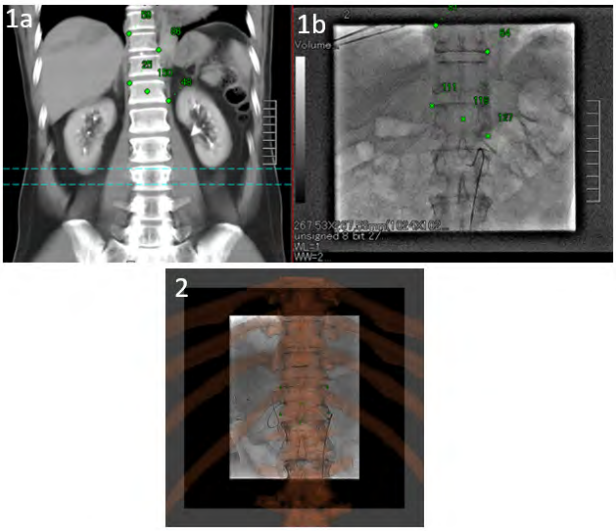


Figure 3: Annotation approaches. On the one hand, the manual labelling approach is represented in images 1a/b, which shows the corresponding landmark points (green points) in both images, slice of he CT image(1a) and X-ray(1b). On the other hand, image 2 represents the 3D road-map tool approach, where is possible to see the X-ray image (gray) in the background and the 3D spine model (orange) over it.

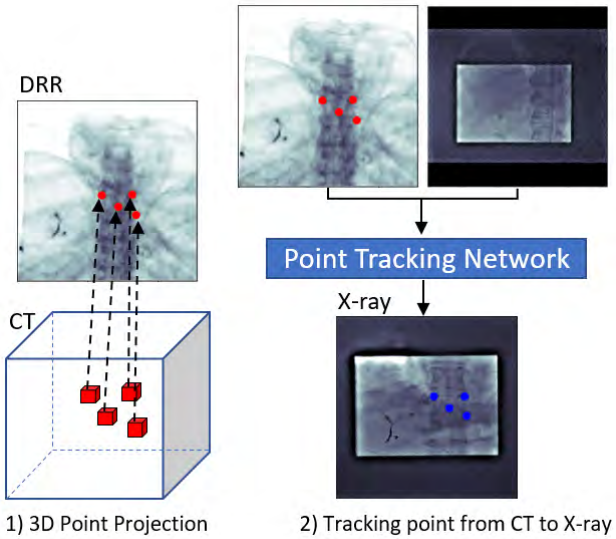


Figure 4: Overview of the proposed method.

each temporal X-ray sequence there is an even wider range from (1 to 11) of associated X-ray frames. This data imbalance may lead our network to overfit towards cases where there is a big amount of data. In an attempt to avoid such biases, it is necessary to balance the datasets as much as possible. To do this, data permutations, which ensure a constant number of DRR/X-ray pairs for each patient, have been introduced. Ideally, each permutation will consist of:

1. Add a random transformation to the CT volume and generate the corresponding DRR.
2. Add random noise to the DRR image.

As a result, all DRR images will be different every time, with some repeats of the associated X-ray. To train the method proposed, the perturbed images have been generated with a maximum of $\pm 10^{o}$ of rotation offset and a 10 mm of translation offset which mimic the movements of the patient may with respect to the operation bed.

3.2. Pre-processing

As previously mentioned, most 2D/3D registration methods in the literature use simulated X-ray projection, generated from CT images, to enable comparison with the target X-ray. Therefore a pre-processing step has been used to get the CT projection into a 2D image (DRR). The DRR images were generated by summing the intensities along a ray cast through the CT volume, which is a first approximation to the log-transformed, linear attenuation of X-rays (assuming a monochromatic spectrum) captured in an X-ray or fluoroscopic image, using Equation (1) (McKetty, 1998).

$$I = I_o e^{-\mu x} \tag{1}$$

where I_o = beam intensity at an absorber thickness of zero, x absorber thickness, μ = attenuation coefficient and I = beam intensity transmitted through an absorber of thickness of x. So, the DRRs generated look similar to X-ray images and ease the registration between them.

Another pre-processing step applied is to bring the range of intensity values of all the images to a normal distribution by a z-score normalization (Equation 2).

$$Z = \frac{X - \bar{X}}{\sigma(X)} \tag{2}$$

where the symbol σ corresponds to the standard deviation and $\bar{X}$ to the mean.

As we can appreciate in Figure 2 the background of the images are different in terms of shape and intensity. This can affects the normalization results. So, to solve this problem, a mask has been generated which can enable the background to be ignored and thus, calculate the mean and the standard deviation corresponding only to the specific region where the image is.

3.3. Methodology

The objective of this work is to find the correspondence between X-ray and DRR points by a point tracking approach. The proposed method is divided in two main parts: 3D points selection and projection and a point tracking network. An overview of the complete approach is shown in Figure 4.

An initial step in the method is the 3D point selection (previously discussed in Section 3), which later, will take part in the training step of the network. Having the DRR and X-ray pairs and the ground truth matrices,

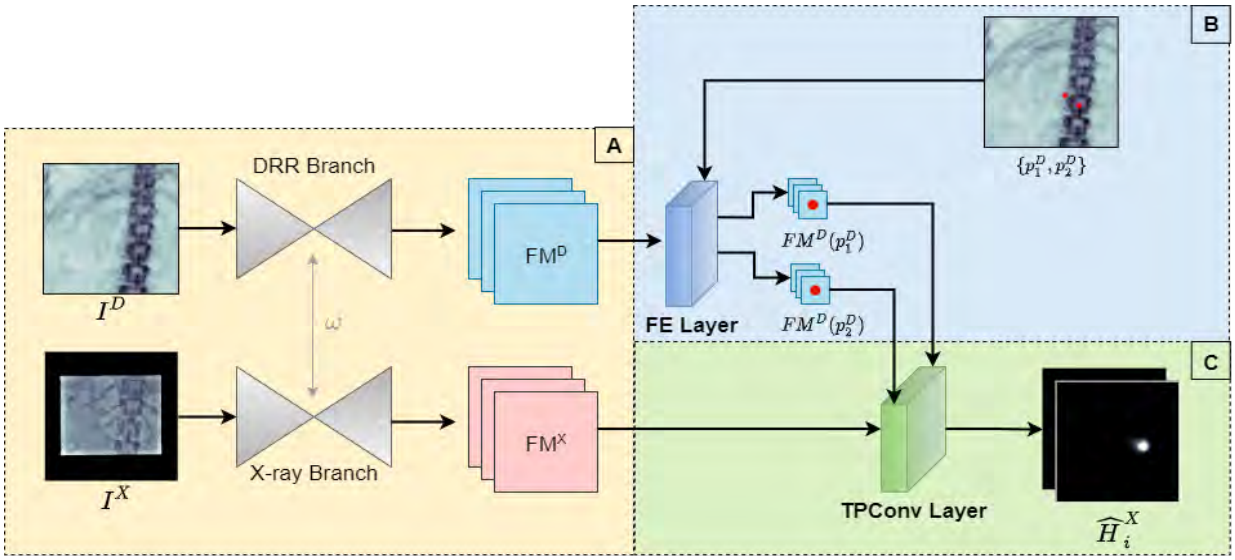


Figure 5: Point tracking approach. The different parts of the network are labeled from A to C. Label A represents Siamese architecture, label B, the Feature Extraction Layer (FE Layer) and label C, the Tracking Point Convolutional Layer (TPConv).

the objective is select a set of 3D points from a CT image and project them onto the corresponding DRR and fluoroscopy images using the Equations (3).

$$\begin{aligned} p3D &= V \cdot v_{3D} \\ [x', y', z', w] &= P \cdot M \cdot p_{3D} \\ p2D &= \left[\frac{x'}{w}, \frac{y'}{w}\right] \end{aligned} \tag{3}$$

where V is a 4x4 matrix to convert the 3D points selected from voxels coordinates (v3D) to millimeters (p3D), P is used to project from 3D volume onto 2D plane, M describes the spatial rotation, translation and p_{2D} is the 2D point projected in pixels. Those parameters are fixed for each pair of image. So, they are not optimized during the training.

Once the 2D points have been obtained, next step is the point tracking network, which is responsible for finding the point-to-point correspondence between DRRs and X-rays pairs. The network is fed with pairs of DRRs and X-ray images (I^D,I^X) and the DRR coordinate points previously projected (P_1^D,P_2^D,...,P_n^D). The output of the network are the corresponding X-ray 2D points as a heatmap ($\widehat{H}_1^X$,$\widehat{H}_2^X$,...,$\widehat{H}_n^X$). The structure of this network is divided in three parts as in (Liao et al., 2019): A siamese architecture and two custom layers, feature extraction layer (FE) and tracking point convolutional layer (TPConv). In Figure 5 is shown a overview of the network.

3.3.1. Siamese Architecture

The siamese architecture consists of two parallel branches, where each branch is a U-Net architecture and whose weights are shared (Figure 6). This specific architecture with shared weights is typically used to learn similarity and dissimilarity between two images. Each branch of the U-net is composed of a contracting path or encoders and a expanding path or decoders, where the first blocks capture the context in the image, and the second blocks enable precise localization. Each pair of encoder-decoder block is connected through a shortcut or skip connection, which helps to train deeper networks avoiding the vanishing problem. In each skip connection, the output of the encoding block is directly connected to the decoding block. In Figure 6 is shown the U-net architecture used in this project. This part of the network extracts the feature maps at pixel-level with the same resolution as the input image. So, if the input image has a shape of MxNx1, the shape of the feature map will be MxNxC where C is the number of channels, 64 for this network. As the Siamese network has one output for the DRR branch and another one for the X-ray branch, the feature maps extracted will be referred as FM^D and FM^X, respectively (Figure 5.A).

3.3.2. Feature Extraction Layer

Once the feature maps have been obtained from the Siamese network, the next step is to get a feature patch for each DRR point (p_i^D) by the extraction of FM^D at p_i^D. The result of this operation is a simple 1x1xC feature vector, which is not able to capture enough relevant information. So, in order to obtain a more representative feature vector, the Feature Extraction Layer (FE) not only extract each channel of FM^D at p_i^D, but also the neighbouring channels of each point, which will change the size of the feature patch from 1x1xC to KxKxC, where K is the kernel or neighborhood size used. The output, i.e. the feature patch, is denoted as $FM^D(p_i^D)$

and an overview of this FE layer is shown in Figure 5.B.

3.3.3. Tracking Point Convolutional Layer

The aim of this Tracking Point Convolutional layer (TPConv) is obtain a probability "heatmap" where the value corresponds with the predicted X-ray point location. In TPConv Layer, the feature patch previously extracted in FE layer is treated as a filter kernel in a convolution. Figure 5.C represents this layer.

To find the correspondence between the points, a similarity operation has to be applied between the feature patch previously extracted, $FM^D(p_i^D)$, and a feature patch, in a certain location x, of the X-ray feature map $FM^X(x)$. The similarity operation is calculated by every single X-ray locations and the location with largest similarity score would be considered the corresponding X-ray point of p_i^D.

As commented before, the $FM^D(p_i^D)$ is treated as a kernel filter therefore this deep search on FM^X can be efficiently executed with a normal convolution, where the input and the filter kernel correspond to FM^X and $FM^D(p_i^D)$, respectively. Thus, the resulting heatmap for each point ($\widehat{H}_i^X$) is computed using the following equation:

$$\widehat{H}_i^X = FM^X * (W \odot FM^D(p_i^D)) \tag{4}$$

where, W, is a trainable weight which picks the features that gives the better similarity. Once the predicted heatmaps $\widehat{H}_i^X$ have been obtained, the next step is to select the corresponding 2D DRR point on the X-ray heatmap. To do so, the x and y coordinates in the heatmap where the intensity value is the highest is selected. The predicted 2D X-ray point is denoted as $\widehat{p}_i^X$.

To measure how good the model is in tracking the points, the following loss function has been used:

$$\begin{aligned} Loss = & \frac{w_1}{n} \sum_i BCE(\sigma(H_i^X), \sigma(\widehat{H}_i^X)) \\ & + \frac{PixelToMM \cdot w_2}{n} \sum_i \left\| \widehat{p}_i^X - p_i^X \right\| \end{aligned} \tag{5}$$

where w_1 and w_2 are weights which help to balance the losses between the tracking and the distance error, H_i^X is the ground truth X-ray heatmap, p_i^X is the ground truth 2D X-ray points and BCE corresponds to the binary cross entropy function, whose inputs should be between [0,1]. Thus, σ symbol represents the sigmoid function which will move the heatmaps values between 0 and 1, using Equation 6.

$$\sigma(\widehat{H}_i^X) = \frac{1}{1 + e^{-\widehat{H}^X}} \tag{6}$$

This work will be evaluate based on the target registration error (TRE), which measure the euclidean distance between the X-ray points before and after the tracking (second part of the Equation (5)).

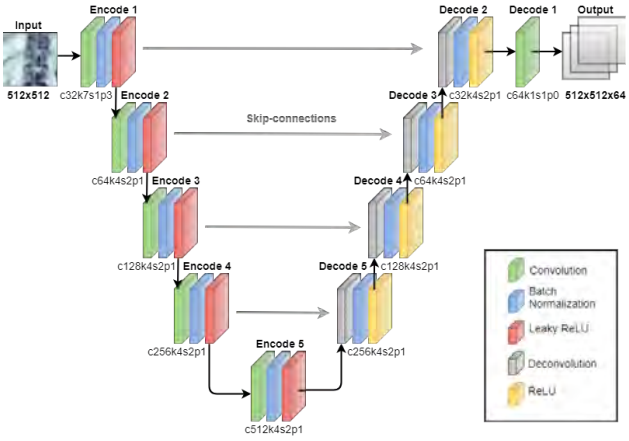


Figure 6: U-Net architecture used in each branch of the tracking point network. The configuration of the convolutional and deconvolutional layers are given by the letters "c", "k", "s", "p", which correspond to channel, kernel, stride and padding size, respectively.

3.4. Implementation and Training details

This method has been completely implemented from scratch, following Liao et al. (2019), in Python using Tensorflow 2.1 framework on an NVIDIA GPU Tesla V100 SXM2 and 32GB of internal memory.

As commented before each Siamese branch, consists in a U-net architecture. Each U-net is composed by five encoding blocks (Convolution, Batch Normalization and Activation) followed by five decoding blocks (Deconvolution, Batch normalization and Activation), connected by skip-connections (Figure 6). In the skip connection occurs a concatenation operation between the feature maps outputs by the "n" encoding block and the "n+1" decoding block.

As described before, the dataset used in the experiments is composed of a selection of CT/Xray pairs over all datasets, to reduce the variability. So, the final dataset is composed by 9 CT images and 18 X-ray images with a total of 57 X-ray frames as previously stated in Table 1. The total number of images available to train and validate the network is 66. For a deep learning project this amount of images is insufficient to make a network extract the good characteristics and generalize well. In order to solve this issue has been generated an artificial data (as described in Section 3).

The original size of the X-ray and CT images are 1024x1024 with a variable pixel spacing. In all the experiments the CT images have been considered as the 3D pre-intervention data, and the respective X-ray images have been treated as 2D intra-intervention data. Note that our images apart from coming from different sources, some of them contains clinical features such us, surgical instruments, contrast in blood vessels, different quality images, etc, which increase the complexity of the dataset and hence the problem to solve.

After a lot of trials it was discovered that some training specifications work better than others. For the op-

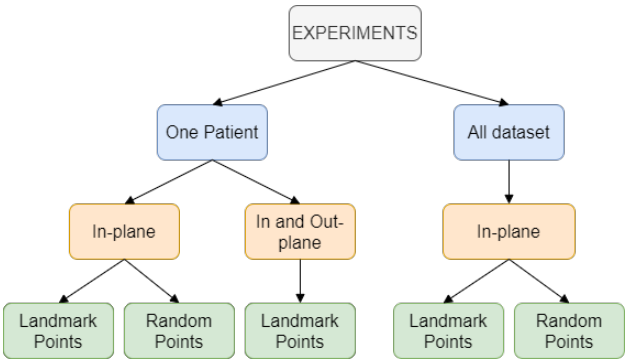


Figure 7: Pipeline followed in the experiments. Blue boxes represent the dataset selected to train and validate. Orange boxes show the transformation parameters applied to generate the perturbed images. And Green boxes indicate which points were included in the training network process.

timization, Adam trained in mini-batches of 8 and with a learning rate of 0.0005 with a decay of 1e-6 in each epoch was used. The loss weights have been set as $w_1 = 100$ and $w_2 = 0.001$. The K size which has given better results is 3 or 5 depending on the experiment and the image resolution used to train is 512x512.

Due to the dimension of the network, GPU memory constraints were faced which limited the values of some parameters of the network, such us the image resolution (Maximum used 512x512) or the batch size (Maximum used 8).

4. Results

Due to the complexity of the dataset and in order to correctly evaluate the network, the experiments have been divided in two main parts. The first experiment is based on training the network in one single patient with different X-ray frames images for training and validating, to simplify the problem and reduce the variability of the dataset. And the second experiment consists in training the network over the entire dataset. Figure 7 shows the pipeline followed in the experiments. The purpose of these experiments is to start from the easiest case, which is training in one patient and validate in the same one, but using different X-ray frames. Then train the network with a complex dataset with more patients which come from different resources and thus, different distributions (This case requires a huge amount of data).

4.1. One Patient experiments

For the first experiment only one patient was used to train and validate. In this experiment, it has to be taken into account that the validation set are composed of different X-ray frames from the training set. Both training and validation sets have been augmented by applying random perturbations.

The 3D transformation of a rigid-body, as it is considered the bones, can be represented by a vector t of 6

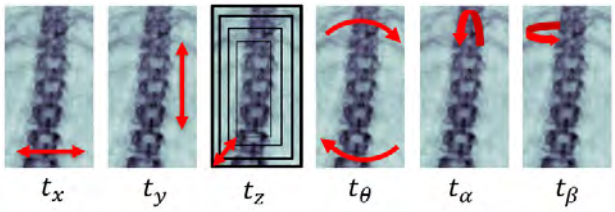


Figure 8: Movements of the 3D transformation parameters

parameters. The 6 components of the vector represent 3 in-plane and 3 out-plane transformations parameters, as it is able to see in Figure 8. Specifically, the in-plane transformation parameters are composed by two translation parameters, t_x and t_y, and one rotation parameter, t_θ, while the out-plane transformation is formed by one out-of-plane translation parameter, t_z, and two out-of-plane rotation parameters, t_α and t_β.

Taking this into account, the one patient experiment has been split in two parts depending on the transformation parameters used to generate the perturbation (Figure 7, orange boxes). In a first part, the transformations applied were only in-plane translations, t_x and t_y, because the in-plane translations are the most significant movements, due to the motion of the patient with regard to the operating bed and also because the effects of those parameters are approximately 2D rigid-body transformations, which will help to simplify the problem. In a second part, the perturbations used are a combination of the three in-plane and three out-plane transformations parameters, which represent a 3D rigid-body transformation.

4.1.1. In-plane experiments

This one-patient experiment will help to do a detailed discussion of the parameters which provide better results. As commented before, the evaluation of the model and the parameters chosen is going to be based on TRE results.

One of the first steps in the point-to-point correspondence network is the selection of the points. In this work we have followed two different points of interest selection approaches. The first approach consists in the use of landmark points as a points of interest because they are typically biologically-meaningful points, which correspond to important anatomical structures, such as the spine in this project. These landmark points have been manually selected and annotated prior to the training of the network. The second approach uses 3D random points which are selected in the CT image and projected into the DRR and the X-ray images using Equation 3. This last approach has the advantage of avoiding the effort of annotation. In Table 2, row 2 and 4 is shown the TRE error for each of the approaches. For this particular experiment the use of landmark points work better than random points. As only one patient is being used to

Experiment	Kernel Size			POI		Weights		TRE
Nº	1	3	5	Lnd	Rnd	w/	w/o	(mm)
1	X			X		X		3.693
2		X		X		X		**1.42**
3			X	X		X		1.54
4		X			X	X		17.5
5		X		X			X	56.82

Table 2: Results with different model options

train and validate, the use of landmarks points can help to extract strong features for that patient and make the network robust for this case.

Another characteristic of the network that has been studied is the parameter K, which represents the neighbourhood size, extracted in the DRR image. This parameter will indicate the neighbourhood information needed to obtain a representative feature of the image, which the network can generalize well to unseen data. In Table 2, rows 1,2 and 3 show the results using a kernel of 1x1, 3x3 and 5x5 respectively. It has to be taken into account that the use of a kernel size of 1x1 does not only contain the information of the current pixel extracted in FM^X and FM^D but also the information which comes from the receptive field of the U-net. The receptive field of a convolutional network is described as the size of the region in the input that produces the feature. Note, that the input region can be the input of the network or the input of a specific unit of the network. So, the unique pixel extracted by the 1x1 kernel will contain information of the current pixel as well as information coming from its neighbourhood.

Analysing the results, the use of a 1x1 kernel does not extract features representative enough for a better similarity measure. However, the use of a 5x5 kernel size makes generalization of the model harder because it takes too much information from its neighbourhood and moreover, the use of a bigger kernel size increases the computation cost. So, the kernel size of 3x3 is the one which extracts the most distinctive features and thus provides the best performance.

Also, it has been studied was the importance of adding a trainable custom layer to track the points. In Table 2, rows 2 and 5, shows the results when a trainable weight (W) is included in Equation 4 or not. As it is possible to see in Figure 9 when a trainable weight is not added in TPConv layer, the training of the model can not learn and thus, decrease the TRE error. So, when the trainable weight is added to the network, the distinctive features are accentuated in the second part of the loss function, which calculate the euclidean distance, and thus, allows the network to learn and reduce the TRE error. So, this experiment shows the importance of using the TPConv custom layer as a trainable layer.

There are parameters which have been settled after some trials such as the loss function weights in Equation 5, the optimiser, the learning rate or the image size. The loss function weights which give better results are $w_1 = 100$ and $w_2 = 0.001$.

Regarding the optimiser, Mini-Batch Stochastic Gradient Descent (SGD), Adadelta and Adam have been tried. Adam was the chosen optimizer not only because it is faster than the others and converges rapidly, but also is an adaptive learning rate method. So, it makes easy the learning rate selection. Even though Adadelta is also an adaptive learning rate method, it was discarded because it takes too much time to converge. In the case of SGD, it was rejected because is needed to choose an optimum value of the learning rate which makes training the model harder.

As was previously commented, due to the size of the model and the GPU used, there have been memory limitations which has not allowed the use of an image resolution bigger than 512x512. Also tried was an image size of 256x256, but the model was not able to extract better features than using higher resolution.

Finally, in order to address the overfitting problem, different regularization methods such us L2 regularization and dropout were compared. Figure 10 shows the results in training and validation for both types of regularization. Clearly the training results using dropout get stacked in a higher loss value than using L2 regularization. This is because, when dropout is applied, a fixed percentage of random neurons inside of the network are discarded while training. So, the complexity and the learning capacity of the network is reduced. Moreover, taking into account the validation results, it is possible to appreciate that using dropout regularization still allows the network to overfit. In this case, to help the model to generalize better, L2 regularization in the first layer of the U-Net with a regularization factor of 0.0005 has been used.

Some visual results of the point tracking model are illustrated in Figure 11. The first row show the pairs of X-ray (a) and DRR perturbed images (b,c) and their corresponding landmark points represented in blue and red,

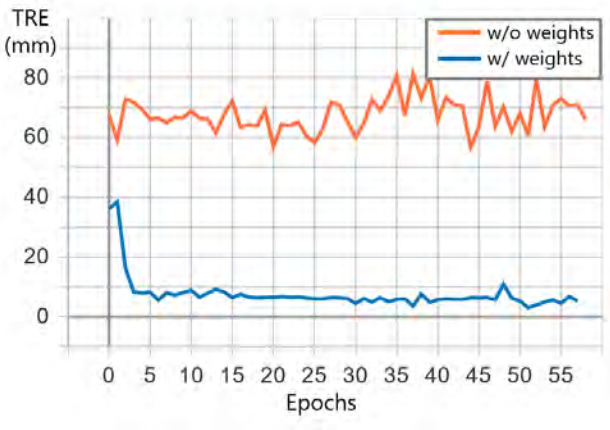


Figure 9: TRE results when TPConv trainable weights are included or not.

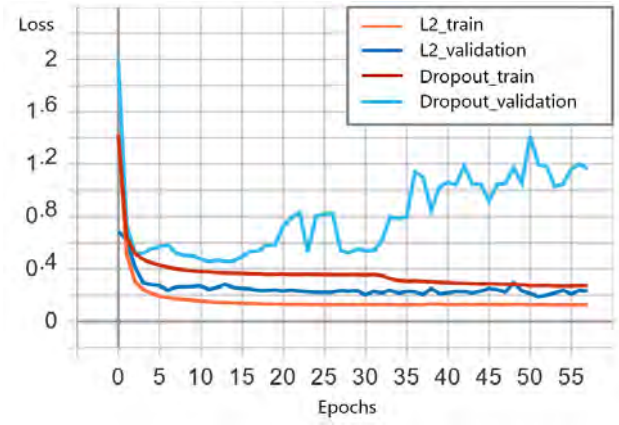


Figure 10: Training and validation loss results using L2 regularization or dropout for the one-patient, in-plane experiment.

respectively. Under the DRRs are a couple of examples of the heatmaps predicted by the model corresponding to a selection of one random landmark point in each image. The white part around the points in the heatmaps represents the output of the network which is used to localize the correspondent point. Moreover, below the heatmaps is possible to see the movement of the DRR points, where the GT X-ray points are represented with blue marks, the DRR points with red marks and the predicted points with green marks. The numbers under the heatmaps represents the TRE before and after apply the point tracking network.

4.1.2. In/Out-Plane Experiments

In this section as in the previous one, the experiments are done in only one of the patient, but in order to increase the complexity of the problem, the perturbed images have been generated modifying the 6 transformation parameters. Taking into account that the main movement is produced via the in-plane translation, the perturbation has been applied within a small range. The maximum in/out plane rotation applied was 10^o and the in/out plane translation was 10 mm.

Regarding the specific points of interest used, the landmark point approach was chosen because it gives better results than the random one. Furthermore, the parameters chosen in this experiment where exactly the same as those selected in the previous experiment, except the kernel size.

As can be appreciated from Table 3 the TRE results

Exp	**Kernel Size**			**TRE**
N^o	**3**	**5**	**7**	**(mm)**
1	X			7.84
2		X		**5.85**
3			X	6.10

Table 3: In/out plane transformation experiments in one patient.

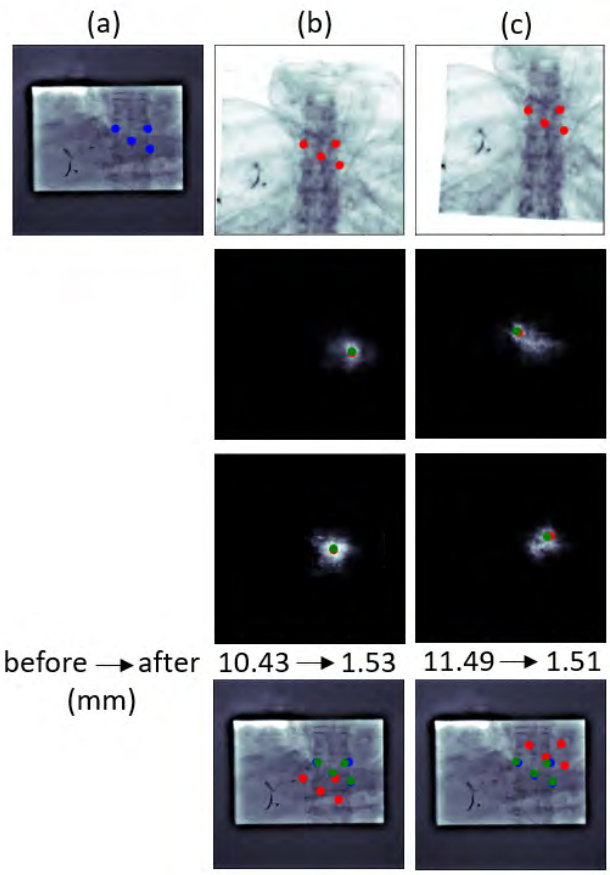


Figure 11: Visual results of the point tracking network. The image (a) represent the X-ray. In (b) and (c) are shown two DRR examples where different in-plane perturbations has been applied. The heatmaps results of two over the four points is vertically aligned with his correspondent DRR image. The two last fluoroscophy images show the GT landmark points (blue marks), the original position of the DRR points (red marks) and the predicted points (green marks).

are better when the kernel size is 5. Moreover,due to the fact that the complexity of the problem has been increased the TRE results are higher than those obtained with one patient and in-plane transformations.

4.2. All Dataset Experiments

The last experiments include all the data shown in Table 1. Seven patients were selected for training and the remaining two for validation. The data were carefully selected in a way to avoid possible bias of the outcomes. First, the model was trained with the same parameters used in the experiments described, at the beginning of this Section, except for the kernel size which was set to 3 in all these experiments. This avoids the issue that a bigger kernel size contains more neighbourhood information and will be prone to overfit faster.

This dataset has been run with the landmark and random points approaches. Figure 12 shows the training and validation values in each epoch for both approaches, and it is possible to see the network starts to memorize

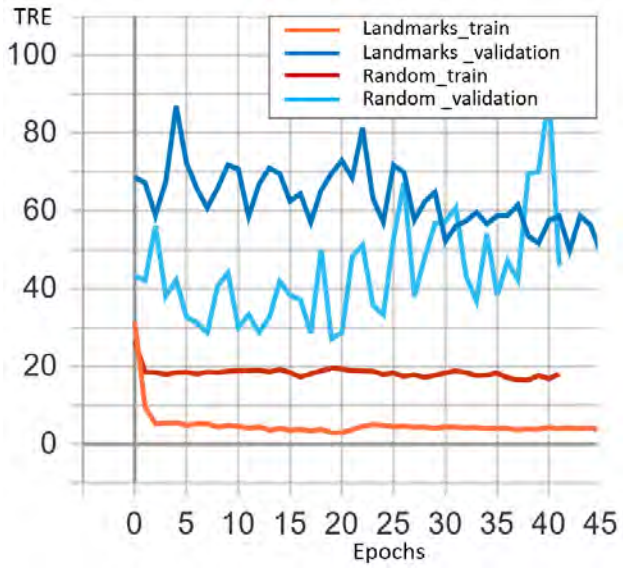


Figure 12: Training and validation TRE results with all the data, using the two different point selection approaches.

from the beginning using the landmark points and a few epochs later when random points are used.

In order to avoid overfitting the following actions have been taken. The first action has been to apply regularization methods such as dropout or L2-regularization. The second action has been to change the complexity of the network by removing some layers and thus reduce the number of trainable parameters. And finally, in order to tackle not only the generalization problem, but also the problem of training convergence when the random point approach is used, different data augmentation strategies have been implemented. These include techniques such us translation, rotation and flipping. Due to the fact that the random points are generated in a step prior to training of the network and not during training, the network will train for the same points on a given image, throughout each epoch. So, to add variability to the points, along with the data augmentation, 4 of the 5 points for each epoch have been randomly selected.

Exp	**POI**		**REG**		**D. AUG**		**TRE**
N°	**Lnd**	**Rnd**	**No**	**Yes**	**w**	**w/o**	**(mm)**
1	X		X			X	47.32
2	X			X		X	39.17
3		X	X			X	27.05
4		X		X		X	**26.87**
5		X		X	X		39.21

Table 4: Results with different actions to avoid overfitting. POI refers to landmark (Lnd) and random (Rnd) points approaches, REG indicates if the regularization is applied or not. D.AUG, refers to data augmentation.

As shown in Table 4, the combined use of random points, as opposed to landmarks, and data augmentation produces the biggest reduction in TRE. Regularization has a substantial impact on the landmark performance but is of minimal benefit when random points are used due to the model is prone to overfit early when landmark points are used. An overall reduction in TRE of around 50% has been obtained using random points but the values remain larger than the value in the initial position (around 10 mm when no registration is applied), as would be expected given the limited number of images in the original data set.

5. Discussion

In this work, has been implemented the 2D/3D point tracking network presented in Liao et al. (2019) to solve the 2D/3D registration problem. The main contribution in this work has been tracking a set of interest points using traditional CT and a only one view X-ray target image instead of multi-view X-rays. As shown during the experiments section, this network is able to learn the most representative features when the variability in the dataset is small enough, but it fails to generalize when the variability increases. The best result has been achieved when the training and the validation has been done with the same patient but different X-ray frames, since, in this case, the variability of the dataset it has been substantially reduced. Notice, that the parameters chosen for this particular patient will not be optimal when the data set changes.

As commented in Section 3, perturbations were applied to the images to help the network learn different transformations and generalize better. The benefits of this were illustrated in the One Patient Experiment Section. However these perturbations do not help tackle the image variability problem.

During the development of this project, a number of decisions were taken to simplify the problem. First, images containing the spine were selected and second, low quality images were discarded. This was based on the assumption that a smaller, high quality, anatomy-specific data set is preferable to a large, low quality, unspecific data set.

Just as other 2D/3D registration approaches have shown, this method needs a large amount of data to learn reliable feature representation. As shown in All Data Experiment section 4.2, the model fails when the dataset is not big enough. The fact that overfitting occurs later using random points makes sense because instead of memorizing the landmark points, which represent the same anatomical structures, random points differ for each image, enabling the model to extract features at a more detailed level. Also, it is possible to see that using random points, the TRE value in training is not able to reach less than 10 mm. This will be because the use of random points increases the variability while training and will include some features that are less stable under these transformations.

The fact that the network overfits is principally because the number of images available in the dataset are insufficient and they come from three different sources. So the images have different intensity distributions.

Another limitation has been the Ground Truth acquisition. Since the aim for this specific project is to minimize the TRE between the DRR and X-ray pair of points, the Ground Truth is crucial to be as accurate as possible. In a separate study the accuracy of the manual landmark selection was estimated for 183 landmarks on 42 fluoroscopy frames and found to have a median value of 6.6mm but a maximum of 188mm. In this project time was taken to ensure that the data used also had accurate ground truth alignments.

6. Limitations and Future work

The major problem with this network's performance is due to the small dataset size and its variability from different sources. So, a proposed option for the future is the acquisition of a considerably larger amount of data. This will also help increase the robustness of the network.

As commented previously, prior to the training of the network, the CT has been projected into a DRR. During training, it is not possible to modify the projection and reduce, if needed, the misalignment between the CT and the patient position. So, If the DRR could be updated by the network, methods could be investigated for integrating this into the training process and reduce this misalignment.

Finally, this project is very challenging due to the added complexity of registering different types of medical images (CT and X-ray) which vary both in their intensity characteristics and their dimensionality. A way to solve this problem could be the use of a Generative Adversarial Network (GAN) to do a domain adaptation between images from the different scanners (Kamnitsas et al., 2017) (Jiang et al., 2018). However we still must take into account that to apply a GAN, the size of the dataset has to be greatly increased first.

7. Conclusions

This project has demonstrated that the method proposed is able to find the 2D point which corresponds to a projected 3D point in certain conditions. This capability has been evaluated using a really challenging and heterogeneous dataset, which is composed by standard CT and single X-ray images.

The experiments proposed show that the network is able to learn the most representative features when the variability of the images is small, with a decrease of around 80% in the TRE error (From 10 mm to 1.42). However, it has been demonstrated also that the method fails using a heterogeneous dataset which does not contain enough images. In this case the method is not able to reach a TRE smaller than 26.87 mm.

8. Acknowledgments

Firstly, I would like to express my gratitude to my supervisor John Hipwell for his perfect guidance and his support, not only in terms of work but also on a personal basis.

Secondly, I would like to thanks to all the members in my work team: Pedro Sanchez, Antonios Perperidis and Christoforos Galazis for helping me and providing really useful suggestions during the development of this thesis.

And last but not least, all the members in CMRE whose have given me the opportunity to be part of a competitive company and develop my career with them.

References

Akter, M., Lambert, A.J., Pickering, M.R., Scarvell, J.M., Smith, P.N., 2015. Robust initialisation for single-plane 3d ct to 2d fluoroscopy image registration. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 3, 147–171.

Fiaz, M., Mahmood, A., Jung, S.K., 2019. Deep siamese networks toward robust visual tracking, in: Visual Object Tracking in the Deep Neural Networks Era. IntechOpen.

Fu, Y., Lei, Y., Wang, T., Curran, W.J., Liu, T., Yang, X., 2020. Deep learning in medical image registration: a review. Physics in Medicine & Biology .

Jaderberg, M., Simonyan, K., Zisserman, A., et al., 2015. Spatial transformer networks, in: Advances in neural information processing systems, pp. 2017–2025.

Jiang, J., Hu, Y.C., Tyagi, N., Zhang, P., Rimner, A., Mageras, G.S., Deasy, J.O., Veeraraghavan, H., 2018. Tumor-aware, adversarial domain adaptation from ct to mri for lung cancer segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 777–785.

Kamnitsas, K., Baumgartner, C., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Nori, A., Criminisi, A., Rueckert, D., et al., 2017. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks, in: International conference on information processing in medical imaging, Springer. pp. 597–609.

Liao, H., Lin, W.A., Zhang, J., Zhang, J., Luo, J., Zhou, S.K., 2019. Multiview 2d/3d rigid registration via a point-of-interest network for tracking and triangulation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12638–12647.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical image analysis 42, 60–88.

Markelj, P., Tomaževič, D., Likar, B., Pernuš, F., 2012. A review of 3d/2d registration methods for image-guided interventions. Medical image analysis 16, 642–661.

McKetty, M.H., 1998. The aapm/rsna physics tutorial for residents. x-ray attenuation. Radiographics 18, 151–163.

Miao, S., Piat, S., Fischer, P., Tuysuzoglu, A., Mewes, P., Mansi, T., Liao, R., 2018. Dilated fcn for multi-agent 2d/3d medical image registration, in: Thirty-Second AAAI Conference on Artificial Intelligence.

Miao, S., Wang, Z.J., Liao, R., 2016. A cnn regression approach for real-time 2d/3d registration. IEEE transactions on medical imaging 35, 1352–1363.

Penney, G.P., Weese, J., Little, J.A., Desmedt, P., Hill, D.L., et al., 1998. A comparison of similarity measures for use in 2-d-3-d medical image registration. IEEE transactions on medical imaging 17, 586–595.

Sheikhjafari, A., Noga, M., Punithakumar, K., Ray, N., 2018. Unsupervised deformable image registration with fully connected generative neural network .

Sherouse, G.W., Novins, K., Chaney, E.L., 1990. Computation of digitally reconstructed radiographs for use in radiotherapy treatment design. International Journal of Radiation Oncology* Biology* Physics 18, 651–658.

Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X., 2020. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Medical Image Analysis , 101693.

Tustison, N.J., Avants, B.B., Gee, J.C., 2019. Learning image-based spatial transformations via convolutional neural networks: A review. Magnetic resonance imaging 64, 142–153.

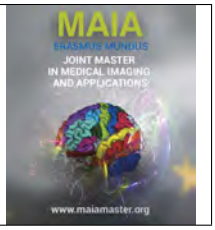
Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H., 2017. End-to-end representation learning for correlation filter based tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2805–2813.

de Vos, B.D., Berendsen, F.F., Viergever, M.A., Sokooti, H., Staring, M., Išgum, I., 2019. A deep learning framework for unsupervised affine and deformable image registration. Medical image analysis 52, 128–143.

de Vos, B.D., Berendsen, F.F., Viergever, M.A., Staring, M., Išgum, I., 2017. End-to-end unsupervised deformable image registration with a convolutional neural network, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 204–212.

Medical Imaging and Applications

Master Thesis, August 2020



MRI Bias Field Correction Using Deep Learning

Aigerim Dautkulova, Jose V. Manjón

IBIME research group, ITACA institute, Universidad Politécnica de Valencia, 46022, Valencia, Spain

Abstract

Magnetic Resonance Images (MRI) are usually affected by intensity inhomogeneities from the MRI acquisition process that difficult the automatic quantification and analysis of medical images. Such intensity inhomogeneity is modeled as a multiplicative low-frequency signal intensity variation across the image commonly referred to as a bias field. In the past, many methods to correct this artifact have been proposed, being most of them based on the optimization of specific image quality criteria. However, optimization methods are time-consuming and sensible to local minima. In this project, we developed different approaches to perform MRI intensity inhomogeneity correction using deep learning. Specifically, we compared two different approaches: supervised and unsupervised. The goal of this work was to implement a new method to automatically correct the bias field using convolution neural networks (CNN). The best results were obtained for the supervised approach. Finally, we show that our proposed supervised approach efficiently outperformed related state-of-the-art methods in terms of accuracy, robustness and efficiency.

Keywords: Magnetic Resonance Imaging, Inhomogeneity Correction, Deep Learning, Convolutional Neural Networks

1. Introduction

Magnetic Resonance Imaging (MRI) is a non-invasive imaging technique that provides detailed images of the interior of the human body allowing the study of its anatomy and structural properties. Its excellent image quality and the use of non-ionizing radiation made it one of the most used image modalities in current clinical practice. However, MR images are affected by different types of artifacts. One of them is the signal intensity inhomogeneity which is produced by imperfections in the radiofrequency coils and object dependent interactions (Sled et al., 1998). Such an artifact is perceived as a multiplicative low-frequency variation of the signal intensity across the image, also known as bias field.

Intensity inhomogeneity in MRI is a major issue, because most automated quantitative methods, such as registration and segmentation, rely on the assumption that a given tissue is represented by similar voxel intensities throughout data. As a result, quantitative parameters computed from corrupted data will likely be erroneous. Therefore, correcting or reducing the effects of this artifact is a crucial preprocessing step for the use of quantitative MRI analysis in research and clinical settings (Belaroussi et al., 2006). Moreover, the benefits of performing the intensity inhomogeneity correction of MRI volumes have been proven in a recent paper (Lee et al., 2018). They showed that there is a significant increase in the accuracy of deep learning-based tumor classification after performing the bias correction of the images.

In this paper, we propose various methods for automatic bias field correction using deep learning. We focused on two different strategies to reach the goal of this work – supervised and unsupervised approaches. The main idea behind the supervised method is to generate the bias-corrected volumes having as an input the original corrupted image and as output its corresponding bias-corrected version. On the other hand, the unsupervised approach is focused on achieving the same goal as supervised but without the use of a ground truth output.

The aim of this work is to improve existing bias-field correction methods and to outperform state-of-the-art approaches in terms of efficiency and effectiveness.

2. State-of-the-art

Intensity inhomogeneity correction is a problem that has been an active research topic and some methods have been proposed over the last few years. These methods can be broadly classified into two different groups – prospective and retrospective.

Prospective correction methods rely on prior information such as factors related to the hardware that corrupt MRI images. Those methods try to minimize the intensity inhomogeneity by acquiring additional images (Axel et al., 1987), merging data obtained from multiple datasets (Liney et al., 1998), combining information from different coils (Murakami et al., 1996), designing dedicated imaging sequences (Deichmann et al., 2002; Mihara et al., 1998). For instance, performing imaging using a combination of volume and surface coils can achieve high uniformity and high signal-to-noise ratio (Narayana et al., 1988). However, the drawback of these methods is the need for additional image acquisition and as a consequence, more prolonged scanning times. Furthermore, the prospective correction methods can eliminate the scanner-related bias field, but not the inhomogeneity that occurs due to the subject's anatomy (Ganzetti et al., 2016).

Retrospective correction methods rely only on image features to remove unwanted intensity inhomogeneities. Theoretically, retrospective algorithms can deal with both scanner and object-related artifacts. Furthermore, those methods usually do not require prior information about the bias field, which makes them more flexible and general than the prospective methods. Therefore, retrospective corrections are widely used and some solutions based on this approach were proposed during the last few years. A number of comparative studies were performed on these methods (Arnold et al., 2001; Likar et al., 2001; Vovk et al., 2007).

Retrospective methods can be further categorized into two subgroups: 1) methods that use a segmentation process to compute the bias field; and 2) those that work directly with the image data (Manjón et al., 2007). In the first group, segmentation based methods are aimed to perform bias correction and tissue classification processes simultaneously. According to different existing segmentation methods, they can be classified into two approaches: 1) expectation-maximization (EM) based algorithms (Guillemaud and Brady, 1997; Leemput et al., 1997; Wells et al., 1996); and 2) Fuzzy C-Means (FCM) based algorithms (Ahmed et al., 1999; Pham and Prince, 1998). In the former approach, the EM algorithm is used for interleaved segmentation and bias correction. These methods use parametric models that are based on a given probability criterion to estimate the bias field. The maximum likelihood (ML) or maximum a posterior (MAP) probability is frequently used as probability criterion. A good example of this type of methods is the well-known SPM12 software (Friston et al., 2011). In the latter – Fuzzy C-Means based – approaches, energy minimization is used to perform simultaneous segmentation and bias correction in which the standard FCM algorithm is used for segmentation (Song et al., 2017). Segmentation-based methods usually require the fixed number of tissue types and they are optimization-based (sensitive to local minima).

The second group of retrospective methods is the algorithms that work directly with the image properties. These methods make minimal assumptions about the image content, such as the number of tissues or locations, which make them more general (Manjón et al., 2007). One of the most used bias correction methods is the Nonparametric Nonuniform Normalization (N3) method (Sled et al., 1998). This method estimates the bias field by sharpening the image histogram using a Gaussian deconvolution and smoothing the obtained bias field estimation by using B-Splines. Due to its performance, this method is called the de facto standard for bias field correction. Recently, Tustison et al. (2010) proposed the N4 method, which is an improved version of the N3 algorithm. In the N4 method, B-spline smoothing strategy was replaced with a modified optimization strategy, which includes a multi-resolution option to capture a range of bias modulation. N4 achieved superior results compared to N3.

The main disadvantage of all of these methods is their parameter dependency. Accordingly, the results of the correction are highly dependent on the correct settings (for example, the variance of the Gaussian filter or the space between B-Spline dots). The Coarse to Fine Bias Corrector (CFBC) method (Manjón et al., 2007) overcome this problem using a progressive course to fine approximation. The other drawback is the iterative nature of the estimation and related optimization problems such as local minima (Manjón et al., 2007).

Over the last few years, deep learning (DL) techniques became state-of-the-art machine learning models across a variety of areas. The development of DL has a huge potential for medical imaging technology, medical data analysis, medical diagnostics, and healthcare in general. In the field of MRI, deep learning has applications at many steps of analysis workflows – image reconstruction (Chen et al., 2018; Hammernik et al., 2017; Yang et al., 2016), segmentation (Akkus et al., 2017; Dalca et al., 2019; Guo et al., 2016), disease prediction (Islam and Zhang, 2018; Lundervold and Lundervold, 2019).

Curiously, in the case of the MRI intensity inhomogeneity correction, there is not much research done using deep learning. Only one paper (Simko et al., 2019) was found regarding this topic, where the authors proposed an Artificial Neural Network called GetNet that is a modified version of ResNet (He et al., 2016). This network was trained on non-medical objects to make the model more general. However, this method was only evaluated in 2D and no extensive validation was per-

formed, which makes it difficult to evaluate its quality.

In this project, we present a parameter-free volume-based 3D convolutional neural network (CNN) that is able to accurately correct the image inhomogeneities in near real-time. The proposed method is insensitive to optimization problems (in fact it can be seen as an amortized optimization) and can be easily integrated into any pipeline for preprocessing purposes.

3. Material and methods

3.1. Problem formulation

An MRI image is usually modeled as follows:

$$Y = xB + n \tag{1}$$

where Y is the observed image, x is true emitted intensity (clean signal), B is the multiplicative bias field which is supposed to be spatially smooth and n is a random additive noise (Manjón et al., 2007). If we don't consider the effect of the random noise n (or we filter it before) the bias correction can be done by dividing the observed intensity Y by the estimation of the bias field B. In this project, to obtain an estimation of the clean signal $\hat{x}$ we multiply observed image Y by the inverse of B for practical reasons (we will directly estimate $1/B$ so we can avoid zero division problems):

$$\hat{x} = Y * (1/B) \tag{2}$$

3.2. Dataset

In this work, we used the publicly available IXI dataset (http://www.brain-development.org/). The data was initially collected as part of the Information eXtraction from Images project. This dataset consists of 580 MRI images from healthy subjects with different sex and ages. Figure 1 shows example brain MRI volumes of the IXI dataset. The images were acquired using two different scanners: Philips Intera 3T and Philips Gyroscan 1.5T. They were collected at three different hospitals in London (UK). The details of the scanner parameters are shown in Table 1.

Table 1: IXI dataset scanner parameters

Parameters	**Philips Intera 3T**	**Philips Gyroscan 1.5T**
Repetition Time	9.6 ms	9.8 ms
Echo Time	4.6 ms	4.6 ms
Flip Angle	8°	8°
Slice Thickness	1.2 mm	1.2 mm
Volume Size	256×256×150	256×256×150
Voxel Dimensions	0.94×0.94×1.2 mm^3	0.94×0.94×1.2 mm^3

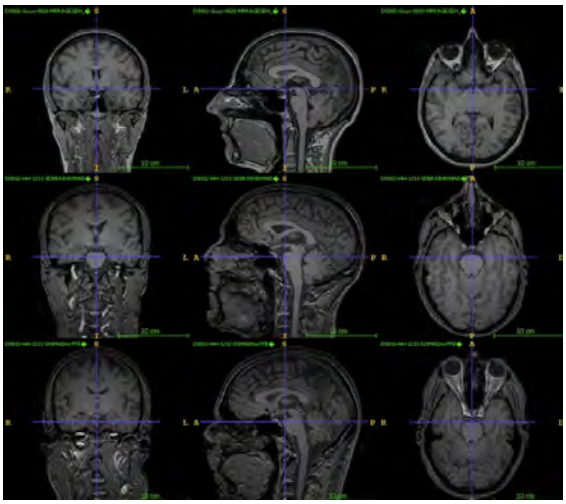


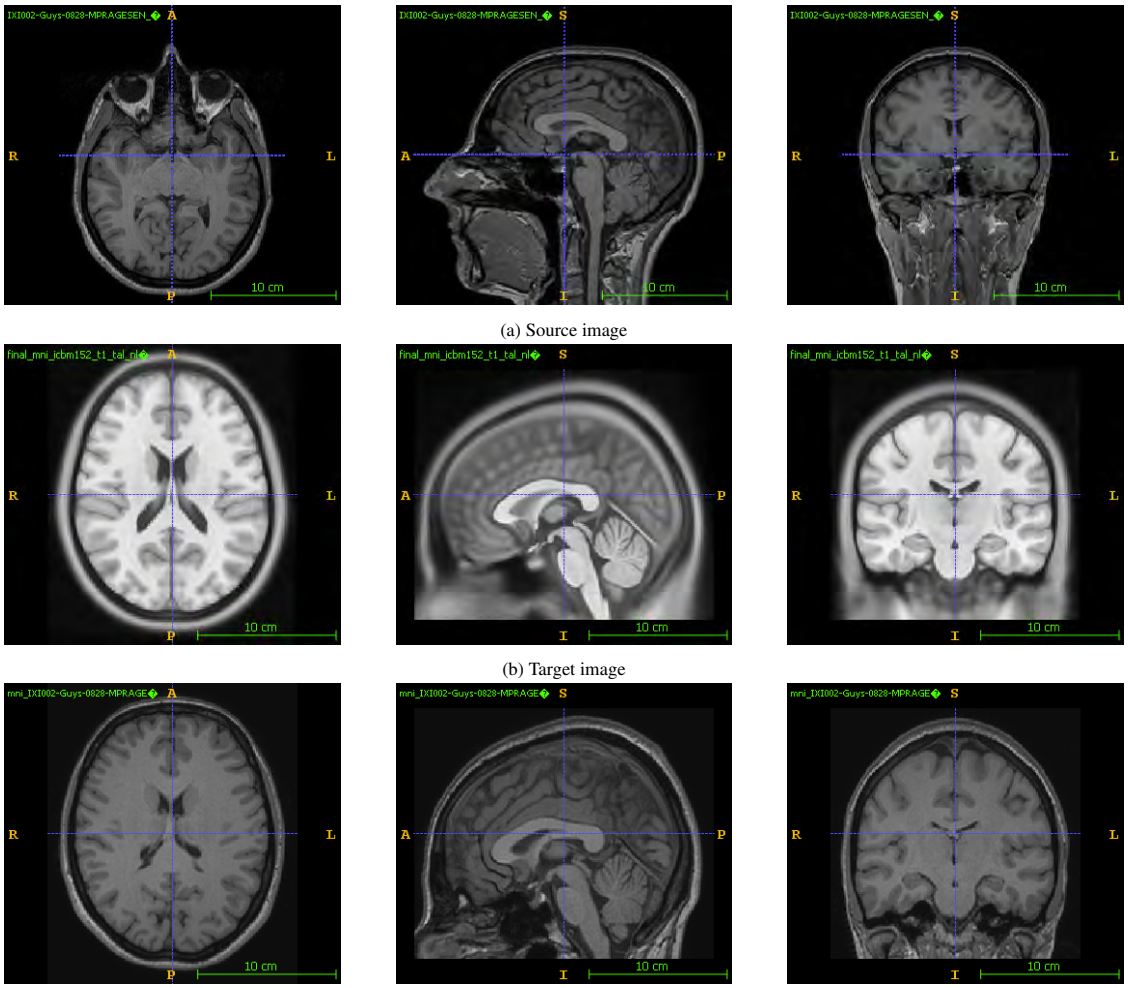
Figure 1: Example brain MRI images of the IXI dataset

3.3. Data preprocessing

MR images are acquired in a large variety of image orientations and resolutions. In order to simplify the bias field correction problem, we preprocessed the images by affine registering the original images to the standard Montreal Neurological Institute (MNI152) space using the ANTS (Avants et al., 2008) software (Figure 2). The standard image in the MNI space has a fixed resolution of 1 mm^3 and dimensions of 181×217×181 voxels. By fixing the orientation and resolution of the images we will require less training data because we do not have to deal with different orientations and resolutions. The corrected image in the native space can be always obtained by applying the corresponding inverse affine transformation. Finally, the intensities of input images were normalized. Two different normalization methods were tested. The first approach is a mean normalization consisting of dividing the volume by its mean value so all the volumes share a mean of one. In this approach, all the intensities of the volume are restricted to positive values. The other approach is the classical z-scoring consisting of subtracting the mean of the volume and dividing by its standard deviation.

3.4. Evaluation metrics

It is generally accepted that the spatial intensity distribution in the MRI volume is piece-wise constant and that each tissue type is represented by similar intensities corresponding to a unique grayscale level. Based on these hypothesis, a valid intensity inhomogeneity correction method should decrease the standard deviation in intensity for each tissue (Belaroussi et al., 2006). The coefficients of variation CV is the ratio of the standard deviation to the mean. It is used for measuring the homogeneity of the WM and GM of the brain. To evaluate the performance of the proposed methods CV of white matter (WM) and grey matter (GM) were employed.



(a) Source image

(b) Target image

(c) Registered image

Figure 2: Example of the image registration to the standard MNI space. (a) Original brain MRI from IXI dataset; (b) MNI volume used as a target for registration; (c) Brain MRI registered to MNI space

The formulas for computing those metrics are the following:

$$CV_{WM} = \frac{\sigma(WM)}{\mu(WM)} \tag{3}$$

$$CV_{GM} = \frac{\sigma(GM)}{\mu(GM)} \tag{4}$$

where σ is the standard deviation and μ is the mean of the tissue intensities.

A modification of the CV value is the Coefficient of Joint Variation (CJV), which also measures the overlap between tissue distributions. The formula for computing the CJV is following:

$$CJV(WM,GM) = \frac{\sigma(WM) + \sigma(GM)}{\mu(WM) - \mu(GM} \tag{5}$$

Lower values of CV_{WM}, CV_{GM}, and $CJV(WM,GM)$ coefficients indicate better corrections. To compute these metrics a posteriori probability maps produced by SPM12 were used. They were thresholded by the value of 0.9 to avoid the partial volume contamination.

3.5. Supervised approach

The overall pipeline for supervised bias field correction approach is depicted in Figure 3. A detailed description of the steps performed is presented in the following sections.

3.5.1. Training data

In the proposed supervised approach, we need the corresponding bias free images as ground truths for our input bias corrupted images. Unfortunately, we have no access to the ideal bias free images as they are unknown. To generate the bias free output images of the network all the images were bias corrected using the SPM12 tool (Friston et al., 2011). We are aware that this is just a surrogate of the real bias free images but we found it effective enough for the purposes of this work. We chose SPM12 method based on the results of the comparison (Figure 4) of three different bias correction methods (SPM12, N4 and CFBC). The Coarse to Fine Bias Corrector (CFBC) is an inductive method that proposes possible bias field models and keeps the most probable

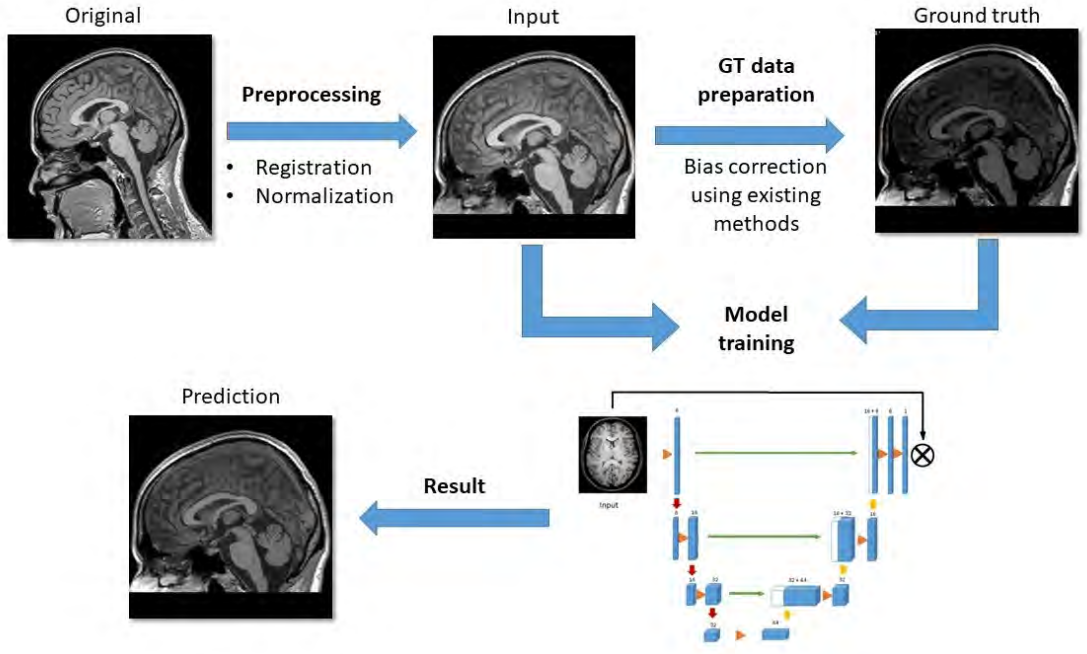


Figure 3: Graphical representation of the general pipeline of the proposed supervised approach for MRI intensity inhomogeneity correction

one instead of obtaining it directly from the data by applying some transformation on it (Manjón, 2006). To measure the homogeneity of the different brain tissues, we used the tissue segmentations provided by SPM12 (white matter, gray matter, cerebrospinal fluid).

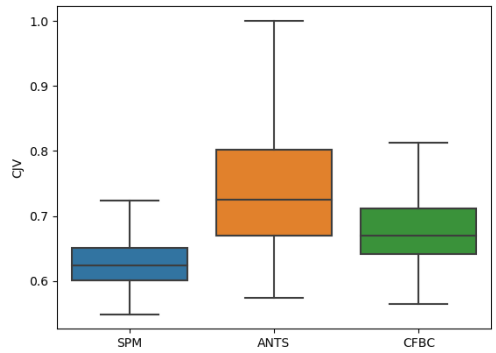


Figure 4: Comparison of three existing bias-correction methods on the basis of the mean coefficient of joint variation (CJV) value computed for the IXI dataset. The smaller box the less dispersed the CJV values

3.5.2. Network architecture

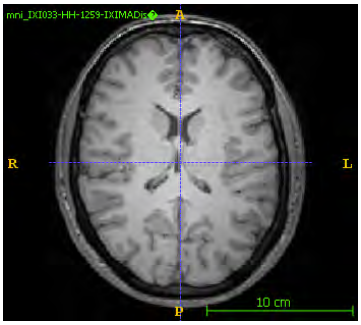
The U-Net is a convolutional neural network that was first proposed by Ronneberger et al. (2015) for biomedical image segmentation. It is basically a modified autoencoder with resolution dependent shortcuts. It consists of four main parts: encoder, bottleneck, decoder and shortcuts. The network architecture is a sequence of convolution plus pooling layers that first reduce the spatial resolution of the image, and then increase it by combining it with the image data and feeds to the consecutive convolution layers. The last layer normally uses a 1×1×C convolution kernel to map the final feature vector to the desired number of classes (C). In this work, we have applied various modifications to the original U-Net architecture, which will be explained in the following sections.

Model 1. Figure 6 illustrates our modified U-Net architecture, where the encoding part of the model is composed of three blocks. Each block is made of a 3D convolution layer (7×7×7) with ReLU activation function and the batch normalization followed by the max-pooling layer. The number of filters in the first resolution level are 8, 16 and 32 in the following levels. The bottleneck is composed of a 3D convolution layer with the ReLU activation, 64 filters and the batch normalization. The decoding part is made of three blocks similarly to the encoding path. Each block is composed of an upsampling layer, concatenation with the corresponding part of the downsampling path, 3D convolution layer with the batch normalization. The number of filters in this path are 32, 16, and 8. An important thing to highlight in Model 1 is that to generate bias-free volume in the last layer we multiply the inverse of the bias field with the original input volume. Thus, the network predicts the inverse bias field and multiplies it to the input volume to generate the bias free output. The resulting network has a total of 25 layers and 2,308,337 trainable parameters.

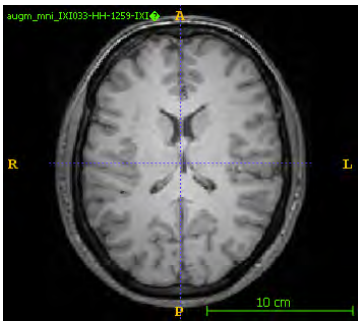
The input of the model is the original raw volume registered to the MNI space presented as an input tensor

and the output of the network the corresponding bias-corrected volume.

We performed data augmentation in order to increase the number of training samples to reduce overfitting and improve generalization of the model. The augmentation is performed within the data generator function. It is done by left-right flipping the input and output images randomly taking benefit from the pseudo-symmetry of the human brain. The example image augmentation is depicted in Figure 5.



(a) Original brain MRI



(b) Augmented brain MRI

Figure 5: Example of the data augmentation. (a) original brain MRI image; (b) augmented brain image acquired by left-right flipping

Model 2. Figure 7 shows another modified version of the U-Net architecture, which is a smaller variation of the Model 1. The only change compared to Model 1 is that its output is modified to have a feature encoding branch that measures feature differences between the predicted image and the target image. This is accomplished using a loss function that not only measures the reconstruction error of the predicted image but also its feature error through the use of a multiscale encoder with shared weights. The Model 2 has 33 layers and 2,769,497 trainable parameters.

Adaptive smoothing layer. Since the proposed network tries to predict a low-frequency bias field, we thought that blurring the input volume will help in the estimation. However, since we do not know the degree of smoothness required we developed a custom adaptive smoothing layer that can be used as the first layer of the network. The adaptive smoothing is a custom layer that performs the Gaussian blurring of the volume with a learnable kernel size.

The main reason to use the smoothing is to increase the signal-to-noise ratio. According to the theory of Matched Filter, smoothing will maximize this coefficient. Based on the idea of this theorem, the Gaussian kernel will give an optimum resolution of signal from the noise that we are looking for. Therefore, the noise which produces intensity inhomogeneity will be best detected after the smoothing process.

3.5.3. Loss Functions

The supervised network learns a mapping between the input image to the target and the loss function should be chosen on the basis of the specific task pursued by the model. In our case, the task is to minimize the intensity inhomogeneity present on the image. In this work, we employed different loss functions, which will be shown in this section.

Mean Absolute Error (MAE). The MAE is calculated by taking the average of the absolute difference between the ground truth value and the model prediction. Compared to the Mean Squared Error loss MAE encourages less blurry and higher quality images (Thomas, 2020).

Custom loss. Combining the MAE of the intensities and the MAE of the gradient difference loss (MAE-G). MAE loss compares the similarity between the images, but does not tell the information about the inhomogeneity of the images. The MAE of the gradient difference (MAE-G) loss function was used in order to capture this information. The gradient of the image is the directional change in the intensity of the image. The bias field distorts the intensity distributions of the MRI, thus, the gradients of the clean and corrupted by intensity inhomogeneity volumes are different. The aim of the Custom loss is to achieve similar intensities and also similar gradients for the compared input and target ground truth volumes.

3.5.4. Feature matching

The Mean Absolute Error loss is not designed to capture the texture of the image and the different frequency sub bands. For this reason, in this work, a Feature loss was introduced. This proposed loss was implemented as an Encoder function, where the image is transferred from the original to the smaller representation. This allows to capture different features of the image while performing the convolution operations. More convolutional layers allow to get different features at different scales. In this work, we use three convolutional layers in the feature encoders (see Figure 7).

We apply the encoder function to both input and target images. Important note here is that the same Encoder should be used to both images. Therefore, if the images are similar the encoding will be similar. Encoding does not measure the intensity but feature similarity. It compares different features of the image on low,

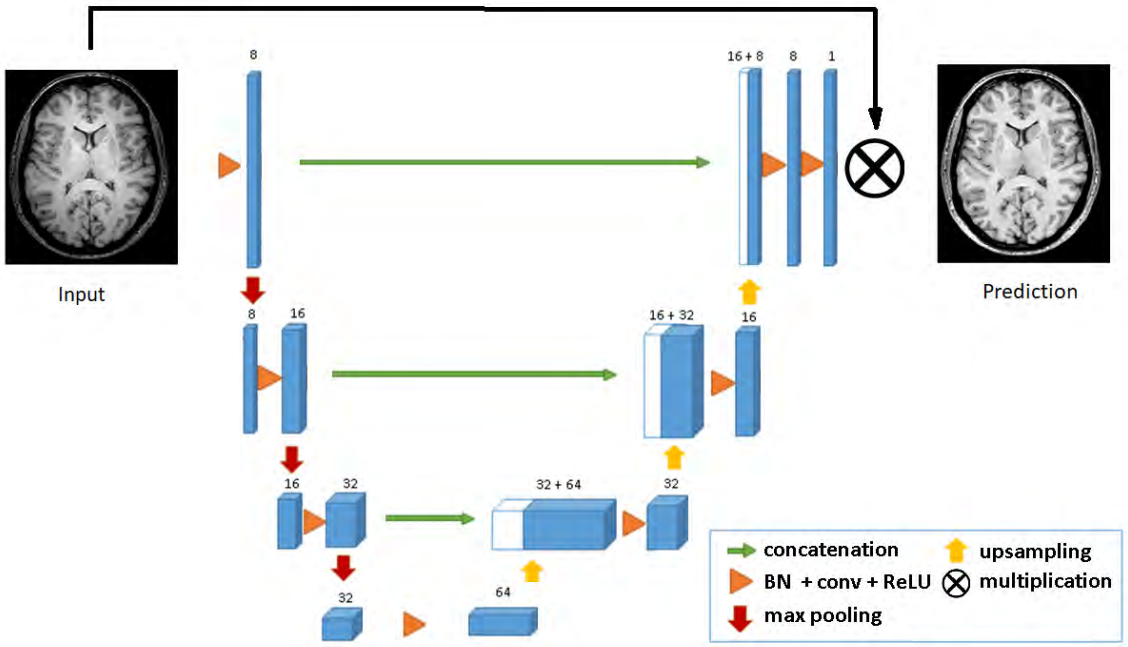


Figure 6: Network architecture: Model 1

medium and high-frequency levels. The Feature loss can be defined as:

$$\begin{aligned} FeatureLoss = &\sum |F1(Pi) - F1(Ti)| \\ &+ \sum |F2(Pi) - F2(Ti)| \\ &+ \sum |F3(Pi) - F3(Ti)| \end{aligned} \tag{6}$$

where $F1$, $F2$ and $F3$ are image features at different scales. P is the predicted image and T is the target image.

3.6. Unsupervised approach

In the unsupervised approach, we aim to train the model without the ground truth labels. The model is forced to learn connections inside the image and make assumptions based on them. Different methods that were used for the unsupervised approach are described in the following subsections.

3.6.1. Model architecture

Model 3. The architecture used in the unsupervised approach has the same structure as Model 1 but with two outputs instead of one, the inhomogeneity corrected volume and the estimated inverse bias field. The input of the model is the original raw volume registered to the MNI space. The output of the model is not known in this case. Therefore, we used loss metrics that are related to enforce the image homogeneity. The last layer multiplies the estimation of the inverse of the bias field with the input image to generate a bias-corrected volume. As a result, we receive the approximation of the bias field and the bias-corrected volume.

R-Model 3. Another version of Model 3 was obtained by slightly modifying its architecture. In Figure 8 the reduced version of Model 3 (R-Model 3) is presented. We jumped over two resolution levels from the decoder path and directly upsampled the image from the lower resolution to the spatial resolution of the original input image. Therefore, the model architecture became an asymmetrical encoder-decoder with simplified decoder part.

3.6.2. Loss functions

Since the output of the unsupervised model has multiple values we use a different loss function. Therefore, the final loss function is made of two parts:

$$L_{total} = \lambda_1 L_{image} + \lambda_2 L_{bias} \tag{7}$$

where L_{image} is a custom loss function that is applied to the bias-corrected image prediction, and L_{bias} is another custom loss function that measures the properties of the estimated bias field of the image, λ_1 and λ_2 are corresponding loss weights. The loss weights are used to balance the contribution of each output loss to the final loss. The default values for loss weights are [1, 1].

For the image loss (L_{image}), we applied functions able to measure the homogeneity of the image: median gradient loss and normalized cross-correlation. This loss can be defined as:

$$L_{image} = (1 - NCC) * \mu(G_{image}) \tag{8}$$

where NCC is the Normalized Cross Correlation and $\mu(G_{image})$ is the mean gradient of the image.

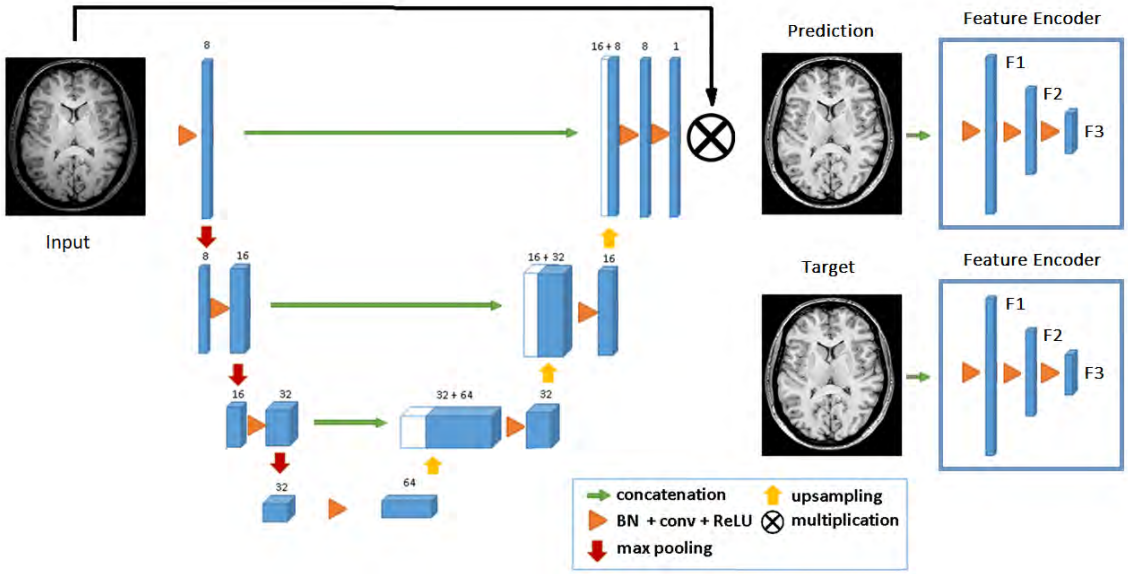


Figure 7: Network architecture: Model 2 including feature encoder modules with shared weights

At the same time, for the bias field (L_{bias}) term we used metrics able to assess the smoothness of the field, the mean gradient loss and amplitude constraint:

$$L_{bias} = L_{AC} * M(G_{bias}) \tag{9}$$

where L_{AC} is the amplitude constraint and $M(G_{bias})$ is the median gradient of the predicted bias field.

Normalized Cross Correlation (NCC). The NCC metric is commonly used in the evaluation of the similarity between two images (Rao and Yerravelli, 2014). The normalized cross correlation can be defined as:

$$CC = \frac{\sum_i \sqsubset (x(i) - mx) * (y(i - d) - my) \sqsupset}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i - d) - my)^2}} \tag{10}$$

where x and y are series between which we compute the correlation, and mx, my is their means. The maximum value occurs when the compared images perfectly match each other, and 0 if the images are uncorrelated.

Gradient loss (G_{image} and G_{bias}). Homogeneous images show well-ordered intensities and well clustered low gradient values in homogenous regions (Manjón et al., 2007). Therefore, we use the gradient of the image as a loss function, which we want to minimize. For the output of the model that is the predicted image, we use the median value of its gradient. We compute the gradient using the differences between neighboring voxels in x, y, z directions. The median ignores outliers in data distribution. For the bias field, we take the mean gradient. Where the gradient G of the three-dimensional image function F can be defined as:

$$G = \nabla F(x, y, z) = \sqsubset \frac{dF}{dx}, \frac{dF}{dy}, \frac{dF}{dz} \sqsupset \tag{11}$$

Amplitude Constraint (L_{AC}). In order to restrict the range of the bias field, we apply the amplitude constraint to the second part of the model output. The formula used for this loss function can be written as:

$$L_{AC} = \|1 - \mu(y_{pred})\|^2 \tag{12}$$

where y_{pred} is the approximation of the bias field given by the model and $\mu(y_{pred})$ is its mean.

4. Experiments and Results

The data processing and all experiments were carried out at IBIME lab at the Polytechnic University of Valencia using a desktop PC with an AMD Ryzen 7 processor with 16 GB RAM running Windows 10. The model was implemented using the Keras 2.3.1 (Chollet, 2015) deep learning library on top of the Tensorflow 1.15 (Abadi et al., 2016) in Python 3.6.

4.1. Training images

The IXI dataset (N=580) was randomly divided into the following subsets: training, validation and testing. In order to load the training set, we use the data generator function with a batch size of 1. Three strategies were followed regarding the bias corrected images used as an output of the model. The testing of all three strategies was performed on the same dataset of 20 brain MRI volumes.

- *SPM12 only:* First we used the MRI volumes corrected by the SPM12 software. In this case, all training images were bias corrected using the same software. 550 images were used for training and 10 for validation.
- *Selected SPM12 cases:* The second strategy was to choose only a subset of the best corrected images

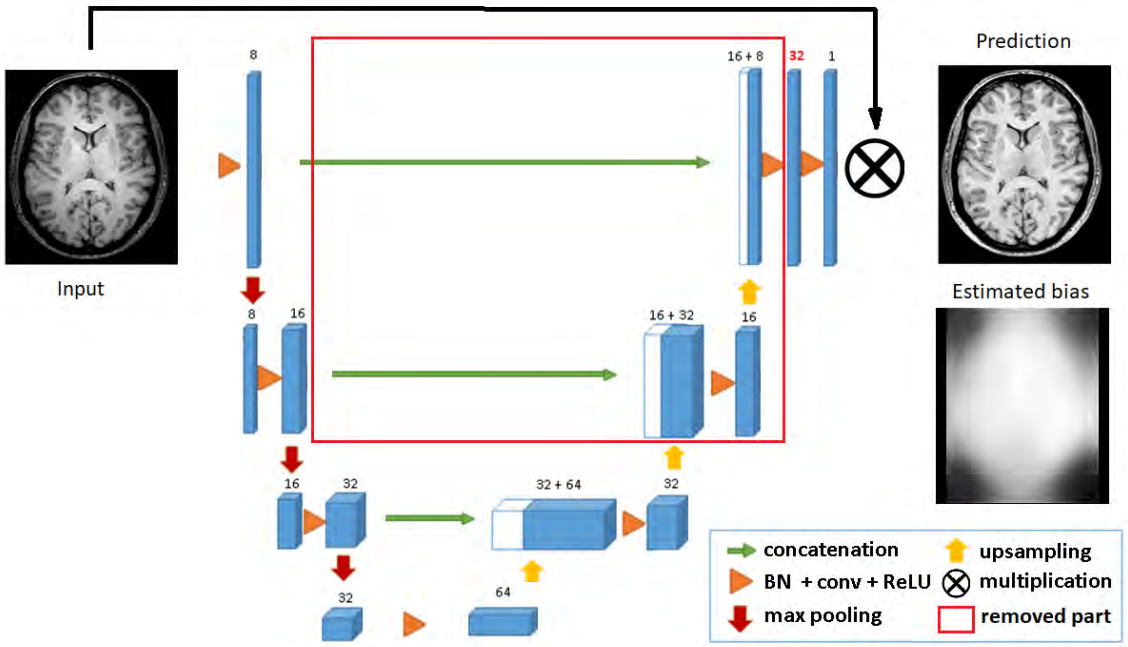


Figure 8: Network architecture: R-Model 3. Inside the red box network parts that were not used in the R-Model 3 architecture compared to the Model 3 network

using SPM12. This was done upon the observation that SPM12 does not correct all the images equally good, due to its optimization based nature. For this case, we drew the histogram of all CJV values of the dataset (Figure 9) and chose a value of 0.9 for the threshold. Using this threshold 303 MRI images in total were chosen. Of the selected volumes 10 were used for validation, and the remaining for training.

- *Best selection:* The third strategy is to select the best bias-corrected images from three different correction approaches (SPM12, ANTS, and CFBC). Based on their CJV metric, from the original 580 images, 496 cases were collected from SPM12, 65 from CFBC, and 19 from ANTS. In this case, 550 images were used for training and 10 for validation.

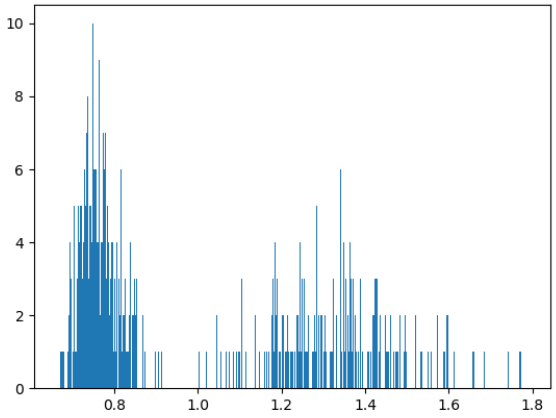


Figure 9: CJV values of the IXI dataset. Note that two clusters are clearly visible. Left one represents more homogenous cases and right ones more heterogeneous (representing suboptimal bias field correction)

4.2. *Intensity normalization and testing strategy*

Some layers (the batch normalization in our case) perform differently during the training and testing phase. In the batch normalization during the training mode in order to rescale the input of the layer the mean and the variance of the mini-batch are used. In the testing mode, a historical moving average and variance are used that was computed during the training of the model. Therefore, this behavior of the layer in the testing mode leads to suboptimal results when using the small size of the batch, in this project N of 1. For that reason, during the testing of the model, we perform the prediction in the training mode. We call this strategy Training Time Batch Normalization (TTBN).

The effect of the pre and post processing in the final outcome of the method was tested. We compared the use of two different normalization techniques: z-scoring and mean-ratio method. In Table 2 the results of the different experiments are shown. In terms of the normalization methods, the mean-ratio normalization approach showed better performance than classic z-scoring. This result was expected, since the mean-ration normalization forces that both input and output images to be positively defined (bias field is positive-definite). Further-

Table 2: Test results from different options of the method using Model 2 architecture and best selection dataset. Lower coefficients indicate better correction. Best results in bold

Metric	Z-scoring	Z-scroing + TTBN	Mean norm.	Mean norm + TTBN
CV_{WM}	0.0853 ± 0.0067	0.0799 ± 0.0101	0.0452 ± 0.0040	**0.0407** ± **0.0053**
CV_{GM}	0.2216 ± 0.0263	0.2229 ± 0.0240	0.1157 ± 0.0103	**0.1132** ± **0.0114**
CJV	0.7433 ± 0.0557	0.7308 ± 0.0456	0.6829 ± 0.0525	**0.6299** ± **0.0354**

more, the use of TTBN compared to classical prediction in both cases showed an improvement in the performance.

4.3. Supervised approach

The experimental results for the supervised approach are given in Table 3. The conducted experiments can be divided into three groups according to the method that was used for the bias correction of the ground truth volumes: 1) SPM12 only; 2) Selected SPM12 cases; 3) Best selection method. All supervised methods presented in the following sections were trained using Adam optimizer (Kingma and Ba, 2015) with default parameters. Data generator function with batch size 1 was used to feed the network during 300 epochs (50 cases per epoch). The kernel size 7 was experimentally chosen as the best fit for this project, and the filter size 8 was used.

4.3.1. Experiments with the SPM12 bias-corrected dataset as ground truth

Model 1. Experiments using the Model 1 network architecture were performed using different loss functions and proposed adaptive smoothing layer The initial experiment was the bias correction using the MAE loss which obtained a mean CJV value of 0.6423. The next change was the use of the adaptive smoothing as a first layer of the model. The method in this experiment with the MAE loss got the CJV of 0.6414, while with the MAE_G loss obtained a value of 0.7442. Transfer learning with a former model to the latter model was applied and got a CJV of 0.6439. This test showed a noticeable difference between the performance of the model with different loss functions.

Model 2. Experiment using Model 2 with the proposed feature matching technique and the combined loss function that is made of MAE and MAE_G loss functions was performed. This model showed the mean CJV value of 0.6974.

The best result for this set of experiments was obtained using Model 1 architecture.

4.3.2. Experiments with the selected SPM12 bias-corrected dataset as ground truth

Model 1. The initial experiment of this set was the use of the MAE loss and training the model from scratch. This method obtained the mean CJV value of 0.7904. After, we trained the same model but the MAE loss was changed to the proposed MAE_G loss function and obtained a value of 0.7849.

Experiments with the reduced SPM12 dataset did not show better performance compared to experiments with the full dataset. Therefore, further experiments that used the selected SPM12 dataset were discarded.

4.3.3. Experiments with the selected best dataset from the three state-of-the-art methods

The best results of three available bias correction methods were combined to one dataset of 580 images.

Model 1. Experiment with the Model 1 network architecture and MAE loss obtained the mean CJV value of 0.6954.

Model 2. Model 2 architecture that was trained using the selected dataset and MAE loss got a value of 0.6416, while the same model with the combined loss (MAE and MAE_G) obtained the mean CJV of 0.6299. The same model as previously used with the adaptive smoothing layer showed the CJV of 0.6459.

From this set of experiments can be seen that the Model 2 network architecture performed better compared to Model 1 using the selected best dataset. Furthermore, the adaptive smoothing layer was not able to improve the result obtained by Model 2.

4.3.4. Comparison of the best obtained result with state-of-the-art

The comparison of the existing bias correction methods with the proposed models is presented in Table 4. As we can see our proposed Model 2 outperformed compared bias correction methods for the CJV and CV_{GM} metrics and for the CV_{WM} value behaved similarly to the SPM12. It is worth to note that the proposed method had also the lowest dispersion (standard deviation).

In Figure 10 each boxplot shows the value of the dispersion of CJV coefficients in the test dataset for four

Table 3: Supervised approach. Mean CJV values obtained for different strategies using three different ground truth datasets in the supervised approach (MAE – mean absolute error; MAE_G – mean absolute error of the gradient difference; AS – adaptive smoothing; FM – feature matching; TL – transfer learning). The best result for each set is shown in bold

No	Model	Loss	AS	FM	TL	Mean CJV
SPM12 only						
1	Model 1	MAE	No	No	No	0.6423
2	Model 1	MAE	Yes	No	No	**0.6414**
3	Model 1	MAE_G	Yes	No	No	0.7442
4	Model 1	MAE_G	Yes	No	Yes	0.6439
5	Model 2	MAE + MAE_G	No	Yes	No	0.6974
Selected SPM12 cases						
6	Model 1	MAE	No	No	No	0.7904
7	Model 1	MAE_G	No	No	No	**0.7849**
Best selection						
8	Model 1	MAE	No	No	No	0.6954
9	Model 2	MAE	No	Yes	No	0.6416
10	Model 2	MAE + MAE_G	No	Yes	No	**0.6299**
11	Model 2	MAE + MAE_G	Yes	Yes	No	0.6459

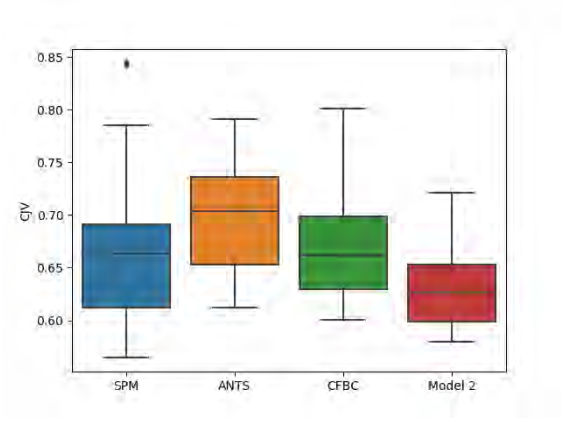


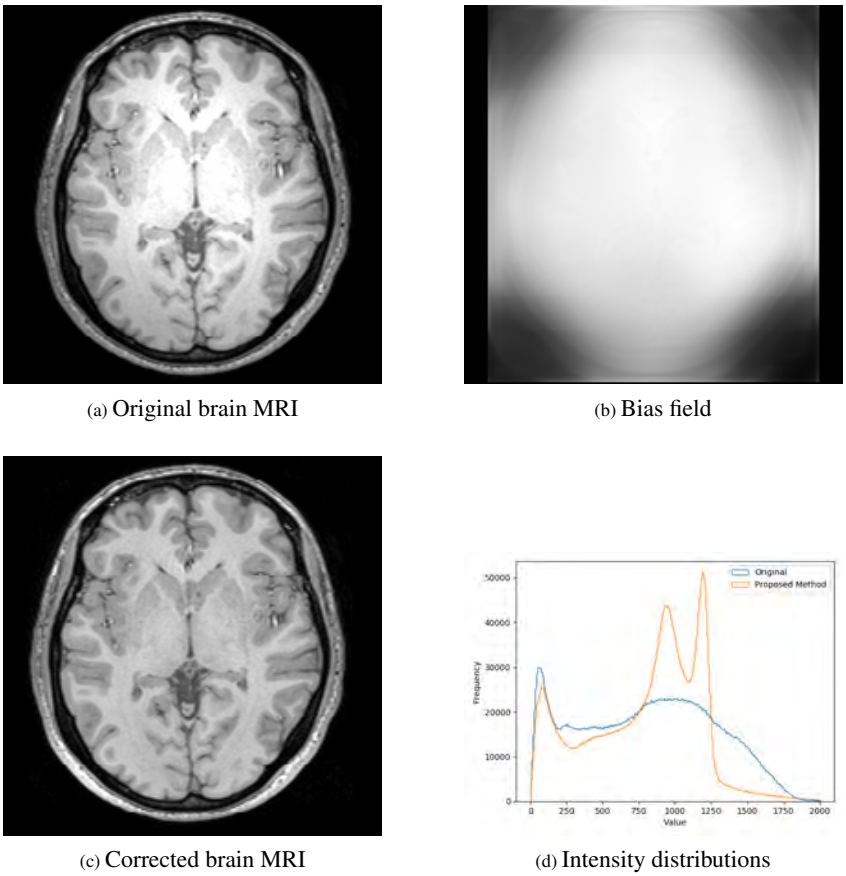
Figure 10: Mean CJV values for the test dataset corrected by three different bias correction methods and by proposed Model 2. The proposed network was trained on 580 images (filtered by an ensemble of methods) for 300 epochs

methods. The smaller box length denotes the less dispersed data. Furthermore, the ends of two whiskers show the range of scores. Thus, the smaller ranges indicate less scattered data. Therefore, we can conclude that the results of Model 2 were less dispersed and scattered compared to other examined methods.

In Figure 11, the example of the bias correction is depicted. We can visually notice the difference between the image containing intensity inhomogeneity and the bias-corrected volume. In Figure 11d the histogram of intensity distributions of this example image and its bias-corrected version is presented. It is readily apparent that the intensity distribution of the bias-corrected volume is more reasonable than the distribution of the original image because three peaks corresponding to the three tissue type can be clearly differentiated.

4.4. Unsupervised approach

We tested the effect of applied changes in Model 3 network architecture. In Table 5 the comparison of the proposed Model 3 and its reduced version R-Model 3 is shown. As can be seen, the latter model showed better performance for all three evaluation metrics. These results can be explained by a fact, that the bias field is



(a) Original brain MRI (b) Bias field

(c) Corrected brain MRI (d) Intensity distributions

Figure 11: Example of the bias correction. (a) original not corrected data; (b) estimated bias field; (c) corrected brain MRI volume; (d) intensity distributions histogram of the original and corrected volume

a low-frequency signal and by upsampling the image from the lower resolution right to the higher resolution we force the model to remove these low-frequency artifacts. Moreover, the execution time of the reduced model is twice faster than Model 3. Therefore, R-Model 3 architecture was used in the conducted experiments.

To train the network we used two different optimizers with their default parameters: 1) Adam; 2) AdaBound (Luo et al., 2019). The loss function used in the training process is custom L_{total} loss. To feed the network during 50 epochs (10 cases per epoch) a data generator function was used with batch size 1. Similarly to the supervised method, data augmentation within the generator was performed. The results of performed experiments with the unsupervised approach are presented in Table 6.

The experiments can be divided into two groups on the basis of the used optimizer:

- *Adam.* The starting point of experiments was the use of default values for the loss weights, which is [1, 1]. The result obtained was the CJV of 0.8483, while for [0.5, 1] we got the CJV value of 0.7843. Afterwards, the adaptive smoothing layer was used in the proposed model as a first layer and the obtained result was a value of 0.8411. These results show that better performance of the model using the Adam optimizer was achieved by giving smaller weights to the L_{image} loss.

- *AdaBound.* Using the AdaBound optimizer training of the model was performed using two different loss weights: [1, 1], [0.5, 1]. For the former case resulting CJV was 0.8869, for the latter we got a value of 0.7889. With the use of the adaptive smoothing layer proposed model obtained the CJV of 0.8541. It can be seen that the model using [0.5, 1] loss weights performed better compared to standard weights.

From the results obtained, we can conclude that both of the tested optimizers perform similarly and do not show a big difference in model results. The use of the adaptive smoothing layer did not show an improvement in performance of the model. However, the best-obtained result was achieved using Adam optimizer.

4.4.1. Comparison of the proposed unsupervised method with the state-of-the-art

The comparison of bias correction methods with the proposed unsupervised method is presented in Table 7. The lower value indicated better correction of the vol-

Table 4: Bias Correction Methods Comparison with the Proposed Supervised Approach. Best results in bold

Metrics	Original	SPM12	ANTS	CFBC	Model 2
CV_{WM}	0.0951 ± 0.0322	**0.0392 ± 0.0074**	0.0443 ± 0.0067	0.0487 ± 0.0118	0.0407 ± 0.0053
CV_{GM}	0.1540 ± 0.0257	0.1144 ± 0.0135	0.1221 ± 0.0126	0.1255 ± 0. 0143	**0.1132 ± 0.0114**
CJV	0.9336 ± 0.2963	0.6677 ± 0.0718	0.6988 ± 0.0534	0.6725 ± 0.0545	**0.6299 ± 0.0354**

Table 5: Test results from proposed models. Lower coefficients indicate better correction. Best results in bold

Metric	Model 3	R-Model 3
CV_{WM}	0.0955 ± 0.0311	**0.0805 ± 0.0272**
CV_{GM}	0.1541 ± 0.0243	**0.1399 ± 0.0185**
CJV	0.9412 ± 0.2942	**0.8483 ± 0.2460**
Trainable params.	2,066,801	1,803,265
Execution time	0.8 s	0.4 s

ume. Compared to the existing bias correction methods proposed method was not able to perform better by the value of CJV and CV_{WM} metric. However, the proposed method obtained a similar mean CJV and lower standard deviation for the CV_{GM} metric compared to the CFBC approach.

5. Discussion

In this paper, we presented a new method for MRI bias field correction using deep learning. Two different approaches were implemented – supervised and unsupervised. It was possible to achieve state-of-the-art results using the supervised training approach. Using the unsupervised approach, we could not reach the same results. However, the experiments showed considerable improvement from the original results.

The dataset used in this project does not contain ground truth of the bias field, therefore, we had to perform the bias correction and make ground truth images before the work with the models. As we know, intensity inhomogeneity correction tools do not perform the bias correction equally good for all the images. The reason for that is the optimization nature of those tools. Initially, SPM12 was used for the bias field correction task of the whole dataset, after we decided to check the performance of different bias correction tools. For this purpose, we compared the corrections of three methods: SPM12, N4, and CFBC. As the experimental results showed, the SPM12 performed better than the others. When interpreting the previously obtained results in percentage terms we observed that the dataset combined from different correction methods contains 85.5% images processed by SPM12, 11.2% by CFBC method, and 3.3% by ANTS.

We trained the models using both datasets and as it was expected that the network trained using a combined dataset performed better than the one corrected only by SPM12 toolbox. The explanation for this is that by training on the ensemble of methods we are getting the best of each method because these approaches have their strengths and weaknesses.

It was somehow unexpected that the proposed approach would perform better than existing bias correction methods as initially it was trained to mimic the SPM12 behavior. However, we believe that the proposed method was able to learn the average concept of the bias field from the training volumes though some of the samples were better corrected than the others (due to the optimization nature of the existing methods).

Moreover, it was experimentally proven that performing the test of the model in the training mode improves the performance of the network. The reason for the better results is that in the training mode batch normalization layers use current values for the mean and standard deviation preventing the suboptimal results when using small batch size.

For the unsupervised method, we were not able to get the result better than the other methods, but the experiments showed great improvement in results. Two different network architectures were tested and a model with fewer layers showed better performance compared to the original one. We think that the reduced model was able to better deal with the low-frequency noise from the input volume. Moreover, the impact of different loss weights was shown during the conducted experiments. Giving a smaller weight to the image loss and bigger weight to the bias field loss has brought significant changes to the model performance.

6. Conclusions

In this paper, we proposed a novel method for MRI bias field correction using deep learning. Two different

Table 6: Unsupervised approach. Mean CJV values obtained for different experiments using the IXI dataset in the unsupervised approach (AS – adaptive smoothing). The best result for each set is shown in bold

Original IXI dataset

No	Model	Loss weights [L_{image}; L_{bias}]	Optimizer	AS	Mean CJV
1	R-Model 3	[1, 1]	Adam	No	0.8483
2	R-Model 3	[0.5, 1]	Adam	No	**0.7843**
3	R-Model 3	[0.5, 1]	Adam	Yes	0.8411
4	R-Model 3	[1, 1]	AdaBound	No	0.8869
5	R-Model 3	[0.5, 1]	AdaBound	No	**0.7889**
6	R-Model 3	[0.5, 1]	AdaBound	Yes	0.8541

Table 7: Bias Correction Methods Comparison with the Proposed Unsupervised Approach. Best results in bold

Metrics	Original	SPM12	ANTS	CFBC	R-Model 3
CV_{WM}	0.0951 ± 0.0322	**0.0392 ± 0.0074**	0.0443 ± 0.0067	0.0487 ± 0.0118	0.0610 ± 0.0069
CV_{GM}	0.1540 ± 0.0257	**0.1144 ± 0.0135**	0.1221 ± 0.0126	0.1255 ± 0. 0143	0.1255 ± 0.0084
CJV	0.9336 ± 0.2963	**0.6677 ± 0.0718**	0.6988 ± 0.0534	0.6725 ± 0.0545	0.7843 ± 0.0830

approaches were followed in this work, supervised and unsupervised.

We used the U-Net model with a multiplicative residual connection. The IXI dataset that was used for the training has no ground truth of the bias field, therefore the intensity inhomogeneity correction was performed using different existing tools (SPM12, ANTS, and CFBC). It was shown that the model trained on the dataset corrected by the ensemble of these methods performed better than the one that uses a single method. From an efficiency point of view, the proposed supervised method is able to correct 3D brain volume in 1 second which is 30 times faster than ANTS and 300 times faster than SPM12. We showcased the impact of proper data normalization on the quality of the proposed method. Moreover, an improved prediction strategy using the batch normalization in training mode at test time (TTBN) was presented.

The proposed approach outperformed the related state-of-the-art methods in both efficiency and performance terms. Furthermore, the proposed bias correction method is fully automatic and does not require any information about the tissue types or intensity distributions, and can be included in complex pipelines as part of the preprocessing.

The other proposed unsupervised method was not able to outperform the existing methods, but it showed promising results during the work on this project. Therefore, future work might be done on the development of this approach.

7. Acknowledgments

I would like to thank my supervisor Jose Manjón for the support and encouragement throughout this project. I would also like to thank Kaisar Kushibar for support and motivation. Finally, I would like to thank IBIME research group (Biomedical Informatic Research Group, Polytechnic University of Valencia) for providing the resources needed for the project implementation.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283.

Ahmed, M.N., Yamany, S.M., Farag, A.A., Moriarty, T., 1999. Bias field estimation and adaptive segmentation of MRI data using a modified Fuzzy C-Means algorithm. Proceedings. 1999

IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149) 1, 250–255. doi:10.1109/42.511747.

Akkus, Z., Galimzianova, A., Hoogi, A., 2017. Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. J Digit Imaging 30, 449–459. doi:10.1007/s10278-017-9983-4.

Arnold, J.B., Liow, J.S., Schaper, K.A., Stern, J.J., Sled, J.G., Shattuck, D.W., Worth, A.J., Cohen, M.S., Leahy, R.M., Mazziotta, J.C., Rottenberg, D.A., 2001. Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects. Neuroimage 13(5), 931–943. doi:10.1006/nimg.2001.0756.

Avants, B.B., Epstein, C.L., Grossman, M., Gee, J.C., 2008. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Medical image analysis 12(1), 26–41.

Axel, L., Costantini, J., Listerud, J., 1987. Intensity correction in surface-coil MR imaging. AJR Am J Roentgenol 148(2), 418–420. doi:10.2214/ajr.148.2.418.

Belaroussi, B., Milles, J., Carme, S., Zhu, Y., Benoit-Cattin, H., 2006. A nonparametric MRI inhomogeneity correction method. Medical image analysis 10(2), 234–246. doi:10.1016/j.media.2005.09.004.

Chen, F., Taviani, V., Malkiel, I., Cheng, J., Tamir, J., Shaikh, J., Chang, S., Hardy, C., Pauly, J., Vasanawala, S., 2018. Variable-Density Single-Shot Fast Spin-Echo MRI with Deep Learning Reconstruction by Using Variational Networks. Radiology 289(2), 366–373. doi:10.1148/radiol.2018180445.

Chollet, F., 2015. Keras. https://github.com/fchollet/keras.

Dalca, A.V., Yu, E., Golland, P., Fischl, B., Sabuncu, M.R., Iglesias, J.E., 2019. Unsupervised Deep Learning for Bayesian Brain MRI Segmentation. MICCAI .

Deichmann, R., Good, C., Turner, R., 2002. RF inhomogeneity compensation in structural brain imaging. Magnetic Resonance in Medicine 47(2), 398–402. doi:10.1002/mrm.10050.

Friston, K.J., Ashburner, J.T., Kiebel, S.J., Nichols, T.E., Penny, W.D., 2011. Statistical parametric mapping: the analysis of functional brain images. Elsevier.

Ganzetti, M., Wenderoth, N., Mantini, D., 2016. Quantitative Evaluation of Intensity Inhomogeneity Correction Methods for Structural MR Brain Images. Neuroinformatics 14, 5–21. doi:10.1007/s12021-015-9277-2.

Guillemaud, R., Brady, M., 1997. Estimating the bias field of MR images. IEEE Trans Med Imaging 16(3), 238–251. doi:10.1109/42.585758.

Guo, Y., Gao, Y., Shen, D., 2016. Deformable MR Prostate Segmentation via Deep Feature Learning and Sparse Patch Matching. IEEE Trans Med Imaging 35(4), 1077–1089. doi:10.1109/TMI.2015.2508280.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M., Sodickson, D., Pock, T., Knoll, F., 2017. Learning a variational network for reconstruction of accelerated MRI data. Magnetic Resonance in Medicine 79. doi:10.1002/mrm.26977.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 770–778.

Islam, J., Zhang, Y., 2018. Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain Informatics 5(2). doi:10.1186/s40708-018-0080-3.

Kingma, D.P., Ba, J.L., 2015. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980.

Lee, C.Y., Chen, G.L., Zhang, Z.X., Chou, Y.H., Hsu, C.C., 2018. Is Intensity Inhomogeneity Correction Useful for Classification of Breast Cancer in Sonograms Using Deep Neural Network? Journal of Healthcare Engineering , 1–10doi:10.1155/2018/8413403.

Leemput, K.V., Maes, F., Vandermeulen, D., Suetens, P., 1997. Automated model-based bias field correction of MR images of the brain. IEEE Trans Med Imaging 18(10), 885–896. doi:10.1109/42.811268.

Likar, B., Viergever, M.A., Pernus, F., 2001. Retrospective correction of MR intensity inhomogeneity by information minimization. IEEE Trans Med Imaging 20(12), 1398–1410. doi:10.1109/42.974934.

Liney, G.P., Turnbull, L.W., Knowles, A.J., 1998. A simple method for the correction of endorectal surface coil inhomogeneity in prostate imaging. Magnetic Resonance Imaging 8(4), 994–997. doi:10.1002/jmri.1880080432.

Lundervold, A.S., Lundervold, A., 2019. An overview of deep learning in medical imaging focusing on MRI. Zeitschrift für Medizinische Physik 29(2), 102–127. doi:10.1016/j.zemedi.2018.11.002.

Luo, L., Xiong, Y., Liu, Y., Sun, X., 2019. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. ArXiv abs/1902.09843.

Manjón, J., Lull, J., Carbonell-Caballero, J., García-Martí, G., Martí-Bonmatí, L., Robles, M., 2007. A nonparametric MRI inhomogeneity correction method. Medical image analysis 11, 336–345. doi:10.1016/j.media.2007.03.001.

Manjón, J.V., 2006. Segmentación Robusta de Imágenes de RM cerebral. Ph.D. thesis. Polytechnic University of Valencia.

Mihara, H., Iriguchi, N., Ueno, S., 1998. A method of RF inhomogeneity correction in MR imaging. Magnetic Resonance Materials in Physics, Biology and Medicine 7, 115–120. doi:10.1007/BF02592235.

Murakami, J.W., Hayes, C.E., Weinberger, E., 1996. Intensity correction of phased-array surface coil images. Magnetic Resonance in Medicine 35(4), 585–590. doi:10.1002/mrm.1910350419.

Narayana, A., Brey, W., Kulkarni, M., Sievenpiper, C., 1988. Compensation for surface coil sensitivity variation in Magnetic Resonance Imaging. Magnetic Resonance Imaging 6, 271–274. doi:10.1016/0730-725X(88)90401-8.

Pham, D.L., Prince, J.L., 1998. An adaptive Fuzzy C-Means algorithm for image segmentation in the presence of intensity inhomogeneities. Proceedings of SPIE - The International Society for Optical Engineering 3338, 555–563. doi:10.1117/12.310864.

Rao, Yerravelli, 2014. Application of normalized cross correlation to image registration. International Journal of Research in Engineering and Technology 03, 12–16. doi:10.15623/ijret.2014.0317003.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS , 234–241doi:10.1007/978-3-319-24574-428.

Simko, A., Löfstedt, T., Nyholm, T., Garpebring, A., Jonsson, J., 2019. A Generalized Network for MRI Intensity Normalization. MIDL Conference .

Sled, J., Zijdenbos, A., Evans, A., 1998. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. IEEE Trans Med Imaging 17(1), 87–97. doi:10.1109/42.668698.

Song, S., Zheng, Y., He, Y., 2017. A review of Methods for Bias Correction in Medical Images. Biomedical Engineering Review 1. doi:10.18103/bme.v3i1.1550.

Thomas, C., 2020. Deep learning image enhancement insights on loss function engineering. https://towardsdatascience.com/deep-learning-image-enhancement-insights-on-loss-function-engineering-f57ccbb585d7.

Tustison, N., Avants, B., Cook, P., 2010. N4ITK: improved N3 bias correction. IEEE Trans Med Imaging 29(6), 1310–1320. doi:10.1109/TMI.2010.2046908.

Vovk, U., Pernus, F., Likar, B., 2007. A review of methods for correction of intensity inhomogeneity in MRI. IEEE Trans Med Imaging 26(3), 405–421. doi:10.1109/TMI.2006.891486.

Wells, W., Kikinis, R., Grimson, W., Jolesz, F., 1996. Adaptive segmentation of MRI data. Magnetic Resonance Imaging 15(4), 429–442. doi:10.1109/42.511747.

Yang, Y., Sun, J., Li, H., Xu, Z., 2016. Deep ADMM-Net for compressive sensing MRI. Advances in neural information processing systems 29, 10–18.

Medical Imaging and Applications

Master Thesis, August 2020

Brain MRI synthesis via pathology factorization and adversarial cycle-consistent learning for data augmentation

Khrystyna Faryna, Kevin Koschmeider, Bram van Ginneken

Diagnostic Image Analysis Group, Radboud University Medical Center, The Netherlands

Abstract

Identifying pathology in medical imaging data is a crucial step for patient diagnosis, treatment and prognosis. Deep learning, particularly convolutional neural networks, has led to breakthroughs in computer-aided diagnosis and detection. Nonetheless, these methods are heavily dependent on large number of training samples, which is not often available in medical imaging field. Moreover, while state-of-art supervised segmentation methods rely on precise voxel-wise annotations, manual lesion delineation in medical images is extremely laborious and time consuming task. Recent advancements in the field of generative adversarial networks (GAN) show promising results in generating realistic data samples for the purpose of augmenting datasets for downstream tasks, however the quality of samples generated by GANs also depends on the variability and size of the training set, particularly for large images. Unlike the majority of recent GAN methods, which focus on generation of either unlabeled samples or data restricted to particular classes, we propose a framework for controllable pathological image synthesis. Our approach is inspired by CycleGAN, where instead of generating images from random noise, we perform cycle-consistent image-to-image translation between two domains: healthy and pathological. Guided by a semantic map, an adversarially trained generator synthesizes pathology on a healthy image in the specified location. We demonstrate our approach in two distinct applications: a public dataset for brain tumors segmentation (BraTS2018) and an institutional dataset of cerebral microbleeds in traumatic brain injury patients. We subsequently utilize synthetic images generated with our method for data augmentation for the detection of cerebral microbleeds. Enriching the training dataset with synthetic images produced by our method exhibits the potential to increase sensitivity of cerebral microbleeds in traumatic brain injury detection system. The model trained only on real samples achieves an average sensitivity of 88% at 20 false positives per patient, after augmenting the training set with synthetic samples the model achieves an average sensitivity of 92% at the same rate of false positives per patient.

Keywords: Generative Adversarial Networks, Data Augmentation, Cerebral Microbleeds, MRI, Detection

1. Introduction

Medical image synthesis is defined as generation of quantitatively accurate and realistic-looking images (Frangi et al., 2018). Synthetic images have proven to be useful in number of medical image analysis problems (Bowles et al., 2017; Shin et al., 2018; Sun et al., 2018).

Generation of realistic medical images is a challenging task. Synthesis of good quality high-resolution images from random noise requires often prohibitively large numbers of training samples. The methods applying noise-to-image synthesis are forced to rely on small patch based approaches, which further limits usability of generated images. Recently, a number of methods synthesizing pathology on normal images with GANs have been proposed (Gupta et al., 2019; Wei et al., 2019). These methods rely on cycle-consistent GANs (Zhu et al., 2017) to perform pathological-to-healthy and healthy-to-pathological synthesis. Pathological-to-healthy (also known as pseudo-healthy synthesis) is defined as generation of subject-specific healthy images corresponding to the diseased ones (Figure 1). It can be useful both for clinical and research purposes. Particularly the resulting pseudo-healthy image can serve as a means for object detection (Sun et al., 2018) or segmen-

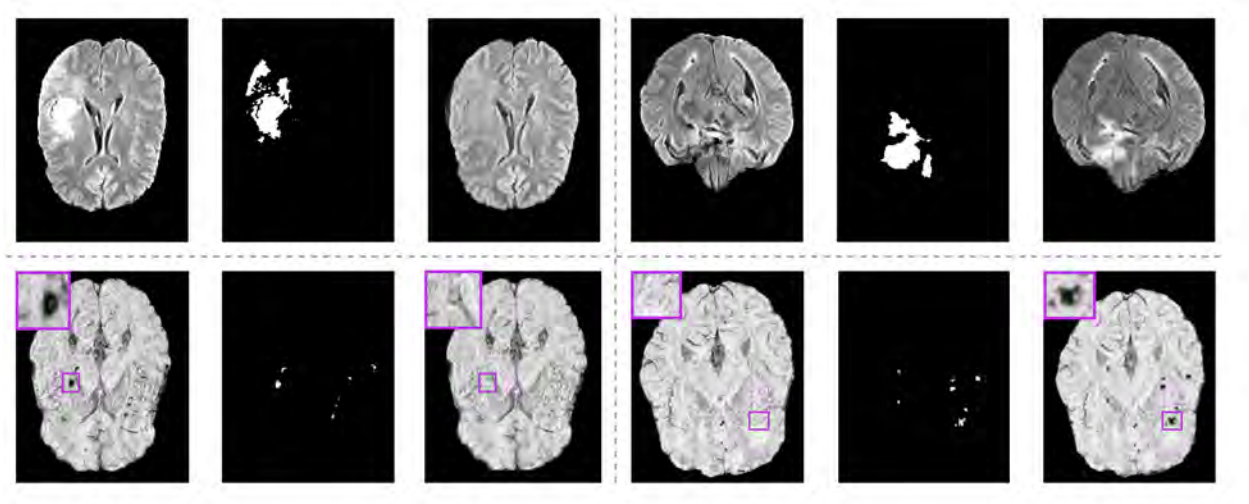


Figure 1: Top left quadrant: pseudo-healthy synthesis from brain tumour images (pathological image, pathology annotation, pseudo-healthy image), top right quadrant: pseudo-pathological synthesis of brain tumour images (healthy image, pathology annotation, pseudo-pathological image), bottom left quadrant: pseudo-healthy CMB image synthesis (pathological image, pathology annotation, pseudo-healthy image), bottom right quadrant: pseudo-pathological CMB image synthesis (healthy image, pathology annotation, pseudo-pathological image).

tation (Vorontsov et al., 2019).

Healthy-to-pathological synthesis (here referred to as pseudo-pathological synthesis) is a generation of images with pathology corresponding to the healthy ones, while preserving subject identity (Figure 1). Generation of these images could be used in such applications as data augmentation (Gupta et al., 2019; Shin et al., 2018; Wei et al., 2019) or modeling of subject specific changes in disease progression (Bowles et al., 2018; Xia et al., 2019).

A recent trend is using synthetic pathological images for data augmentation to improve generalization of classification, detection or segmentation networks (Shin et al., 2018). Particularly, the small number of positive pathological samples in comparison to normal ones is a common challenge in the field of medical image analysis. Image transformations (Hussain et al., 2017) and various sampling strategies (Dubey et al., 2014) have been adopted as a classical ways to tackle class-imbalance. However, the improvement in performance they introduce is limited, as the issue of a small training set not fully representing the underlying data distribution is not addressed. In contrast, the use of synthetic data can augment and increase diversity in the training set (Mariani et al., 2018). One of the possible ways to tackle the class imbalance is by using GANs for synthesis of positive samples from negative ones. Several recent studies (Gupta et al., 2019; Wei et al., 2019) proposed applying CycleGAN (Zhu et al., 2017) to perform healthy-to-pathological synthesis to create pathological images from healthy ones. Although CycleGAN demonstrates the ability to successfully learn mapping functions between two distinct probability distributions, it was shown (Chu et al., 2017) that in case of one-to-many mapping (detailed description in *Section 3.3.*) the synthesis problem is ill-posed.

Furthermore, the area of medical image synthesis where the generation process could be controlled by an input semantic mask is underexplored. Majority of papers currently focus on generation of either unlabeled data or images belonging to restricted classes, rather than simultaneously generating per pixel/voxel annotation –although the latter would bring more benefit to a number of applications.

In pseudo-healthy or pseudo-pathological synthesis, we not only aim to generate images but also to preserve subject identity, which leads to a number of new challenges. Supervised methods require paired data (ground truth), which is not possible to obtain in case of pseudo-healthy or pseudo-pathological synthesis, as a given subject can not be healthy and have a pathology at the same time. Meanwhile, longitudinal data could be scarce and contain ageing-induced changes.

In a recent work by Xia et al. (2020), a cycle-consistent GAN is used to perform pathological-to-healthy and healthy-to-pathological synthesis. To address the one-to-many problem, the pathology is explicitly factorized during training of the networks. The method has shown promising results in pseudo-healthy synthesis, however its ability to generate high quality pseudo-pathological images is not explicitly evaluated. Moreover, the method is 2D, which limits its applicability in the range of medical image analysis tasks, where pathology often requires the 3D context to be properly identified. Overall, there is a large number of papers demonstrating impressive results with 2D natural image synthesis (Brock et al., 2018), whereas the progress on the generation of 3D medical images is still limited.

1.1. Overview

In this paper, we focus on synthesis of 3D medical images through pathology factorization and adversarial cycle-consistent learning. Inspired by Xia et al. (2020), we perform pathological-to-healthy and healthy-to-pathological image synthesis, factorizing pathology into a semantic map. Our approach is based on CycleGAN, extended with pathology annotation as additional input and *abnormality mask* (Sun et al., 2018) loss which enforces preservation of identity in regions outside of the pathology map. We additionally modify the identity preservation pathway of a CycleGAN enforcing the lesions to be synthesized (in-painted) exclusively in the area specified by the annotation provided. Unlike the majority of recent methods in literature, our approach is 3D. We demonstrate our solution on two different datasets: Brain Tumor Segmentation (BraTS2018) dataset (Bakas et al., 2018; Menze et al., 2015) and an institutional dataset of cerebral microbleeds in traumatic brain injury (TBI) patients (Figure 1).

1.2. Cerebral microbleeds in traumatic brain injury

The clinical prognosis for patients with TBI currently is estimated using Glasgow Coma Scale and the assessment of large hemorrhages on Computed Tomography images. The prognostic precision of these two methods is limited (van den Heuvel et al., 2016a).

Recent clinical findings suggest (Werring, 2011) that clinical prognosis for patients with TBI is related to cerebral microbleeds (CMBs). CMBs are caused by leakages of small blood vessels, where hemosiderin deposits lead to focal dephasing of the MRI signal (Roob et al., 1999). Susceptibility Weighted Imaging (SWI), a sequence of MRI, is known to have a high sensitivity in detection of CMBs (Liu et al., 2014). In size, CMBs have a diameter of less than ten millimeters. As described by Greenberg et al. (2009), on the SWI sequence CMBs appear as "spherical hypointense lesions". In TBI cases, CMBs could also have an elongated shape.

Detection of CMBs can provide useful information for the clinical prognosis of patients with TBI. However, this task is not trivial: due to their small size and visual similarity with blood vessels on 2D projections, estimated time for manual annotation of CMBs is approximately 1 hour per scan (van den Heuvel et al., 2016a), which is prohibitively time consuming in clinical settings. While recently a number of methods for automatic detection of CMBs have been proposed (Dou et al., 2016; van den Heuvel et al., 2016a), these methods are limited by the small number of training samples.

In this paper, we investigate how an automatic CMB detection system can benefit from synthetic data generated with our approach. We train a model to synthesize CMBs on healthy scans. Afterwards, we enrich the real training dataset with synthetic images produced by our model.

1.3. Key contributions

The contributions of this work are the following:

1. We extend the method of Xia et al. (2020) to 3D. Leveraging 3D context expands applicability of the approach to a wider range of tasks, particularly in the medical domain.
2. We propose an alternative approach of pathological-to-healthy and healthy-to-pathological synthesis, capable of synthesizing high quality pseudo-pathological images. Our approach, guided by semantic binary map, allows controllable synthesis of pathological images from healthy ones.
3. We demonstrate our solution on two datasets: high-grade glioblastoma (HGG) images from the BraTS2018 challenge and an institutional dataset of CMB in TBI patients.
4. We utilize the generated synthetic CMB images for data augmentation in CMB detection task in TBI patients.

2. State of the art

In this section, we review literature related to our study. Firstly, we discuss non-deep learning and deep learning methods focusing on pathological-to-healthy synthesis (*Section 2.1*). Secondly, we describe recent methods which synthesize pathology on healthy data (*Section 2.2*). Thirdly, we review several related methods applying factorized representation in medical imaging (*Section 2.3*). We then describe recent methods using GAN synthesized images for data augmentation in down-stream tasks.

2.1. Pathological-to-healthy synthesis

2.1.1. Non-deep learning methods

Tsunoda et al. (2014) have proposed a pseudo-normal synthesis method to extend the application of temporal subtraction system for lung nodule enhancement in absence of previous patients scans. In their approach, the normal image is selected from a broad database of other patients, and deformed to fit the target patient. A number of early methods for pseudo-normal image synthesis relied on patch based, dictionary learning approach (Cao et al., 2012; Roy et al., 2011; Ye et al., 2013). First, the dictionary of source and target domain patches is created, then image synthesis is performed as a propagation of patches based on a specific similarity metric.

Bowles et al. (2016) have proposed a method that relies on voxel-wise kernel regression to synthesise a subject's pseudo-healthy FLAIR image from the respective T1-weighted image. The method is based on learning local relationships between intensities in T1-weighted and FLAIR image pairs of healthy subjects. One of the prerequisites assumed in the method is that the pathology is not prominent in T1-weighted images, which does not hold true in all of the cases.

2.1.2. Deep learning methods

Recent advancements in deep learning have allowed to perform more elaborate non-linear mappings and increase the scale of image synthesis methods. This progress is largely attributed to GANs and variational autoencoders (VAE).

The majority of autoencoder methods rely on learning density estimates of healthy data in the latent space. Uzunova et al. (2019) designed an unsupervised method for pathology detection training a conditional variational autoencoder on exclusively healthy data, conditioned on relative position of patches. Baur et al. (2018) used pseudo-healthy synthesis as a means for brain lesion detection applying adversarial training and deep representation learning. Their model was trained only a set of normal data and without any labels. Schlegl et al. (2017) proposed to create a generative model of healthy local anatomical appearance, and subsequently use it for pathology detection. The majority of the autoencoder methods are trained exclusively on data without pathology to ensure synthesized samples belong to the distribution of healthy samples and subsequently used for pathology detection. However, a recent study by (Nalisnick et al., 2018) suggests against using the learned manifolds from deep generative models to identify inputs similar to the training distribution.

GANs proposed by Goodfellow et al. (2014) and, have been widely adopted for medical image synthesis task. A number of methods focus on pseudo-healthy synthesis to aid pathology segmentation or detection tasks. Certain diseases can introduce global changes to the organ which can not be estimated by only pathology mask, for instance, the brain affected by neurodegenerative diseases, such as Alzheimer's disease. Baumgartner et al. (2017) propose visual feature attribution map to aid detection and visualization of disease effect. (Sun et al., 2018) proposed a method for semi-supervised image segmentation based on adversarial image synthesis. The abnormal-to-normal translation is performed to generate how a normal medical image would look like given its abnormal counterpart. The method is based on cycle-consistent GAN and relies on abnormality mask loss to learn which parts of the image should be inpainted or preserved. The resulting pseudo-healthy images are subsequently subtracted from their pathological originals and binarized to obtain a segmentation prediction. Vorontsov et al. (2019) proposed a semi-supervised method combining image-to-image translation between weak binary labels (indicating the presence of lesions), with fully supervised segmentation on a fraction of samples. Andermatt et al. extended CycleGAN to translate images between the domains of healthy and pathological images with residual generators explicitly modeling the pathology segmentation. In order to specify a single representation of the pathology, the translation from the healthy to the pathological domain is performed with a variational autoencoder.

The study of (Cohen et al., 2018) demonstrates how distribution losses (as used in CycleGAN) can hallucinate or hide a pathology in translation between domains tasks (FLAIR to T1), when one of the classes (pathological or healthy) are under- or over-represented.

2.2. Healthy-to-pathological synthesis

In the work of Xia et al. (2019), the learnt joint distribution of brain MRI scans and corresponding ages was used to simulate subject-specific aged images by a network conditioned on patients age, thus helping to distinguish healthy ageing from accelerated one. Bowles et al. (2018) used a GAN to model Alzheimer disease related features and show their correspondence with the changes observed over a longitudinal examination. Gupta et al. (2019) used a CycleGAN that synthesizes bone lesions on images without pathology to mitigate class imbalance in a bone X-ray dataset. Similarly, (Wei et al., 2019) used a CycleGAN to map from the source domain of normal colonic mucosa images to synthetic colorectal polyp images.

2.3. Factorized representation

In factorized representation, learning "different explanatory factors of the data which tend to change independently of each other" (Bengio et al., 2013) are disentangled. Chartsias et al. (2019) hypothesized that medical images are naturally composed of spatial factors and factors that denote the imaging characteristics, and demonstrated how anatomy could be disentangled from medical imaging modality. Joyce and Kozerke (2019) proposed a controllable data synthesis method that learns a disentangled representation of 3D medical data. Particularly, the model learns the spatial structure of the data, represented by an 'anatomical factor', and an anatomy deformation represented by a 'rendering factor'. Xia et al. (2020) proposed a method of pseudo-healthy synthesis where pathology is explicitly factorized from anatomy through the segmentation network.

2.4. Data augmentation

The limited number of training samples is a common challenge for deep learning methods in the medical domain, where pathology is uncommon by definition. Even when the data is available, obtaining manual annotations is labor-intensive, time-consuming and requires expertise. Recently, multiple studies have been done on utilizing GANs to create synthetic data for the purpose of augmenting the training dataset. Frid-Adar et al. (2018) applied classical data augmentation to a training set of liver CT images to train a GAN, subsequently combining the classical augmentation with images synthesized by GAN to train a liver lesion classifier. Yang

et al. (2018) proposed a class-aware adversarial synthesis method to generate lung nodules on CT scans. The framework is composed of a patch-inpainter conditioned on random latent variables and the target nodule label with class-aware discriminators. Sandfort et al. (2019) used a CycleGAN to synthesize contrast CT images from non-contrast ones and subsequently using the last for data augmentation in training a segmentation network. Han et al. (2019) used a 3D multi-conditional GAN to generate lung nodules and place those on lung CT images to augment the dataset for 3D object detection. Shin et al. (2018) proposed a framework to generate synthetic pathological images with brain tumors from healthy brain segmentations and tumour annotations.

3. Materials and methods

3.1. Data

In this paper, we demonstrate our method on two datasets: BraTS2018 and an institutional dataset of TBI.

3.1.1. BraTS2018

We use fluid-attenuated inversion recovery (FLAIR) data of the BraTS2018 challenge dataset as this sequence has a better representation of HGG then T1 and T2 and is used by radiologists for tumour annotation (Menze et al., 2015). The BraTS2018 HGG dataset consists of 210 images, it is released skull-stripped, resampled to isotropic resolution of 1 mm^3, with tumour annotations provided. We split the dataset into training, validation and testing subsets consisting of 110, 30, 70 scans correspondingly.

3.1.2. Cerebral microbleeds in Traumatic Brain Injury

The TBI dataset used in this study initially consisted of scans from 33 patients with varying TBI severity (moderate to severe) and 18 healthy subjects. For each subject, an SWI scan has been acquired using a 3T MRI scanner (Siemens Magnetom Trio). All scans had the resolution of 0.98 0.98 1 mm^3, a flip angle of 15, were acquired with a repetition time of 27 ms, an echo time of 20 ms and a bandwidth of 120 Hz/pixel. Manual delineation of CMBs in TBI scans is extremely time consuming (1 hour/scan–van den Heuvel et al. (2016b)). Initially, a single trained expert annotated all 33 pathological scans. A subset of 10 patient scans has been manually annotated by 6 trained experts following the Microbleed Anatomic Rating Scale (MARS) guidelines (Gregoire et al., 2009) to be later used for evaluation of a CAD system and measurements of inter-reader variability. The remaining scans have been used to train the CMB detector (van den Heuvel et al., 2016b). The trained detector has been inferred again on the larger set of data (including original training set) to obtain new annotations, which subsequently have been refined by a trained expert (neuroradiologist). These annotations are used further in our study. The dataset consisted of 85 scans belonging to 51 patients, with some patients having scans at two different timepoints. We subsequently split the dataset into training (46 pathological and 14 healthy scans), validation (11 pathological and 4 healthy) and testing (10 pathological). There were no overlapping patients between the subsets of the dataset.

3.1.3. Preprocessing

The images were clipped between 0 and 99.5 percentile of intensity and then rescaled to the range [0,1]. We sample the scans perpendicular to the axial plane as it carries the highest amount of information in brain MRI.

In the 2D setting, we select 60 middle slices from each brain MRI scan and crop them to the width of 160 and length of 208 pixels. Using a ground truth we label a slice as pathological if its annotation contains at least 1 pixel of lesion, otherwise it is labeled as healthy.

The 3D setting imposes memory and computational constraints. In order for the GAN to develop a good grasp of a brain's anatomy, we need to preserve the full axial plane. Resampling to lower resolution leads to a loss of image quality and details, which is undesirable taking into account that the CMB lesions are normally smaller that 10 mm in size. Thus, we first crop the volumes to the size (160,176,150) to minimize the empty background. Subsequently, we extract patches of size (160,176,32) with a 50% overlap in the direction perpendicular to the axial plane.

3.2. Problem statement and notation

Here, we denote a healthy sample x_{h_i} with empty pathology mask y_{h_i}, where x_{h_i} belongs to the healthy data distribution, $x_{h_i} \sim H$. A pathological sample is denoted as x_{p_i} with the corresponding pathology mask y_{p_i}, where x_{p_i} belongs to the pathological data distribution, $x_{p_i} \sim P$. Our objective is two-fold: given x_{h_i} and a random pathology mask y_{p_i}, generate a synthetic image $\tilde{x}_{p_i}$ such that $\tilde{x}_{p_i} \sim P$, and given x_{p_i} (and a suitable pathology mask y_{p_i}), generate a synthetic image $\tilde{x}_{h_i}$ such that $\tilde{x}_{h_i} \sim H$. The index i specifies a subject identity and is omitted further for simplicity purposes.

3.3. Pathology factorization necessity

Healthy-to-pathological synthesis is a one-to-many problem. While it is generally presumed that there exists a single $\tilde{x}_h$ corresponding to x_p, the reverse problem has infinitely many solutions. In other words, there could be many versions of pathological samples (lesions of different sizes, shapes and at different locations) corresponding to a healthy one. When a CycleGAN translates x_p to $\tilde{x}_h$, the pathology information should be lost, however the cycle-consistency loss forces the network

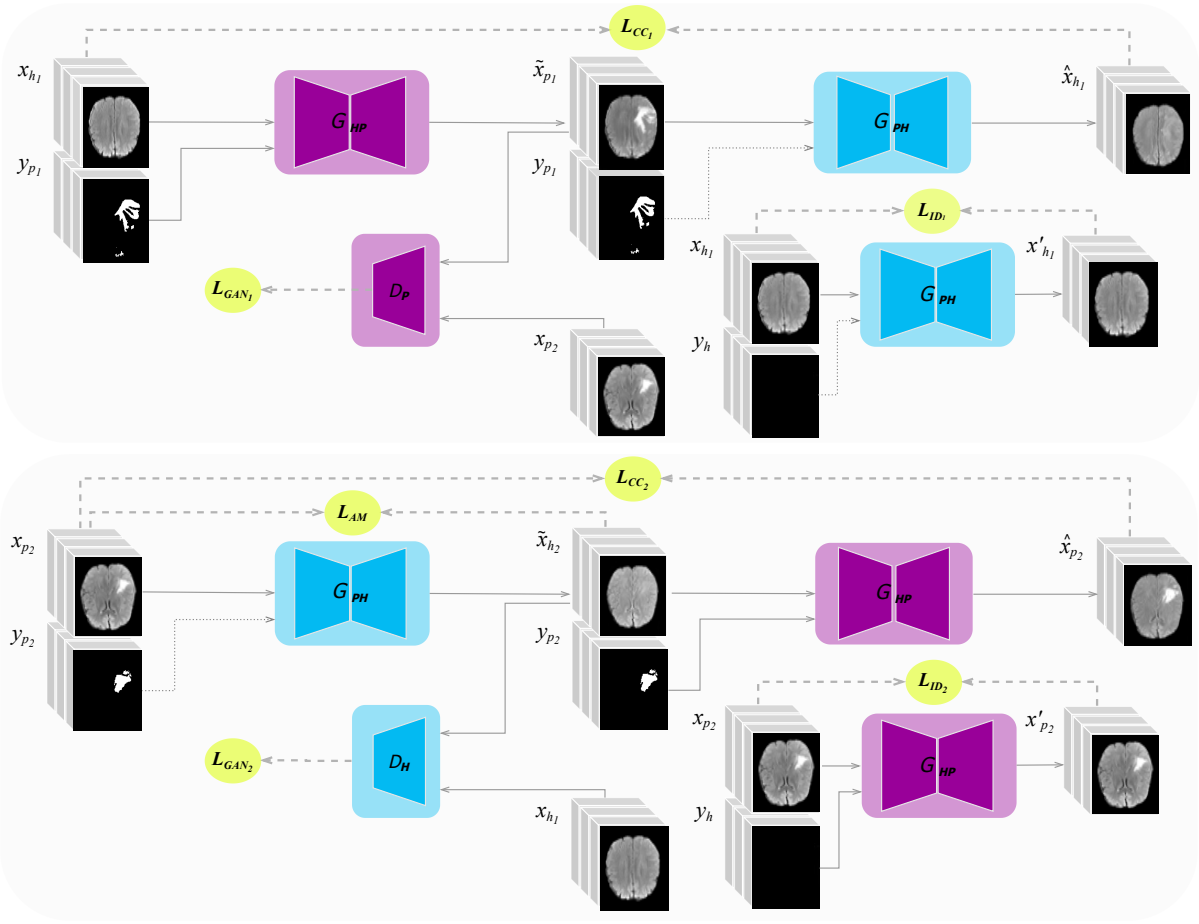


Figure 2: An overview of our method: in the HPH cycle (top) G_{HP} synthesizes a pseudo-pathological image $\tilde{x}_{p_1}$ from a real healthy image x_{h_1}. The critic D_P learns to distinguish between real (x_{p_2}) and fake ($\tilde{x}_{p_1}$) pathological samples and encourages G_{HP} to generate more realistic samples, in the PHP cycle (bottom) G_{PH} synthesizes a pseudo-healthy image $\tilde{x}_{h_2}$ from a real pathological image x_{p_2}. The critic D_H learns to distinguish between real (x_{h_1}) and fake ($\tilde{x}_{h_2}$) healthy samples and encourages G_{PH} to generate more realistic samples.

to still be able to reconstruct the initial input. Thus, CycleGAN has to either encode the information within the generated $\tilde{x}_h$ samples (Chu et al., 2017; Xia et al., 2020) or within the generators capacity.

3.4. Proposed approach and model training

The schematic of the proposed approach along with training losses is shown in Figure 2. The inter-domain translation is performed in a CycleGAN fashion: the model consists of two generators and two discriminators. The configuration comprises of two cycles: healthy-to-pathological (HPH) synthesis and pathological-to-healthy (PHP) synthesis.

In the HPH cycle, the healthy-to-pathological generator G_{HP} receives as input the healthy image x_{h_1} and a random pathology mask y_{p_1}, concatenated along the channel dimension. The task of G_{HP} generator is to synthesize a pseudo-pathological image $\tilde{x}_{p_1}$, such that the pathology would be located in the areas provided by the mask y_{p_1}, while the rest of the image should remain unchanged. The goal of the discriminator D_P is to distinguish between real (x_{p_2}) and fake ($\tilde{x}_{p_1}$) pathological samples. The obtained pseudo-pathological image $\tilde{x}_{p_1}$ is then concatenated with its pathology mask y_{p_1} and passed to the pathological-to-healthy generator G_{PH}. The task of G_{PH} is to reconstruct the input image $\hat{x}_{p_2}$. Additionally, to enforce preservation of identity and reassure that G_{PH} does not modify a healthy image the x_{h_1} concatenated with y_h and passed through G_{PH}.

In the PHP cycle, the concatenated x_{p_2} and y_{p_2} are passed through the pathological-to-healthy generator G_{PH}. The goal of G_{PH} generator is to synthesize a pseudo-healthy image $\tilde{x}_{h_2}$, i.e. 'hide' the pathology specified by y_{p_2}. The goal of the discriminator D_H is to distinguish between real (x_{h_1}) and fake ($\tilde{x}_{h_2}$) healthy samples. The resulting pseudo-healthy image $\tilde{x}_{h_2}$ is then concatenated with the pathology mask y_{p_2} and passed to the healthy-to-pathological generator G_{HP}, which reconstructs the original image $\hat{x}_{p_2}$. To ensure that no lesion is created outside the mask provided a pathological image x_{p_2} concatenated with y_h and passed through G_{HP}. The abnormality mask loss is used to stabilize the

training and enforce in-painting only the area of an image specified by mask.

3.5. Networks architecture

For a 2D setting, we used models proposed by Xia et al. (2020) for both configurations. The details of the networks architecture for 3D setting are described below.

3.5.1. Generators

The generators G_{HP} and G_{PH} have an architecture consisting of an encoder and decoder (see Figure 3), inspired by Xia et al. (2020). We use two long skip connections from the encoder to the decoder branch, in order to better preserve details of the image. Two downsampling steps are performed in each generator. We use strided convolution to perform downsampling (stride=2) instead of max-pooling as it is trainable (Springenberg et al., 2014). The encoder is followed by 6 residual blocks, to alleviate gradient vanishing. Upsampling steps are performed with interpolation followed by convolution. We avoided using transposed convolution layer as those are known for causing checkerboard artifacts (Odena et al., 2016). Each convolutional layer, besides the last one, is followed by a leaky ReLU activation function (negative slope=0.2). The last convolutional layer of the generator is followed by a sigmoid activation function, as our input is scaled between 0 and 1. The detailed description of convolutional (CB) and residual (RB) blocks is shown in Figure 3.

3.5.2. Discriminators

The architecture of the discriminators D_P and D_H is shown in Figure 3. It is composed of five convolutional layers with isotropic filters (kernel size = 4). Downsampling is performed using strided convolution (stride=2). Each convolutional layer is followed by a leaky ReLU activation function (negative slope=0.2). The last convolutional layer does not have an activation function.

3.6. Losses

The model is trained with four different types of losses: *identity loss*, *cycle consistency loss*, *adversarial losses*, and *abnormality mask loss*. The detailed description of each is provided below.

3.6.1. Identity loss

The main goal of identity loss is to discourage the generators G_{HP} and G_{PH} from modifying input images which already belong to a target domain.

In the PHP cycle, where the pathology map is used, the generator G_{PH} is trained to in-paint only the areas specified by the abnormality map.

$$L_{ID_1} = E_{x_{h_1}\sim H, y_h\sim M_h}\left[\left\|G_{PH}(x_{h_1}, y_h) - x_{h_1}\right\|\right] \quad (1)$$

In the PHP cycle without a pathology map, the G_{PH} generator is tasked to not only to in-paint the pathology but also to localize it.

$$L_{ID_1} = E_{x_{h_1}\sim H}\left[\left\|G_{PH}(x_{h_1}) - x_{h_1}\right\|\right] \quad (2)$$

In the HPH cycle, we enforce the pathology to be synthesized in regions provided by a binary semantic map. If the provided map is empty the image should not be modified.

$$L_{ID_2} = E_{x_{p_2}\sim P, y_h\sim M_h}\left[\left\|G_{HP}(x_{p_2}, y_h) - x_{p_2}\right\|\right] \quad (3)$$

3.6.2. Cycle consistency loss

Cycle consistency loss is a key factor in preservation of subject identity. In the PHP cycle, the main task of cycle consistency loss is to encourage the generator G_{HP} to reconstruct the pathological image x_{p_2} such that the resulting $\hat{x}_{p_2} = G_{HP}(\tilde{x}_{h_2}, y_{p_2})$ is as close as possible to the original input pathological image $\hat{x}_{p_2} \approx x_{p_2}$

$$L_{CC_2} = E_{x_{p_2}\sim P}\left[\left\|\hat{x}_{p_2} - x_{p_2}\right\|\right] \quad (4)$$

In the HPH cycle, the purpose of the cycle consistency loss is to enforce reconstruction of the healthy image x_{h_1}, by generator G_{PH}, such that the resulting $\hat{x}_{h_1} = G_{PH}(\tilde{x}_{p_1}, y_{p_1})$ is as close as possible to the original input healthy image $\hat{x}_{h_1} \approx x_{h_1}$.

In the HPH cycle without an input pathology mask, the cycle consistency loss encourages G_{PH} to reconstruct the healthy image x_{h_1} such that the resulting $\hat{x}_{h_1} = G_{PH}(\tilde{x}_{p_1})$ is as close as possible to original input healthy image $\hat{x}_{h_1} \approx x_{h_1}$.

$$L_{CC_1} = E_{x_{h_1}\sim H}\left[\left\|\hat{x}_{h_1} - x_{h_1}\right\|\right] \quad (5)$$

3.6.3. Adversarial loss

To control the generation of synthetic healthy and pathological images we use *Wasserstein* loss with gradient penalty (Gulrajani et al., 2017).

$$\begin{aligned} L_{GAN_1} = E_{x_{p_2}\sim P}[&D_p(x_{p_2}) - D_p(\tilde{x}_{p_1}) \\ &+ \lambda_{GP}(\left\|\nabla_{\dot{x}_{p_2}}(\dot{x}_{p_2})\right\|_2 - 1)^2] \end{aligned} \quad (6)$$

where x_{p_2} is a batch of pathological images, $\tilde{x}_{p_1}$ is a batch of pseudo-pathological images, D_P is the discriminator used to distinguish real and synthetic pathological images. ($\dot{x}_{p_2}$ is defined as ($\dot{x}_{p_2} = (1-\alpha)G_{HP}(x_{p_2}) + \alpha x_{p_2}, \alpha \sim U[0,1]$.

$$\begin{aligned} L_{GAN_2} = E_{x_{h_1}\sim H}[&D_H(x_{h_1}) - D_H(\tilde{x}_{h_2}) \\ &+ \lambda_{GP}(\left\|\nabla_{\dot{x}_{h_1}}(\dot{x}_{h_1})\right\|_2 - 1)^2] \end{aligned} \quad (7)$$

Similarly, here x_{h_1} is a batch of healthy images, $\tilde{x}_{h_2}$ is a batch of pseudo-healthy images, D_H is the discriminator used to distinguish between real and synthetic healthy images. ($\dot{x}_{h_1}$ is defined as ($\dot{x}_{h_1} = (1 -$

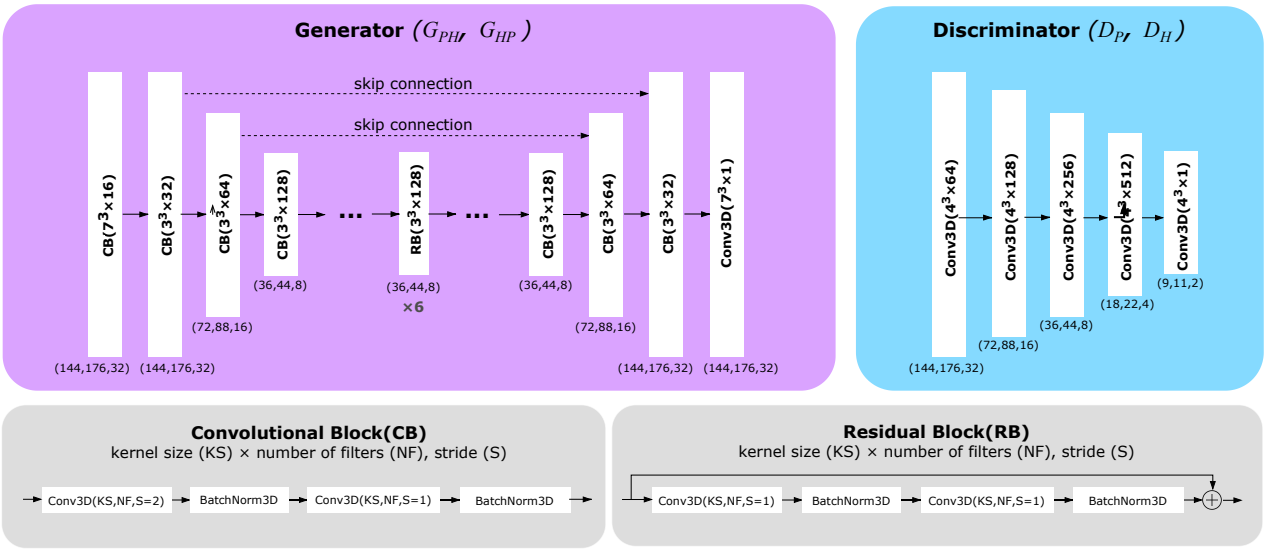


Figure 3: The architecture of the generator networks G_{PH} and G_{HP}, and the discriminator networks D_P and D_H (top). A structure of residual and convolutional blocks (bottom).

$\alpha)G_{PH}(x_{h_1}) + \alpha x_{h_1}, \alpha \sim U[0,1]$. The first two terms, in both aforementioned equations measure the Wasserstein distance between real and synthetic pathological and healthy images correspondingly. The last, in both equations is a gradient penalty loss. As in (Gulrajani et al., 2017; Xia et al., 2020) the parameter $\lambda_{GP} = 10$.

3.6.4. Abnormality mask loss

The abnormality mask loss enforces preservation of subject identity and stabilizes the training (Sun et al., 2018).

$$L_{AM} = E_{x_{p_2}\sim H, y_p \sim M_p}\left[\left\|(x_{p_2} - \tilde{x}_{h_2}) * (1 - y_{p_2})\right\|\right] \quad (8)$$

3.6.5. Total loss

The overall loss is a weighted sum of the aforementioned individual losses:

$$\begin{aligned} L_{total} =& \lambda_{GAN_1} L_{GAN_1} + \lambda_{GAN_2} L_{GAN_2} \\ &+ \lambda_{CC_1} L_{CC_1} + \lambda_{CC_2} L_{CC_2} \\ &+ \lambda_{ID_1} L_{ID_1} + \lambda_{ID_2} L_{ID_2} + \lambda_{AM} L_{AM}, \end{aligned} \quad (9)$$

where $\lambda_{GAN_1} = \lambda_{GAN_2} = 1$, $\lambda_{CC_1} = \lambda_{CC_2} = 10$, $\lambda_{ID_1} = \lambda_{ID_2} = 5$, and $\lambda_{AM} = 10$.

4. Experimental setup

4.1. Baseline

As a baseline, we adopt the semi-supervised method of pseudo-healthy image synthesis proposed by Xia et al. (2020), which consists of two cycles: pathological-to-healthy (PH) and healthy-to-healthy (HH). In the PH cycle, the pathology is disentangled through the segmentation network, while the generator network is adversarially trained to synthesize a pseudo-healthy image from a pathological one. The pseudo-healthy image is then concatenated with the pathology segmentation, and the reconstruction network is trained to recover the original pathological image in a cycle-consistent fashion. The HH cycle is designed to stabilize training and prevent the reconstructor network from inventing a pathology if the input mask is empty.

We subsequently extend the method of Xia et al. (2020) to 3D. Following their training scheme, we convert the network layers to 3D, introducing several modifications to meet the computational constraints without major sacrifices in image quality. The detailed description of 3D networks used is provided in *Section 3.5.*

4.2. Training details

2D and 3D models where trained for 300 and 150 epochs respectively, for the translation task on the BraTS2018 dataset. In the case of CMB synthesis, the models were trained for 400 epochs. The critics and generators have been updated in alternating fashion. Following (Baumgartner et al., 2017; Gulrajani et al., 2017; Xia et al., 2020), we update the critics 50 times more than generators in the first 25 epochs; after 25 epochs, we perform 5 critic updates per 1 update of the generator. The networks were trained using Adam optimizer (Kingma and Ba, 2015) with a learning rate 0.0001, β_1 of 0.5, and β_2 of 0.99. In the 2D setting, we used a batch size of 10, while in 3D setting due to computational constraints we use a batch size of 4.

4.3. Evaluation metrics

4.3.1. Pathological-to-healthy synthesis

Xia et al. (2020) proposed metrics to evaluate the quality of synthesized pseudo-healthy images: *healthiness* (h) and *identity* (iD).

The synthetic image *healthiness* is estimated through evaluation of segmented pathology of pseudo-healthy images. It employs a segmentation network S_{pre}, pretrained on pathology annotations and is defined as follows:

$$h = 1 - \frac{E_{x_h \sim H}\left[\left\|N(S_{pre}(\tilde{x}_h))\right\|\right]}{E_{x_p \sim P}\left[\left\|N(S_{pre}(x_p))\right\|\right]}, \quad (10)$$

where x_p is a pathological image and x_h is a synthetic pseudo-healthy image corresponding to the pathological one. $N(...)$ refers to the number of pathology voxels detected by S_{pre} in the input image.

The aim of *identity* metric is to evaluate how well the subject identity is preserved through comparing areas outside of pathology mask with the Multi-Scale Structural Similarity Index (MS-SSIM). Formally, *identity* is defined as:

$$iD = MS - SSIM\left(x_p(1-y_p), \tilde{x}_h * (1-y_p)\right), \quad (11)$$

where x_p, is a pathological image, y_p is a pathology ground truth and $\tilde{x}_h$ is a synthetic pseudo healthy image corresponding to the pathological one.

Unlike originally proposed, we compute all the metrics in 3D on the entire scan. The idea behind providing comparison in 3D is that the preservation of structures in axial, sagittal and coronal planes could be better described with a 3D comparison. To obtain 3D volumes from 2D methods, we infer the generative models on each slice in a volume and reconstruct the full volume by stacking the slices and zero padding it to the original shape.

To reconstruct 3D patches to full volumes, we perform linear blending of patches along the direction perpendicular to the axial plane with 50% intersection between neighbouring slabs.

The segmentation network used in calculation of healthiness scores is a modification of the 3D U-Net proposed by Buda et al. (2019) with batch normalization following every convolutional layer. We decreased the number of channels by the factor of two to reduce overfitting. The final model used for evaluation was trained for 400 epochs, and achieved a mean Dice coefficient of 0.80 on our testing set for a binary tumor segmentation task.

4.4. Data augmentation in CMB detection

A summary of our data augmentation pipeline is shown on Figure 4.

We firstly train a proposed 3D GAN to perform translation between two domains: healthy and TBI. A trained healthy-to-pathological generator is subsequently inferred on healthy images from the dataset, guided by semantic binary CMB annotations. We recycle the annotations from the training dataset, performing the following image transformations: horizontal flip, rotation by a small angle (1 to 5 degrees) and dilation by one voxel. In this study, we use a detection framework proposed by Koschmeider **?**, based on 3D-UNet (Özgün Çiçek et al., 2016). We compare the performance of a detector trained in three different settings: only real data, only synthetic data and combined real and synthetic data. Identical set of hyperparameters is used across the experiments, and the models were trained for same number of epochs.

5. Results and discussion

In this section, we present the final results of our study as well as discuss its outcomes and limitations. We provide *healthiness* and *identity* metrics only for

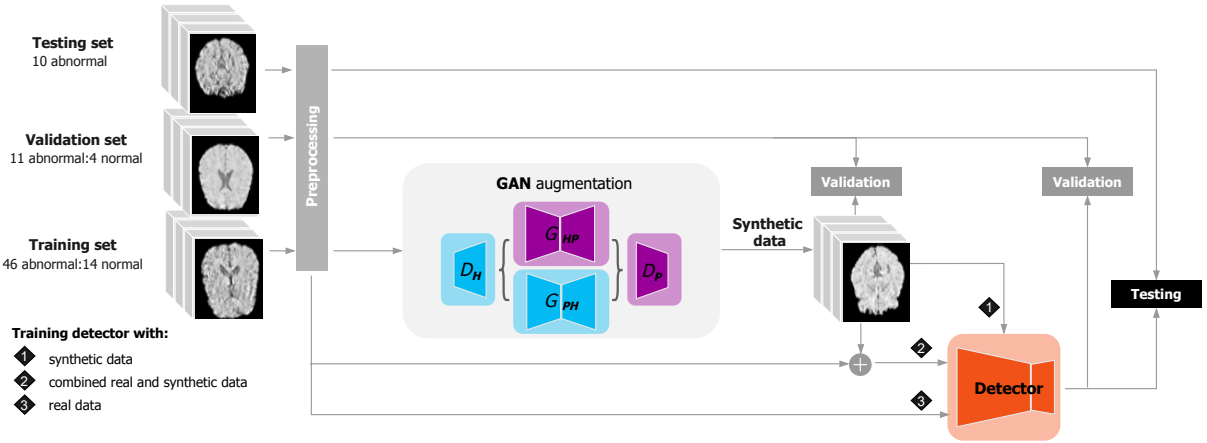


Figure 4: Data augmentation pipeline.

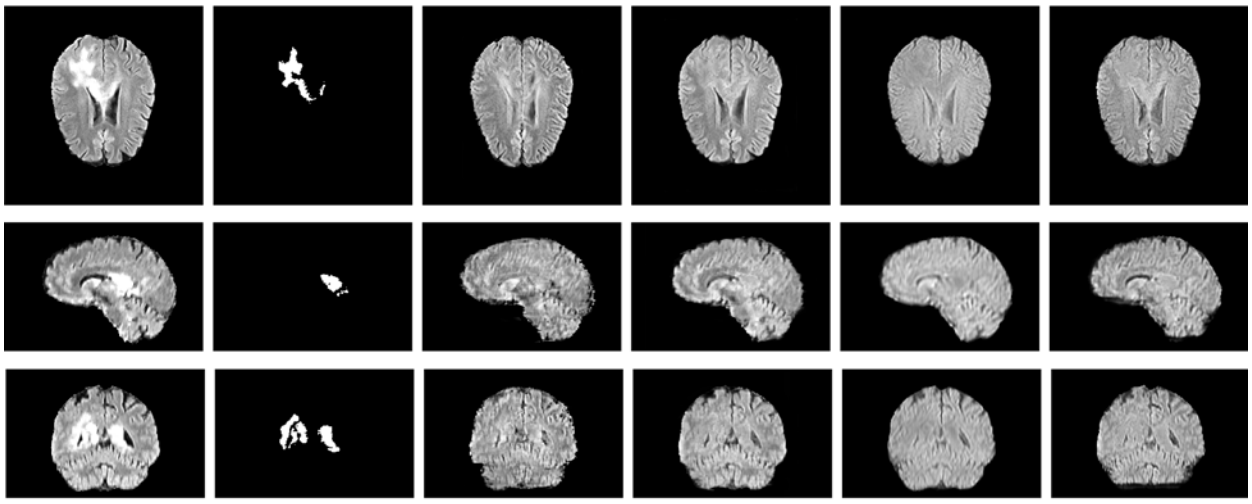


Figure 5: Axial, sagittal and coronal view of pseudo-healthy images produced by different methods. From left to right: original pathological image, pathology annotation, pseudo-healthy image produced by Xia et al. (2020) method (2D), our method (2D), 3D adaptation of Xia et al. (2020), our method (3D).

Table 1: Healthiness and identity metric scores of evaluated methods with 95% CI

Method	**Healthiness (h)**	**95%CI**	**Identity (iD)**	**95%CI**
Xia et al. (2020)	0.85	(0.82–0.88)	0.91	(0.90–0.91)
3D adaptation of Xia et al. (2020)	0.73	(0.67–0.78)	0.95	(0.94–0.95)
Proposed 2D	0.79	(0.75–0.84)	0.96	(0.96–0.97)
Proposed 3D	0.75	(0.70–0.80)	0.94	(0.93–0.94)

pseudo-healthy images synthesized from brain tumors (BraTS2018). Due to the small size of CMB lesions, any scores obtained with the aforementioned metrics are not informative.

Bootstrapping was used to estimate statistical significance of achieved *healthiness* and *identity* metric scores. We boot-strapped the samples 1000 times to generate a distribution of each metric. The obtained 2.5 and 97.5 percentiles were used as 95% confidence intervals (CI). We provide a qualitative comparison of synthetic pseudo-pathological brain tumor images, as well as pseudo-healthy CMB images. Finally, we present the results of synthetic data augmentation in the CMB detection task.

5.1. BraTS2018

5.1.1. Pseudo-healthy synthesis for brain tumours

The results of pseudo-healthy synthesis on BraTS2018 dataset evaluated with *healthiness* and *identity* metric are shown on Table 1. Note that the main focus of our proposed method is pseudo-pathological synthesis, as it uses the pathology mask to synthesize pseudo-healthy images. The *healthiness* and *identity* scores for pseudo-healthy images generated with our method are reported solely for a purpose of synthetic image quality comparison.

The example of pseudo-healthy images generated with different methods is shown in Figure 5, more figures are provided in *Appendix*. The *identity* metric, being based on MS-SSIM index, reflects the overall quality of synthetic images and is mostly influenced by two factors: structural information and contrast properties of the image. Overall, both 3D methods achieve a better preservation of structure, this is particularly prominent in sagittal and coronal views (Figure 5), and partially reflected in *identity* metric scores Table 1. The method proposed by Xia et al. (2020), despite demonstrating qualitatively good results in axial view, suffers from discontinuities between slices and loss of structure in sagittal and coronal view. The images produced by both 3D methods exhibit loss of contrast. This could be attributed to 3D generators having considerably higher number of parameters while being trained on the same amount of data. Our proposed 2D model obtained the highest *identity* score, the key reason behind it is AM loss, which enforces preservation of structures outside of the abnormality mask region. This type of loss is particularly useful in pseudo-healthy synthesis tasks where the pathology is localized and does not affect the sur-

rounding tissue.

The method of Xia et al. (2020) obtained the highest *healthiness* score. Overall, both 3D methods achieve lower *healthiness* scores. We assume that it might be due to the fact that images produced by 2D methods have discontinuities between slices which might hinder the pretrained segmentation network from identifying abnormality, more detailed discussion of this point is provided in *Section 5.5.*.

Main limitations of the methods in terms of pseudo-healthy synthesis in BraTS2018 images are discussed further. Firstly, all of the above methods assume a localized nature of pathology: slices of the brain whose corresponding annotation mask is empty are assumed to not be influenced by the pathology. While that might hold true for a number of pathologies, HGG is causing deformations in brain structures which extends to slices labeled 'healthy' (Xia et al., 2020).

Secondly, in pseudo-healthy images produced with all of the methods the cerebrospinal fluid ventricles appear smaller than in the original pathological images. The reason behind it is a biased distribution of the data. Brain tumours are relatively large lesions, and in the BraTS2018 dataset the majority of slices where the cerebrospinal fluid ventricles are present also have lesion tissue present (corresponding annotation is not empty). Thus, the 'healthy' domain data is mostly comprised of parts of the brain which do not have ventricles, allowing the generators to learn this biased distribution. Both of the above limitations could be alleviated with the use of additional healthy scans. During the study, we experimented with the use of OASIS-3 (LaMontagne et al., 2019) dataset (FLAIR sequence) as an additional healthy domain, however the Jensen–Shannon divergence between the histograms of healthy slices of BraTS2018 and OASIS-3 was higher than the divergence between histograms of pathological and healthy Brats2018 slices, forcing the generators to learn the difference between imaging protocols rather than pseudo-healthy or pseudo-pathological synthesis.

5.1.2. Pseudo-pathological synthesis for brain tumours

The examples of synthesized pseudo-pathological images from BraTS2018 by method proposed by Xia et al. (2020), ours 2D method and ours 2D method without pathology mask input during pseudo-healthy synthesis are shown on Figure 6. All of the methods 'insert' pathology in the specified place. Both of our methods produce better preservation of original image details than the baseline. We assume that this is due to the fact that in Xia et al. (2020) approach the only loss controlling the quality of generated pseudo-pathological images is cycle-consistency, which is not sufficient for producing high quality images. On the other hand, in our implementation the quality of synthetic pathological images is adversarially enforced with the discriminator D_P.

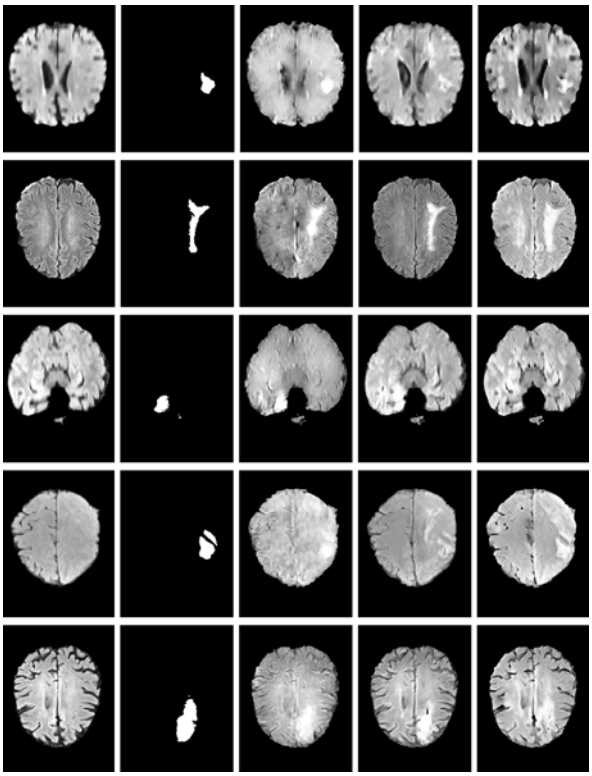


Figure 6: Pseudo-pathological synthetic brain tumour images produced by three 2D methods. The first column shows healthy axial slices from the Brats2018 dataset, the second column shows pathology annotation, the third, fourth and fifth columns show pseudo-pathological images synthesized by Xia et al. (2020), our 2D method with and our 2D method without segmentation input in pseudo-healthy synthesis task correspondingly.

5.2. Cerebral microbleeds in traumatic brain injury

In 2D projection, CMBs can appear indistinguishable from blood vessels. Thus, in this study only 3D methods are considered for synthesis of pseudo-pathological CMB images.

In the method proposed by Xia et al. (2020), the pathology is disentangled via the segmentation network. The synthesis of pseudo-pathological images (reconstruction) relies on the output of a segmentation network to guide where the lesion should be 'inserted'. HGG tumors are relatively large pathologies, usually having a single instance per scan, they are easier to identify by segmentation network than CMBs. On the other hand, CMB lesions have multiple instances, are on average few millimeters in size and have similarities with other structures in the brain (blood vessels). During CMB segmentation, the network produces false positives, over- or underestimates the lesion areas, destabilizing the training of the reconstruction network. Only after the segmentation network crosses the point of overfitting, the reconstructor starts receiving a meaningful input.

The reconstruction network eventually converges to majorly overestimating the sizes and merging together

clusters of lesions. Meanwhile, the feedback that the generator receives from adversarial losses is not sufficient to encourage pseudo-healthy synthesis due to extreme class imbalance. Thus, the pseudo-healthy synthesis converges to identity. Even though the method is capable of producing pseudo-pathological samples, further study needs to be done on how to make it applicable to tasks with such extreme class imbalance. .

Figure 7 illustrates the synthesized pseudo-pathological TBI images obtained with the 3D adaptation of the method proposed by Xia et al. (2020).

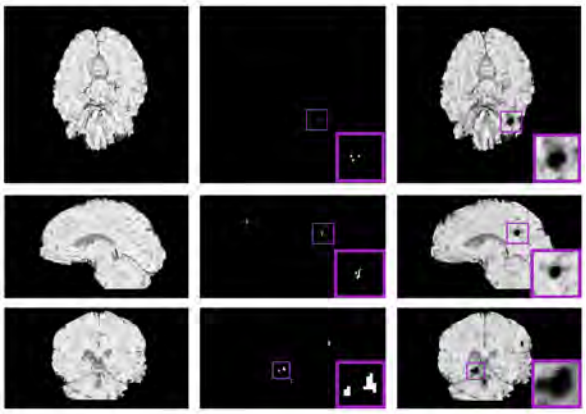


Figure 7: Pseudo-pathological synthetic CMB images produced by the 3D adaptation of (Xia et al., 2020). Pseudo-pathological synthetic sample in axial, sagittal and coronal view. From left to right: healthy brain SWI scan, pathology annotation, synthesized pseudo-pathological SWI scan.

5.2.1. *Pseudo-healthy synthesis for cerebral microbleeds*

The examples of synthesized with our method pseudo-healthy samples from TBI images are shown on Figure 8. Additional examples are provided in *Appendix*. As can be seen from the Figure 8, while the microbleeds are in-painted while the similar structures (blood vessels) are preserved.

5.2.2. *Pseudo-pathological synthesis for cerebral microbleeds*

The example of synthesized pseudo-pathological images with our method are shown on Figure 9, additional examples are provided in *Appendix*. The results of experiments with synthetic data augmentation are shown on Figure 10.

As mentioned in *Section 4.4.*, the synthetic data augmentation experiment consists of 3 main pipelines. Firstly, we train the detection model with only real data to determine the baseline performance. Secondly, the detection model is trained with only synthetic data to identify whether the detector is capable of learning from solely synthetic images. Finally, the detection model is trained on combined synthetic and real images. The proportion of real to synthetic images in this experiment is

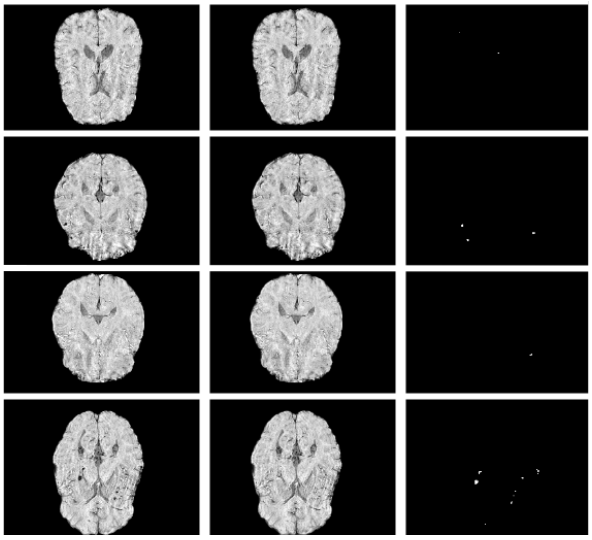


Figure 8: Pseudo-healthy synthetic CMB images produced by our 3D method. The first row shows pathological axial slices from the CMB dataset. The last row shows CMB annotations, the middle row shows synthesized pseudo healthy brain SWI slices.

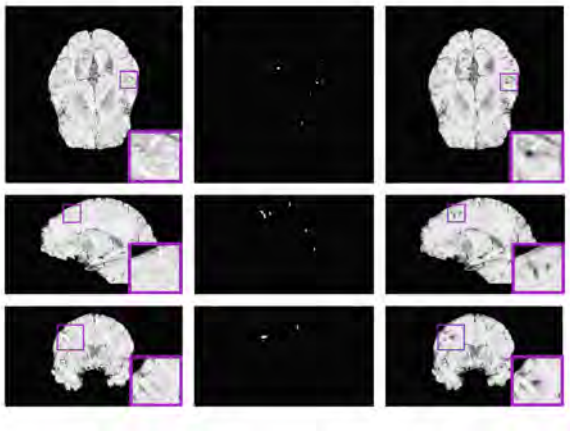


Figure 9: Pseudo-pathological synthetic sample in axial, sagital and coronal view. From left to right: healthy brain SWI scan, pathology annotation, synthesized pseudo-pathological SWI scan.

1:1. The detection model performance results trained in all three setting are evaluated using Free-Response Operating Characteristic (FROC). Our test set consists of 10 SWI scans manually annotated by 6 experts. The performance of each expert is shown against the majority voting of the rest of experts.

The model trained on synthetic data only is capable of producing meaningful predictions while tested on real samples. It demonstrates comparatively worse performance, at 20 false positives per patient it reaches a sensitivity of less then 75%. We attribute it to the fact that synthetic data still has certain differences in image characteristics from real. The images appear less sharp, what, we assume, is caused by restoration of full volume from individual patches that introduces blurring. A further work would be to investigate less aggressive ways

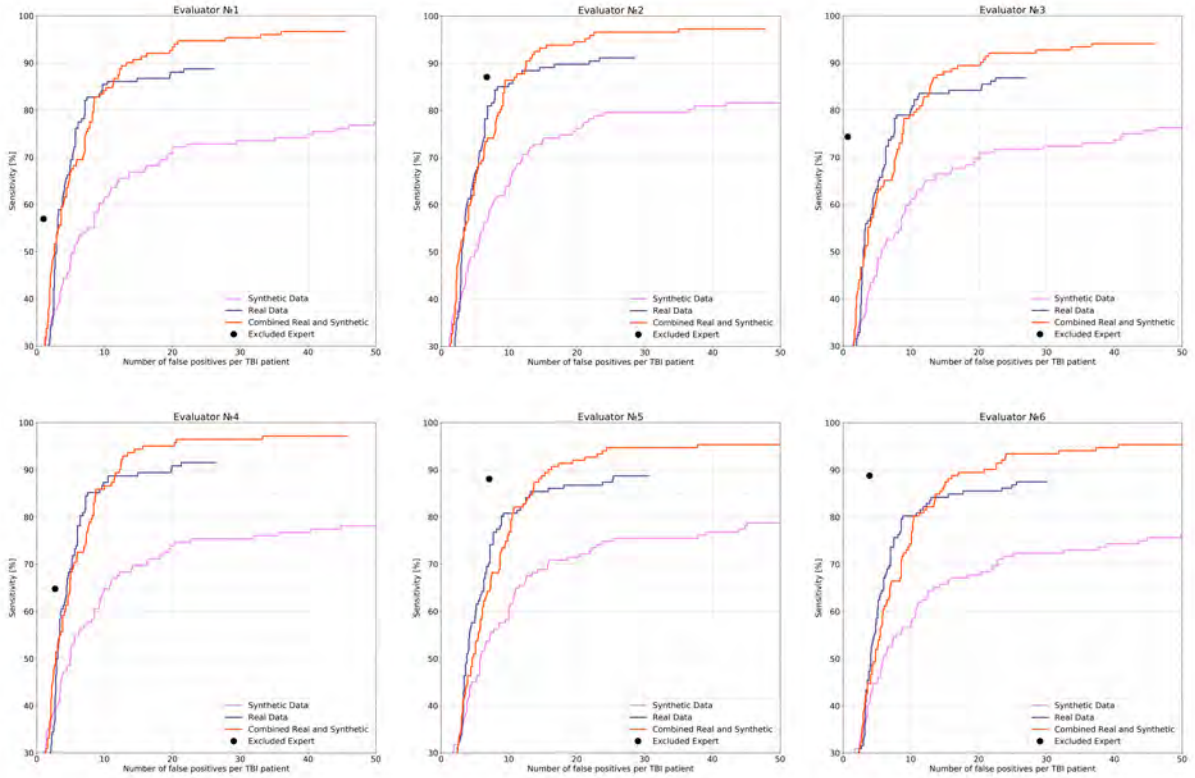


Figure 10: Detection performance of the model trained on real, synthetic and combined data.

of patch reconstruction. Additionally, we observed that lesion boundaries in synthetic images are less sharp then in real ones. In future work, we aim to introduce additional regularization to enhance the weight of the signal from lesion areas. The model trained with real data only achieves a sensitivity of < 90% at 20 false positives per patient across 6 experts. While combining the real samples with synthetic ones with 1:1 ratio, the model reaches an average sensitivity of 92% at 20 false positives per patient. Overall, enriching the training dataset with synthetic images produced by our method exhibits the potential to increase sensitivity of a cerebral microbleeds in traumatic brain injury detection system.

5.3. The impact of adversarial loss

In the initial stage of the experiments we used L2 adversarial loss function (Mao et al., 2016) along with Patch-GAN (Isola et al., 2016) discriminator. We observed fluctuations in image quality, unstable training and convergence, as well as mode collapse. We then switch to using Wasserstein loss with gradient penalty (Gulrajani et al., 2017) function. We noticed overall improvement in GAN performace, particularly stability of training, improved image quality which was consistent with loss values.

5.4. Effectiveness of 'healthiness' metric

Metrics that can benchmark generative models and directly quantitatively validate medical images synthesized by GANs are in high demand. The *healthiness* metric proposed by Xia et al. (2020) relies on pretrained segmentation network to judge how healthy a synthesized pseudo-healthy image is. The question that arises is: *Does the segmentation network know what it does not know?* In other words, if the pathology could not be identified by the pretrained segmentation network, does it mean that the synthetic pseudo-healthy image resembles healthy anatomy? We illustrate that it does not hold true in all of the cases with a toy example (Figure 11).

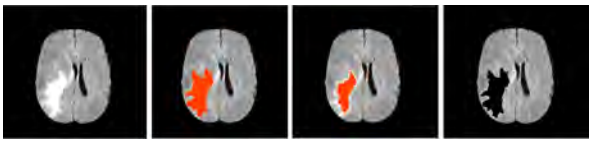


Figure 11: From left to right: pathological brain image, pathological brain image + tumour ground truth, pathological brain image + tumour segmentation, pathological brain image with tumour area turned to zero + segmentation. The sample achieved *healthiness* score of 0.95.

We set the lesion area provided by ground truth of the BraTS2018 scan to zero and compute the healthiness metric. While the healthiness metric is misled with this

example the human evaluator would easily identify it. While proposed metric gives rough estimate of whether a features of specific pathology (HGG) could be identified on the image, the overall 'healthiness' of the image is not reflected. Moreover, the fact that the metric is based on auxiliary segmentation network and is sensitive to its performance challenges its applicability in benchmarking.

6. Conclusion

In this paper, we present our semi-supervised method of controllable 3D medical image synthesis through pathology factorization and adversarial cycle-consistent learning. We perform pathological-to-healthy and healthy-to-pathological image synthesis, guided by a pathology annotation. Our approach is shown in 2D and 3D settings with two distinct applications: Brain Tumor Segmentation (BraTS2018) dataset and institutional dataset of cerebral microbleeds in traumatic brain injury patients. We provided quantitative and qualitative comparison of generated images. We utilize pseudo-pathological images synthesized with our method for data augmentation of CMB detection task. Supplementing the training dataset with synthetic images generated with our method indicates a potential of increasing sensitivity of cerebral microbleeds in traumatic brain injury detection system. The model trained with real data only, on average, achieves a sensitivity of 88% at 20 false positives per patient. Whereas after combining the real samples with synthetic ones, the model reaches an average sensitivity of 92% at 20 false positives per patient across 6 experts.

7. Acknowledgement

This thesis is the conclusion of my participation in Erasmus Mundus Joint Master Degree in Medical Imaging and Applications (MAIA) program. I would like to express my deepest appreciations to European Commission for the financial support during this two years, MAIA coordination committee for giving me opportunity to join this program and my supervisors Kevin Koshcmeider and Bram van Ginneken, for valuable contributions into this thesis work.

References

Andermatt, S., Horváth, A., Pezold, S., Cattin, P., . Pathology segmentation using distributional differences to images of healthy origin.

Bakas, S., Reyes, M., et. al, A.J., 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge.

Baumgartner, C.F., Koch, L.M., Tezcan, K.C., Ang, J.X., Konukoglu, E., 2017. Visual feature attribution using wasserstein gans. CoRR abs/1711.08998. URL: http://arxiv.org/abs/1711.08998.

Baur, C., Wiestler, B., Albarqouni, S., Navab, N., 2018. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. CoRR abs/1804.04488. URL: http://arxiv.org/abs/1804.04488.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1798–1828.

Bowles, C., Gunn, R., Hammers, A., Rueckert, D., 2018. Modelling the progression of Alzheimer's disease in MRI using generative adversarial networks, in: Angelini, E.D., Landman, B.A. (Eds.), Medical Imaging 2018: Image Processing, International Society for Optics and Photonics. SPIE. pp. 397 – 407. URL: https://doi.org/10.1117/12.2293256, doi:10.1117/12.2293256.

Bowles, C., Qin, C., Guerrero, R., Gunn, R., Hammers, A., Dickie, D., Hernández, M., Wardlaw, J., Rueckert, D., 2017. Brain lesion segmentation through image synthesis and outlier detection. NeuroImage: Clinical 16, 643–658. doi:10.1016/j.nicl.2017.09.003.

Bowles, C., Qin, C., Ledig, C., Guerrero, R., Gunn, R., Hammers, A., Sakka, E., Dickie, D.A., Hernández, M.V., Royle, N., Wardlaw, J., Rhodius-Meester, H., Tijms, B., Lemstra, A.W., van der Flier, W., Barkhof, F., Scheltens, P., Rueckert, D., 2016. Pseudo-healthy image synthesis for white matter lesion segmentation, in: Tsaftaris, S.A., Gooya, A., Frangi, A.F., Prince, J.L. (Eds.), Simulation and Synthesis in Medical Imaging, Springer International Publishing, Cham. pp. 87–96.

Brock, A., Donahue, J., Simonyan, K., 2018. Large scale gan training for high fidelity natural image synthesis.

Buda, M., Saha, A., Mazurowski, M.A., 2019. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Computers in Biology and Medicine 109, 218 – 225. doi:https://doi.org/10.1016/j.compbiomed.2019.05.002.

Cao, T., Zach, C., Modla, S., Powell, D., Czymmek, K., Niethammer, M., 2012. Registration for correlative microscopy using image analogies, in: Dawant, B.M., Christensen, G.E., Fitzpatrick, J.M., Rueckert, D. (Eds.), Biomedical Image Registration, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 296–306.

Chartsias, A., Joyce, T., Papanastasiou, G., Williams, M., Newby, D.E., Dharmakumar, R., Tsaftaris, S.A., 2019. Factorised representation learning in cardiac image analysis. CoRR abs/1903.09467. URL: http://arxiv.org/abs/1903.09467.

Chu, C., Zhmoginov, A., Sandler, M., 2017. Cyclegan, a master of steganography. URL: https://arxiv.org/abs/1712.02950.

Cohen, J.P., Luck, M., Honari, S., 2018. Distribution matching losses can hallucinate features in medical image translation, in: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2018, Springer International Publishing, Cham. pp. 529–536.

Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V.C., Shi, L., Heng, P., 2016. Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks. IEEE Transactions on Medical Imaging 35, 1182–1195.

Dubey, R., Zhou, J., Wang, Y., Thompson, P.M., Ye, J., 2014. Analysis of sampling techniques for imbalanced data: An n=648 adni study. NeuroImage 87, 220 – 241. doi:https://doi.org/10.1016/j.neuroimage.2013.10.005.

Frangi, A.F., Tsaftaris, S.A., Prince, J.L., 2018. Simulation and synthesis in medical imaging. IEEE Transactions on Medical Imaging 37, 673–679.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H., 2018. Synthetic data augmentation using gan for improved liver lesion classification.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, pp. 2672–2680. Cited By 9768.

Greenberg, S.M., Vernooij, M.W., Cordonnier, C., Viswanathan, A., Al-Shahi Salman, R., Warach, S., Launer, L.J., Van Buchem, M.A., Breteler, M.M., 2009. Cerebral microbleeds: a guide to detection and interpretation. The Lancet Neurology 8, 165 – 174.

doi:https://doi.org/10.1016/S1474-4422(09)70013-4.

Gregoire, S.M., Chaudhary, U.J., Brown, M.M., Yousry, T.A., Kallis, C., Jäger, H.R., Werring, D.J., 2009. The microbleed anatomical rating scale (mars). Neurology 73, 1759–1766. URL: https://n.neurology.org/content/73/21/1759, doi:10.1212/WNL.0b013e3181c34a7d.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. Improved training of wasserstein gans, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA. p. 5769–5779.

Gupta, A., Venkatesh, S., Chopra, S., Ledig, C., 2019. Generative image translation for data augmentation of bone lesion pathology, in: Cardoso, M.J., Feragen, A., Glocker, B., Konukoglu, E., Oguz, I., Unal, G., Vercauteren, T. (Eds.), Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, PMLR, London, United Kingdom. pp. 225–235. URL: http://proceedings.mlr.press/v102/gupta19b.html.

Han, C., Kitamura, Y., Kudo, A., Ichinose, A., Rundo, L., Furukawa, Y., Umemoto, K., Li, Y., Nakayama, H., 2019. Synthesizing diverse lung nodules wherever massively: 3d multi-conditional gan-based ct image augmentation for object detection.

Hussain, Z., Gimenez, F., Yi, D., Rubin, D.L., 2017. Differential data augmentation techniques for medical imaging classification tasks. AMIA ... Annual Symposium proceedings. AMIA Symposium 2017, 979–984.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2016. Image-to-image translation with conditional adversarial networks.

Joyce, T., Kozerke, S., 2019. 3d medical image synthesis by factorised representation and deformable model learning, in: Burgos, N., Gooya, A., Svoboda, D. (Eds.), Simulation and Synthesis in Medical Imaging, Springer International Publishing, Cham. pp. 110–119.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. CoRR abs/1412.6980.

LaMontagne, P.J., Benzinger, T.L., Morris, J.C., Keefe, S., Hornbeck, R., Xiong, C., Grant, E., Hassenstab, J., Moulder, K., Vlassenko, A., Raichle, M.E., Cruchaga, C., Marcus, D., 2019. Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease. medRxiv doi:10.1101/2019.12.13.19014902.

Liu, J., Kou, Z., Tian, Y., 2014. Diffuse axonal injury after traumatic cerebral microbleeds: An evaluation of imaging techniques. Neural Regeneration Research 9, 1222–1230. doi:10.4103/1673-5374.135330. cited By 37.

Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., 2016. Multi-class generative adversarial networks with the L2 loss function. CoRR abs/1611.04076. URL: http://arxiv.org/abs/1611.04076.

Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C., 2018. Bagan: Data augmentation with balancing gan.

Menze, B.H., Jakab, A., et. al, S.B., 2015. The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging 34, 1993–2024.

Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B., 2018. Do deep generative models know what they don't know?

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill URL: http://distill.pub/2016/deconv-checkerboard, doi:10.23915/distill.00003.

Roob, G., Schmidt, R., Kapeller, P., Lechner, A., Hartung, H.P., Fazekas, F., 1999. Mri evidence of past cerebral microbleeds in a healthy elderly population. Neurology 52, 991–991. URL: https://n.neurology.org/content/52/5/991, doi:10.1212/WNL.52.5.991.

Roy, S., Carass, A., Prince, J., 2011. A compressed sensing approach for mr tissue contrast synthesis, in: Székely, G., Hahn, H.K. (Eds.), Information Processing in Medical Imaging, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 371–383.

Sandfort, V., Yan, K., Pickhardt, P.J., Summers, R.M., 2019. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. Scientific Reports 9, 16884. doi:10.1038/s41598-019-52737-x.

Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G., 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.T., Shen, D. (Eds.), Information Processing in Medical Imaging, Springer International Publishing, Cham. pp. 146–157.

Shin, H.C., Tenenholtz, N.A., Rogers, J.K., Schwarz, C.G., Senjem, M.L., Gunter, J.L., Andriole, K., Michalski, M., 2018. Medical image synthesis for data augmentation and anonymization using generative adversarial networks.

Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M., 2014. Striving for simplicity: The all convolutional net.

Sun, L., Wang, J., Huang, Y., Ding, X., Greenspan, H., Paisley, J., 2018. An adversarial learning approach to medical image synthesis for lesion detection.

Tsunoda, Y., Moribe, M., Orii, H., Kawano, H., Maeda, H., 2014. Pseudo-normal image synthesis from chest radiograph database for lung nodule detection, in: Advanced Intelligent Systems.

Uzunova, H., Schultz, S., Handels, H., Ehrhardt, J., 2019. Unsupervised pathology detection in medical images using conditional variational autoencoders. International Journal of Computer Assisted Radiology and Surgery 14, 451–461. URL: https://doi.org/10.1007/s11548-018-1898-0, doi:10.1007/s11548-018-1898-0.

van den Heuvel, T., van der Eerden, A., Manniesing, R., Ghafoorian, M., Tan, T., Andriessen, T., vande Vyvere, T., van den Hauwe, L., ter Haar Romeny, B., Goraj, B., Platel, B., 2016a. Automated detection of cerebral microbleeds in patients with traumatic brain injury. NeuroImage: Clinical 12, 241–251. doi:10.1016/j.nicl.2016.07.002.

van den Heuvel, T., van der Eerden, A., Manniesing, R., Ghafoorian, M., Tan, T., Andriessen, T., vande Vyvere, T., van den Hauwe, L., ter Haar Romeny, B., Goraj, B., Platel, B., 2016b. Automated detection of cerebral microbleeds in patients with traumatic brain injury. NeuroImage: Clinical 12, 241–251. doi:10.1016/j.nicl.2016.07.002.

Vorontsov, E., Molchanov, P., Byeon, W., Mello, S.D., Jampani, V., Liu, M.Y., Kadoury, S., Kautz, J., 2019. Boosting segmentation with weak supervision from image-to-image translation. ArXiv abs/1904.01636.

Wei, J., Suriawinata, A., Vaickus, L., Ren, B., Liu, X., Wei, J., Hassanpour, S., 2019. Generative image translation for data augmentation in colorectal histopathology images.

Werring, D.J., 2011. Cerebral microbleeds: Pathophysiology to clinical practice.

Xia, T., Chartsias, A., Tsaftaris, S.A., 2019. Consistent brain ageing synthesis, in: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.T., Khan, A. (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, Springer International Publishing, Cham. pp. 750–758.

Xia, T., Chartsias, A., Tsaftaris, S.A., 2020. Pseudo-healthy synthesis with pathology disentanglement and adversarial learning. Medical Image Analysis 64, 101719. doi:https://doi.org/10.1016/j.media.2020.101719.

Yang, J., Liu, S., Grbic, S., Setio, A.A.A., Xu, Z., Gibson, E., Chabin, G., Georgescu, B., Laine, A.F., Comaniciu, D., 2018. Class-aware adversarial lung nodule synthesis in ct images.

Ye, D.H., Zikic, D., Glocker, B., Criminisi, A., Konukoglu, E., 2013. Modality propagation: Coherent synthesis of subject-specific scans with data-driven regularization, in: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 606–613.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks.

Özgün Çiçek, Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation.

8. Appendix

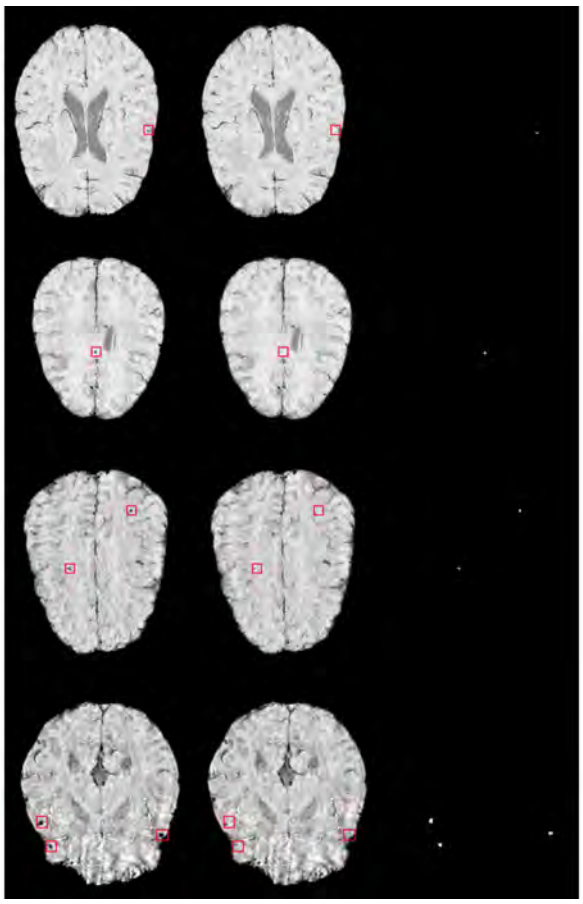


Figure 13: Pseudo-healthy synthetic CMB images produced by our 3D method. The first row shows pathological axial slices from the CMB dataset. The last row shows CMB annotations, the middle row shows synthesized pseudo healthy brain SWI slices.

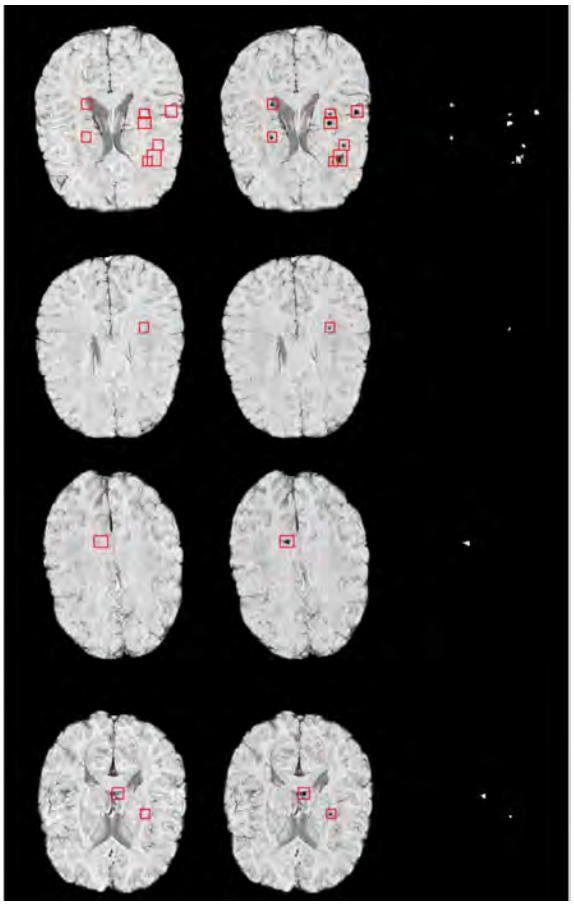


Figure 14: Pseudo-pathological synthetic CMB images produced by our 3D method. The first row shows healthy axial slices from the dataset. The last row shows CMB annotations, the middle row shows synthesized pseudo-pathological brain SWI slices.

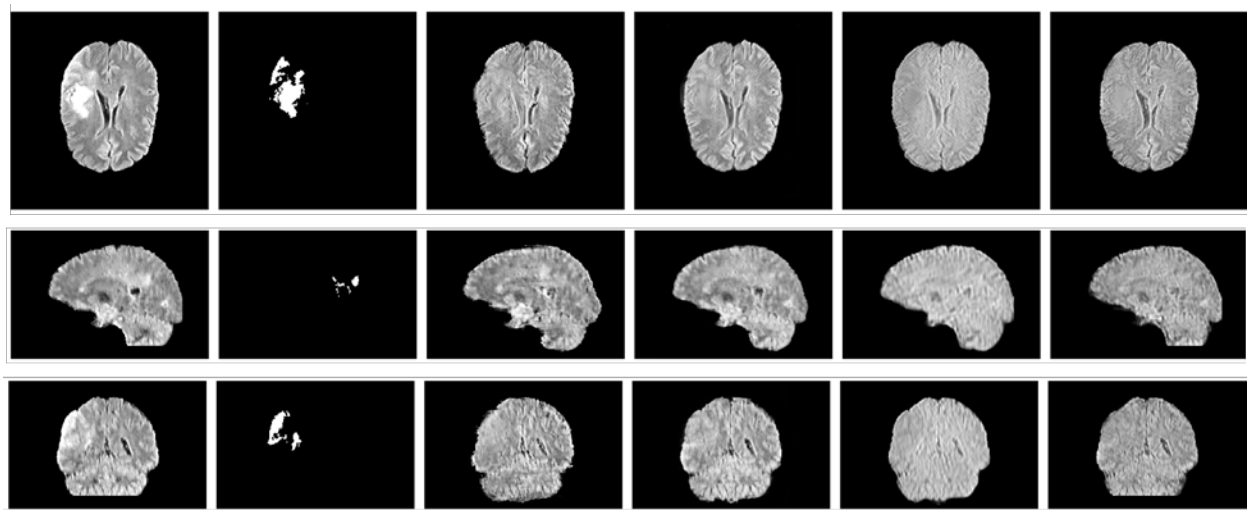


Figure 12: Axial, sagittal and coronal view of pseudo-healthy images produced by different methods. From left to right: original pathological image, pathology annotation, pseudo-healthy image produced by Xia et al. (2020) method (2D), our method (2D), 3D adaptation of Xia et al. (2020), our method (3D).

Medical Imaging and Applications

Master Thesis, August 2020

Performant GPU based image processing pipelines

Jhon Mauro Gomez Benitez, Koen Brants

Agfa Gevaert NV, Septestraat 27, 2640 Mortsel, Belgium

Abstract

Image Processing Pipelines are defined as structures which relates a set of image processing, computer vision, and machine learning algorithms along with multiple purpose operations which, on its whole, performs more complex tasks as segmentation, real time tracking, and also execute technical operations as rendering. This technique is used in the new modality-calibration technology for X-ray machines which is based in Augmented Reality, demanding high computational resources from the workstations to reach almost real-time processing. The current execution model lacks of parallelism and under-use the available computational resources, while the existing parallelism technologies do not fulfil the existing problem constraints. The development of the project is orientated towards the investigation of modern programming techniques for efficiently computing Image Processing Pipelines efficiently exploiting the available computational resources both on CPU and CUDA enabled GPUs. The design of a multi-threaded job scheduler is provided along with the improvement of existing Image Processing algorithms based in computation on the GPU. Experimental results with the existing Angulated Overlay pipeline shows an execution speed-up of 2.03x.

Keywords: MAIA master, performance, cuda, pipelines, scheduling, multithreading

1. Introduction

The novelties in Medical Imaging and in techniques for Computer Aided Diagnosis have increased the demand of computational resources and, in consequence, the efforts of the hardware manufacturers for delivering reliable systems tending to fulfil this requirement. This objective is not exclusive on the hardware industry. During the last years the research community have turn their attention on looking for an efficient usage of the available computational resources at software level (Banalagay et al. (2014)).

Although, the ultimate goal when it comes to improving the performance of an application is to reduce the computation times, the approach to follow varies according to the available computational resources and the nature of the application.

Per AGFA, the core of the different X-Ray imaging modalities is the Workstation, which is specially arranged to deploy the software that supports the entire image acquisition process and also provide tools for image enhancement, segmentation, modality calibration, image storage, and other core operations. The acquisition workflow is supported by image and video processing algorithms arranged in different pipelines in charge of performing high level operations.

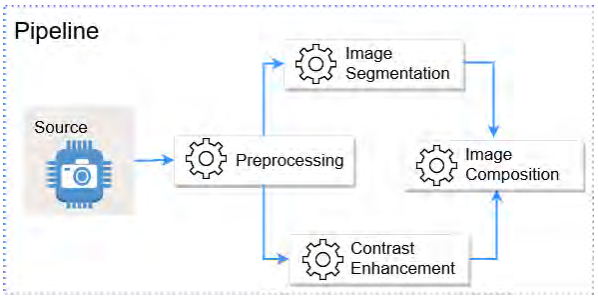


Figure 1: Schematic example of an image processing pipeline

The figure 1 exemplifies a pipeline where an input image is presented to the *Preprocessing* algorithm, which results are forwarded to both the *Image Segmentation* and the *Contrast Enhancement* algorithms. The results of both executions are combined through the *Image Composition*. These execution schemes are flexible and work always, but lack of parallelism as they currently

stand.

Despite the AGFA's pipeline structure was not conceived to be executed in parallel, from a Computer Science perspective it fits the National Institute of Standards and Technology (NIST) definition of Directed Acyclic Graph (DAG) in Black (2004), typically used in scheduling problems for coordinating parallel executions. This kind of Graph structures have been widely studied (Kwok and Ahmad (1999)) nevertheless, the task of finding inner paths for ease the scheduling process is recognised as a NP-Hard problem (Basagni et al. (1997)), which opens the doors to researchers looking for new DAG exploration techniques.

Moreover, an implementation-related issue comes out when working in parallel computing. This is addressed in Vetter et al. (2018) as the *Extreme Heterogeneity* of systems problem, where the parallelism models variate not only across the existing hardware architectures but also providers. This can be evidenced on how the conventional programming model for C++ applications is not entirely applicable in GPU based implementations specially with NVIDIA CUDA.

This, along with the lack of software solutions which provide reliable interfaces that decouple the implementation from the hardware architecture itself, increase the difficulty on delivering parallel systems, rising the need of defining a problem-scoped less-invasive solution which allows the exploration of the available hardware parallelism capabilities, without the obligation of re-architect the existing software.

We have listed, so far, the challenges to face to extend the parallelism capabilities in the existing Virtual DR solution. Application specially designed to assist the calibration process of the DR-600 AGFA X-ray modality and to assess in real-time the adequate x-ray dosage based on the patient thickness. This is an Augmented Reality application which measures the environment characteristics though a 3D camera, demanding enough computational resources to reach real-time processing capabilities.

In this thesis, a job scheduler adapted to the existing AGFA solution is presented, along with the analysis of several scheduling models over DAGs, all of this towards an efficient coordination of algorithms execution in a multi-thread environment. Also we present a set of GPU implemented alternatives to a selection of bad performing image processing algorithms in the system, looking for boost their execution times. Section 3 describes the methodologies and implementations made to explore the DAGs, also explains the final scheduler design and the improvements made to support GPU parallelism on different algorithms. The results in section 4 are presented for a common set of micro-benchmark tests used for approach the final solution, and the conclusive outcomes are presented based on the execution times using the proposed scheduler over one of the pipelines of the existing software, also measuring individual execution times for the GPU improved algorithms. The results are discussed and the conclusions are presented in sections 5 and 6 respectively.

2. State of the art

Lamport (2015) situates the beginning of the studies in *concurrency* from the middle 60's in the known Mutual Exclusion Problem exposed by Dijkstra (1965). The research in this field has experimented a big interest among the Computer Scientist community in line with the new challenges, rising every day mainly by the increasing demand of computational resources in both research and productive environments. The scale of the problem posed by this master thesis and the available hardware resources, settle our research effort in the *Parallel Computing* spectra, which diverges from the *Distributed Computing* in how the input data stands in memory. While in our problem the data persists across a shared memory where all the available CPUs have access, in the distributed model all the computing entities are independent and the shared data only coexists within each single entity (Raynal (2015)).

2.1. Parallel Computing Models

Culler et al. (1998) defines three different parallelism models based in the the evolution of the microprocessor's architecture, following the Moore's law and the Flynn's Taxonomy (Flynn (1972)):

Bit-Level Parallelism. Defined as the most primitive parallelism model, look for reducing the number of performed operations when a variable exceeds the microprocessor word size. This model evolves only with the microprocessor architecture and its efficiency is limited to the compiler design.

Instruction-Level Parallelism. In this model the program structure is evaluated at compilation and execution level to determine execution branches that can run independently. This model is mostly transparent to the developer and boost the application performance when the implementation is efficient and the compiler exploits such characteristic.

Task-Level Parallelism. This model takes advantage on the multi threading and multi core capabilities of the modern microprocessors, allowing the distribution and concurrent execution of weakly coupled tasks. According to Cook (2013), this model is extensible to the CUDA programming paradigm, where the tasks execution follow a fined-grained parallelism approach, performing small tasks over little chunks of data distributed in a large number of threads, in an architecture which supports fast communications through shared memory.

Our current project stands in the Task-Level parallelism spectra since the pipelines execution model is

based in loosely-coupled tasks and the main goal is to speed up its processing time by an effective task coordination in a multi-threading environment. Moreover the parallelism is exploited at fine-grained level through GPU-based computations in specific image processing algorithms.

2.2. Solutions supporting Task-level parallelism

A set of frameworks have been developed to support the development of parallel-based systems at task-level:

Open MP. Specified in Dagum and Menon (1998), provides with an open source API for multi-process execution in shared memory systems. The programming model is based in clauses for defining data sharing attributes, synchronisation policies looking for race conditions control, and scheduling policies within `for` and `do` loops. Core operations as thread creation, works distribution and flow control, rely in compilation directives and static annotations where the execution flow is explicitly defined, posing a high difficulty dealing with dynamically created pipelines where its structure is not known at compilation time.

Thread Building Blocks (TBB). This technology, defined by Intel (2006) together with the release of their first multicore processors, is a widely used C++ library for parallel programming in multithread and shared memory environments. Its programming model is template-based and provide tools for thread management, synchronisation, memory allocation and task scheduling. The tool-set comes with parallel implementations of generic algorithms and also a set of classes to abstract data flows as DAGs. From a programming standpoint, its task graph description language is complex to manage and requires a conscious rearchitecture of the existing AGFA framework towards an efficient library adaptation. Moreover Contreras and Martonosi (2008) shows that the dynamic management of TBB, carries some parallelism overhead and the Random Stealing thread coordination model is less effective as application heterogeneity and core counts increase.

Cpp-Taskflow. Huang et al. (2020) define a new framework with Dynamic Flow processing, supporting large and complex task dependencies and also with a comprehensive API for easy development. It is mainly intended for High Performance Computing where millions of tasks are involved, specially in shared memory systems. The implementation is programming friendly, but is not easily extendable to support early terminations on a single task failure, management of default and external dependencies, which is already supported by the AGFA framework, and the execution of decoupled tasks. The scheduling model is based in a Random Stealing thread coordination, potentially facing the same drawbacks than TBB in this matter.

2.3. The Scheduling Problem

The main objective in the scheduling problem is to map a set of jobs onto concurrent workers and manage its execution in a way where all the dependencies are satisfied, all of this at a minimum execution cost.

Hagras and Janecek (2003) classifies the problem in *Static*, where the problem characteristics as interdata dependencies and execution times are known a priori (At compilation time), and *Dynamic* where the jobs to scheduled are created on demand and no information is known beforehand.

In our project scope the jobs to execute are known before any scheduling-related step, but the DAGs and related jobs are created in execution time, nonetheless we can get a set of problem-specific characteristics before the scheduling is performed. That is the reason in which we will elaborate on top of static scheduling.

Topcuoglu et al. (2002) group the different scheduling algorithms between *Guided Random Search based* where the scheduling is performed based in a random selection of available jobs, and *Heuristic Based* which studies the DAG characteristics to make a deterministic execution. The *Heuristic Based* algorithms can be classified in *List Scheduling, Clustering* and *Task Duplication*.

Among them, the *List Scheduling* excels on reducing the scheduling overhead while the *Clustering algorithms*, on maximising the resources usage.

On the List Scheduling algorithms, Wu and Gajski (1990) define a model based in the analysis of a modified version of the Critical Path, having the worst performing branch within an execution schema, as the determinant of an efficient scheduling. El-Rewini and Lewis (1990) defied a heuristic where the DAG and an arbitrary interconnect topology of a target machine are taken into account. This scheduling strategy is efficient in multiprocessor systems arranged with different topologies where the communication brings an overhead over the general execution, although in the highly coupled multi core systems the communication overhead is negligible.

Huang et al. (2017) propose three different clustering heuristics based in the analysis of the *Critical Path*, the *Larger Edge* and the *Critical Child*, looking for a reduction in the communication time between paths.

Although the presented methods have demonstrated to be theoretically efficient, the applicability is limited to specific scopes, as in Rho et al. (2017) which is focused in communication contention for automotive, scheduling over image-processing based jobs have been barely explored and the literature does not show significant work over functional systems.

2.4. CUDA and Thread Supporting Libraries

2.5. CUDA

(NVIDIA (2007)) Is defined as a parallel computing platform which provides an API model and a tool-set for use CUDA-enabled GPUs for general purpose processing. Has a high applicability mainly in image and graphics processing and in Artificial Intelligence applications with Neural Networks. Now stands as the state-of-the-art solution for General-Purpose computing on Graphics Processing Units. One of its main drawbacks is the expensive data copying operations between host and device memory and, as shown in Dashti and Fedorova (2017), choosing the right data transfer model depends on the problem characteristics and is left up to the programmers expertise. The new Unified Memory Access model try to overcome the data transferring overhead by keeping a common address space shareable between CPU and GPU, but Zhang et al. (2014) evidences that this model brings an increasing execution overhead and restricts flexibility for adding future optimisations.

2.6. Thread Support Libraries

QT [1] provides a platform-independent thread API implemented over platform-native threading APIs as pthreads or Win32. This implementation is written inline with the QT Meta Object System, supporting inter-object communication through Signals and Slots. On top of it stands QConcurrent, a scalable high-level API for writing multi-threaded programs transparent to low-level primitives as mutex, locks, wait conditions or semaphores. Thread support has been proven efficient as it has evolved with QT since its first implementation, but is completely bind to the QT subsystem and is not convenient when decoupling and library independent implementations are wanted.

C++ (ISO/IEC_14882:2011 (2011)) provides built-in support for threads and concurrent execution since C++ 11, which brings code portability as its bigger benefit over the already existing platform-dependant libraries. Although is not a subsystem for job scheduling, the new API provides a set of essential tools for asynchronous execution as *Futures Management*, along with concurrency control mechanisms as *Condition Variables*, *Mutual Exclusions* and *Atomics*.

3. Material and methods

For this master thesis two pathways are evaluated looking for an efficient execution of a pipeline based in image and video processing algorithms. The first one is aimed towards an efficient coordination of the execution of nodes within the pipeline, exploiting the parallelism capabilities of the modern CPUs. The second one looking for speed up each node execution by rewriting some of the existing algorithms, looking for an effective benefit of the computation capabilities of the Graphical Processing Units (GPUs).

[1] https://doc.qt.io/qt-5/threads.html

3.1. Hardware and software

The experimentation process is elaborated on top of a in-house developed framework which supports the definition and execution of pipelines. This framework is built on C++ 11 using QT5 to support the graph definition and a coordinated execution. The image and video processing algorithms are supported by OpenCV 3.4.5. The Visual Studio 2015 IDE and its compilers support the construction of the entire *VirtualDR* software and provides most of the tools for measuring the execution performance. CUDA Toolkit 10.0 is required to compile and run the CUDA based implementations in NVidia GPU processors.

	Workstation 1	**Workstation 2**
Chipset	Intel Core i5 8500	Intel Core i7 9850H
Cores	6 / 6 threads	6 / 12 threads
RAM	16 GB	32 GB
GPU		NVIDIA Q T2000
Compute		7.5

Table 1: Workstations and their setups

On what hardware is related, the experimentation is performed on two different workstations, specified in the Table 1.

The initial experimentation is performed in the Workstation 1, using custom pipeline topologies mainly for micro-benchmarking exploration, while the experimentation on CUDA based algorithms and the tests over production-enabled pipelines are executed in the *Workstation 2* which a has GPU enabled configuration.

3.1.1. Framework

The framework was conceived as a backbone to support the execution of pipelines in an structure called `Network` which is abstracted as *Directed Acyclic Graphs (DAG)* $G(V, E)$, where:

- V is the set of nodes. Each $v_i \in V$ represents a single algorithm or operation (Indistinct of its nature).
- E is the set of directed edges or connections. An edge e_{ij} connects the output properties of the node v_i with the input properties of v_j in a parent-child relationship, conditioning the v_j execution on v_i termination.

A node can lack of inputs (*Entry nodes*) or outputs (*Exit nodes* or *Leafs*), whereas others as *Pipeline Blockers* are conditioned to the execution of a set of nodes which are not explicitly related or connected by properties. The usage of the QT Property System ease the

process of propagating the outputs of the node execution to its dependants. As the framework is currently designed, the nodes are executed sequentially following a topological order, making sure that the dependencies are successfully satisfied and emergency halting the pipeline execution when a failure happens.

3.2. The Angulated Overlay pipeline

In X-rays, the Bucky-Potter grid is a device positioned opposite to patient from the x-ray tube and is helpful in reducing the quantity of scattered x-rays that reach the detector, increasing the image's contrast resolution. This grid has three different Automatic Exposure Controls (AEC) in charge of controlling the exposure when a certain amount of radiation has been received.

The VirtualDR application uses the pattern in the Bucky-Potter grid for automatically detecting the location of the AECs and the x-ray field and, in that way, calibrate the modality, determine whether the patient is well positioned in front of the x-ray tube, calculate the patient thickness and the adequate x-ray dosage for reaching a good quality image.

The Angulated Overlay is one of the main pipelines in the VirtualDR application, designed to detect the patient thickness based on a depth sensor outcome attached to the modality. Also, through image processing algorithms and the output form a grey-scale camera, detects the AECs and the x-ray field, overlaying them over the patient on real time to guide the image acquisition process.

The Figure 2 represents the Bucky-Potter grid with its guides, where the AECs are highlighted in grey and the x-ray field in yellow.

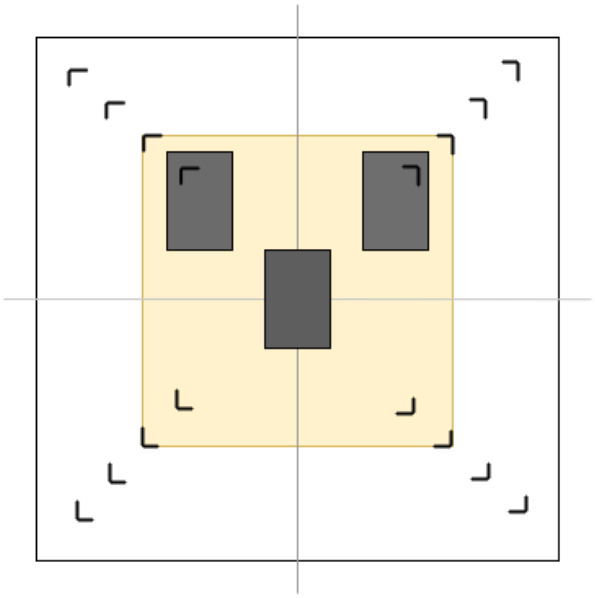


Figure 2: Bucky-Potter grid representation

This pipeline is composed of 138 nodes of Algorithms, but during the experimentation process only 134 were used.

3.3. Job Scheduler

The Job Scheduler is a module which coordinates the execution of a set of jobs exploiting the parallelism capabilities of the available computational resources. The problem scope is limited to a single workstation with a multi-core CPU. Towards the definition a reliable model that covers the problem specifications, the next architectural aspects are considered:

- Decouple the definition of *Jobs* as execution units, from *nodes* as algorithms or operations. A job performs the algorithm or operation of one node at least.
- The execution of a single node is coordinated from a single thread.
- The production of the jobs should be decoupled to the execution itself.
- The coordination needs to be effective in the way the nodes within the jobs to be executed have all their dependencies satisfied.
- Beware on avoiding deadlocking scenarios.

3.3.1. Producer - Consumer pattern

The Producer - Consumer design pattern is presented as a solution to the homonym problem in Dijkstra (1972) for multiprocess synchronisation, where a *Producer* is in charge of generating data and feeding a common buffer, and a *Consumer* is responsible of consuming the data from the buffer and processing it. This pattern set the basis of a decoupled oriented design on modern systems where the old complex monolithic applications are now abstracted as a set of loosely-couple light services. In line with the pattern specifications, in our scheduler design the entity *Producer* produces execution ready jobs that are delivered to a *Single Access Queue* which behave as a buffer and where the *Task Manager* consume from to proceed with its execution.

3.3.2. Single Access Queue - Buffer

The *Single Access Queue* behaves as a synchronised buffer, where the *Producer* places the jobs containing the nodes ready to be executed. In our design, this entity is integrated within the *Task Manager* and provides an interface which allows the insertion of new jobs from different producers. As a centralised resource, the integrity of the insertion and extraction operations are controlled using a single *Mutex*, making sure that only a single access to the Queue can be performed at the time. Every successful insertion is notified to the *Consumer* so the jobs can be extracted and executed. The queue design diverges from the original conception posed by Dijkstra (1972) in the way that the absence of available

jobs will not lead to a blocking scenario during the extraction, instead a `NULL` job is returned, delegating the decision over the *Consumer*. The queue size is not explicitly limited because the nature of the problem in this project scope will never lead to an uncontrolled creation of jobs. This architectural decision reduces the execution overhead by removing unnecessary checks and potential locking scenarios. The queue structure responds to a Linked List, where the time complexity of the insertion and non-indexed extraction operations are constant ($O(1)$).

3.3.3. Task Manager - Consumer

The *Task Manager* is the entity which consumes the jobs ready for execution from the *Single Access Queue* and also holds a set of workers which performs the jobs execution in single and independent threads. The set of worker threads are defined using the Thread Pool pattern (Garg and Sharapov (2001)) leaving a fixed number of threads during the whole program execution while they are only destroyed on program termination. This pattern shows an advantage over other thread management approaches, by reducing the latency of constantly creating and destroying threads, which negatively impacts the general system performance. All the workers are allocated during the application start-up stage and remain in waiting state.

The task on determining how to proceed when there are no jobs available for execution in the Single Access Queue, is delegated to the workers itself.

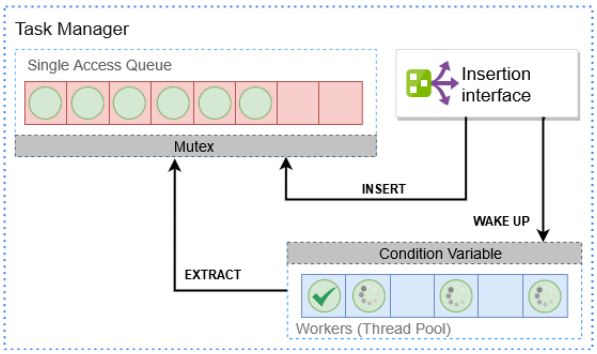


Figure 3: Task Manager schematic representation

The design is depicted in Figure 3 and the behaviour of this module can be described as follows:

1. The workers are in `waiting` state.
2. The *Task Manager*, as holder of the Queue insertion interface, notifies the availability of new job to its workers through a common concurrent `condition_variable` (Arpaci-Dusseau and Arpaci-Dusseau (2018)), waking the workers up.
3. Each worker ask for jobs to the *Single Access Queue* and they get either a job ready to be executed or a NULL instance which means that all the available jobs were consumed.
4. In case of a `NULL` job, the worker go back to waiting state until new jobs are inserted.
5. When a valid job is retrieved, the worker proceed with its execution and once is done, ask the queue for new available jobs.

The *Consumer* is always up regardless on the availability of jobs to be executed or the state of the application since there is no awareness on the running *Producers*. As it is not practical to have multiple *Task Managers* across the application, we make sure that only a single instance lives during the entire execution of the application, following the guidelines of the Singleton Design Pattern.

3.3.4. Network Setup

The problem characteristics and the current framework have some constraints that needs to be overcome towards an effective parallel execution.

Graph Representation. The `Network` structure defined by AGFA is accurate in defining a DAG, but is not efficient when traversing the graph is required. There are two representations from the Graph Theory helpful in easing the graph exploration and evaluation (Cormen et al. (2001)):

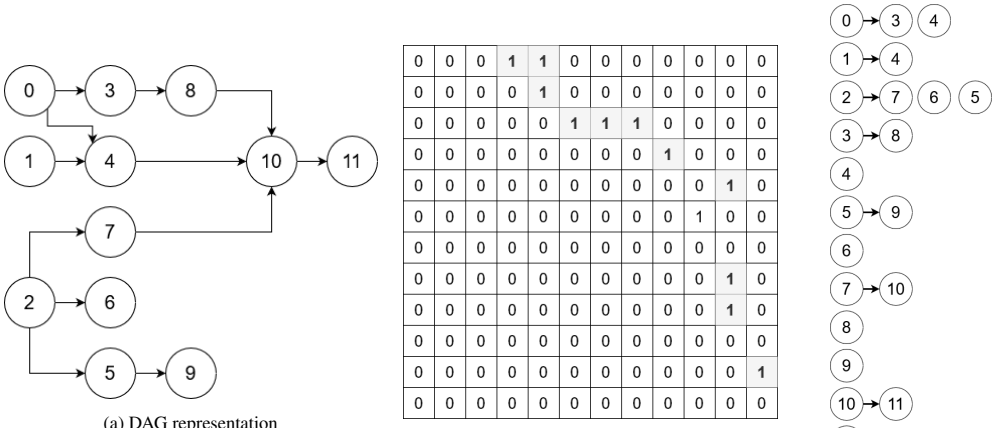
The *Adjacency Matrix* relates the graph nodes through an indexed matrix M where $m_{ij} \neq 0$ represents a directed connection from v_i to v_j.

The *Adjacency List* relates a node with its children through a list L where L_i is the sub-list of the children of v_i. This representation is more space efficient than the Adjacency Matrix specially in sparse graphs where the nodes are not densely connected. In terms of computational complexity, traversing the graph is better through adjacency lists since the node children are retrieved in constant time $O(1)$ instead of linear time $O(n)$ which is the case for the Adjacency Matrix.

The Figure 4 shows an example of a DAG and its representations.

Pipeline Blockers. This is an exceptional behaviour of the pipelines where a non-connected node v_i conditions its execution after the successful termination of a set of nodes $v_0 ... v_{i-1}$. This feature breaks with the definition of a graph, since the *Blocker* node is not part of the graph as long as is not explicitly connected to any graph node. The usefulness comes out in scenarios like the network dump, where the graph state is saved in the latest stages of a pipeline execution. To preserve this behaviour in a concurrent execution, such nodes are connected following the algorithm 1

This strategy behaves as a Graph Cut by limiting the concurrent execution of a section of the graph, to the successful termination of the pipeline blockers.



(a) DAG representation

(b) Adjacency Matrix

(c) Adjacency List

Figure 4: DAG and its representations

Algorithm 1: Insertion of Pipeline Blockers

input : The topologically ordered Adjacency List L of the Graph $G(V, E)$
input : The Pipeline Blocker nodes B
for $v_i \leftarrow v_0$ **to** $v_n \in L$ **do**
  $BA \leftarrow$ The list of $b \in B$ that must be executed before v_i
  for $ba_i \leftarrow ba_0$ **to** $ba_n \in BA$ **do**
    $connect(G, ba_i, v_i)$
  end
end

3.3.5. Scheduler Algorithm

Analysing the existing pipelines, an heuristic is proposed to reduce the coordination overhead by clustering a set of nodes within a single job. Based on the followed approach, this scheduling algorithm can be classified as Static Clustering-based. It is not dynamic since is not evolved across the pipeline execution but the heuristic is applied only once prior execution.

Background. The jobs production and consumption have an implicit overhead related with the coordination process. In the proposed scheduling approach most of this overhead comes from the insertion and extraction operations from the *Single Access Queue*, the dependencies propagation and verification process and the *Producer* termination control. We can reduce the overall execution time by smartly reducing the coordination effort or removing it where is not necessary.

Heuristic. The heuristic for the proposed scheduler algorithm, explores the graph looking for maximal paths $p_i(V)$ of weakly connected vertexes $v \in p_i$, where all of them are continuously dependant and only have a single parent and a single children. The starting and ending point of a path can be analysed as follows:

- Having $v_{i-1}, v_{i-2}...$ as the parent nodes of v_i, if $indeg(v_i) \neq 1$ (Number of parent nodes), then v_i can be considered the starting point of a path.
- Having $v_{j+1}, v_{j+2}...$ as the children nodes of v_j, if $outdeg(v_j) > 1$ (Number of children nodes), then v_j can be considered as the ending point of the path.
- If $indeg(v_i) = 1$, then v_i is a potential inner node of a path.

The Algorithm 2 depicts the procedure that returns a list with the selected paths which, latterly, will be added into single execution jobs.

The Figure 5 shows a DAG with the scheduling routine. The vertexes inside the dotted red boxes denotes the nodes that can be joined within a single job. The *Makespan* of the scheduling process is defined as the overall time required to execute the graph. In this hypothetical scenario where the execution time t of every node is the same along with the coordination overhead time c, the makespan of the graph in Figure 5 can be presented as $makespan(G) = 5t + 4c$.

The Figure 6 represents the graph of jobs in the Figure 5 with the jobs unified after the heuristic application. The execution time of the jobs C, D and H is $2t$ as each job groups two nodes, while for the I job is $3t$. In that way, $makespan(G) = 5t + 2c$.

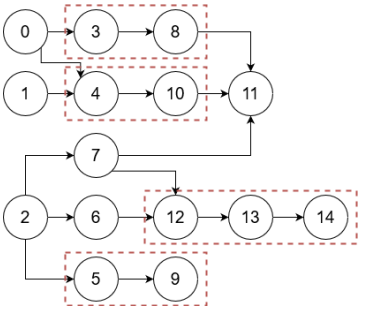
The defined heuristic theoretically reduces the makespan by relieving the coordination overhead, increasing the continuous usage of the workers and reducing the threads idle time.

3.3.6. Producer

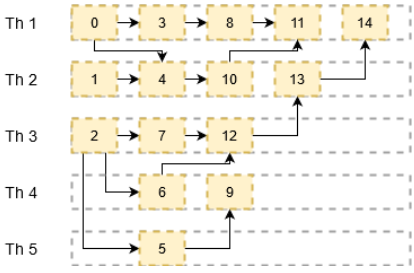
The *Producer* entity is in charge on coordinating how the network nodes will be executed. Is defined on top of the `Network` definition of the existing framework, having direct access to the different algorithms and connec-

Algorithm 2: Finding heuristic-accepted paths within the graph

input : The Adjacency List C of the Graph $G(V, E)$
input : The Adjacency List of parents P of the Graph $G(V, E)$
output: List of paths LP
Function `build_bridge`(*node v, path p*)**:**
  if *v has more than 1 parent in P* **then**
    return
  p.insert(v)
  if *v has only one children in C* **then**
    build_bridge(C[v], p)
end
for $v_i \leftarrow v_0$ **to** $v_n \in G$ **do**
  if v_i *has only one parent in C* **then**
    continue
  let $p \leftarrow$ empty_path
  if v_i *meet the characteristics to be an starting point* **then**
    build_bridge(v_i, p)
  if p *is not empty* **then**
    LP.insert(p)
end



(a) DAG representation

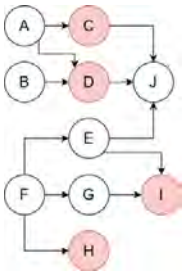


(b) Scheduling diagram

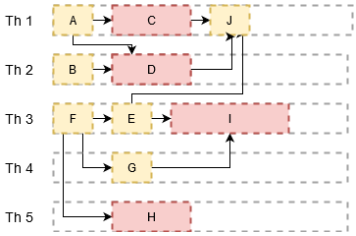
Figure 5: DAG and its scheduling representations

tions which composes the pipelines. Towards an effective coordination on the jobs production, the next design aspects were evaluated:

Special Dependency Management. The framework already provides a support to verify whether the node dependencies has been satisfied, but this approach is failure susceptible to an exceptional feature called *Default*



(a) DAG representation



(b) Scheduling diagram

Figure 6: DAG of Figure 5 with unified jobs and its scheduling representations

Properties. These are special kind of properties which have a predefined value and are, by definition, already satisfied even when a parent node changes its value during execution. This problem is not visible during a sequential execution following a topological order, but can be experienced during a parallel execution. The followed strategy consists in creating an Adjacency List where child nodes are related to its parents and check whether the child's parents has been already processed, then run the framework's dependency check. If both checks are successfully accomplished, we say that the node is ready for scheduling,

Error management and stop criteria. The framework as defined per AGFA, halts the pipeline execution when an algorithm next to be executed is not able to satisfy all its dependencies. This also happens when the algorithm execution fails or throws an exception. To cover this behaviour, the producer is implanted with an execution state which is verified and updated after every job execution. An algorithm failure stops the jobs production process and wait for all the concurrently running jobs to terminate, before halting the pipeline execution. Is considered a successful execution when all the nodes are successfully executed.

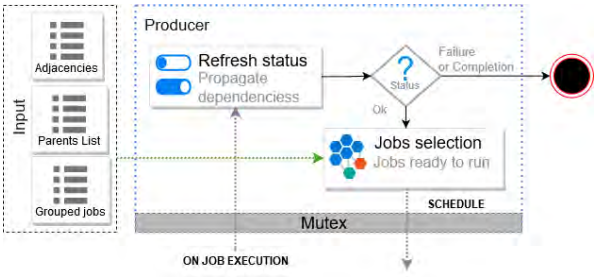


Figure 7: Producer schematic representation

Behaviour. The *Producer* behaviour as it is represented in the Figure 7 can be depicted as follows:

1. The *Producer* is initialised with the Adjacency Lists (Parents P and children C) and the list of jobs T obtained from the application of the scheduling strategy.

2. The *Entry nodes* are listed to be executed since they do not have dependencies to be satisfied.
3. If the node is the starting point of a job $t_i \in T$, then t_i is sent for execution, otherwise a new job containing the node is created and also sent for execution.
4. The Producer is notified of a job termination through an attached callback function.
5. If the job termination was successful and there are more nodes to evaluate:
   (a) Check the pipeline status. If it is in failure status then nothing should be done.
   (b) Extract the latest node executed by the job.
   (c) Propagate the node output throught the pipeline.
   (d) Extract those children nodes where the dependencies has been successfully satisfied.
   (e) Per every extracted node, evaluate if the node is the starting point of a job $t_i \in T$, then t_i is sent for execution, otherwise a new job containing the node is created and also sent for execution.
6. If the job termination was unsuccessful or there are not more nodes to evaluate.
   (a) In case of unsuccessful termination, fail the pipeline evaluation so all the running jobs get aware that no more processing should be performed.
   (b) Wait for the termination of the nodes in execution statie.
   (c) Declare the termination.

3.3.7. Integration

The *Scheduling Heuristic* and the *Network Setup* build the set of structures required for performing the scheduling process, thus the construction of the adjacency lists and the application of the heuristic need to be performed before the scheduling process is started. This process, albeit is not executed during the scheduling process, represents an overhead in the global system performance. As long as the pipeline structure does not change across the application execution, these setup steps are only performed once following a Lazy Initialisation approach where the structures are built if and only if they have not been constructed before.

The Figure 8 gives an schematic overview of how the adjacency lists and the list of group jobs are built.

Finally the general overview of the presented job scheduler is presented in the Figure 9.

3.4. CUDA Support

As we mentioned before, the existing pipelines implemented by AGFA are mostly composed by image processing algorithms which can potentially get performance benefit when are computed in GPU enabled systems. The usage of GPUs to run concurrent operation

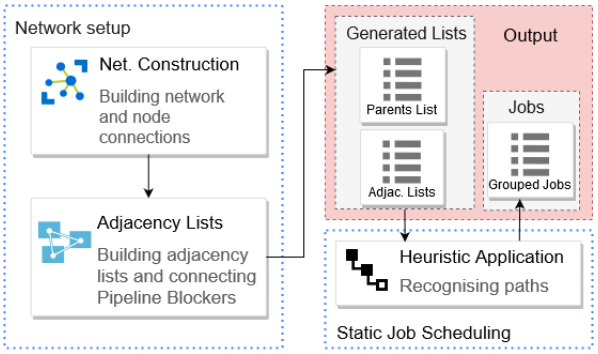


Figure 8: Pipeline transformation - Schematic representation

has shown outstanding result on what computation time is related, however an overall performance degradation can be evinced when the data needs to be transferred from and to the Device memory.

Looking for a global performance improvement, we evaluated a set of memory management techniques and also delivered the CUDA implementation of a set of slow performing algorithms.

3.4.1. Memory Management

Towards a further successful integration into a pipeline, it is necessary to place some attention on how the memory is allocated in the Device and how the memory is transferred from the Host to the Device.

Memory Allocation. The process of allocating memory in the Device generates an overhead that surpasses the execution time in CPU of most of the algorithms. This factor make pointless the effort of rewriting some image processing algorithms in CUDA.

To tackle this issue we took advantage of the pipelines structure and how the nodes input mutates across multiple executions. We noted in most of the nodes where image processing algorithms are executed, that the matrices dimensions do not change regardless of their content and how many times the node is executed in the pipeline life-cycle. From this observation it is concluded that a single device memory allocation per node can be performed on its first execution, making reusable the memory space and reducing the re-allocations only when the size of the node input changes.

On the context of the OpenCV GPU Matrices, this memory re-usage strategy reduces the computational complexity of the object creation from linear $O(n)$ to constant $O(1)$, preserving the space where the matrix data is placed and limiting the object creation when there are structural changes, eliminating the Device allocation overhead.

Host - Device data moving. One of the most promising features of the recent CUDA enabled devices is the im-

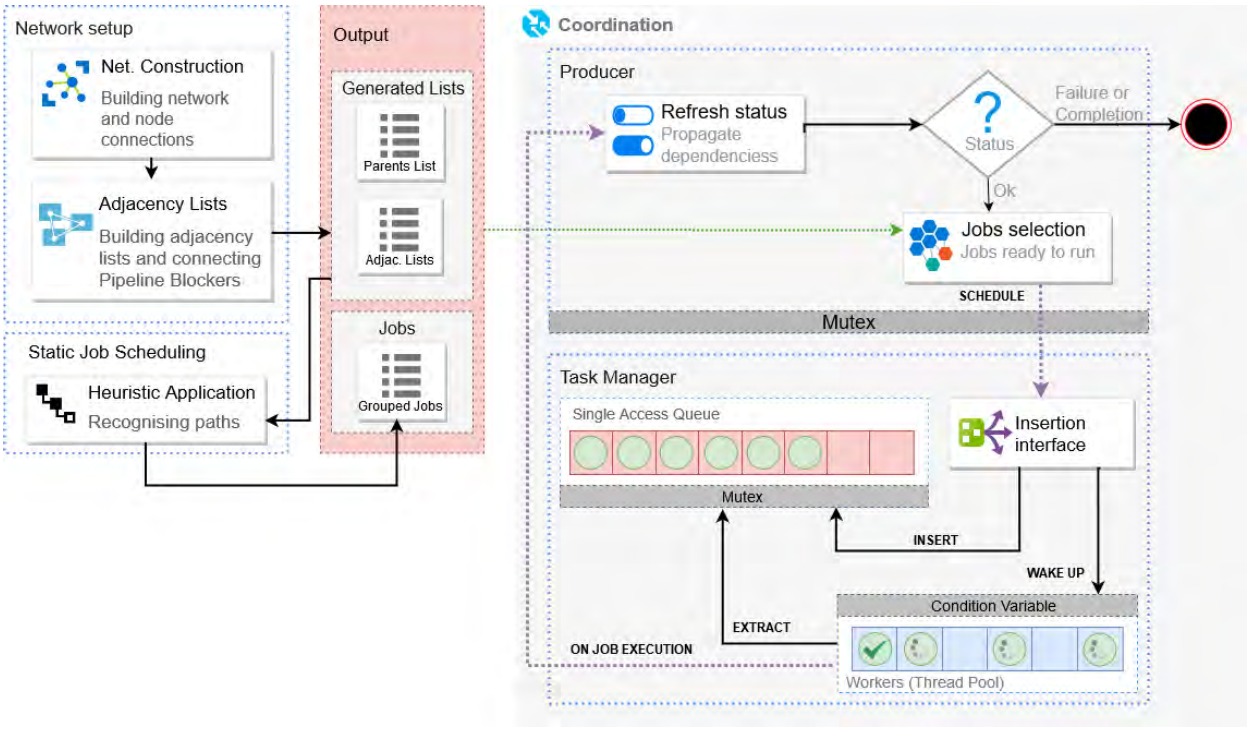


Figure 9: General representation of the proposed scheduling process

plementation of the Unified Memory Access as a common memory space shared between CPU and GPU. Albeit this new memory model simplifies how memory is managed across Host and Device, Zhang et al. (2014) exposes its limitations and demonstrate through micro-benchmark tests that the performance improvement is limited to the kernel design itself, increasing the code complexity of very simple models, reaching in most of the cases non outstanding results. This memory management model is relatively new and there is no backward compatibility, what limits their usage reducing the software compatibility.

Using the non-cached CPU Write Combined memory looking for speed up the data moving has not shown any performance benefit and instead has shown a big drawback on its temporary allocation characteristic which wipe-out in very short cycles.

Getting attached to our previous approach based on pre-allocations we experimented a good performance improvement moving data from CPU to GPU when the destination has a pre-alocated the space of memory where the new data will be placed, relieving the allocation overhead.

3.4.2. Algorithms

The Table 2 shows the list of algorithms, in the context of the Angulated Overlay pipeline, which are currently implemented in CUDA. OpenCV provides already reliable and efficient implementations of most of them, having the same output of their CPU-based counterparts, while the blue-highlighted in the Table 2 lack of CUDA-based implementation.

Algorithms	
BitwiseAnd	Remap
CalibratedCoordinates	Rotate
ConvertTo	RotatedCoordinates
CopyTo	ShiftDecentered
CountNonZero	ShiftedInRange
ExtractChannel	Subtract
InRange	Zeros
MinMaxLoc	

Table 2: CUDA implemented algorithms. Blue-highlighted the algorithms which required an implementation from scratch

In this section we will only emphasise in the set of algorithms implemented from scratch and their design aspects.

Global Characteristics. The compute capability of the CUDA enabled device is 7.5, where having blocks of 32×32 with 1024 threads per block, maximises the occupancy up to 32 wraps (Figure 10).

The number of registers per thread are as less as possible, trying not to exceed 64 registers avoiding the stress in the wrap occupancy and reaching better computation times.

The grid size is set to change according to the algorithm's input size reducing the number of wasted threads, and the kernels are designed to perform operations over a single pixel.

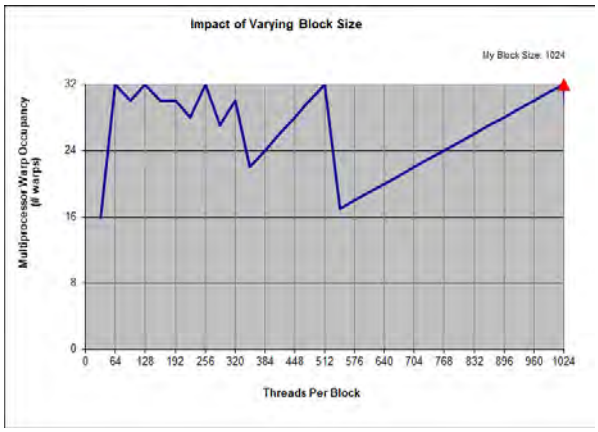


Figure 10: Impact of varying block size for CUDA Compute Capability 7.5

The implementation is supported also with the CUDEV module of the OpenCV library that provides tools for extending the CUDA support to the `cv::cuda::GpuMat` class. This class is the CUDA counterpart of the `cv::Mat`, where the data is allocated in the GPU Device.

Rotation. The GPU implementation of this algorithm is already supported by OpenCV but is not extensible to 2 channel matrices, which are widely used across the pipeline.

The new implementation supports up to 4-channel matrices with rotation angles of 90, 180 and 270 degrees, as the CPU counterpart. The algorithm kernel depicted in the Algorithm 3 performs a clockwise rotation using flips and transpositions to calculate the new pixel position as follows:

- *90 degrees:* Vertical flip, then transpose.
- *180 degrees:* Horizontal flip, then vertical flip.
- *270 degrees:* Horizontal flip, then transpose.

Finally the pixel information is copied into its new location. This pixel coordinates are extracted from the *block* and *thread* assigned to run the kernel in the execution grid.

Calibrated Coordinates. The GPU implementation of this algorithm is not supported by OpenCV,

This function depicted in the Algorithm 5 is designed to calibrate a set of coordinates based in the scene scale and the relation between the depth information obtained from the depth sensor and the theoretical distance between the camera and the scene, calculated with the Field Of View as $d = 1/\tan(FOV/2)$.

Algorithm 4 shows the generic kernel used for this algorithm, where a custom function run over each pixel and writes its output in another matrix.

Algorithm 3: Kernel for clockwise rotation

```
input : A pointer S to the matrix containing the
        source data
input : A pointer D to the matrix containing the
        destination data
input : The rotation angle a
(x, y) ← Obtain the x and y coordinates from the
  block and thread.
if x ≥ S.cols || y ≥ S.rows then
    return
switch The value of a do
    case 90 do
        newy ← S.rows − 1 − y;
        newx ← newy; newy ← x;
        break;
    case 180 do
        newx ← S.cols − 1 − x;
        newy ← S.rows − 1 − y;
        break;
    case 270 do
        newx ← S.cols − 1 − x;
        newy ← newx; newx ← y;
        break;
    otherwise do
        newx = newy = 0;
    end
end
D(newy, newx) ← S(y, x)
```

Algorithm 4: Generic Kernel

```
input : A pointer S to the matrix containing the
        source data
input : A pointer D to the matrix containing the
        destination data
input : op as a custom function to apply over a
        single pixel
(x, y) ← Obtain the x and y coordinates from the
  block and thread.
if x ≥ S.cols || y ≥ S.rows then
    return
D(y, x) ← op(S(y, x))
```

InRange. The GPU implementation of this algorithm is not supported by OpenCV, but is supported in its CPU version.

It is a classical thresholding algorithm which verifies if the pixel stands within a range, regardless of the matrix' number of channels. This implementation in Algorithm 6 is a generalised version of the thresholding algorithm where the pixel rejection criteria is left to a custom made function which verifies if the pixel value is within the input range.

RotatedCoordinates. This algorithm is designed to rotate matrices containing Cartesian coordinates. The in-

Algorithm 5: Function for calibrating 2D coordinates based in depth information

input : The 3D point $p(i, j) \in M$ which contains the x and y coordinates to calibrate as (p_x, p_y) and its depth information in p_z
input : s as the projection scale
output: c as the calibration of (p_x, p_y)
$fov\_h \leftarrow$ Horizontal Field Of View angle
$fov\_v \leftarrow$ Vertical Field Of View angle
$tan\_fov\_h \leftarrow tan(fov\_h/2) * 2$
$tan\_fov\_v \leftarrow tan(fov\_v/2) * 2$
$normalizedX \leftarrow p_x/M.cols - 0.5$
$normalizedY \leftarrow 0.5 - p_y/M.rows$
$c.x \rightarrow normalizedX * p_z * scale * tan\_fov\_h$
$c.y \rightarrow normalizedY * P_Z * scale * tan\_fov\_v$
$c.z \rightarrow p_z * scale$

Algorithm 6: In Range kernel

input : A pointer S to the matrix containing the source data
input : A pointer D to the matrix containing the destination data
input : d and u as the down and up range limits
input : A function fun which determines if a pixel is within the d - u range
$(x, y) \leftarrow$ Obtain the x and y coordinates from the block and thread.
if $x \geq S.cols \parallel y \geq S.rows$ **then**
  return
$D(y, x) \leftarrow (fun(S(y, x), d, u))?255 : 0;$

put is a 2-channels matrix where the first channel represents the x coordinate, and the second channel the y coordinate in a Cartesian representation where the origin is located at the centre of the matrix. The algorithm rotates the matrix in 90, 180 and 270 degrees clockwise, but also transforms the coordinate at each pixel to follow the same rotation taking into account its origin.

The same kernel of the rotation operation is used (Algorithm 3), but is extended with the coordinates transformation function in the Algorithm 7

Shifted InRange. This is basically an extension of the InRange algorithm where the ranges are shifted. For this implementation, the same kernel than the InRange algorithm is used and the ranges are shifted prior the algorithm computation.

4. Results

4.1. Remarks

This section presents the results obtained from the experimentation process. The execution times might be presented either in seconds or milliseconds based on

Algorithm 7: Decision function for the Rotated Coordinates algorithm

input : p as the coordinates to transform
input : a as the rotation angle
output: c as the coordinates transformed
switch *The value of a* **do**
  case *90* **do**
    $swap(p_x, p_y)$;
    $p_y \leftarrow -p_y$;
    break;
  case *180* **do**
    $p \leftarrow -p$; break;
  case *270* **do**
    $swap(p_x, p_y)$;
    $p_x \leftarrow -p_x$;
    break;
end
$c \leftarrow p$

the test characteristics. Every scenario was calculated 10 different times and the minimum (MIN), maximum (MAX) and the average (AVG) execution times are presented, except for the tests performed in CUDA where the average execution time is more relevant. The Performance Improvement PI metric is calculated over the average execution times as $PI = t_{seq}/t_n$ where t_{seq} is the average execution time in the sequential model, and t_n is the execution time on n threads.

4.2. Micro-benchmark

The micro-benchmark process is performed over controlled scenarios to verify the feasibility of a set of hypothesis.

4.2.1. Number of Threads

Goal. Verify how the parallel execution of a set of decouple tasks gets benefit of the amount of available thread.

Setup. Pipeline with 20 independent jobs with $O(n^2)$ computational complexity (Random square matrices multiplications of 4000, 2000, 1000 and 500 rows/cols).

Outcome. The Figure 11 shows the performance improvement in each scenario. Regarding the maximum reached improvement: The pipeline with matrix multiplications of 4000×4000 runs $4.92x$ faster on 12 threads. The pipeline with matrix multiplications of 2000×2000 runs $5.21x$ faster on 12 threads. The pipeline with matrix multiplications of 1000×1000 runs $5.06x$ faster on 11 threads. The pipeline with matrix multiplications of 500×500 runs $4.42x$ faster on 8 threads.

4.2.2. Execution strategies

Goal. Verify the parallelism benefit obtained from three different parallel execution strategies.

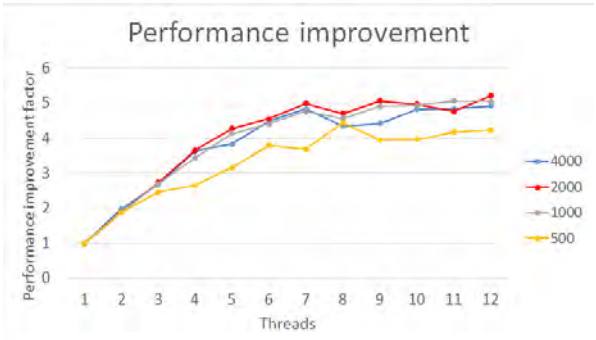


Figure 11: Performance improvement on executing random square matrices multiplications of 4000, 2000, 1000 and 500 rows/cols in a pipeline of 20 jobs over 1 - 12 threads

Setup. Pipeline represented by the graph in Figure 12 with 12 nodes with $O(n^2)$ computational complexity (Random square matrices multiplications of 4000, 2000, 1000 and 500 rows/cols). The nodes execution is constrained by dependencies.. Execution is performed in the Workstation 1 (See Table 1).

The critical path as the maximal sequence of nodes that cannot be executed in parallel is: $0 \rightarrow 3 \rightarrow 8 \rightarrow 10 \rightarrow 11$ with an average execution time of 44.230786 seconds.

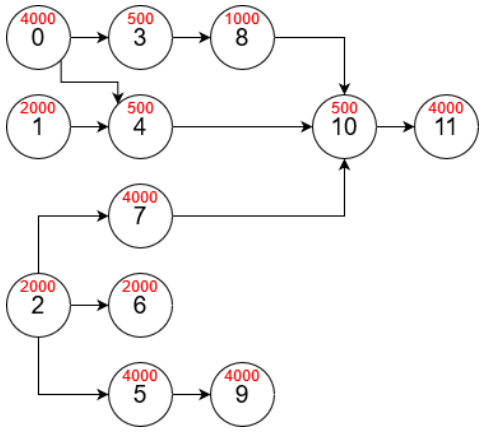


Figure 12: DAG for testing execution strategies. Square random matrix multiplications with rows/cols size indicated in red colour.

Strategy 1. Execute the DAG concurrently following a topological order. Pause the scheduling of new tasks when the dependencies of the next node to send are not satisfied. Resume the scheduling once the dependencies has been satisfied. The figure 13 shows the result of the execution.

Strategy 2. Execute the DAG concurrently, continuously exploring all the nodes and scheduling only those whose dependencies are satisfied. The exploration is limited in each iteration on the set of nodes which has not been scheduled. The exploration is over when the latest node is scheduled. The figure 14 shows the result of the execution.

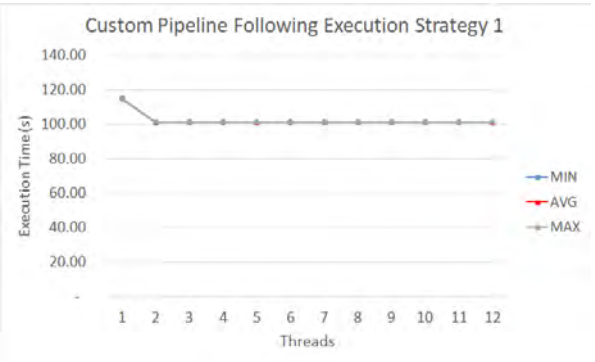


Figure 13: Execution results for the pipeline in Figure 12 following the Strategy 1

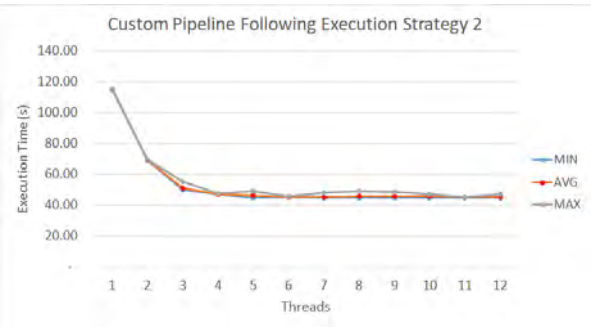


Figure 14: Execution results for the pipeline in Figure 12 following the Strategy 2

Strategy 3. Execute the DAG concurrently in a chain-like way. Get the initial nodes where their dependencies are satisfied and schedule their execution. Wait for the successful execution of a node and explore and schedule its children nodes whose dependencies are satisfied. The exploration is over when the latest node is scheduled. The figure 15 shows the result of the execution.

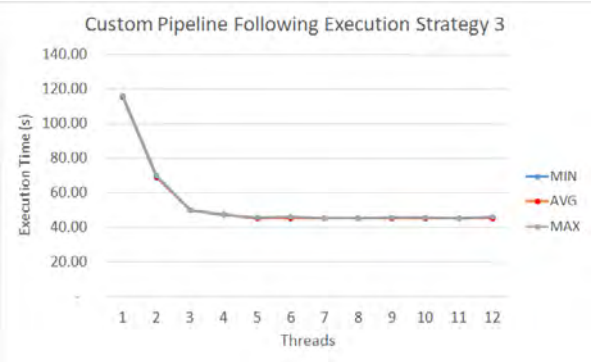


Figure 15: Execution results for the pipeline in Figure 12 following the Strategy 3

Outcome. The Figure 16 shows the performance improvement in each strategy. Regarding the maximum reached improvement: The Strategy 1 runs 1.14x faster on 2 threads and its minimum execution time is 101.22s. The Strategy 2 runs 2.55x faster on 11 threads and its minimum execution time is 44.93s. The Strategy 3 runs 5.56x faster on 5 threads and its minimum execution time is 44.92s.

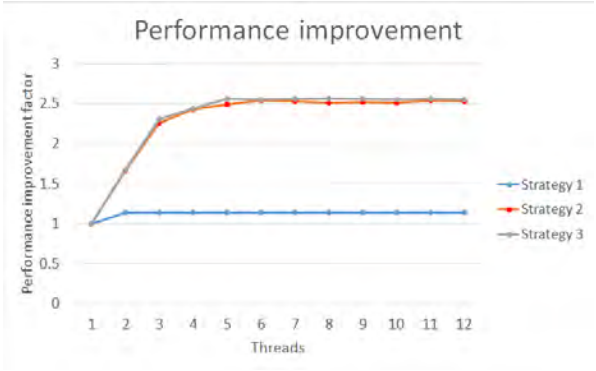


Figure 16: Performance improvement on the execution of the Strategies 1, 2 and 3

4.3. Execution Over The Angulated Overlay Pipeline

The next set of tests were performed over the Angulated Overlay Pipeline. It is important to remark that the execution times are related with the pipeline execution while the entire application is in execution state.

The execution strategy chosen for this test set is the Strategy 3, which is the one used in the final scheduler described in the section 3.3.6, also including the Scheduling Heuristic described in 3.3.5. Execution is performed in the Workstation 2 (See Table 1).

Goal. Verify how the proposed scheduler benefits the Angulated Overlay pipeline execution with and without the Scheduling Heuristic.

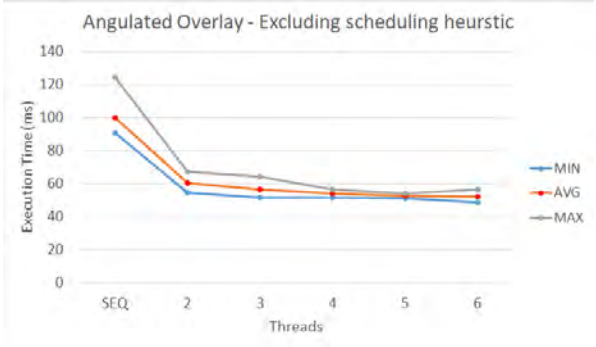


Figure 17: Parallel execution results for the Angulated Overlay pipeline without the Scheduling Heuristic

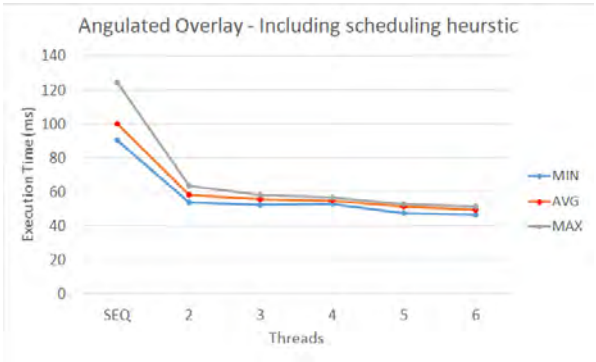


Figure 18: Parallel execution results for the Angulated Overlay pipeline with the Scheduling Heuristic

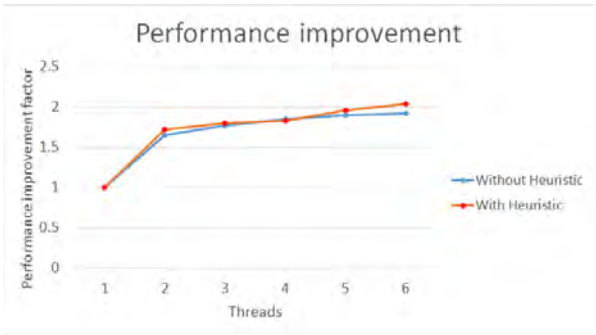


Figure 19: Performance improvement on the execution over the Angulated Overlay pipeline with and without the Scheduling Heuristic

Outcome. The Figured 17 and 18 shows the pipeline execution times on different number of threads without and with the heuristic implemented. Finally the Figure 19 shows the related performance improvement of both tests. Without the heuristic, the maximum performance improvement factor is 1.91x with a minimum execution time of 48.7743ms at 6 threads, while when the heuristic is included, it reaches a performance improvement of 2.03x with a minimum execution time of 46.5982ms also at 6 threads.

Finally the heuristic was able to group 52 nodes within 21 jobs, distributing the pipeline in a total of 107 jobs.

4.4. CUDA

Execution is performed in the Workstation 2 (See Table 1).

4.4.1. Memory Management Micro-benchmark

This test was intended to show the benefits of memory pre-allocation in the process of creating and performing host-to-device memory copying on GpuMats, which is the C++ class of OpenCV that supports the creation of matrices in the GPU. The Table 3 shows the tests results using two experimental matrix sizes used in the Angulated Overlay pipeline: 32 bit Float matrix of 3 channels with sizes 480×640 and 43×43.

Op.	Matrix Size.	Execution Time (ms)		
		Mat	*GpuMat*	*GpuMat Prealloc.*
Create	43×43	0.00232	0.00818	0.00345
	480×640	0.01335	0.37169	0.00363
Upload	43×43		0.42612	0.02905
	480×640		0.87705	0.43635

Table 3: Average creation time of an OpenCV Mat and GpuMat, and host-to-device memory copying.

4.4.2. Algorithms Execution

The Table 4 shows the execution times of the five algorithms implemented in CUDA from scratch, in its worst performing node. The nodes are part of the Angulated Overlay pipeline and the input matrix dimensions on each node are specified in terms of *rows* $\times$ *cols* $\times$ *channels*.

The Rotated Coordinates algorithm in the node `rot_coords` shows an average performance improvement of 18.45x. The Calibrated Coordinates algorithm in the node `calibratedcoordinates` shows an average performance improvement of 5.37x. The Shifted In Range algorithm in the node `aec1_project_inpaint` shows an average performance improvement of 2.76x. The In Range algorithm in the node `detector_project` shows an average performance improvement of 3.09x. The Rotate algorithm in the node `rotated_depth` shows an average performance improvement of 1.08x.

4.4.3. Algorithms Micro-benchmark

The next set of tests evaluate the CUDA algorithm implementations, by using randomly generated square matrices of 30, 100, 200, 400, 800, 1600 and 3200 rows, comparing its execution time against the CPU implementation.

Rotated Coordinates Algorithm. Figure 20 shows the average execution times of the Rotated Coordinates algorithm in CPU and GPU using randomly generated `Float 32bit` matrices of two channels, with values in the range of -1000 and 1000. The rotation angle is 90 degrees as it is the worst performing scenario in the CPU execution (See Table 4).

Calibrated Coordinates Algorithm. Figure 21 shows the average execution times of the Calibrated Coordinates algorithm in CPU and GPU using randomly generated `Float 32bit` matrices of 3 channels, with values in the range of 0 and 1500. The other input parameters as the Fields of View and the Scale were randomly generated as they are not relevant in the performance of the algorithm.

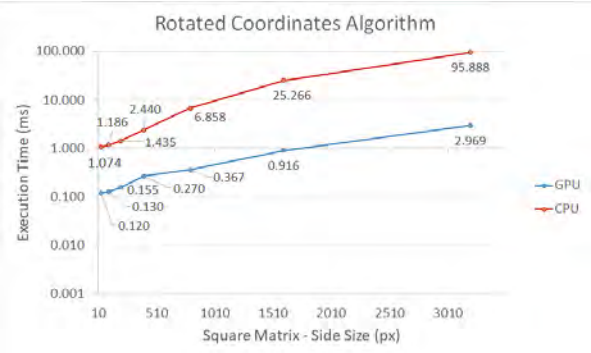


Figure 20: Average execution time for the Rotated Coordinates algorithm in CPU and GPU, using a rotation angle of 90 degrees

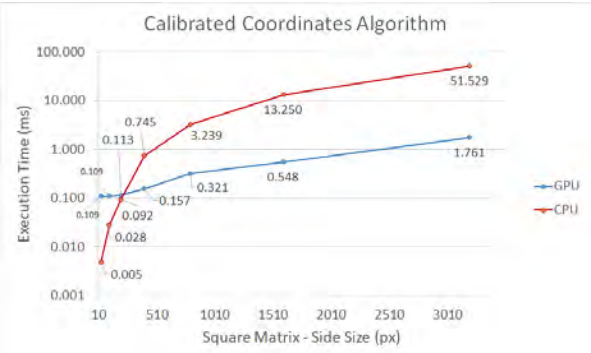


Figure 21: Average execution time for the Calibrated Coordinates algorithm in CPU and GPU

In Range Algorithm. Figure 22 shows the average execution times of the In Range algorithm in CPU and GPU using randomly generated `Float 32bit` matrices of two channels, with values in the range of -200 and 200. The filtering ranges are -104 to -112 for the channel 1, and -54 to -27 for the channel 2.

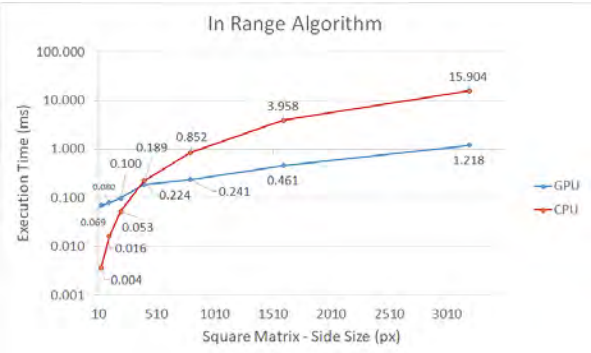


Figure 22: Average execution time for the In Range algorithm in CPU and GPU

Rotate Algorithm. Figure 23 shows the average execution times of the Rotate algorithm in CPU and GPU using randomly generated `Float 32bit` matrices of one channel, with values in the range of 0 and 1500. The

Node	Algorithm	Input	Type	Execution Times (ms)	
				CPU	*GPU*
rot_coords	Rotated Coordinates - 90 deg.	480x640x2	Float 32	4.29266	0.23262
calibratedcoordinates_inpaint	Calibrated Coordinates	480x640x3	Float 32	1,46079	0.27165
aec1_project_inpaint	Shifted In Range	640x480x2	Float 32	0.49932	0.18067
detector_project	In Range	640x480x2	Float 32	0.49048	0.15870
rotated_depth	Rotate - 90 deg.	480x640x1	Float 32	0.36573	0.33680

Table 4: Execution time of the CUDA re-implemented algorithms in its worst performing node

rotation angle is 90 degrees as it is the worst performing scenario in the CPU execution (See Table 4).

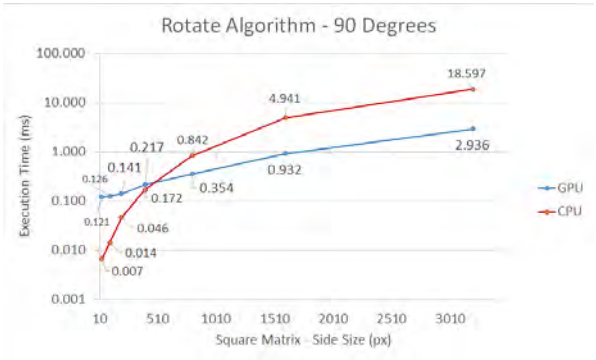


Figure 23: Average execution time for the Rotate algorithm in CPU and GPU, using a rotation angle of 90 degrees

Shifted In Range Algorithm. This algorithm was not deeply evaluated since its implementation is based in the In Range Algorithm, but shifting the input ranges before execution.

5. Discussion

5.1. Micro-benchmark Process

From the initial steps of the micro-benchmark process in Figure 11 it can be seen that the overall performance improves when the number of allocated threads is closer to the number of CPU cores, but it deserves to emphasise than it might not always the case. The threads are allocated and managed by the Operating System (OS) so, the full availability of CPU cores is hypothetical but not practical, explaining why having more threads than CPU cores still show some improvement. However the graph is clear in showing that all the scenarios experiment a performance improvement up to 6 threads, where at least the 85% of the maximum experimented performance improvement is reached. Right after the results might vary from having a plateau to facing a performance degradation by the exhaustion of the computational resources. For instance, the pipeline where matrices of 500×500 are multiplied, faced a performance degradation running over 7 threads, while the other tests faced such situation over 8 threads. To conclude, the maximum number threads to use to tackle a parallel problem can be related with the number CPU cores, and although a higher performance can be reached adding threads beyond such number, the risk of facing a sudden performance degradation increases.

The initial outcome from the experimentation in the section 4.2.2 is that maximising the threads usage benefits the parallel execution. The Strategy 1 increases the threads idle time by following a sequential-like topological order and pausing the entire scheduling until the explored node satisfies its dependencies. The strategies 2 and 3 overcome this problem by scheduling nodes on demand based on a deeper graph exploration, showing better execution results in Figures 14 and 15 where both of them are close to the expected computation time for the critical path of the graph in Figure 12, rather than the best execution results for Strategy 1 in Figure 13.

Another observation is that the parallel execution is constrained by the DAG topology which is the main reason of why a parallel execution of fully decoupled jobs as in Figure 11 experiments a bigger performance improvement. This lead to conclude that there are topologies (as the Directed line graph) where little or no performance improvement can be experimented regardless the available CPU threads and the quality of the Scheduling heuristic. Of course when the parallelism capabilities are exploited over each node execution, the overall performance can be improved.

Figure 16 shows that the Strategy 3 has the biggest performance improvement but is still very close to the Strategy 2. This is understood in the way that Strategies 2 and 3 nearly reach the execution time of the critical path. The difference which makes the strategy 3 better is the reduction of scheduling overhead by only exploring the graph following the children of the successfully executed nodes. The exploration proposed by the strategy 2 might present a big overhead as the number of DAG nodes increases.

5.2. Execution on the Angulated Overlay Pipeline

Figures 17 and 18 show that the pipeline execution benefits on the availability of concurrent resources, reducing its execution time in less than a half which is the case for the scheduler including the scheduling heuristic running over 6 threads. There is not a significant improvement right after the execution over 2 threads,

which can be associated with limitations related with the pipeline's DAG topology.

Although when looking at the execution time the benefit of the scheduling heuristic is not clear, the Figure 19 comparing the performance improvement of both flavours, favours the heuristic showing a higher improvement when the pipeline runs over 2, 3, 5 and 6 threads, over the scheduling strategy without the heuristic. We can conclude from this testing step that the reduction of the scheduling overhead with smart scheduling strategies towards the maximum exploitation of the parallelism capabilities of a pipeline, can still reduce its execution time.

5.3. CUDA Memory micro-becnhmark

The results in the Table 3 shows that in all the cases the preallocation benefits the creation of a GpuMat, and exceeds in performance the Mat creation in longer inputs. The upload operation gets some benefit also by removing the overhead generated when allocating space for copying the data from Host to Device. We can conclude that maximising the memory re-usage and minimising the number of new allocations leads to create efficient CUDA based implementations.

5.4. CUDA implementations

All the GPU implementations of the listed algorithms in the Table 4 are more efficient and faster in each scenario, but the individual tests allowed us to find that there are cases where the GPU implementations might not have better execution times.

The GPU overhead generating during the execution of the first kernel and the initial allocation of the threads to perform the parallel execution, reduces the algorithm's efficiency for little inputs. That can be evidenced in Figures 21, 22 and 23 where the CPU implementation outperforms their CUDA counterparts when the input square matrix has less than 200 rows. This shows that no always choosing a CUDA implementation is efficient.

The case in Figure 20 is exceptional since in our experimentation the CUDA implementation shows a better performance in all the inputs. The main reason behind this is the inefficient implementation of the CPU version of the Calibrated Coordinates algorithm. As this is described in the Section 3.4.2, the coordinates after rotated are transformed. In the presented CUDA implementation, the matrix is explored only once and the rotation and transformation are performed pixelwise at the same iteration, while the CPU implementation perform both operations separately through different OpenCV functions, traversing the matrix multiple times. The CUDA implementation of this algorithm runs $32,29x$ faster than its CPU version with matrices of 3200×3200.

The overall conclusion of this test set is that the CUDA implementations with the usage of pre-allocated memory have shown an outstanding performance improvement over its CPU counterparts, demonstrating than the multi-threading capabilities of the GPUs along with a good problem granulation for a good kernel definition are clue for efficient CUDA approached implementations.

6. Conclusions

In this project, we proposed the a modular architecture of a job scheduler that decouples the job generation from its execution.

The highest performance improvement in the Angulated Overlay pipeline was $2.03x$, reducing in more than a half the computation time over 6 threads. This result was reached combining an scheduling strategy that reduces the pipeline's nodes exploration, with an scheduling heuristic that groups the execution of consecutive nodes into single jobs.

Furthermore, looking for a GPU extension of the existing pipelines, a set of algorithms were re-implemented using the CUDA framework, reaching outstanding results as it was for the Rotated Coordinates algorithm, with an improvement factor of $18.45x$. The results of this first approximation is an incentive for developing better performing pipelines exploiting the hardware heterogeneity, looking for real-time level computations.

This project open three different investigation branches for further studies: First, towards an efficient usage of the CPU resources exploring modern features e.g. branch prediction, usage of cached memory, etc. Second, towards improving the heuristic for clustering jobs in a DAG reducing the scheduling overhead. Third, towards the implementation of the existing algorithms, looking for efficient CPU and CUDA based implementations, reducing the execution time of every node.

7. Acknowledgments

I would like to thank my supervisor Koen Brants for his friendly help during my stay in Belgium and to following up this project elaboration. To all the MAIA team, the academics for the knowledge acquired during this two years, and to the coordination team for their diligence and unconditioned help in the legalities and those bureaucratic matters that no one likes. And off course a mention to the new friends I got and the people I met during this master. Finally but not least ... a mi familia por su consejo y soporte desde la distancia.

An special ungrateful mention to the SARS CoV2 for messing this year up ... better days are yet to come.

"Success is not final, failure is not fatal: It is the courage to continue that counts." - Winston Churchill.

References

Arpaci-Dusseau, R.H., Arpaci-Dusseau, A.C., 2018. Operating Systems: Three Easy Pieces. CreateSpace Independent Publishing Platform, North Charleston, SC, USA.

Banalagay, R., Covington, K., Wilkes, D., Landman, B., 2014. Resource estimation in high performance medical image computing. Neuroinformatics 12. doi:10.1007/s12021-014-9234-5.

Basagni, S., Bruschi, D., Ravasio, F., 1997. On the difficulty of finding walks of length k. RAIRO - Theoretical Informatics and Applications - Informatique Theorique et Applications 31, 429–435.

Black, P.E., 2004. Directed acyclic graph. URL: https://xlinux.nist.gov/dads/HTML/directAcycGraph.html. online; accessed 20 April 2020.

Contreras, G., Martonosi, M., 2008. Characterizing and improving the performance of intel threading building blocks, in: 2008 IEEE International Symposium on Workload Characterization, pp. 57–66.

Cook, S., 2013. Chapter 2 - understanding parallelism with gpus, in: Cook, S. (Ed.), CUDA Programming. Morgan Kaufmann, Boston. Applications of GPU Computing Series, pp. 21 – 36. doi:https://doi.org/10.1016/B978-0-12-415933-4.00002-8.

Cormen, T., Leiserson, C., Rivest, R., Stein, C., 2001. Introduction to Algorithms, Second Edition.

Culler, D., Singh, J.P., Gupta, A., 1998. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. chapter 1. pp. 15–16.

Dagum, L., Menon, R., 1998. Openmp: an industry standard api for shared-memory programming. IEEE Computational Science and Engineering 5, 46–55.

Dashti, M., Fedorova, A., 2017. Analyzing memory management methods on integrated cpu-gpu systems. SIGPLAN Not. 52, 59–69. URL: https://doi.org/10.1145/3156685.3092256, doi:10.1145/3156685.3092256.

Dijkstra, E.W., 1965. Solution of a problem in concurrent programming control. Commun. ACM 8, 569. URL: https://doi.org/10.1145/365559.365617, doi:10.1145/365559.365617.

Dijkstra, E.W., 1972. Information streams sharing a finite buffer. Inf. Process. Lett. 1, 179–180.

El-Rewini, H., Lewis, T.G., 1990. Scheduling parallel program tasks onto arbitrary target machines. J. Parallel Distributed Comput. 9, 138–153.

Flynn, M.J., 1972. Some computer organizations and their effectiveness. IEEE Transactions on Computers C-21, 948–960.

Garg, R.P., Sharapov, I.V., 2001. Techniques for optimizing applications: High performance computing, pp. 426–431.

Hagras, T., Janecek, J.J., 2003. Static vs. dynamic list-scheduling performance comparison, in: Acta Polytechnica - Journal of Advanced Engineerng, pp. 16–21.

Huang, K.C., Gu, D.S., Liu, H.C., Chang, H.Y., 2017. Task clustering heuristics for efficient execution time reduction in workflow scheduling. Journal of Computers 28, 43–56. doi:10.3966/199115592017022801004.

Huang, T.W., Lin, D.L., Lin, Y., Lin, C.X., 2020. Cpp-taskflow v2: A general-purpose parallel and heterogeneous task programming system at scale. arXiv:2004.10908.

Intel, 2006. Intel threading building blocks. URL: https://www.threadingbuildingblocks.com/. online; accessed 11 July 2020.

ISO/IEC_14882:2011, 2011. Programming Languages - C++. Standard. International Organization for Standardization. Geneva, CH.

Kwok, Y.K., Ahmad, I., 1999. Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Comput. Surv. 31, 406{471. URL: https://doi.org/10.1145/344588.344618, doi:10.1145/344588.344618.

Lamport, L., 2015. Turing lecturethe computer science of concurrency: The early years. Commun. ACM 58, 71{76. URL: https://doi.org/10.1145/2771951, doi:10.1145/2771951.

NVIDIA, 2007. About cuda - nvidia developer. URL: https://developer.nvidia.com/about-cuda. online; accessed 25 April 2020.

Raynal, M., 2015. Parallel computing vs. distributed computing: A great confusion? (position paper), in: Hunold, S., Costan, A., Giménez, D., Iosup, A., Ricci, L., Gómez Requena, M.E., Scarano, V., Varbanescu, A.L., Scott, S.L., Lankes, S., Weidendorfer, J., Alexander, M. (Eds.), Euro-Par 2015: Parallel Processing Workshops, Springer International Publishing, Cham. pp. 41--53.

Rho, J., Azumi, T., Nakagawa, M., Sato, K., Nishio, N., 2017. Scheduling parallel and distributed processing for automotive data stream management system. Journal of Parallel and Distributed Computing 109. doi:10.1016/j.jpdc.2017.06.012.

Topcuoglu, H., Hariri, S., Min-You Wu, 2002. Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transactions on Parallel and Distributed Systems 13, 260--274.

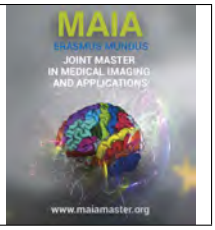
Vetter, J.S., Brightwell, R., Gokhale, M., McCormick, P., Ross, R., Shalf, J., Antypas, K., Donofrio, D., Humble, T., Schuman, C., Van Essen, B., Yoo, S., Aiken, A., Bernholdt, D., Byna, S., Cameron, K., Cappello, F., Chapman, B., Chien, A., Hall, M., Hartman-Baker, R., Lan, Z., Lang, M., Leidel, J., Li, S., Lucas, R., Mellor-Crummey, J., Peltz Jr., P., Peterka, T., Strout, M., Wilke, J., 2018. Extreme heterogeneity 2018 - productive computational science in the era of extreme heterogeneity: Report for doe ascr workshop on extreme heterogeneity. USDOE Office of Science (SC) doi:10.2172/1473756.

Wu, M.., Gajski, D.D., 1990. Hypertool: a programming aid for message-passing systems. IEEE Transactions on Parallel and Distributed Systems 1, 330--343.

Zhang, T., Landaverde, R., Coskun, A., Herbordt, M., 2014. An investigation of unified memory access performance in cuda. 2014 IEEE High Performance Extreme Computing Conference, HPEC 2014 2014. doi:10.1109/HPEC.2014.7040988.

Medical Imaging and Applications

Master Thesis, August 2020



Brain Image Analysis using Spatially Localized Neural Networks

Ahmed Gouda, Corné Hoogendoorn

Canon Medical Research Europe Ltd., Edinburgh, United Kingdom

Abstract

Parcellation of brain MRI is a powerful tools for characterization of normal and pathological tissues. Recently, three-dimensional Deep Convolution Neural Network (CNN) have quickly turned into the state-of-the-art in a variety of brain image parcellation applications. However, it brings with it various challenges. Specifically, these are posed by GPU memory limitations for high resolution volumes, the complexity of the segmentation task and low numbers of manually annotated training data. These challenges have been addressed in this thesis through two main brain MRI segmentation approaches. The first approach proposes an alternative strategy for Spatially Localized Atlas Network Tiles (SLANT). It simplified the brain segmentation task into registered localized sub-volumes, leveraging the brains symmetry and voxel spatial locations. The scarcity of annotated data is addressed through another semi-supervised proposed approach by combining a feature matching GAN model with SLANT. The second approach is an implementation for unsupervised deep learning approach combining atlas base segmentation and deep learning-based registration, without the need for any annotated volumes during training. The proposed models are trained on a collection of datasets from various sources to classify 31 anatomical structures. The SLANT approach using UNet model achieves the best segmentation results, reaching percentage Dice similarity scores ranging from 79.4% to 96.6% on selected parcels of interest for neurodegenerative diseases.

Keywords:
Brain Segmentation, Network Tiles, Deep CNN, GAN, Atlas Segmentation

1. Introduction

Automated imaging segmentation has opened new horizons in brain imaging applications, as it is an essential stage for measuring and visualizing anatomical structures of tissue-volumes derived from Magnetic Resonance Images (MRI). MRI quantitative and qualitative analysis has been used extensively for analysis of brain disorders, which helped clinical specialists to diagnose, monitor progression and therapeutic response for various neurodegenerative and neurodevelopmental disorders, tumors and psychiatric disorders. Moreover, segmentation is used extensively in intervention planning and guidance.

Delivering critical information about the shapes and volumes of brain structures is a very challenging segmentation tasks. Manual delineation for Brain lobes provides very precise brain parcels but it is time-consuming, complex and a lack of reproducibility process. Enormous progression in brain MR imaging has contributed to generating high quality MR volumes that makes manual delineation not feasible for clinical use. Consequently, many computerized-based segmentation algorithms, both semi-automated and fully automated, have been proposed to facilitate the delineation process.

Semi-automated segmentation requires medical specialist intervention to guide initialization and/or interaction. In interactive methods, the generated labels after non-precise computerized segmentation are corrected manually. By contrast, manually initialized methods require manually initialized seed points or contour that roughly represents the boundary of a target brain structure. Manually initialized methods can be divided into two primary sub-categories which are region-based and boundary-based. In region-based approaches, each voxel is assigned to membership according to homogeneity of the adjacent voxels, as in region growing and merging algorithm (Zhu and Yuille,

1996). Boundary-based approaches attempt to deform the initialized boundaries seeds around the objects by minimizing the energy function that measures the variation in gradient features near to the boundary, such as snake and balloon algorithms (Kass et al., 1988), (Staib and Duncan, 1992), (McInemey and Terzopoulos, 1999). Although semi-automated segmentation approaches facilitates the delineation process, they have been deployed in small-scale medical applications.

Fully-automated segmentation is the preferred technique in medical imaging fields, as it does not require human intervention through the segmentation stages, and it is therefore easy to be deployed in clinical application. Classical fully-automated unsupervised clustering algorithms such Fuzzy c-mean (Zhang and Chen, 2004), k-mean (Dhanachandra et al., 2015) and Expectation Maximization (EM) (Zhang et al., 2001) have been widely used for MRI brain segmentation. These algorithms are effective to classify a group of tissues with similar pixels without accounting the spatial location. Hence, they have been used to segment the main brain tissue classes with significant intensity differences, which are White Matter (WM), Gray Matter (GM), and CerebroSpinal Fluid (CSF).

Supervised segmentation approaches are applied to segment some specific brain anatomical structures, steered by a model of the shape and/or appearance of these structures like Active Shape Model-based approaches (Van Ginneken et al., 2002), and level sets segmentation based approaches (Baillard et al., 2001) (Wang et al., 2014).

Image artifacts such as bias field and partial volume effects present important challenges for fully-automated segmentation due to the variety of anatomical brain structures that may share the same tissue contrast. Therefore, probabilistic atlas-based (Aljabar et al., 2009) segmentation algorithms are widely used as they exploit prior anatomical information to make the segmentation task more robust. In this approach, previously delineated labels for reference MRI images (atlases) are manipulated as a prior knowledge to segment target image. The image segmentation problem is cast as a registration problem by registering the reference atlas images to the domain of the target image. The relevant atlas labels are then propagated onto the target image. Single-atlas (Guimond et al., 2000) (Wu et al., 2007) is the basic framework in atlas-based segmentation approaches, as it uses single atlas image. However, its performance degrades in high anatomy variation between the atlas image and the target image. Therefore, multi-atlas (Rohlfing et al., 2004) (Heckemann et al., 2006) segmentation approaches address this problem by registering multiple atlases images, while the conflict between propagated label are harmonized using multi-atlas label fusion techniques (Warfield et al., 2004) (Heckemann et al., 2006) (Wang et al., 2012) (Asman and Landman, 2013) (Iglesias and Sabuncu, 2015). The different atlas based segmentation approaches have been regarded as robust brain volume segmentation standards, and they are still used in many recent medical parcellation tools (Mikhael and Pernet, 2019).

Recently, deep Convolutional Neural Networks (CNN) approaches have been deployed in large-scale for computer-aided medical imaging systems. Recent advances in semantic segmentation using three-dimensional kernels have enabled to segment three-dimensional brain structure. Though semantic segmentation achieves state-of-the-art volume segmentation results. It includes specific challenges that need to be addressed, such as the scarcity of labelled data, the high class imbalance found in the ground truth labels and the memory limitation problems for three-dimensional images. In this thesis works, we compare three recent MRI volume segmentation approaches to analyze brain structures based on supervised, semi-supervised and unsupervised learning methods. These approaches employ CNN segmentation and registration models to leveraging some advantages of regionalized network specialization while mitigating the memory limitations for high resolution images.

2. State of the art

In the last decade, deep CNN have outperformed other machine learning approaches in many visual recognition tasks (Bengio et al., 2013). Although convolutional networks have already existed for a long time (Lawrence et al., 1997) (LeCun, 1998), their success was restricted due to implementation scale of the networks size which is limited by he lack of computational power, and the complexity of the problem that could be addressed. The increased availability and power of GPU technology allowed the applicability of deep CNNs for large scale medical imaging applications.

The state-of-the-art CNN models for supervised image segmentation are variants of conventional encoder-decoder architecture like UNet (Ronneberger et al., 2015) (Çiçek et al., 2016) and V-Net (Milletari et al., 2016). In the encoder part, the input image is down-sampled into the latent space using strided convolution and max pooling layers. In the decoder part, the compressed image is up-sampled and concatenated to the same level encoding layers via skip connections, in order to make the decoder output follow the spatial structure of the input. These skip connections constrain the reconstruction process at the same-scale feature maps of the encoder and decoder layers.

An alternative semi-supervised learning approach utilizes Generative Adversarial Networks (GANs) to use a very few annotated training examples (Mondal et al., 2018). The segmentation model is trained with labeled and unlabeled scans by extracting few-shot patches from these volumes. The adversarial networks consists of a generator and discriminator. The generator network

tries to produce realistic fake patches, while the discriminator tries to distinguish the generated fake patches from the true patches.

Another recent unsupervised volume segmentation approach combines a conventional Bayesian probabilistic atlas-based segmentation with deep learning (Dalca et al., 2019). This approach comprises a deformable medical image registration framework using VoxelMorph (Balakrishnan et al., 2018), a probabilistic atlas and the global statistics of image intensity classes to efficiently estimate the deformation field and the scan-specific likelihood intensity parameters for the input image. One of the major advantage of this approach that the segmentation model does not require any ground truth data during training.

The basic technique to apply a full volume brain segmentation is to fit the complete MRI volume to a 3D CNN. Despite the decent results of this technique, it faces a GPU memory limitation for high resolution 3D brain volumes. Therefore, other approaches have been proposed to solve this problem by performing 2D slice segmentation of the 3D volume (Dong et al., 2017). The predicted output for each slice separately is then combined into a volume. This approach may have limited accuracy because it does not consider the inter-slice information between the neighboring slices. Hence, different 2.5D[1] segmentation approaches (Roth et al., 2014) (Angermann and Haltmeier, 2019) have been proposed to address the missing information in 2D segmentation and memory limitations of 3D segmentation. Although the 2.5D approaches can provide good segmentation results for some medical case studies, it is not standard technique to express volume images.

Patch-based segmentation is one of proposed solutions to deal with memory and computational requirements. In this approach, the brain region of interest is divided into similar-sized 3D overlapped sub-regions (patches). All the generated patches are then used to train a 3D CNN model. The testing volumes are predicted through non-overlapped 3D patches that cover the whole brain region. This approach also addresses the problem of labeled database limitation through few-shot learning (Fei-Fei et al., 2006). However it is not robust to segment a high number of anatomical parcels for full volume brain structure, since each patch will cover only a subset of brain regions, and this can seriously exacerbate the class imbalance problem.

Spatially Localized Atlas Network Tiles (SLANT) (Huo et al., 2018) (Huo et al., 2019) approach addresses the problem of limited GPU memory and simplifies the complex problem of high number of anatomical labels segmentation into simpler problems, better suited to limited training data. In this approach, the issue of arbitrary per-patch coverage of the brain regions is addressed by registering the training and testing volumes into a standard template. Then, the whole volume is divided into overlapped fixed sub-volumes (tiles), each one being processed by a different UNet.

[1] 2.5D is a general term for methods that conceptually lie somewhere between 2D and 3D.

3. Material and methods

The systems pipelines for this work are based on two existing systems. The first pipeline system proposes an alternative strategy for SLANT approach. This approach comprises an implementation for two sub-approaches using UNet model as a supervised segmentation, and feature matching GAN model as semi-supervised segmentation. The second pipeline system builds on unsupervised volume segmentation approach, combining VoxelMorph registration network with Bayesian probabilistic atlas.

3.1. Dataset Description and Pre-processing

The input images for these pipeline systems are collected from various resources with different isotropic spatial resolution for single modality MRI T1 weighted brain scan. The dataset is composed of raw clinical MRI brain images from different scanners and MRI brain datasets from different challenges. Whole dataset is divided with fixed proportion into Training set and Testing set, as shown in Table 1.

Dataset Name	Training Set	Testing Set
BGM Atlas	64	19
ADNI MCI	34	10
MRICloud	33	10
MICCAI 2012 MR	25	8
ADNI AD	21	7
Volunteers (Canon)	19	6
IBSR MR	14	4
Total	210	64

Table 1: Number of training and testing T1w Brain MRI volumes per dataset (Wu et al., 2016) (de Vent et al., 2016) (Mori et al., 2016) (Landman and Warfield, 2012) (Frazier et al., 2007).

In order to unify all disparate scans, they are affinely registered using Elastix toolbox (Klein et al., 2009) to MNI ICBM 152 space (Lancaster et al., 2007). Then, N4ITK was applied to suppress the bias field noise (Tustison et al., 2010). Since the acquired scans from MRI devices are non-scaled, the scans intensities are not harmonized over different scanners, and even different scans across the same scanner. Therefore, each scan is normalized by truncating the intensities outside the percentile range 5% to 95%. This harmonization technique eliminate the outliers intensities, and stretches the major brain intensities information over the histogram. Furthermore, it is a relatively simple approach in comparison to existing approaches (Madabhushi and Udupa, 2006) (Schaap et al., 2009) (Simkó et al., 2019).

3.2. Ground Truth Generation

The ground truth labels have been generated using brain parcellation application (Murphy et al., 2014). It was developed based on traditional image analysis techniques and comprises five stages. In the first stage, the left-right volume direction is rotated towards the mid-sagittal plane. In the second stage, 99 atlas volumes are affinely registered and propagated to align the input volume. In the following stage, mutual information metric is computed be between the registered atlases and input volume in order to select the best aligned atlases to proceed to the fourth stage. Then, the selected atlases are non-rigidly registered to the input volume space, and the relevant labels are propagated using the affine and non-rigid deformation fields. In the last stage, the priory probability distribution are generated over the structure labels, and they are refined using the EM algorithm for final assignment.

This brain parcellation application generates 290 brain parcels including the background region. The parcellation protocol defines five levels of increasing refinement. The first level includes main divisions of the forebrain, midbrain and hindbrain, as well as CSF and the skull, while the fifth level include a detailed brain classes that was generated by the parcellation application, as shown in Table 2. Merging the left and right counterparts into a single label leverages the brain symmetry and can lead to greater memory savings.

Parcellation Level	Without L/R Merging	With L/R Merging
Level 1	9	8
Level 2	23	14
Level 3	57	31
Level 4	139	72
Level 5	290	149

Table 2: Number of ground truth parcels for each level, with and without left-side and the right-side labels merging.

3.3. Brain Atlas Generation

VoxelMorph based atlas segmentation approach requires generating the probabilistic atlas priors for all brain parcels. As illustrated in Figure 1, an affine registration followed by B-splines registration have been preformed using Elastix toolbox to propagate the training volumes to the MNI ICBM 152 space. Then, bias field noise was removed using N4ITK. Using the brain parcellation application, left-side and right-side combined labels were generated for L3 parcels. The prior probabilities of observing the propagated parcels for the training volumes are firstly computed. Afterwards, the left-side and right-side prior probabilities for the axial direction are mirrored and combined to generate symmetric probabilistic atlas.

3.4. Spatially Localized Atlas Network Tiles (SLANT)

The proposed SLANT pipeline system is shown in Figure 2. This pipeline is fed with the pre-processed volumes in the MNI space, and the corresponding generated ground truth.

3.4.1. Network Tiles

The brain region of interest of the MNI space is constrained within a bounding-box (160, 192, 160). The entire bounding box region is covered by $k_x \times k_y \times k_z$ equally spaced, equally sized 3D tiles, extending $d\%$ of the bounding box size along each side. The amount of overlap between tiles can be varied through combinations of k and d.

In the original SLANT system (Huo et al., 2018) (Huo et al., 2019), each tile subspace trained with specific CNN model using 3D UNet architecture. We propose exploiting the symmetry of the brain across the midsagittal plane. The right-side tiles are horizontally mirrored and augmented with the left-side tiles. However, medial common tiles are horizontally mirrored and augmented with themselves to keep number of training cases balanced as in the side tiles. This technique boosts the accuracy, and reduces the number of the trainable tiles.

3.4.2. UNet Network Model

Besides dedicating a UNet model to each individual tile, another technique is proposed in this research by training all the SLANT tiles using a single UNet model. The 3D coordinates for each voxel can be added as

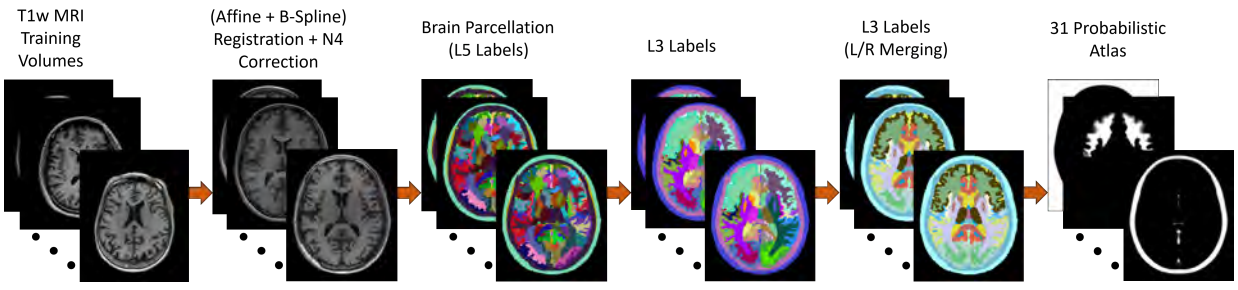


Figure 1: Probabilistic atlas generation for 31 brain parcels.

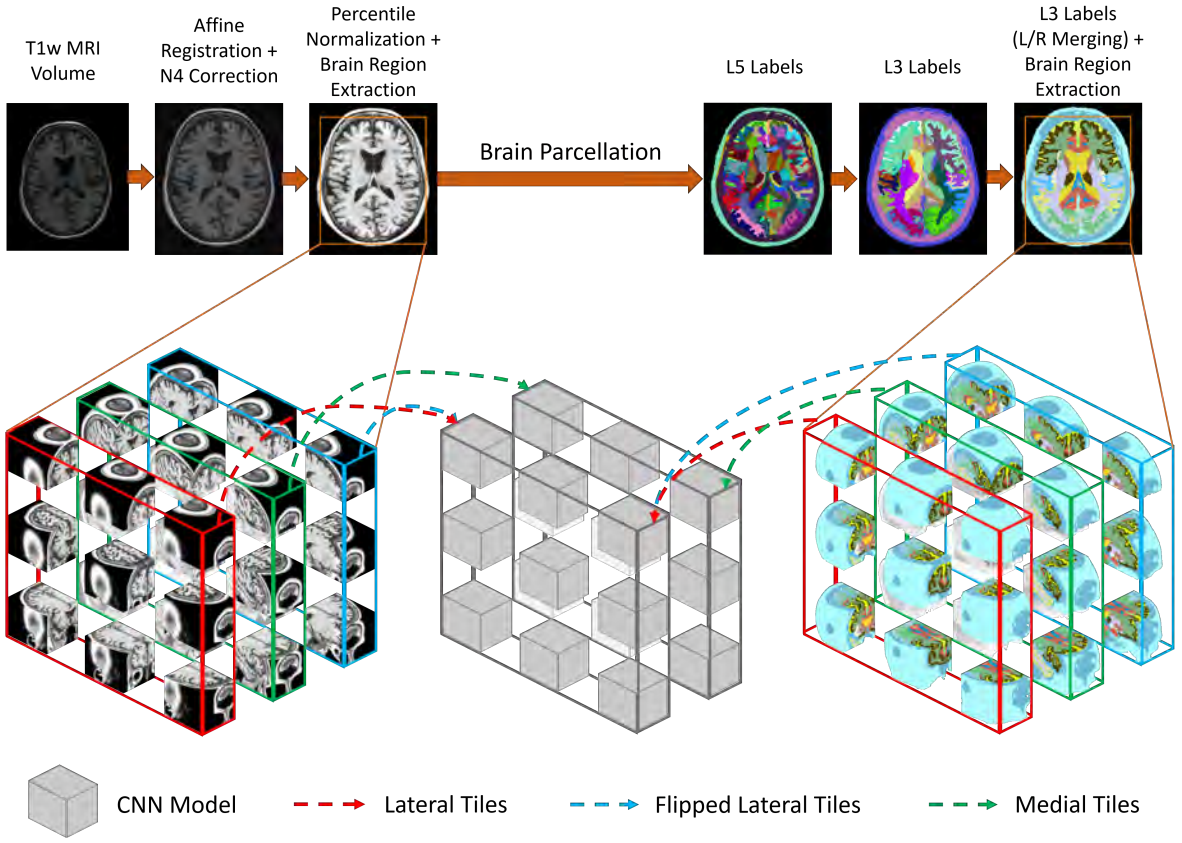


Figure 2: The proposed SLANT($3, \frac{2}{3}$) pipeline system includes canonical medical image pre-processing and ground truth generation. Each tile covers 66.67% of the bounding-box region that covers the brain volume.

spatial feature, as shown in Figure 3. In order to decrease the model complexity is this configuration, the spatial feature map size is downsampled by a factor of 2. The volume image is centred by normalizing the spatial features between -1 and 1. These 3D coordinates feature map is divided into subspace relevant to each tile, and concatenated with the second level of the UNet model. This proposed architecture exploit the advantages of few-shot learning SLANT approach. In addition, it decreases the number of training models. The UNet network model in the original SLANT paper uses voxel-wise Dice loss function between predicted parcels A_i and the ground truth parcels B_i, ignoring the background. Using M as the total number of parcels, the DSC and its derived loss function are defined as

$$DSC_i = \frac{2\,|A_i \bigcap B_i|}{|A_i| + |B_i|} \quad i = 1, ., ., M \tag{1}$$

and

$$L_{dice} = 1 - \frac{\sum_{i=1}^{M} DSC_i}{M} \tag{2}$$

3.4.3. *Feature Matching GAN Network Model*

This model was built based on existing work (Mondal et al., 2018) as illustrated in Figure 4. It proposes a novel combination between SLANT system architecture

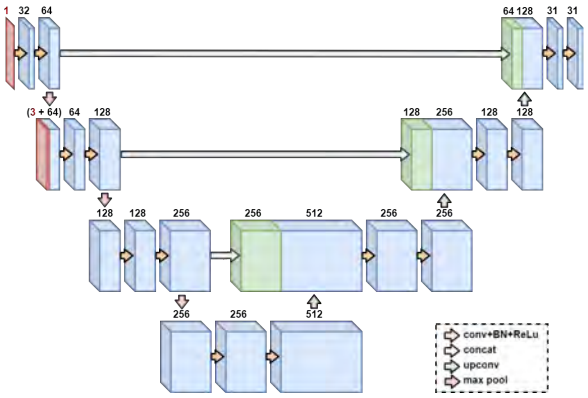


Figure 3: 3D UNet architecture with 4 inputs represented by red boxes. The input volume image in the first level and the three coordinates in the second level. Green boxes represent copied feature maps from the encoder and concatenated to the decoder at the same level.

and semi-supervised adversarial deep learning. To implement this model, the training set is split into labeled and unlabeled volumes, and each volume is divided into SLANT tiles. The labeled, unlabeled and the generated fake tiles are included during the training process.

The conventional GANs algorithmic architectures uses two CNNs, pitting one against the other. The gen-

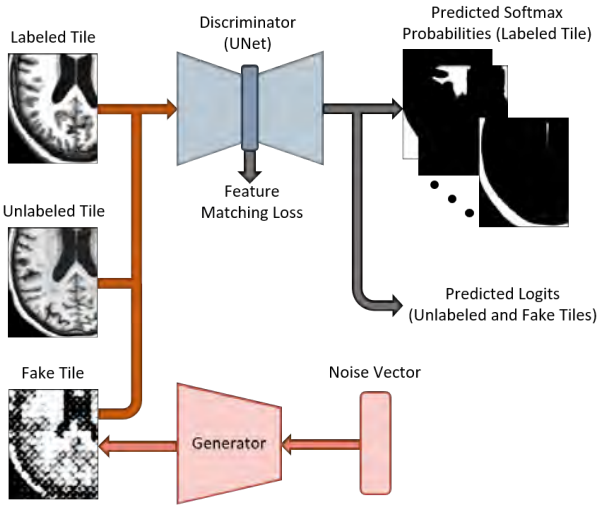


Figure 4: Feature matching GAN Model is an adversarial setup consisting of Generator, and Discriminator in UNet architecture. Both networks are trained simultaneously.

erator G_{θ_G} is trained to map a randomly generated noise vector $z \in \mathbb{R}^d$ with uniform distribution into a synthetic image vector $\tilde{x} = G(z)$. Meanwhile, the discriminator D_{θ_D} is trained to differentiate between real tiles $x \sim p_{data(x)}$ and synthesized tiles $\tilde{x} \sim p_{G(z)}$. Both of the generator and discriminator networks are two players in a min-max optimization game, as shown in the following function $V(D_{\theta_D}, G_{\theta_G})$.

$$\min_{G_{\theta_G}} \max_{D_{\theta_D}} \mathbb{E}_{x\sim p_{data(x)}}[logD_{\theta_D}] + \mathbb{E}_{z\sim noise}[1 - D_{\theta_D}(G_{\theta_G}(z))] \tag{3}$$

The labeled tiles in sub-volume space $x_{H\times W\times D}$ are trained using standard 3D UNet segmentation model to the output space $y_{H\times W\times D}$ with M logit classes $[l_{i,1}, ..., l_{i,M}]$. Using the Softmax function, the output can be represented by class probabilities.

$$p_{model}(y_i = j|x) = \frac{exp(l_{i,j})}{\sum_{m=1}^{M} exp(l_{i,m})} \tag{4}$$

A voxel-wise Dice loss function $L_{labeled}$ are computed between the predicted segmentation probabilities $p_{model}(y_i = j|x)$ using Softmax function and the ground truth of the labeled tile. Since the generator model G predict the realistic synthesized tile, the discriminator D requires an additional class to distinguish if the generated fake tile is true. This additional class can be recast back within the M classes by maximizing the following equation.

$$\mathbb{E}_{x\sim p_{data(x)}} \sum_{i=1}^{H\times W\times D} log\ p_{modal}(y_i \in 1, ..., M|x) \tag{5}$$

From the last equation, the unlabeled and fake tiles can be also trained using the same UNet model. Using the normalized logits strategy in (Salimans et al., 2016), the loss functions for the fake and unlabeled tiles $L_{unlabeled}$ and L_{fake} can be directly calculated by employing the normalized logits in the Softmax function of Equation (4), where $Z_i(x) = \sum_{m=1}^{M} exp[l_{i,k}(x)]$, as shown in the following equations.

$$L_{unlabeled} = -\mathbb{E}_{x\sim p_{data(x)}} \sum_{i=1}^{H\times W\times D} log\left[\frac{Z_i(x)}{Z_i(x)+1}\right] \tag{6}$$

$$L_{fake} = -\mathbb{E}_{x\sim p_{data(x)}} \sum_{i=1}^{H\times W\times D} log\left[\frac{1}{Z_i(G_{\theta_G}(z))+1}\right] \tag{7}$$

The yield Dice loss functions from labeled tiles $L_{labeled}$ are weighted by parameter α to stimulate the UNet segmentation predictions, and it is combined with obtained loss output from fake and unlabeled tiles in a discriminator loss function.

$$L_{discriminator} = \alpha L_{labeled} + L_{unlabeled} + L_{fake} \tag{8}$$

The generator uses a feature matching (FM) strategy for calculating the loss, which aims to match the expected values of features $f(x)$ in an intermediate layer of the discriminator. $f(x)$ is the output from the second last of the UNet encoder models, as it provides a higher performance than using the last layer (Mondal et al., 2018).

$$L_{generator} = \left\|\mathbb{E}_{x\sim p_{data(x)}}(x)f(x) - \mathbb{E}_{z\sim noise}f(G_{\theta_G}(z))\right\|_2^2 \tag{9}$$

Following the FM GAN authors' steps, the 3D UNet architecture is modified to adapt the GAN framework in order to make the training more stable. The weight normalization is used instead of the batch normalization.Alseo, Leaky ReLUs is used for activation functions because it is robust to small negative outputs (logits), as they will still provide a gradient whereas standard ReLU would not. Since sparse gradients are induced by max pooling which are not good for GANs, the authors replaced them with average pooling.

3.4.4. Predicted Labels Reconstruction from Network Tiles

During the reconstruction process for the predicted labels, the common voxels within the overlapped regions would be segmented more than once from multiple models. Since the predicted probabilities from the softmax activation function of the model output layer represents the membership of each voxel towards all classes, the robust trained models may provide high probability variance between the predicted classes. Therefore, in this work the different predicted probabilities for the overlapped voxels are summed together.

This technique provides a soft decision for the maximum argument classification of each class. The reconstructed function for the 3D image in the MNI space S_{MNI} at voxel point (i_x, i_y, i_z) is shown in the following equation, where S_t is the sub-space tile at voxel point (j_x, j_y, j_z), and T the total number of tiles.

$$S_{MNI}(i_x, i_y, i_z) = \underset{l\in\{0,1,\ldots,M-1\}}{argmax} \sum_{t=1}^{T} p\left(l \mid S_t(j_x, j_y, j_z)\right) \tag{10}$$

The predicted probabilities addition technique is better than the majority voting method which is not sensitive to the predicted probability outputs form the more robust trained models. In addition, it is less complex because it does not require an additional classification stage for the overlapped regions.

3.5. VoxelMorph Based Atlas Segmentation (VMBAS)

VoxelMorph Based Atlas Segmentation approach is carried out utilizing Bayes' rule for segmentation, and merging it with an unsupervised deep learning-based registration framework. The network model architecture $g_{\theta_C}(I, A) = (\theta_S, \theta_I) = (v, \mu, \sigma^2)$ of this system was designed using the 3D UNet based architecture for VoxelMorph (Balakrishnan et al., 2018). As demonstrated in Figure 5, the model reads two inputs: an MRI volume I and the probabilistic atlas A. This UNet includes 32 convolutional filters in both the encoder and decoder stages using a kernel size of 3, stride of 2, and LeakyReLu activation functions. At the end point of the UNet, a pair of convolutional layers are attached. The first convolutional layer to output stationary velocity field v. This layer is followed by scaling and squaring (Arsigny, 2006) (Dalca et al., 2018) (Krebs et al., 2019) integration layer to calculate the deformation field $\phi = exp(v)$, which yield the diffeomorphic flow loss parameter. Then, the probabilistic atlas A is warped using an additional spatial transform layer. The second convolutional layer to output the Gaussian intensity parameters μ, σ^2, which is combined with input MRI volume I to provide the likelihood maps. These maps with warped atlas enable computation of the data loss term.

From the generated probabilistic atlas, the prior probability A for each ground truth label l at the spatial voxel location $x_j \in \Omega$ is given by $A(l, x)$. The probabilistic atlas map is deformed by the diffeomorphic transform function ϕ_v, and by using the stationary velocity field parameter v which parametrizes the prior $\theta_s = v$. The S_j represents the segmentation at each voxel j, as shown in the following equation.

$$p(S|\theta_s; A) = p(S|v; A) = \prod_{j\in\Omega} A\left(S_j, \phi_v(x_j)\right) \tag{11}$$

The spatial location of the voxels j in image I are displaced by deformation field ϕ_v by the spatial gradient ∇u_v, where $\phi_v = Id + u_v$. The deformation term in the loss equation is weighted by the parameter λ.

$$p(\theta_S; \lambda) = p(v; \lambda) \propto exp\left[-\lambda \|\nabla u_v\|^2\right] \tag{12}$$

The likelihood parameters $\theta_I = \{\mu, \sigma^2\}$ are represented by Gaussian distribution function $\mathcal{N}(.; \mu_{S_j}, \sigma^2_{S_j})$ for the voxel intensity I at location j, where μ and σ^2 are the means and variances of voxels intensities under each class.

$$p(I|S, \theta_I) = p(I|S, \mu, \sigma^2) = \prod_{j\in\Omega} \mathcal{N}(I_j; \mu_{S_j}, \sigma^2_{S_j}) \tag{13}$$

Since the number of training volumes are limited, data was augmented to N volumes by horizontal flipping

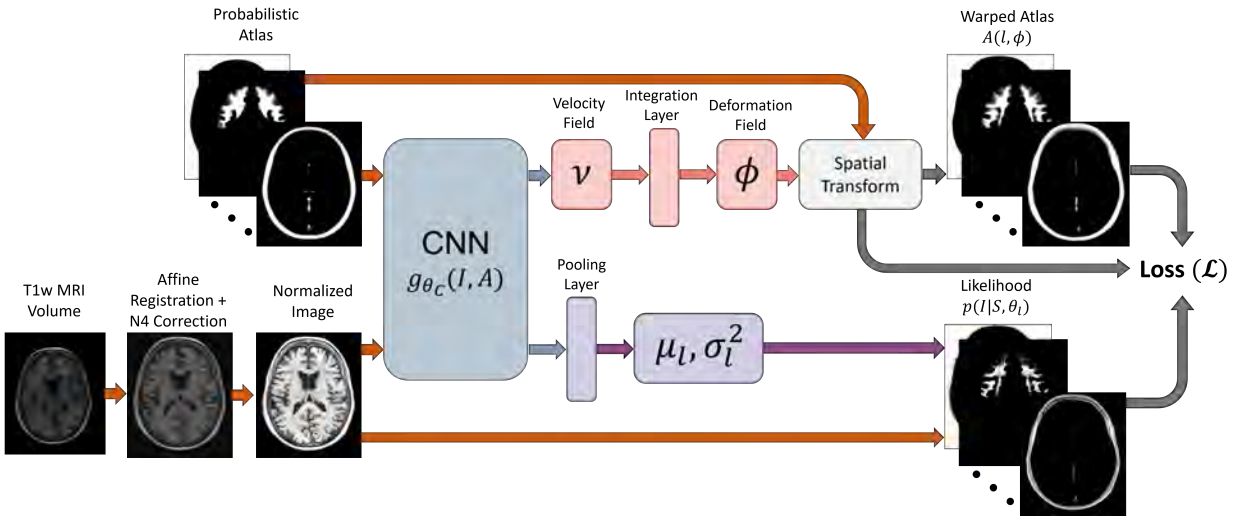


Figure 5: VoxelMorph Based Atlas segmentation pipeline. The network model $g_{\theta_{C(.,.)}}$ provides the deformation velocity field v which propagates the atlas to the input image space. and the intensity likelihood parameters μ, σ^2 that resample the likelihood map per each parcel.

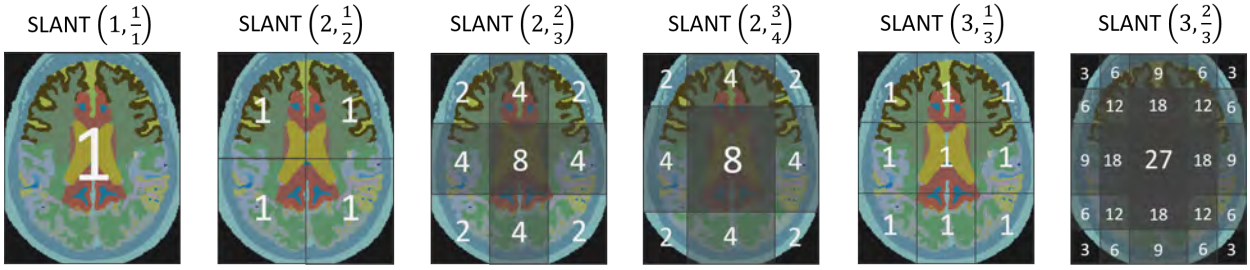


Figure 6: Different experimental setup for the SLANT approach. It show the number tiles of overlays in the axial cross sectional center.

of the images. The total loss function can be expressed as shown in the following equation. The log-partition function $K(\lambda)$ is controlled by the hyper-parameter λ, keeps the probability distribution with a proper values without affecting the optimization.

$$\begin{aligned}\mathcal{L}(A,I) &= -\sum_{n=1}^{N} log\ p(v^n,\mu^n,[\sigma^2]^n I^n;A,\lambda)\\ &= -\sum_{n=1}^{N}\sum_{j\in\Omega} log\left[\sum_{l=1}^{M}\mathcal{N}(I_j^n;,\mu_l^n,[\sigma_l^2]^n)A\left(l,\phi_{v_m}(x_j)\right)\right]\\ &\quad -K(\lambda)+const\end{aligned} \tag{14}$$

This approach does not require the ground truth during the training. Consequently, it is contrast adaptive to MRI volumes with unobserved contrast. Given a new testing volume and the probabilistic atlas, the trained model predicts the deformation field $\hat{v}_t$ and the intensity parameters $\hat{\theta}_t$. The optimal segmentation can be computed from the maximum argument of the wrapped atlas and the likelihood values according to the following equation.

$$\hat{S}_j = \underset{l}{argmax}\mathcal{N}(I_j;\hat{\mu}_t;\hat{\sigma}_t^2)A(l,\phi_{\hat{v}}(x_j)) \tag{15}$$

3.6. Implementation Details

Figure 6 shows the different experimental setup for SLANT approach using UNet model which addresses three different SLANT cases. The first case SLANT$(1,\frac{1}{1})$ uses the entire volume, which is equivalent to not applying SLANT. The second case is without overlapping tiles, as illustrated in SLANT$(2,\frac{1}{2})$ and SLANT$(3,\frac{1}{3})$. The last case is by using overlapping tiles with different sizes. Two additional setups were implemented in SLANT$(3,\frac{2}{3})$ configuration, using a single model with and without the three dimensional spatial features. In order to make fair analysis for these various settings, the same hyper-parameters tuning was set for all experiments using batch size = 1, optimizer = "Adam" (Kingma and Ba, 2015) and learning rate = 0.0001. In addition, all the network models are trained over 30 epochs in all experiments. Besides the initial normalization technique using percentile for the entire volume, another normalisation was applied for the extracted sub-volume tiles before training using mean and standard deviation.

The plot in Figure 7 illustrates the model loss per epoch in the case of SLANT$(3,\frac{2}{3})$ with UNet configuration. It shows lower training and validation Dice loss for the center tile 2_2_2 more than the corner tile 1_1_1. Since the loss function does not account the background portion loss which having a large portion of background, it may lead to higher loss values in absolute terms. However, it shows a higher training and validation accuracy for the same corner tile more than the center tile as shown in Figure 8.

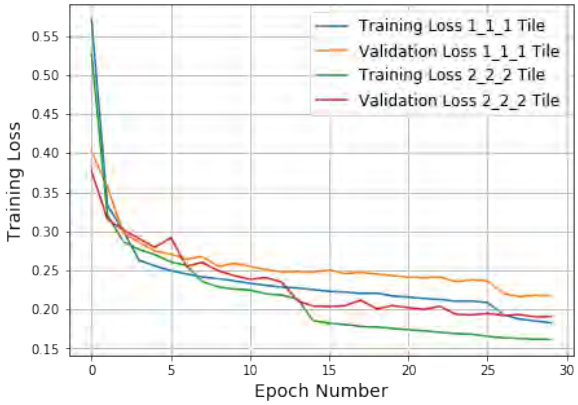


Figure 7: The training and validation Dice loss of the first fold cross validation for SLANT$(3,\frac{2}{3})$, and using UNet. The tiles 1_1_1 and 2_2_2 are located at the corner and the center respectively.

The hyper-parameters configuration for feature matching GAN uses the same patch size as in UNet, the "Adam" optimizer configures the generator and discriminator with learning rate 0.0001 and a momentum of 0.5. The labeled loss weight parameter α is set to 15. Two experimental configuration during the training stage based on the ratio of labeled and unlabeled volumes. The training set is divided into two equivalent portions of labeled and unlabeled volumes in the first configuration. The number of the training epochs in this case is set to 120 with 70 training volumes per epoch. In the second configuration, the training set is divided as quarter for labeled and three-quarter for unlabeled.

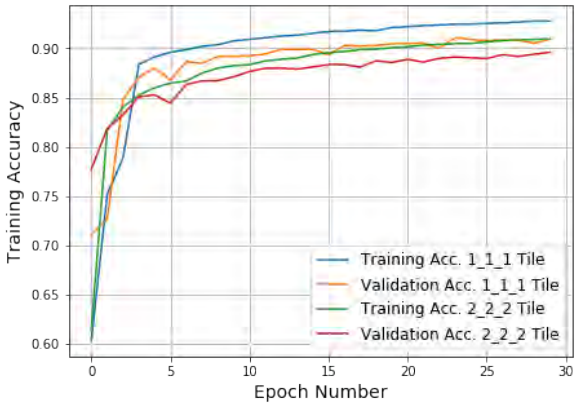


Figure 8: The training and validation accuracy of the first fold cross validation for SLANT($3,\frac{2}{3}$), and using UNet. The tiles 1_1_1 and 2_2_2 are located at the corner and the center respectively.

Since the number of training volume samples per epoch increased to 105 in the second configuration, the number of the training epochs is decreased to 80.

As shown in the loss plot in Figure 9, the Dice loss for labeled image is decreasing smoothly while the unlabeled and fake losses overall appear unstable. The center tile 2_2_2 shows lower labeled loss than the corner tile 1_1_1. However, Figure 10 shows higher accuracy for the corner tile 1_1_1 than the center tile 2_2_2.

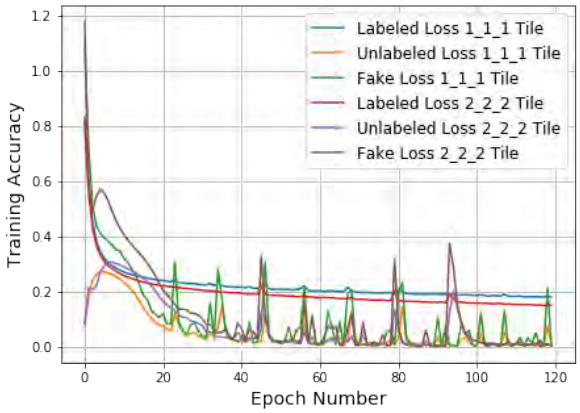


Figure 9: The training Dice losses of the first fold cross validation for SLANT($3,\frac{2}{3}$) using a half to half labeled and unlabeled data, and using FM GAN. The tiles 1_1_1 and 2_2_2 are located at the corner and the center respectively.

Atlas based VoxelMorph experiments are also configured using the same SLANT hyper-parameters. Meanwhile, the registration parameter λ is set to 10 which is set empirically. Figure 11 shows a decay in the data loss during training the model, while diffusion loss increasing. On the other hand, Figure 12 shows unstable and fast overfitting for the validation accuracy because the network model was trained using loss function to estimate the deformation field, without using spatial segmentation loss function between the predicted and the ground truth images.

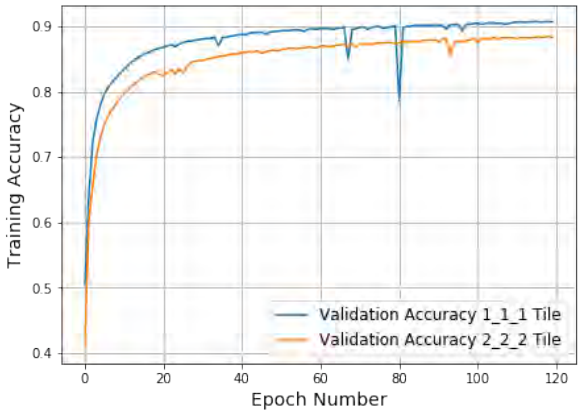


Figure 10: The validation accuracy of the first fold cross validation for SLANT($3,\frac{2}{3}$), and using FM GAN. The tiles 1_1_1 and 2_2_2 are located at the corner and the center respectively.

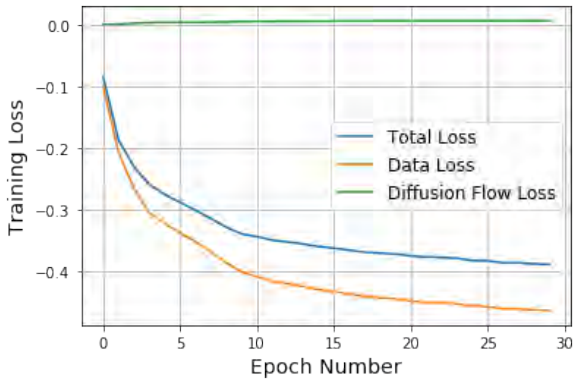


Figure 11: The training loss of the first fold cross validation for Atlas Voxelmorph. Total Loss = Data Loss + λ (Diffusion Flow Loss) , where $\lambda = 10$.

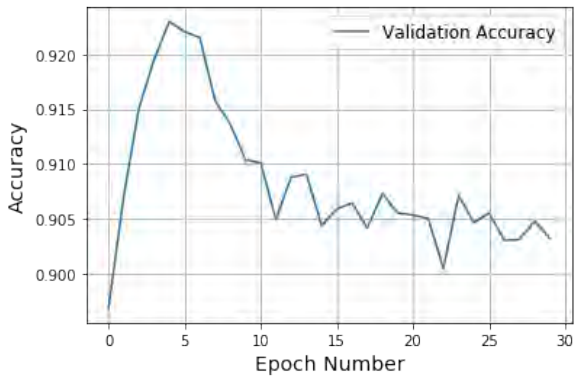


Figure 12: The validation accuracy of the first fold cross validation for VMBAS.

All the pre-processing and registration methods are kept the same for all SLANT and VMBAS experiments.

All training and testing was done on an Nvidia DGX-1[2] machine with Tesla V100 SXM2 GPUs 32GB memory.

3.7. Evaluation Criteria

In order to obtain robust evaluation for the predictive models results, the training set is shuffled in fixed seed, and then split into 3 equal portions for 3-fold cross-validation. The model which provides the greatest validation accuracy in each fold is selected for predicting the testing volumes. Then, the affine registration parameters from the pre-processing stage are inverted, and both of the ground truth and the predicted images are propagated from the MNI space to the original volume space. In the testing stage, post-processing techniques have been applied on the predicted images to sparse false predicted labels due to the noisy background. This technique is carried out in object-level by keeping the biggest connected component which represents the brain region, and discarding the tiny disconnected components.

All the experimental results have been evaluated using Dice Similarity Coefficient (DSC) for voxel-level metric functions according to Equation (1). Symmetric Hausdorff Distance (HD) is another applied metric function which is carried out by calculating the highest of all the distances from a point a in the predicted segmentation parcel A of specific parcel to the closest point b in the relevant ground truth parcel B. The Symmetric Hausdorff Distance is measured in the patient space.

$$HD = max(D(A, B), D(B, A)) \tag{16}$$

where

$$D(A, B) = max_{a \in A} min_{b \in B} \|a - b\| \tag{17}$$

4. Results

Quantitative and qualitative analysis have been performed to evaluate the segmentation results. The testing volumes are predicted three times from the best selected models in each cross validation fold. Using the metric functions, the predicted images are evaluated numerically over all parcels collectively and individually. Then, the three predicted results for each testing image are averaged. The box-plot diagram in Figure 13 provides high-level summaries of the performance of all the models. In order to obtain a precise analysis for the brain region, the background is not considered during computing the DSC, as it changes from image to another. Overall, SLANT approach outperforms VMBAS approach in DSC.

The segmentation performance increases by using overlapped tiles. It is affected by two factors which are the number of overlaid models and overlapped region size. In non-overlapped SLANT models, using a bigger tile size in SLANT($2,\frac{1}{2}$) model achieves higher performance than SLANT($3,\frac{1}{3}$). However, the segmentation performance degrades by fitting the brain volume model in one model as in SLANT($1,\frac{1}{1}$). In addition, it provides poor segmentation performance in many testing image cases. Increasing the number of overlaid models makes the overlapped regions to be predicted from multiple models at once, and it boosts the segmentation accuracy as in SLANT($3,\frac{2}{3}$). Meanwhile, increasing overlapped region size offers the possibility for bigger regions to be trained and predicted from different models.

The single SLANT model is trained on a higher number of tiles and it is less computationally and resource expensive in comparison to multi-model SLANT. Moreover, it achieves higher evaluation performance than SLANT($1,\frac{1}{1}$) because the model is trained on relatively few number of training examples, and it see the overlapping parts more often during training. Despite these benefits, the segmentation accuracy of a single SLANT model fails to attain the multi-model-SLANT. Adding the spatial location features produces a more robust segmentation performance but it still cannot exceed the multi-model SLANT.

The FM GAN model is a semi-supervised segmentation approach (contrasting with SLANT using UNet model, which is fully supervised), as it uses a partially labeled training set. Consequently, it provides lower segmentation performance in comparison to the UNet model using the same SLANT($3,\frac{2}{3}$) configuration.

In the case of using quarter to three-quarter labeled to unlabeled data, it gives a very slightly elevated results than half to half data. Hence, this method is robust to a reduction from 50% labeled to 25% labeled. Using three-quarter of unlabeled data makes the model trained on more various unlabeled examples in each epoch. Accordingly, it drives the model to better generalize to unseen instances.

Per-parcel analysis is reported for a set of parcels of specific interest for neurodegenerative disorders. These parcels are divided under six groups of Level 1, and they are listed in Tables 3 through 8. All experiments provide high Dice scores within bigger size parcels in general, while the implemented systems sometimes fail to detect small parcels. VMBAS approach shows the lowest DSC output in the major parcels. On the other hand, this approach and SLANT($3,\frac{2}{3}$) configuration provide relatively precise HD scores, in comparison to other SLANT configurations.

As illustrated in Table 3, the segmented parcels under Telencephalon group achieves the best DSC score in SLANT($3,\frac{2}{3}$) UNet configuration. However, Basal Ganglia parcel shows a slightly higher score with single model configuration, and it is the highest score among Telencephalon parcels. Conversely, Limbic White Matter shows the lowest DSC score. In Table 4,

[2]https://www.nvidia.com/en-gb/data-center/dgx-1/

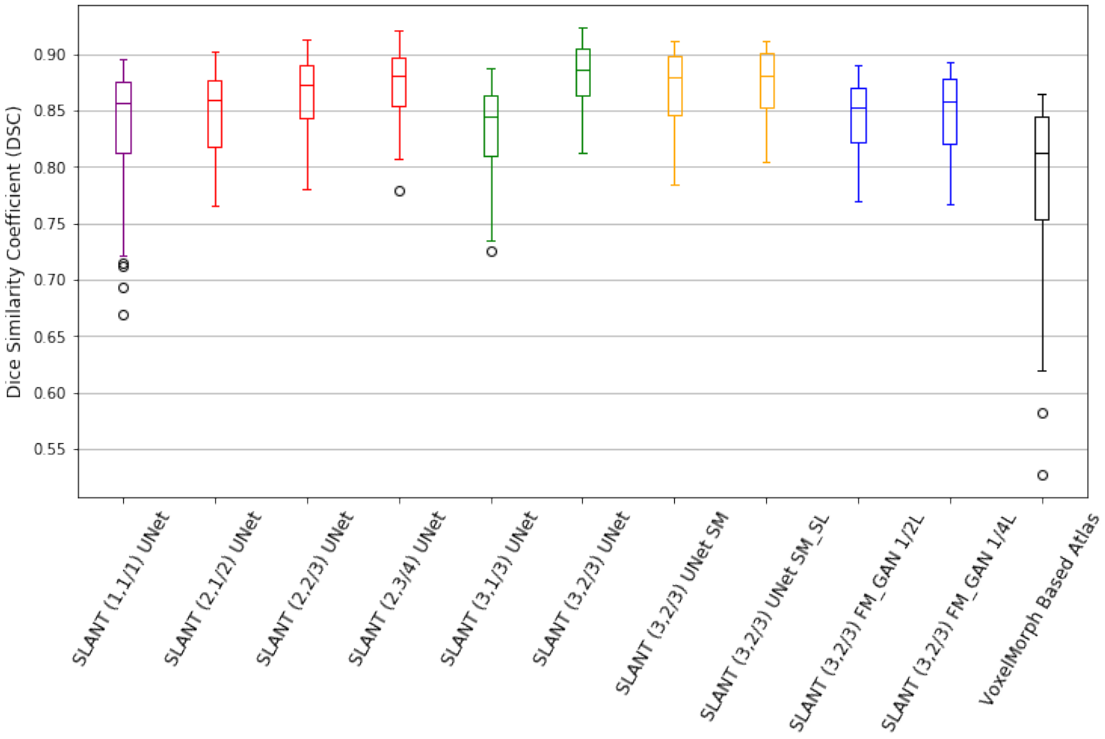


Figure 13: Statistical analysis using Dice Similarity metric function for the entire brain region. SM: Single Model, SL: Spatial Location and L: Labeled Volumes.

SLANT(3,$\frac{2}{3}$) UNet configuration obtain the lowest HD only Limbic White Matter parcel, while the same configuration with FM GAN model and half labeled data has the lowest score in Insula and Basel Ganglia parcels. However, the rest of Telencephalon parcels attain the lowest HD with VMBAS. SLANT(1,$\frac{1}{1}$) UNet configuration is unable to predict the Insula parcel in the two of cross validation models.

Table 5 lists the average DSC scores for four Brain tissue groups, which are Diencephalon, Mesencephalon, Metencephalon and Myelencephalon. SLANT(3,$\frac{2}{3}$) UNet provides the highest DSC scores in the parcels under the all four groups except the Thalamus parcel under Diencephalon group. The highest DSC score for Thalamus parcel achieved in single model configuration in SLANT(3,$\frac{2}{3}$). Meanwhile, this configuration fails to detect the Medulla parcel in one of cross validation models. In Table 6, the lowest HD values in Basal Forebrain, Midbrain and the Pons parcels are obtained in SLANT(3,$\frac{2}{3}$) UNet, while Cerebellum parcel achieves the lowest HD in VMBAS. Medulla parcel shows the lowest HD in single model SLANT(3,$\frac{2}{3}$) with spatial features. The lowest HD score among all the parcels is obtained in the Thalamus parcel by using (3,$\frac{2}{3}$) FM_GAN $\frac{1}{2}$L configuration.

SLANT(3,$\frac{2}{3}$) UNet configuration carries out the highest DSC scores in the CSF tissue group, as shown in Table 7. It also provides the lowest HD values in III_Ventricle and IV_Ventricle parcels as shown in Table 8, while the lowest HD value in LateralVentricle is attained by VMBAS.

Figures 14 and 15 illustrate the parcellation results for a predicted testing image sample. Comparing the predicted images visually with the ground truth reveals some segmentation challenges in the different experimental cases. As shown in the predicted images using the SLANT approach, some parcels are bleeding as they are falsely segmented as the neighboring parcels. Meanwhile, the predicted parcels in the overlapped regions are detected more precisely. The predicted parcels boundaries from the VMBAS are less accurate. Moreover, they completely fail to classify some brain voxels.

5. Discussion

Many sources of error have an impact on the segmentation. For instance, the delineation error from the parcellation application that generates a noisy pseudo ground truth. It affects the training model and the prediction results, and it may underestimate the tested methods due to the noise that pseudo GT inevitably. Therefore, an iterative certainty metrics such as cross-fold validation can be applied to reduce the pseudo GT error impact, and to obtain more meaningful results.

Affine registration process can has an indirect effect on the segmentation process, which may be correlated with similarity between the registered image and the MNI template image. Consequently, the distribution of

Experiment	Telencephalon							
	Parietal	**Limbic**	**Insula**	**BasalGangl**	**InferiorWM**	**Frontal**	**Temporal**	**LimbicWM**
$(1,\frac{1}{1})$ UNet	77.00	80.99	*26.79	85.53	84.85	82.72	83.74	72.18
$(2,\frac{1}{2})$ UNet	71.84	84.34	81.22	86.20	87.75	84.70	86.49	75.51
$(2,\frac{2}{3})$ UNet	79.39	85.49	88.50	89.94	88.60	85.61	87.60	72.89
$(2,\frac{3}{4})$ UNet	80.05	85.84	88.92	90.14	88.92	85.94	87.82	77.88
$(3,\frac{1}{3})$ UNet	75.53	84.13	87.16	83.69	84.41	84.12	86.16	75.42
$(3,\frac{2}{3})$ UNet	**81.00**	**86.99**	**89.84**	91.47	**89.93**	**86.98**	**88.62**	**79.39**
$(3,\frac{2}{3})$ UNet SM	78.82	86.39	89.17	**91.57**	89.08	86.02	87.14	79.10
$(3,\frac{2}{3})$ UNet SM_SL	79.59	86.66	89.16	91.12	89.43	86.03	87.47	78.84
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	75.66	81.05	83.91	89.89	87.82	81.61	81.54	74.46
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	75.90	80.06	82.68	88.13	86.90	82.68	82.48	73.12
VMBAS	70.12	76.32	78.44	76.64	79.36	78.37	80.28	60.89

Table 3: Percentage average Dice Similarity Coefficient for Telencephalon parcels group. BaselGangl: Basel Ganglia and WM: White Matter. (*) Failed to calculate Dice score in two of the cross fold experiments.

Experiment	Telencephalon							
	Parietal	**Limbic**	**Insula**	**BasalGangl**	**InferiorWM**	**Frontal**	**Temporal**	**LimbicWM**
$(1,\frac{1}{1})$ UNet	54.05	34.77	–	27.02	26.73	72.17	51.48	18.31
$(2,\frac{1}{2})$ UNet	68.38	28.31	44.60	21.15	29.47	62.30	41.64	19.82
$(2,\frac{2}{3})$ UNet	32.00	24.15	10.13	14.69	22.05	28.09	39.30	13.87
$(2,\frac{3}{4})$ UNet	20.17	19.90	9.49	8.09	16.18	26.68	20.60	14.85
$(3,\frac{1}{3})$ UNet	36.84	32.92	56.15	49.08	38.88	35.95	48.74	23.71
$(3,\frac{2}{3})$ UNet	10.33	10.24	6.00	7.10	11.28	13.18	13.66	**10.73**
$(3,\frac{2}{3})$ UNet SM	30.91	22.00	11.50	10.12	14.11	30.51	31.74	13.89
$(3,\frac{2}{3})$ UNet SM_SL	25.21	23.77	11.32	8.13	17.66	20.15	27.08	12.16
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	9.34	9.54	**4.74**	**6.70**	12.03	11.37	9.44	14.58
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	11.55	11.64	5.43	6.84	12.79	11.66	11.74	11.53
VMBAS	**8.71**	**8.38**	5.73	7.04	**11.19**	**8.68**	**7.96**	11.49

Table 4: Average Hausdorff Distance for Telencephalon parcels group. BaselGangl: Basel Ganglia and WM: White Matter. (–) Failed to calculate HD in two of the cross fold experiments.

Experiment	Diencephalon		Mesencephalon	Metencephalon		Myelencephalon
	Thalamus	**BasalForebrain**	**Midbrain**	**Pons**	**Cerebellum**	**Medulla**
$(1,\frac{1}{1})$ UNet	89.21	79.79	91.84	94.54	94.15	92.89
$(2,\frac{1}{2})$ UNet	91.87	77.78	92.57	94.93	94.96	75.65
$(2,\frac{2}{3})$ UNet	91.45	84.59	93.11	94.91	96.20	91.61
$(2,\frac{3}{4})$ UNet	92.12	86.01	94.32	95.85	96.01	94.05
$(3,\frac{1}{3})$ UNet	71.90	76.89	88.69	93.45	94.85	87.76
$(3,\frac{2}{3})$ UNet	92.99	**87.39**	**94.58**	**96.27**	**96.56**	**94.94**
$(3,\frac{2}{3})$ UNet SM	**93.57**	87.28	94.10	95.41	95.55	*62.62
$(3,\frac{2}{3})$ UNet SM_SL	93.38	86.63	94.53	95.67	95.93	93.90
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	93.14	83.73	93.24	94.64	92.71	93.09
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	91.87	82.45	93.03	94.82	94.80	92.89
VMBAS	83.60	65.10	80.97	86.71	90.02	83.94

Table 5: Percentage average Dice Similarity Coefficient for Diencephalon, Mesencephalon, Metencephalon and Myelencephalon parcels groups. (*) Failed to calculate Dice score in one of the cross fold experiments.

the training data will be centered on the template at least as far as the spatial arrangement of the anatomy is concerned. In addition, it changes the different parcels size.

Another challenging issue is the lack of differentiability between the intensity distributions/textures of neighboring parcels which cause inaccurate segmentation boundaries between the parcels or completely false segmented parcels. Therefore, strong spatial priors can overcome this problem as in VMBAS approach. In addition, the non-brain tissue classes especially that include a combined different types of face and head tissues in a single parcel. This creates a class with variable features that interfere with the multiple primary features of other classes. Therefore, the non-brain tissue classes

Experiment	Diencephalon		Mesencephalon	Metencephalon		Myelencephalon
	Thalamus	BasalForebrain	Midbrain	Pons	Cerebellum	Medulla
$(1,\frac{1}{1})$ UNet	12.87	32.90	8.31	12.71	51.82	26.37
$(2,\frac{1}{2})$ UNet	29.63	75.48	49.23	35.12	97.62	105.98
$(2,\frac{2}{3})$ UNet	10.92	14.07	30.00	6.05	21.86	93.12
$(2,\frac{3}{4})$ UNet	5.45	7.33	5.68	9.79	28.60	46.77
$(3,\frac{1}{3})$ UNet	56.24	56.77	62.26	56.56	42.46	100.39
$(3,\frac{2}{3})$ UNet	4.35	**4.75**	**5.05**	**4.59**	8.30	51.54
$(3,\frac{2}{3})$ UNet SM	10.70	10.69	6.90	5.99	14.46	–
$(3,\frac{2}{3})$ UNet SM_SL	21.65	8.33	14.31	6.08	14.10	**3.70**
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	**3.24**	5.37	9.73	17.61	12.61	25.47
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	3.47	5.97	10.17	6.91	12.53	14.65
VMBAS	4.63	5.21	5.18	5.16	**7.74**	4.41

Table 6: Average Hausdorff Distance for Diencephalon, Mesencephalon, Metencephalon and Myelencephalon parcels groups. (–) Failed to calculate HD in one of the cross fold experiments.

Experiment	CSF		
	LateralVentricle	III_Ventricle	IV_Ventricle
$(1,\frac{1}{1})$ UNet	90.35	86.28	90.55
$(2,\frac{1}{2})$ UNet	92.73	87.04	90.71
$(2,\frac{2}{3})$ UNet	93.90	90.30	76.46
$(2,\frac{3}{4})$ UNet	94.10	90.17	92.35
$(3,\frac{1}{3})$ UNet	91.69	40.35	85.25
$(3,\frac{2}{3})$ UNet	**94.56**	**91.25**	**92.75**
$(3,\frac{2}{3})$ UNet SM	94.21	90.14	91.24
$(3,\frac{2}{3})$ UNet SM_SL	94.13	90.37	91.29
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	93.25	89.22	90.86
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	92.81	89.88	90.55
VMBAS	85.42	74.98	79.54

Table 7: Percentage average Dice Similarity Coefficient for CSF parcels group. III: The Third and IV: The Fourth

Experiment	CSF		
	LateralVentricle	III_Ventricle	IV_Ventricle
$(1,\frac{1}{1})$ UNet	22.49	12.48	20.17
$(2,\frac{1}{2})$ UNet	25.03	25.23	41.75
$(2,\frac{2}{3})$ UNet	20.96	11.54	30.07
$(2,\frac{3}{4})$ UNet	17.42	7.23	4.42
$(3,\frac{1}{3})$ UNet	31.43	63.68	88.38
$(3,\frac{2}{3})$ UNet	16.16	**6.22**	**3.42**
$(3,\frac{2}{3})$ UNet SM	20.26	14.65	4.35
$(3,\frac{2}{3})$ UNet SM_SL	16.29	9.86	7.66
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{2}$L	15.72	5.45	3.91
$(3,\frac{2}{3})$ FM_GAN $\frac{1}{4}$L	18.58	8.22	4.73
VMBAS	**13.35**	6.85	5.02

Table 8: Average Hausdorff Distance for CSF parcels group. III: The Third and IV: The Fourth

are scripted from the database in many implementations before the processing.

Furthermore, unbalanced parcels size makes detecting and segmenting tasks very hard for small parcels. Thus, this problem can be handled by using a loss function during the training stage which normalize each parcel loss based on the its size, such as in Dice Similarity loss function, or multiplying each parcel loss by a certain weight. The drawback of these techniques that it can make a single false predicted pixel in tiny parcels can have the same effect as missing nearly a whole large parcels.

In the SLANT approach, the network segmentation model suffers from disperse false predicted segmentation voxels, and accordingly they deteriorate the HD error. These errors are decreased in the high number of

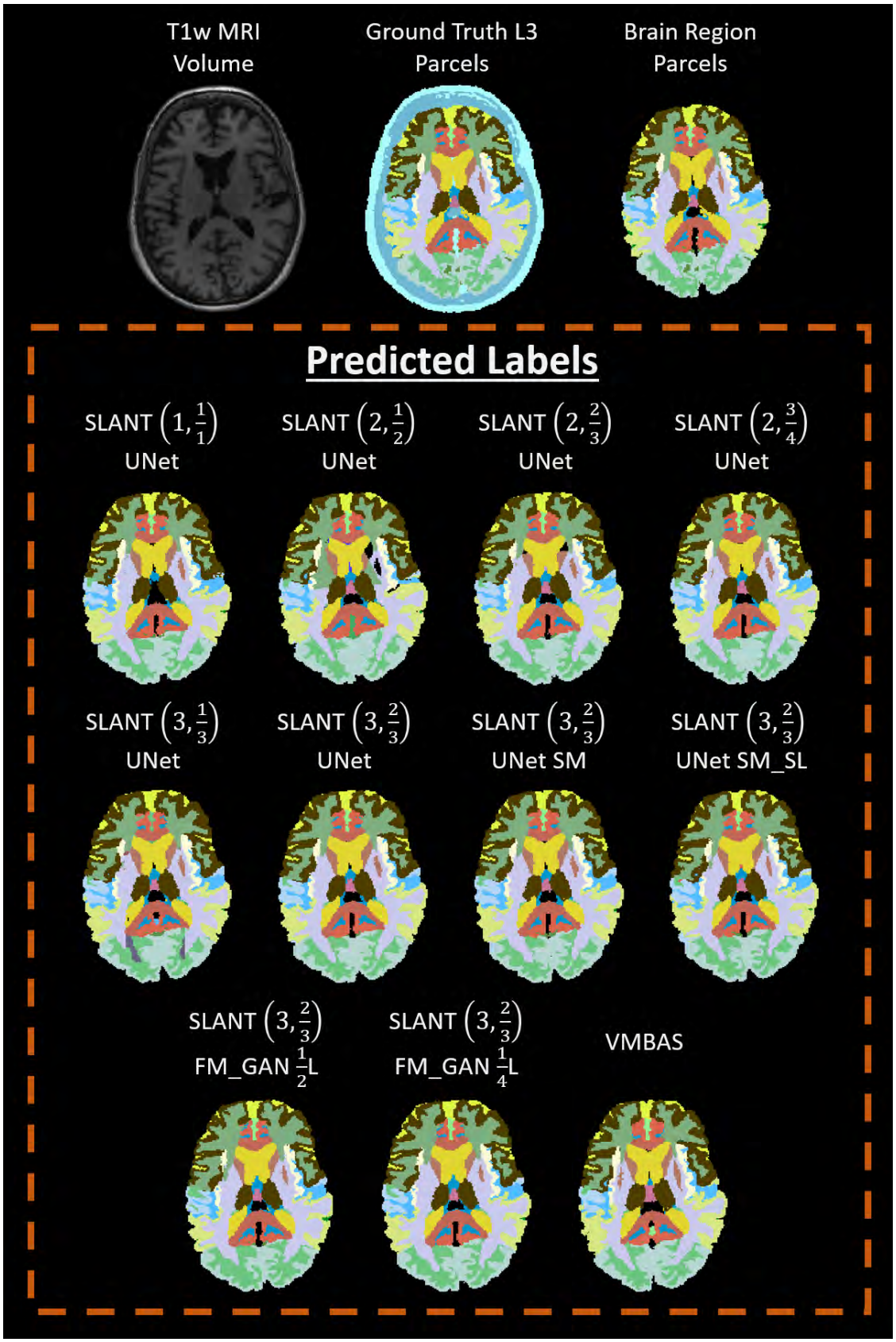


Figure 14: Axial position for selected medium quality testing image. The parcels are predicted using the best selected models from the same cross validation.

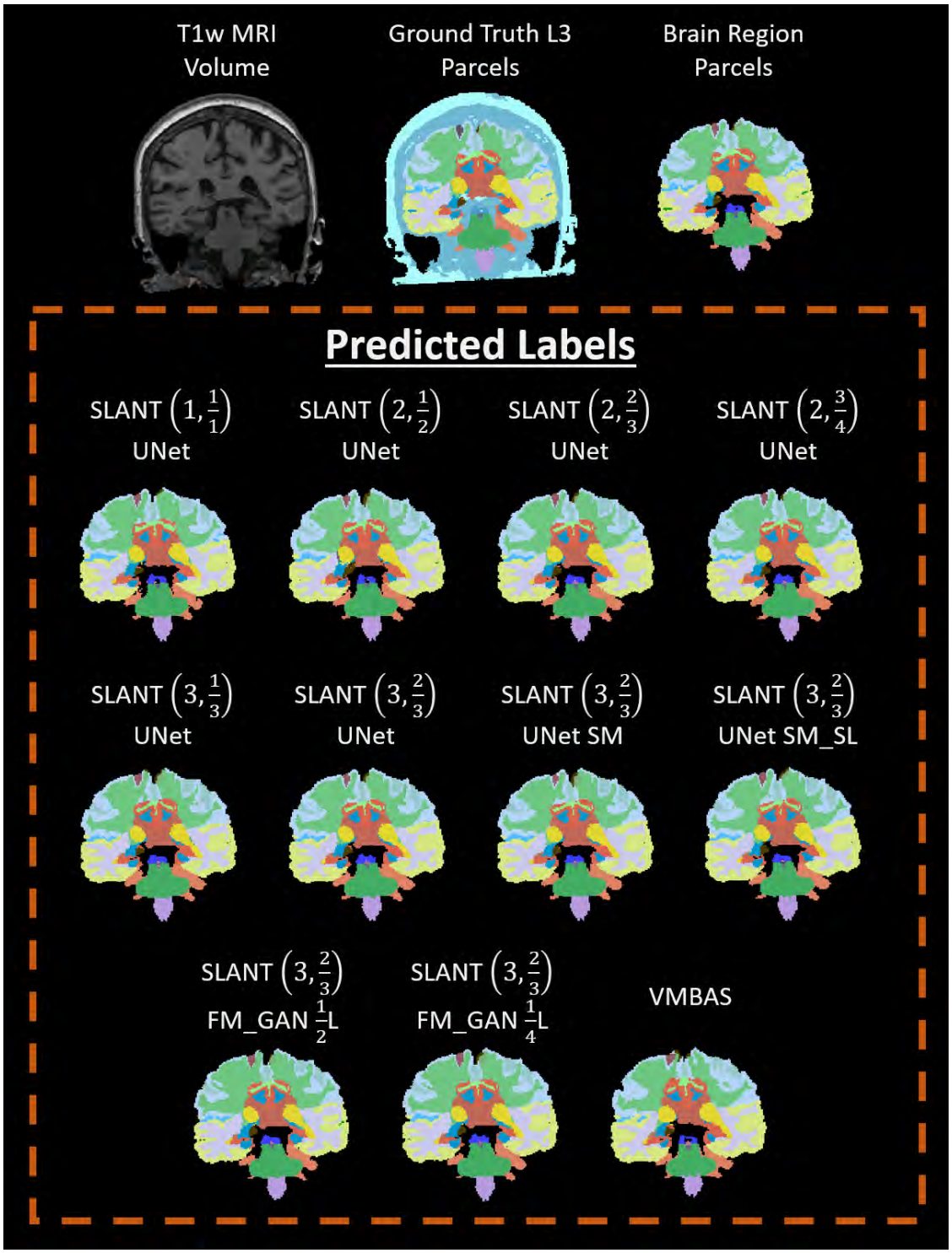


Figure 15: Coronal position for selected medium quality testing image. The parcels are predicted using the best selected models from the same cross validation.

overlapped regions as they are predicted from different models that corrects the combined prediction probabilities. Therefore, the parcels in the overlapped regions towards the image center as in SLANT$(3,\frac{2}{3})$ configuration have smaller HD error.

The depth of the generator and the discriminator in the FM GAN model are not adaptive to the variation of the nonuniform tiles size. Thus, it generates more pixeled fake images when the tile size increases. Using more deep network can overcome this problem as shown in Figure 16, however it can make the model over-fits very fast, and it requires more GPU memory

size. Moreover, having a precise generator can actually degenerate the training performances since in this case the model will not be able to distinguish between unlabeled and fake tiles.

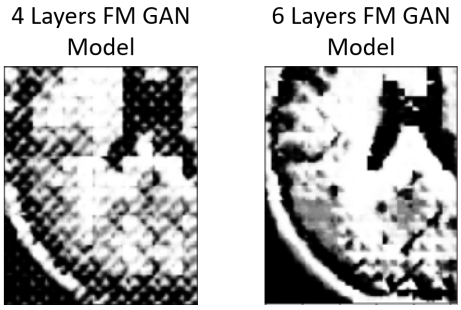


Figure 16: The generated fake tiles using different depth of discriminator and generator networks.

The predicted parcels from VMBAS are spatially constrained with deformed atlas probabilities which prevents the segmentation from having dispersed false predicted segmentation voxels, and therefore it has comparatively low HD error. On the other hand, this approach is reliant on intensity variation for likelihood parameters which makes it not robust to segment high numbers of parcels with similar intensity characteristics. Furthermore, its level of supervision during training does not consider the class size imbalance between parcels. Consequently, it provides comparatively low Dice scores.

6. Conclusions

This research employs two recent approaches which combine medical image processing in MNI space with Deep Learning for full brain volume segmentation. The first approach is substitutional implementation for the SLANT approach by merging the left side and right side tiles while training each model, exploiting the symmetry property of the brain. In addition, preforming analyses using model single model for all tiles with and without feeding the training model with the 3D spatial location feature. All the annotated training data are used for fully supervised training technique using UNet model. Another semi-supervised learning technique is proposed to use a small portion of the annotated training data, combining the tiles-based method in the SLANT approach with the FM GAN model. The second approach is an implementation for unsupervised segmentation principled approach based on combining VoxelMorph registration network with Probabilistic atlas priors. The experimental works are performed on a combination of MRI datasets from several medical resources, and the ground truth are annotated using an automatic parcellation application.

SLANT approach using UNet model has the strongest supervision during training, and it is therefore have the highest segmentation performance. Semi-supervised learning model using FM GAN can be trained on very few number of labeled data and providing comparably acceptable segmentation results. The overlapped regions in SLANT improves segmentation, partly overcoming the lack of a strong spatial prior. Increasing this overlaid regions provided more robust results. Using single model and adding the spatial features requires Less computational resources. However, it does not outperform the original SLANT configuration. VMBAS has the lowest segmentation performance, since it is unsupervised segmentation approach.

References

Aljabar, P., Heckemann, R.A., Hammers, A., Hajnal, J.V., Rueckert, D., 2009. Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. Neuroimage 46, 726–738.

Angermann, C., Haltmeier, M., 2019. Random 2.5 D U-Net for fully 3D segmentation, in: Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting. Springer, pp. 158–166.

Arsigny, V., 2006. Processing data in Lie groups: an algebraic approach. application to non-linear registration and diffusion tensor MRI. Ph.D. thesis.

Asman, A.J., Landman, B.A., 2013. Non-local statistical label fusion for multi-atlas segmentation. Medical image analysis 17, 194–208.

Baillard, C., Hellier, P., Barillot, C., 2001. Segmentation of brain 3D MR images using level sets and dense registration. Medical image analysis 5, 185–194.

Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V., 2018. An unsupervised learning model for deformable medical image registration, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9252–9260.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35, 1798–1828.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 424–432.

Dalca, A.V., Balakrishnan, G., Guttag, J., Sabuncu, M.R., 2018. Unsupervised learning for fast probabilistic diffeomorphic registration, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 729–738.

Dalca, A.V., Yu, E., Golland, P., Fischl, B., Sabuncu, M.R., Iglesias, J.E., 2019. Unsupervised deep learning for Bayesian brain MRI segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 356–365.

Dhanachandra, N., Manglem, K., Chanu, Y.J., 2015. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science 54, 764–771.

Dong, H., Yang, G., Liu, F., Mo, Y., Guo, Y., 2017. Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks, in: annual conference on medical image understanding and analysis, Springer. pp. 506–517.

Fei-Fei, L., Fergus, R., Perona, P., 2006. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence 28, 594–611.

Frazier, J., Caviness, V., Kennedy, D., Worth, A., Haselgrove, C., Caplan, D., Makris, N., 2007. Internet brain segmentation repository (IBSR) 1.5 mm dataset. Collections 10, C6RC85.

Guimond, A., Meunier, J., Thirion, J.P., 2000. Average brain models: A convergence study. Computer vision and image understanding 77, 192–210.

Heckemann, R.A., Hajnal, J.V., Aljabar, P., Rueckert, D., Hammers, A., 2006. Automatic anatomical brain MRI segmentation combining label propagation and decision fusion. NeuroImage 33, 115–126.

Huo, Y., Xu, Z., Aboud, K., Parvathaneni, P., Bao, S., Bermudez, C., Resnick, S.M., Cutting, L.E., Landman, B.A., 2018. Spatially localized atlas network tiles enables 3D whole brain segmentation from limited data, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 698–705.

Huo, Y., Xu, Z., Xiong, Y., Aboud, K., Parvathaneni, P., Bao, S., Bermudez, C., Resnick, S.M., Cutting, L.E., Landman, B.A., 2019. 3D whole brain segmentation using spatially localized atlas network tiles. NeuroImage 194, 105–119.

Iglesias, J.E., Sabuncu, M.R., 2015. Multi-atlas segmentation of biomedical images: a survey. Medical image analysis 24, 205–219.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International journal of computer vision 1, 321–331.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic gradient descent, in: ICLR: International Conference on Learning Representations.

Klein, S., Staring, M., Murphy, K., Viergever, M.A., Pluim, J.P., 2009. Elastix: a toolbox for intensity-based medical image registration. IEEE transactions on medical imaging 29, 196–205.

Krebs, J., Delingette, H., Mailhé, B., Ayache, N., Mansi, T., 2019. Learning a probabilistic model for diffeomorphic registration. IEEE transactions on medical imaging 38, 2165–2176.

Lancaster, J.L., Tordesillas-Gutiérrez, D., Martinez, M., Salinas, F., Evans, A., Zilles, K., Mazziotta, J.C., Fox, P.T., 2007. Bias between MNI and Talairach coordinates analyzed using the ICBM-152 brain template. Human brain mapping 28, 1194–1205.

Landman, B., Warfield, S., 2012. MICCAI 2012 multi-atlas labeling challenge, in: MICCAI 2012 Workshop on Multi-Atlas Labeling, pp. 1–164.

Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D., 1997. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks 8, 98–113.

LeCun, Y., 1998. The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ .

Madabhushi, A., Udupa, J.K., 2006. New methods of mr image intensity standardization via generalized scale. Medical physics 33, 3426–3434.

McInemey, T., Terzopoulos, D., 1999. Topology adaptive deformable surfaces for medical image volume segmentation. IEEE transactions on medical imaging 18, 840–850.

Mikhael, S.S., Pernet, C., 2019. A controlled comparison of thickness, volume and surface areas from multiple cortical parcellation packages. BMC bioinformatics 20, 55.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision (3DV), IEEE. pp. 565–571.

Mondal, A.K., Dolz, J., Desrosiers, C., 2018. Few-shot 3D multimodal medical image segmentation using generative adversarial learning. arXiv preprint arXiv:1810.12241 .

Mori, S., Wu, D., Ceritoglu, C., Li, Y., Kolasny, A., Vaillant, M.A., Faria, A.V., Oishi, K., Miller, M.I., 2016. MRICloud: delivering high-throughput mri neuroinformatics as cloud-based software as a service. Computing in Science & Engineering 18, 21–35.

Murphy, S., Mohr, B., Fushimi, Y., Yamagata, H., Poole, I., 2014. Fast, simple, accurate multi-atlas segmentation of the brain, in: International Workshop on Biomedical Image Registration, Springer. pp. 1–10.

Rohlfing, T., Russakoff, D.B., Maurer, C.R., 2004. Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation. IEEE transactions on medical imaging 23, 983–994.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.

Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E., Summers, R.M., 2014. A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 520–527.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans, in: Advances in neural information processing systems, pp. 2234–2242.

Schaap, M., Metz, C.T., van Walsum, T., van der Giessen, A.G., Weustink, A.C., Mollet, N.R., Bauer, C., Bogunović, H., Castro, C., Deng, X., et al., 2009. Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. Medical image analysis 13, 701–714.

Simkó, A., Löfstedt, T., Garpebring, A., Nyholm, T., Jonsson, J., 2019. A generalized network for MRI intensity normalization. arXiv preprint arXiv:1909.05484 .

Staib, L.H., Duncan, J.S., 1992. Boundary finding with parametrically deformable models. IEEE Transactions on Pattern Analysis & Machine Intelligence , 1061–1075.

Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C., 2010. N4ITK: improved N3 bias correction. IEEE transactions on medical imaging 29, 1310–1320.

Van Ginneken, B., Frangi, A.F., Staal, J.J., ter Haar Romeny, B.M., Viergever, M.A., 2002. Active shape model segmentation with optimal features. IEEE transactions on medical imaging 21, 924–933.

de Vent, N.R., Agelink van Rentergem, J.A., Schmand, B.A., Murre, J.M., Huizenga, H.M., Consortium, A., et al., 2016. Advanced Neuropsychological Diagnostics Infrastructure (ANDI): A normative database created from control datasets. Frontiers in Psychology 7, 1601.

Wang, H., Suh, J.W., Das, S.R., Pluta, J.B., Craige, C., Yushkevich, P.A., 2012. Multi-atlas segmentation with joint label fusion. IEEE transactions on pattern analysis and machine intelligence 35, 611–623.

Wang, L., Shi, F., Li, G., Gao, Y., Lin, W., Gilmore, J.H., Shen, D., 2014. Segmentation of neonatal brain MR images using patch-driven level sets. NeuroImage 84, 141–158.

Warfield, S.K., Zou, K.H., Wells, W.M., 2004. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. IEEE transactions on medical imaging 23, 903–921.

Wu, D., Ma, T., Ceritoglu, C., Li, Y., Chotiyanonta, J., Hou, Z., Hsu, J., Xu, X., Brown, T., Miller, M.I., et al., 2016. Resource atlases for multi-atlas brain segmentations with multiple ontology levels based on T1-weighted MRI. Neuroimage 125, 120–130.

Wu, M., Rosano, C., Lopez-Garcia, P., Carter, C.S., Aizenstein, H.J., 2007. Optimum template selection for atlas-based segmentation. NeuroImage 34, 1612–1618.

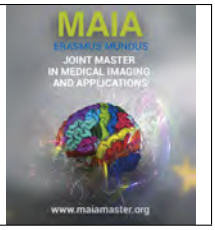
Zhang, D.Q., Chen, S.C., 2004. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. Artificial intelligence in medicine 32, 37–50.

Zhang, Y., Brady, M., Smith, S., 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE transactions on medical imaging 20, 45–57.

Zhu, S.C., Yuille, A., 1996. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE transactions on pattern analysis and machine intelligence 18, 884–900.

Medical Imaging and Applications

Master Thesis, August 2020



Prediction of the Histological Grading of Meningiomas Using Magnetic Resonance Images

Nur Adhianti Heryanto, Henning Müller

Institute of Information Systems
University of Applied Sciences Western Switzerland (HES-SO)
3960-Sierre, Switzerland

Abstract

Histological grading is one of the factors considered when determining the treatment of meningioma. Assessment of histological meningioma grading still relies on an invasive surgical procedure for histopathology. Magnetic resonance imaging (MRI) is the standard imaging modality for diagnosis and surveillance of meningioma. Prediction of the grading from MRI can thus improve treatment planning of meningioma without a need for an invasive procedure. This work aimed to predict the histological grading of meningioma using convolutional neural networks (CNN) from multiple MRI weightings. Besides utilizing it for end-to-end classification, CNNs were also used as feature extractor at the feature and decision level. Machine learning classifiers were trained on the features extracted and compared to hand-crafted radiomics features. The dataset used in this study focused on 3T MRI scans (for homogeneity) consisting of contrast-enhanced T1-weighted (T1-CE) sequences and apparent diffusion coefficient (ADC) maps. A total of 105 cases were included, composed of 78 WHO Grade I, 23 WHO Grade II, and 4 WHO Grade III. The data was grouped into low (WHO Grade I) and high grade (WHO Grade II/III) meningioma. Tumor masks were manually annotated for T1-CE images. Slices were cropped to the tumor masks as a region of interest. For each patient, the slice with the largest tumor area and one following slice before and after it were selected for input to the neural network. Transfer learning with fine-tuning was performed using a pre-trained ResNet-18 model. Each MRI weighting was passed to the model and trained separately. Using ADC maps, end-to-end CNN reached 0.71±0.19 sensitivity and 0.7±0.11 accuracy in 5-fold cross-validation. Deep features extracted from ADC maps using the trained CNN at feature level and logistic regression as the classifier achieved 0.89±0.15 sensitivity and 0.86±0.07 accuracy. Even though the end-to-end strategy using CNNs could not achieve good generalization on unseen data, it was able to learn features that showed superior performance to hand-crafted radiomics in differentiating low and high grade meningioma.

Keywords: Convolutional neural network, histological grading, meningioma, prediction

1. Introduction

Meningioma is one of the most common primary intracranial tumors. According to the most recent survey by the Central Brain Tumor Registry of the United States (CBTRUS), the incidence rate of meningioma is approximately 8.6/100,000 persons, accounting for more than 37.6% of the primary brain tumors in 2012-2016 (Ostrom et al., 2019). The incidence increases with age (median = 66 years) and is predominant in females. Most meningiomas are benign and slowly growing tumors, while a minority (10-20%) show aggressive behaviour with increasing risks of recurrence. Even though the majority of these tumors are considered benign, meningiomas are often associated with increased morbidity including focal neurological deficits, seizures, and decreased quality of life (Buerki et al., 2018). It is their intracranial location (although about 5% can be found in spinal meninges), that can lead to fatal outcomes. Based on their histological features, the World Health Organization (WHO) categorizes meningioma into three types, which are benign (WHO grade I), atypical (WHO grade II), and anaplastic or malignant (WHO grade III) (Louis et al., 2016).

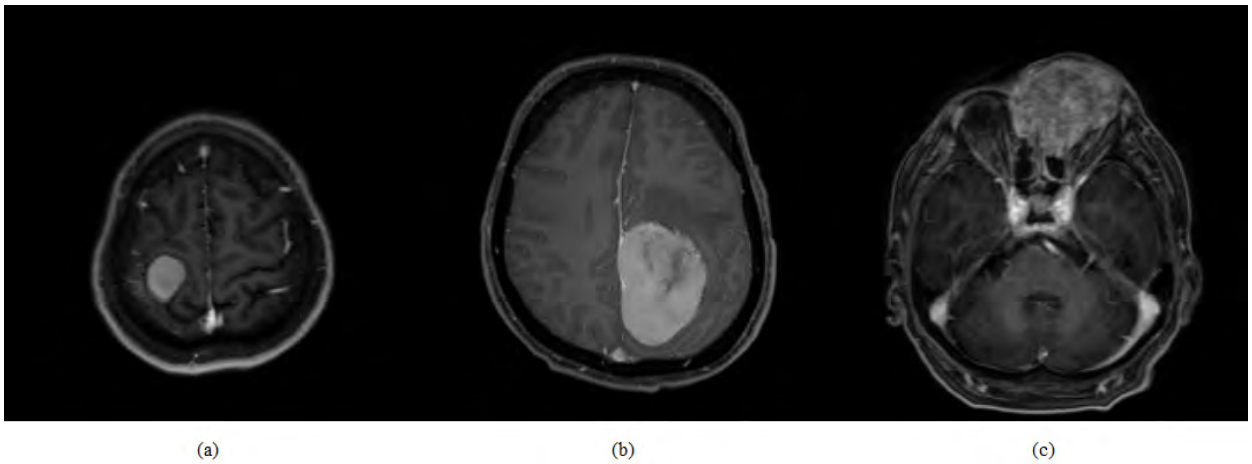


Figure 1: Meningioma of different grades and locations in axial contrast-enhanced T1-weighted images (left to right): (a) WHO Grade I, (b) WHO Grade II, and (c) WHO Grade III. Images shown are slices with the biggest tumor area in the volume.

Among the risk factors for meningioma, exposure to high dose ionizing radiation is known to have the strongest evidence (Claus et al., 2005). Another factor that can increase the risk for meningioma is certain mutations in the neurofibromatosis gene (*NF*2). Hormones are also hypothesized as a risk factor since women are twice as likely as men to develop meningiomas. The etiology of meningioma is still relatively understudied compared to malignant glial tumors, which is due to several factors described in Wiemels et al. (2010). It is a relatively rare disease hence requiring large or multicenter studies for sufficiently large numbers. It al in cde fg in a long latency of 20-30 years or more. In addition, many cases of meningioma are discovered incidentally through brain imaging. Such cases are usually managed through observation without surgical treatment.

Definitive diagnosis of meningioma, including histological grading, requires biopsy or a surgical procedure to obtain tumor tissue for histopathology (Goldbrunner et al., 2016). However, if a meningioma is already strongly indicated in imaging, histological verification is not mandatory. Histological grading is an important key to take into account for prognosis and treatment of meningioma. The standard theurapetic management for meningioma is surgical resection with the aim of gross total resection (GTR). The extent of resection heavily affects prognosis and recurrence rates. Several studies by Komotar et al. (2012), Aizer et al. (2014), and Wang et al. (2017) reported that GTR and adjuvant radiotherapy post-surgery is associated with improved overall survival. Therapy of meningioma should be personalised considering the degree of varying consequences of different treatment for different patients. Presurgical evaluation of the tumor grade thus may help and improve treatment planning.

MRI remains the preferred imaging modality for radiological diagnosis and surveillance of meningioma. Many research projects have been conducted to assess meningioma grading from various preoperative MRI sequences (Huang et al., 2019). Most of them are limited in the number of samples (up to hundreds) and single center studies. These studies suggest that the MR sequences can be useful in assessing tumor grades. There is still no clear parameter that could best differentiate the grades. This implies the task to distinguish lower and higher grades meningioma is still challenging. Approaches using parameters measured from diffusion weighted imaging (DWI) and diffusion tensor imaging (DTI) reported variable findings, some indicating statistical significance to the histological grade while others concluded the contrary (Yin et al., 2012) (Watanabe et al., 2013) (Aslan et al., 2018). Studies using texture analysis and radiomics features showed promising diagnostic ability in meningioma grading. They evaluated the diagnostic value of classification methods using the features extracted (Hamerla et al., 2019) (Laukamp et al., 2019a) (Ke et al., 2019).

Radiomics is a field of research aiming to convert images into mineable high-dimensional data (Gillies et al., 2016) (Lambin et al., 2017). They include quantitative features that are usually grouped into texture parameters: shape, first order, second order, and higher order statistics. These features can capture the structural relationships between pixel intensities and spectral properties that may be visually imperceptible (Kassner and Thornhill, 2010). However, as radiomics features are hand-crafted and require machine learning, it can be time consuming. Deep learning (or CNNs, more specifically) has become state-of-the-art for image classification in the computer vision domain and certain tasks in the medical imaging domain such as segmentation. It is known to be able to extract high-level features that may be a tedious task in traditional machine learning. It has not been largely explored for prediction of meningioma grades, possibly due to the fact that most studies only have limited data, which is not favorable when us-

ing deep learning. For other tasks such as meningioma segmentation or detection (differentiating meningioma from other tumor types, not the grades), it was reported that deep learning could perform well (Laukamp et al., 2019b) (Laukamp et al., 2020).

In contribution to existing research, this master thesis was focused on using two MRI weightings, T1-contrast enhanced and ADC maps, for prediction of meningioma grades. The main approach was using a convolutional neural network. Not only using it for an end-to-end prediction, the neural network was also used as a feature extractor. For comparison, traditional hand-crafted features were extracted and used for classification. With the features extracted, machine learning classifiers were trained and the performance of the different approaches was observed. As a summary, this study aimed to:

- Evaluate the performance of CNNs for the prediction of histological grading of meningioma prediction using single and multiple MRI weightings.
- Evaluate the performance of high-level deep learning-based features and hand-crafted features with machine learning classifiers for prediction of histological grading of meningioma

The thesis is organized as follows: Section 2 describes existing research related to meningioma grading with machine learning and deep learning approaches. Section 3 explains in detail the data used for this work along with the strategies employed in each step. Results are reported in Section 4, whereas analysis of the data is presented in Section 5. Finally, future works and conclusions are given in Section 6.

2. State of the art

2.1. Traditional machine learning

Research by Yan et al. (2017) is among the pilot studies that used texture analysis and machine learning classifiers on preoperative MRI to differentiate lower and higher grade meningioma. Their work is based on a single center study of 131 patients, with 21 high grade or WHO Grade I and 110 low grade or WHO Grade II/III cases. Features were extracted from contrast-enhanced T1-weighted images (T1-CE) with tumor lesion as the region of interest. Three shape features and three texture features were selected through a statistical test (Mann-Whitney), indicating they were significantly different between low and high grade meningiomas. Classification performance was evaluated through a ten-fold cross-validation. SVMs were their best performing model with 0.86 AUC and 0.87 diagnostic accuracy.

Coroller et al. (2017) not only used radiomics features but also semantic (qualitative) features from T1-CE sequences to predict low and high grade meningiomas. It is another single center study of 175 patients composing of 103 low grade and 72 high grade cases. The dataset was split into training (131 cases) and an independent validation dataset (44 cases). Semantic features were selected through uni-variate analysis whereas radiomics features were chosen with a multivariate analysis. Eight radiomics features and four semantic features were found to be the most predictive for meningioma grades. Random forests were used for classification with a nested cross validation in the training step for model tuning and selection. Combining the features increased the performance from AUC of 0.77 and 0.79 on using each features alone to 0.86 AUC.

Chen et al. (2019) published another research based on T1-CE images. Their research is among the few that classify all three meningioma grades, not just differentiating low and high grade tumors. It is still a single center study of 150 patients, consisting of 61 WHO Grade I, 59 WHO Grade II, and 30 WHO Grade III cases. Features extracted were evaluated using Pearson's correlation coefficient. They experimented with several feature selection methods and different multi-class classifiers. A 5-fold cross validation was used in the training step. Their best performing model is linear discriminant analysis (LDA) using 15 radiomics features that were selected using the least absolute shrinkage and selection operator (LASSO). An accuracy of 75.6% with 0.603 kappa value was obtained on the validation set.

Lu et al. (2019) presented another work that classified three grades of meningiomas using ADC values and ADC maps. Their data was obtained from a single study center with 152 cases of WHO Grade I, 48 cases of WHO Grade II, and 15 cases of WHO Grade III (total 215 cases). The data was assigned to a training and testing set with a ratio of 0.7:0.3. The ROI in ADC maps is in reference to the T1-CE images. Extracted features were evaluated through chi-squared and Fisher's exact test. Two step feature selection was then performed. First, three different algorithms yielded ten features each and then, using a recursive feature elimination, 23 features with the lowest misclassification rates were selected. Tree classifiers with a nested cross validation were used to build the model. The best model with a robust performance was a decision forest. They found that the ADC value alone was not significant to the grades. Texture features from ADC maps increased the diagnostic value to over 70% accuracy and a kappa value of over 0.5. Combining the two improved the results even more with over 80% accuracy and kappa value of 0.63.

Multiple studies worked with multi-parametric MRI for prediction of meningioma grades (Hamerla et al., 2019) (Laukamp et al., 2019a) (Ke et al., 2019) (Park et al., 2019). All of them only classified low and high grade meningiomas and are limited in size. They used routine MR sequences with or without diffusion weighted/tensor imaging and reported consistent conclusions. Overall the diagnostic ability is proven to

be increased by using features combined from several MR modalities. Amongst them, Hamerla et al. (2019) used a multi-center study but the number of cases is still relatively small with a total of 147 patients (102 low grade, 45 high grade) obtained from 5 international centers. Five sequences were included in the study: ADC maps, T1W, T1-CE, substraction maps, T2-weighted fluid-attenation inversion recovery (FLAIR), and T2-weighted. Extracted features were ranked with the Mann-Whitney test and selected through random forest with a 4-fold cross-validation. Later, different classification models were built with a 10-fold cross-validation. During the training step, an oversampling method, the synthetic minority oversampling technique (SMOTE), was applied to address the class imbalance. Extreme gradient boosting (XGBoost), which is a tree-based classifier, has the best performance using 16 features from all sequences, obtaining 0.97 AUC.

2.2. Deep learning approaches

Zhu et al. (2019b) utilized deep learning to extract high-level features of meningioma from T1-weighted images. For comparison, the authors also built models using hand-crafted radiomics (HCR) features. A total of 181 patients from two hospitals were included for this study, which they divided into 99 training cases (77 low grade, 22 high grade) and 82 validation cases (69 low grade, 13 high grade). Different approaches were used for deep learning and hand crafted features. For deep learning, transfer learning strategies from a pretrained model on the ImageNet dataset (Xception) was employed to extract 2048 high-level features (Chollet, 2016). Feature selection consisted of a selection through random forest, where features with importance value of at least 0.001 were retained, and a sequential backward selection based on the F-measure. These features were fed into an LDA classifier. A bagging method was applied in both feature selection and model tuning to consider data imbalance. HCR features were followed by two-stage feature selection, Mann-Whitney test and gradient boosting. Classification was then performed with SVMs. There were 39 and 6 features that remained after selection for deep features and HCR respectively. Deep features had superior results compared to HCR, with AUC of 0.81 and 0.72 each. Combining both features and classifying with SVMs only slightly improved the results (from 0.811 to 0.816 AUC).

Although with a lack in detailed information on the data, Zhu et al. (2019a) developed a study using an end-to-end network to predict three grades of meningioma. They have 222 cases in total, split for 190 training cases and 32 validation cases. The MR sequences used and the exact numbers for the data distribution were not specified. However, based on the images they provided, T1-CE sequences were used. Offline data amplification from simple augmentation techniques (mirroring, rotation) was performed to increase training data. They modified the LeNet-5 architecture by adding convolutional, pooling, and softmax layer to build the model which was then trained from scratch. Model performance was evaluated through a 10-fold cross validation. They reached an accuracy of 83.3% on the validation dataset.

This master thesis is primarily referring to the following articles that used the same data source. They used an end-to-end network with different strategies. Due to the number of cases, only low grade and high grade meningiomas were predicted in Banzato et al. (2019), including ADC maps and T1-CE images in their work. Two ImageNet pretrained models, Inception-V3 and AlexNet were used. Images were pre-processed (which included manual intensity corrections) according to the input criteria for network. Data augmentation of random rotation, cropping, flipping, and/or mirroring was also performed. Diagnostic accuracy was evaluated through a leave-one-out cross validation procedure (LOOCV) with another 10-fold cross validation in the training step. Models were initialized with the pretrained weights on every iteration of the leave-one-out case. None of the weights of layers were frozen during training to consider basic features relevant to the dataset. Recall of 84% and AUC of 94% on the validation set was achieved using ADC maps and InceptionV3.

A more recent work by Wodzinski et al. (2020) used T1-CE images to predict low and high grades of meningioma. They highlighted the heterogeneity of the dataset, where images were acquired from 26 different scanners without a single protocol that can be considered as majority. After offline pre-processing, weighted oversampling and strong augmentation (affine transformation, random cropping and flipping, random changing of hue, contrast, and saturation) were performed on every batch generation. Only slices where the lesion is present and cropped to the tumor region were selected. They also used a model pretrained on ImageNet, ResNet-18, and trained the last convolution and fully connected layer while freezing the remaining weights of all layers. Model performance was evaluated through a nested 5-fold cross validation. With proper fine tuning, they were able to achieve 74% for both accuracy and recall.

It is impossible to compare approaches in existing studies or compare our work to them since most use different datasets and different ways to evaluate the approaches used. Evaluation include cross-validation that has the risk of including test data during training and can lead to a biased performance, which is undesirable as the goal is to learn models with good generalization ability on unseen data.

3. Material and methods

3.1. Dataset

The dataset for this work consists of meningioma cases from patients who were admitted to the Neuroradiology and Neurosurgery units of Padua University Hospital. Sequences included were routine MRI scans (although not all are available for every case) with all of them being anonymized. Tumor segmentation masks were manually annotated. In this work, segmentation masks that were provided on T1-CE sequences were used.

Previous work that used data from the same source for this study noted that the data was highly heterogeneous. Therefore, to limit heterogeneity, cases used in this study were those acquired from 3 Tesla MRI scanners only. In total, the dataset consisted of 105 cases 78 WHO Grade I, 23 WHO Grade II, and 4 WHO Grade III with both T1-CE and ADC maps modalities present. These modalities were included to observe performance improvement compared to the work by Wodzinski et al. (2020). With this distribution, it was apparent that the data have to be grouped into low (WHO Grade I) and high grade (WHO Grade II/Grade III). The number of WHO Grade III cases was not sufficient for validation in training the model. Figure 2 compares the data distribution between this study and previous work.

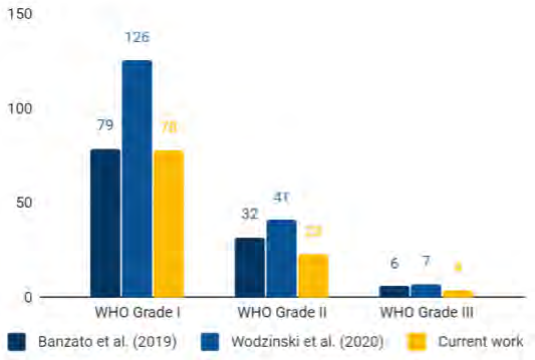


Figure 2: Dataset distribution in the current study and previous work.

There were 3 different scanner models used: Philips Ingenia, Philips Ingenia CX, and General Electric SIGNA Pioneer with Philips Ingenia being used for the the majority of the cases (103 cases). However, similar to previous work, the acquisition protocols varied and it is difficult to find a common ground. For ADC maps, pixel spacing ranges from 0.72mm to 0.98mm and slice thickness is either 3mm or 4mm. The minimum intensity is 0 except for one case where it is -256, while the maximum intensity is mostly 4095, except for two cases being 3.87 and 7320. and For T1-CE, pixel spacing varies from 0.21mm to 1.1mm while slice thickness is from 0.47mm to 4mm. The minimum intensity is 0, while the maximum intensity ranges from 377 to 11568 with an average of 1659. Refer to Table 3 in the Appendix for detailed information of the dataset.

3.2. Data preparation

We have seen that the dataset is not uniform not only inter-patient but also intra-patient (between modalities) due to the different acquisition protocols. Preprocessing is required as model performance is affected by the input data especially in deep learning. The following steps are general pre-processing to both deep learning and machine learning approach. Each approach may require further different pre-processing steps. Since tumor masks were available for T1-CE images, we registered ADC maps to its corresponding T1-CE sequences. This step can be easily applied to other modalities if they need to be included for experiments. In this case, segmentation masks of T1-CE can then be used for ADC maps or other modalities too. This also avoids the need to register segmentation masks where problems may arise (either artifacts appear or masks completely disappear) due to lesion masks not being continuous or present in consequent slices, as previously noted. With this scenario, T1-CE was the reference image in the registration.

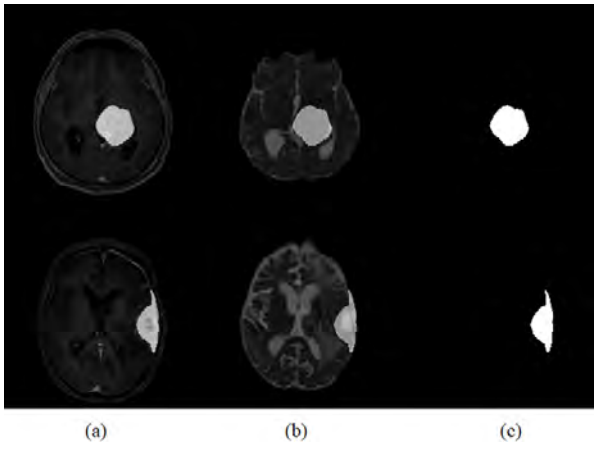


Figure 3: Region of interest on low grade (top) and high grade (bottom) meningioma after pre-processing (left to right): (a) T1-CE sequence, (b) ADC maps, (c) tumor mask. Images shown are slices with the biggest tumor area in the volume.

Image registration uses inverse mapping from the fixed image to the moving image to ensure that every pixel in the fixed image is associated with the moving image. The transformation is defined from the fixed image to the moving image. The registered moving image will have the same characteristic (dimensions and voxel spacing) as the fixed image. Considering this in our application, we performed the general preprocessing steps to T1-CE sequences as other modalities will later be registered to them. Sequences are highly anisotropic, especially for T1-CE where slice thickness varies in a wide range. Voxel resampling was

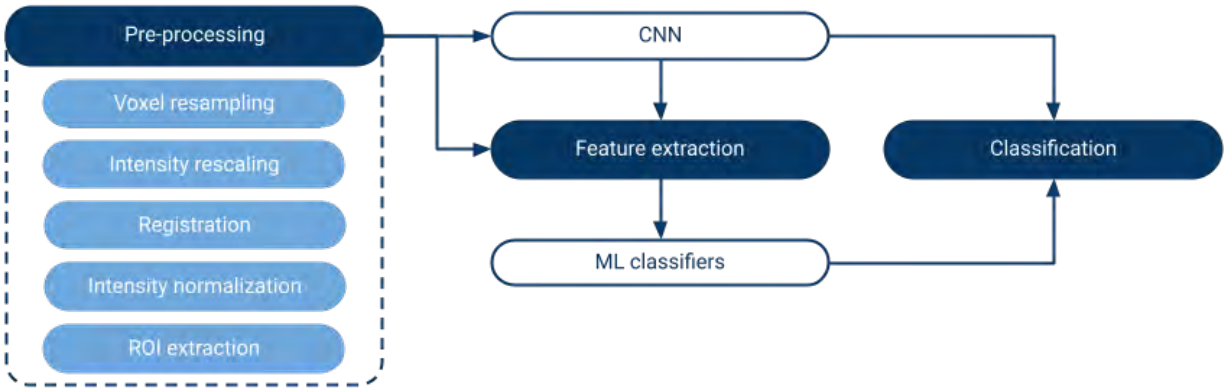


Figure 4: An overview of workflow in this study.

performed to unit spacing of 1x1x1mm^3. Intensity distribution also varies as different scanners were used. In this general pre-processing, intensities were linearly rescaled to [0,255]. For registration with ADC maps, T1-CE sequences were skull stripped. ADC maps were registered to the pre-processed T1-CE images using a rigid transformation (with 6 degrees of freedom). Lastly, in each approach, only slices where tumor is present was selected. The region of interest was extracted by creating a bounding box on slices where tumor is present. Figure 4 presents an overview of the workflow in this study.

3.3. Machine learning approaches

3.3.1. Data pre-processing

Additional pre-processing for the machine learning approach was normalizing images centered at the mean and standard deviation. There are two types of ROI for this approach: (1) using all tumor slices, and (2) a 3-channel image composing of the slice with the largest tumor area plus one slice before and after it. The second type of ROI was used for deep learning that will be further explained. Therefore, for comparison this ROI was also applied in the machine learning approach.

3.3.2. Feature extraction

Radiomics features were extracted as hand-crafted features. 2D features were extracted considering that tumor masks were annotated slice by slice, and to compare with deep learning approaches where 2D inputs and network are being used. Filters were applied to have various types of images to derive radiomics features. The images used were: original, exponential, gradient, logarithmic, 2D local binary pattern (LBP), square, square root, and wavelet images (refer to Figure 5). 3D features were also extracted for comparison when all tumor slices are being used, to see the model performance with supposedly more information of the tumor in the volume.

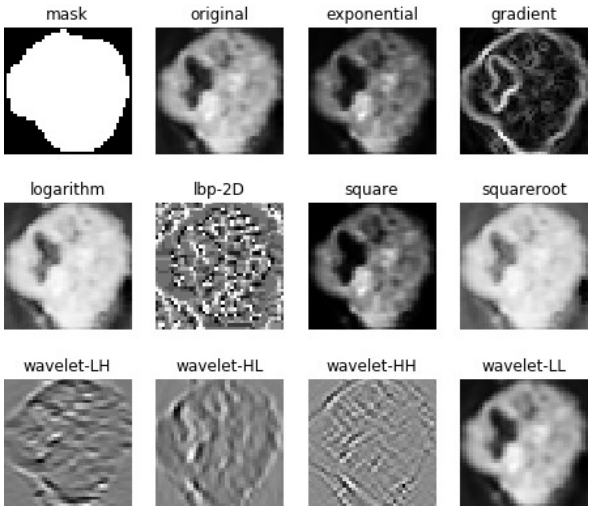


Figure 5: Different images from T1-CE sequence of a low grade meningioma for radiomics features to be extracted from. Images shown are slices with the biggest tumor area in the volume.

The following features were extracted: first order statistics, features based on gray level co-occurrence matrices, gray level run length matrices, gray level size zone matrices, neighboring gray tone difference matrices, and gray level dependence matrices. In both ROI types, a total of 1023 2D features were extracted from each MRI sequence. For features including 3D features, there were 1395 features extracted.

3.3.3. Feature selection

Generally, feature vectors should not be larger than the number of samples. The contrary usually leads to decreasing quality of the model. Redundant features or those with strong correlation to each other that can also decrease model performance should not be included. Feature engineering is performed to retain features that are the most predictive to the target output. Feature selection for this study was a two stage process, with noting that the steps described are only applied to the training data in every cross-validation fold. At first, LASSO

regression with L1-regularization was used to select features. L1-regularization introduces the absolute value of magnitude of feature coefficients (or weights) as a penalty to the loss function being computed, which is mean squared error. With LASSO regression, less important features are shrunk to zero and so we keep the features with non-zero coefficient. Prior to this step, constant and highly correlated features (absolute Pearson correlation coefficient > 0.9) were removed. Features were normalized (centered to mean and standard deviation) with respect to the training data only, and the same transformation was then used for the corresponding validation data. For every complete cross-validation (since nested CV was used), a majority voting was performed to select the final top performing features. The selection process was applied for each MRI sequence. A combination of selected features was also implemented. Different settings resulted in different sets of selected features, refer to Table 4 in Appendix for detailed information. For 2D features from selected slices as the ROI, 14 and 11 features were selected from T1-CE and ADC map each.

3.3.4. Machine learning classifiers

Different linear and non-linear classifiers were investigated: logistic regression, support vector machines, and random forest. Logistic regression models the probability of a certain class by using logistic (sigmoid) function to map data points between 0 and 1 and gives classification based on a certain decision threshold, which is 0.5 for binary classification. Penalty loss or the regularization parameter was included as the hyperparameter to fine-tune. Random forest is an ensemble of decision trees built from random subset of features. Prediction is generated from a majority voting of the predicted class given by the decision trees. Hyperparameters included for fine-tuning are maximum depth of the tree and number of trees. Support vector machines find the hyper-plane that can separate features or data points into classes with a maximum margin. It is called the optimal separating hyper-plane, which acts as the decision boundary for classification. Hyperparameters to fine-tune included regularization parameter and different type of kernels (linear, radial-basis function, and polynomial).

3.4. Deep learning approach

3.4.1. Data pre-processing

Since transfer learning with fine-tuning from pre-trained networks that were trained on ImageNet was the strategy employed for deep learning, input images have to be processed accordingly. Pre-trained ImageNet models are trained on a huge dataset with their own characteristics to classify 1000 classes of images. They have specified certain input criteria (usually shape and normalization parameters) for fine tuning the model or transfer learning. Different pre-trained models were observed: ResNet-18, ResNet-50, and DenseNet121. ResNet-18 was the chosen model for further experiments because of its performance on the validation data. The input criteria they have is a 3-channel image (RGB) with a minimum shape of (224,224) and intensity range between [0,1]. For each patient, the ROI with the largest tumor area, and the following slices before and after it were selected. These three slices make up the 3-channel image requirement for the input of the network. They were reshaped to the minimum size and normalized to the intensity required, in this case using mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225). Figure 6 shows an example of input images to the network.

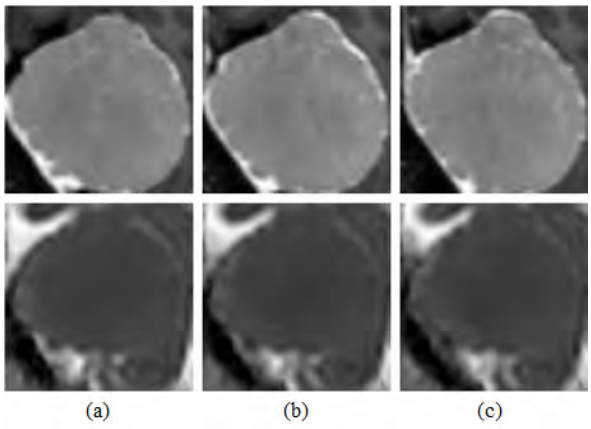


Figure 6: An example of input images for the pre-trained network of T1-CE (top) and ADC map (bottom) sequences. Three slices are selected to have a 3-channel image. The middle slice in the image (b) is the slice with the biggest tumor area in the volume, whereas (a) and (c) are slices consequent it.

Further processing steps were data augmentation and oversampling, which were performed when generating a batch input for the network and only applied to the training data. They are necessary to limit over fitting that can easily happen when using small datasets. Augmentation included random horizontal and vertical flipping, and random rotation between -30° and 30°. A typical sampling technique used to handle data imbalance is oversampling (Buda et al., 2018). The goal is to generate a balanced data class on every batch and hence balance weights update in the network during training. A custom oversampler was created by sampling cases belonging to the majority class exactly once. Cases from the minority class were oversampled to balance the majority cases within a batch.

3.4.2. Convolutional neural network

The chosen model for experiments in this study, ResNet-18, consists of 5 convolutional blocks (with filter size 7x7 for the first block and size 3x3 for the remaining) (He et al., 2016). Transfer learning with fine

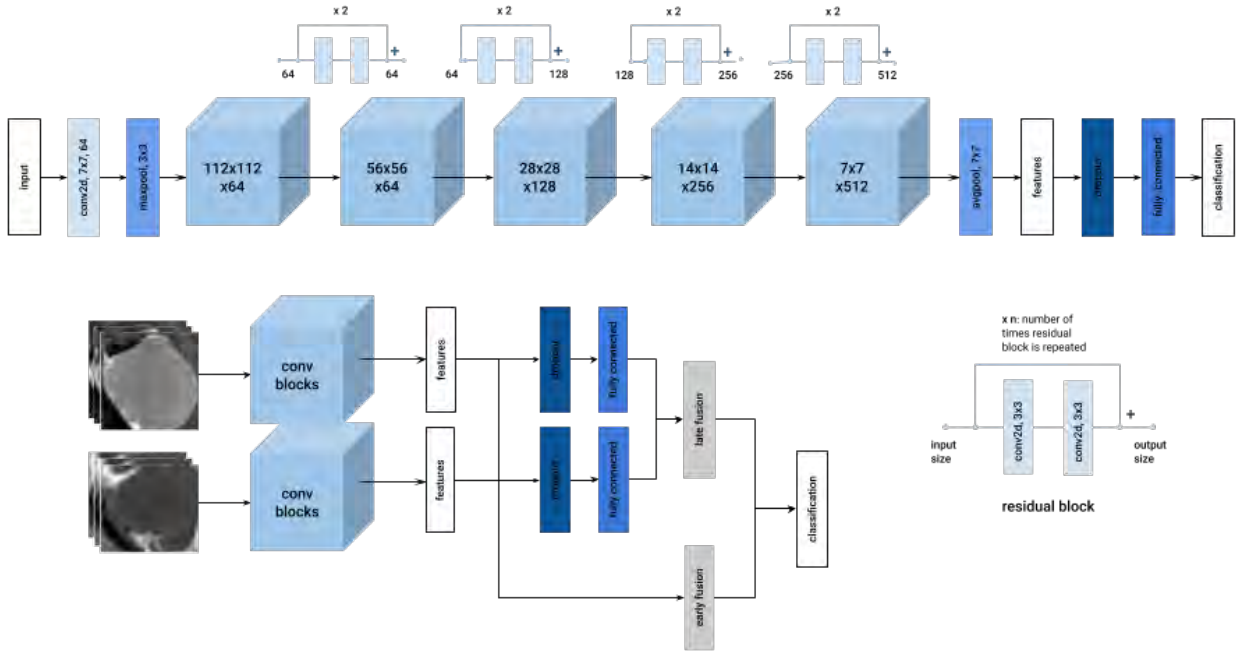


Figure 7: An overview of ResNet-18 architecture. The upper figure is the model used to train single MRI weighting. Two different strategies were employed when using multiple MRI weightings for input: (1) early fusion where features are combined before the fully connected layer, (2) late fusion where features are combined at the decision level.

tuning was used by modifying the output for binary classification. Training different layers were investigated, which are fully connected layer and last convolutional layer. The hidden layers other than these layers were frozen, meaning that pre-trained weights of the corresponding layers were not updated during training. Since the model is already pre-trained, training with a high learning rate is not necessary especially when our data is limited, as it will either lead to overfitting or poor generalization. The learning rate were set to $1e^{-4}$ for the fully connected layer and $1e^{-6}$ for the convolutional layer. With augmentation, overfitting was hardly observed on the training data so we did not decay the learning rate with scheduling. Adding a dropout layer was also observed. Adam optimizer was used considering its stability. Weighted binary cross entropy was the loss function used as our task was a binary classification. Although class distribution was made to be balanced within a batch by oversampling, a slightly higher weight was given to the minority class seeing that class imbalance is present in real case. Loss, sensitivity, and f1-score on the validation data were observed for best model selection in every validation fold.

Each MR weighting was passed to the network and trained separately. CNN was also used as a feature extractor to extract high-level or deep features. The same pipeline as hand-crafted features for feature engineering and classification using machine learning classifiers was adopted. Deep features were extracted at two levels: at feature level (from the convolutional layer, before passed into the fully connected layer), and at decision level (after the fully connected layer). They can be combined in methods known as early fusion, where features are combined at the feature level, and late fusion, where features are combined at the decision level. Features from each modality were combined and passed into machine learning classifiers. For early fusion, with the feature engineering adopted, 15 and 9 features were selected from 512 convolutional features of T1-CE and ADC maps sequences each. For late fusion, strategies included concatenation (and later classified with machine learning classifiers), averaging and weighting. Classification for averaging and weighting in late fusion was performed by passing the decision into a softmax layer. Prediction was obtained by taking the class with the higher probability. Refer to Figure 7 for the architecture of the network used.

3.5. Validation, metrics, and statistical analysis

Nested stratified 5-fold cross-validation (without replacement) was used for all training procedures in both approaches. Nested meaning that in the training set of each validation fold, another cross-validation is performed. The data was randomly divided this way and used for both approaches for comparison. When the dataset is limited in size, cross-validation is a common method to evaluate model performance. It is not intended for selection of a model or best parameters, but rather estimate generalization performance on unseen data. Data is split into k-folds, where one fold is be-

ing used for validation and the other $k-1$ folds are used for training. The process is repeated k times until all data is used as a hold-out data. Stratification equally distributes data based on the class ratio. It keeps the class ratio on each fold similar to the class ratio of the whole dataset. Higher number of folds gives a less biased measure on model performance but at the same time can lead to higher variance due to data subset being different in every fold. Five or ten folds are commonly used, as they are known to have a low bias with moderate variance. Since the task in this study is a binary classification, we focus on the confusion matrix and the following metrics: precision, sensitivity, specificity, and accuracy. Statistical significance test was performed to compare model performance. Paired McNemar test was used on 5-fold cross validation scores.

3.6. Implementation details

This work was implemented using Python programming language with PyTorch as the deep learning framework. Automated skull stripping was done through the software available (ROBEX) with a method proposed by Iglesias et al. (2011). Image registration was performed using SimpleElastix, an extension of SimpleITK in Python that includes elastix C++ library. Radiomics feature were extracted using pyradiomics library for Python for reproducibility. The hardware used for training the network is NVIDIA Quadra P2000 with 32GB RAM.

4. Results

Results using traditional and deep learning approach are summarized in Table 1. The average and standard deviation of metrics (precision, sensitivity/recall, specificity, and accuracy) across all validation folds are reported. The same training and validation data in each fold was used for both approaches in order to be able to compare them. Validation contains 20% data of the complete dataset, so there were 21 validation cases that included 5 to 6 high-grade tumor cases in every fold. We compare: (1) classifiers, (2) different ROI selection (all tumor slices vs. selected tumor slices), (3) feature type (2D vs. 3D), and (3) MRI weightings (T1-CE, ADC maps, or combination of both).

CNN as the main approach in this study was the reference for performance comparison through statistical tests. The network was separately trained using each MRI weighting. Using ADC maps, CNN was able to reach 0.71±0.19 sensitivity and 0.70±0.11 accuracy. Results were lower for T1-CE with 0.46±0.32 sensitivity and 0.59±0.05 accuracy. Precision was low in both sequences, 0.27±0.17 for T1-CE and 0.45±0.11 for ADC maps. The average confusion matrix from all 5-fold is summarized in Table 2. By looking at the confusion matrix, it can be seen that the model trained on T1-CE produced higher number of false positives indeed. Overall results on T1-CE was lower than the model trained on ADC maps, but this was not statistically significant (p-value = 0.089 > 0.05).

For machine learning approach, model performance was compared to the end-to-end network trained on ADC maps. Most of deep features models that were fine-tuned were statistically significant to results from CNN. This was not the case for tree-based classifiers nor models using radiomics features. It was found that tree-based models always had low sensitivity, averaging mostly lower than 0.5. SVM with non-linear kernels did not show consistency under different experiment settings. Classifiers with the best performance on various settings were SVM with a linear kernel and logistic regression with L2-regularization.

T1-CE		Prediction	
		Grade I	Grade II/III
Actual	Grade I	2.6±1.9	2.8±1.5
	Grade II/III	5.8±1.6	9.8±1.9
ADC maps		Prediction	
		Grade I	Grade II/III
Actual	Grade I	3.8±0.8	1.6±1.1
	Grade II/III	4.8±1.3	10.8±1.8

Table 2: Confusion matrix of end-to-end classification using different MR sequences, averaged across 5-fold cross validation

Note that significance reported in the following are results from models trained on ADC maps, considering their higher performance than on T1-CE. Significance of deep features still holds when compared to hand-crafted radiomics, although with different classifiers. High-level features using SVM and logistic regression were significant to 2D radiomics. Deep features with logistic regression was significant to 2D radiomics extracted from all tumor slice, and deep features with SVM was significant to 3D radiomics extracted from all tumor slice. None of ROI selection (selected tumor slice vs. all tumor slice), radiomics feature type (2D vs. 3D), MRI weighting used (T1-CE vs. ADC maps and single vs. multiple weighting), and fusion methods (early vs. late) showed statistical significance in the model performance.

The highest sensitivity and accuracy in the experiments was obtained in deep features combined from T1-CE and ADC maps using logistic regression, with 0.97±0.07 sensitivity (indicating very low number of false negatives) and 0.87±0.06 accuracy. This is significant to results from end-to-end CNN using ADC maps (p-value = 0.003 < 0.05). Precision is at 0.69±0.13 while specificity is 0.83±0.1. Deep features from ADC maps using logistic regression have slightly better results in terms of lower false positives, indicated by 0.72±0.19 precision. Sensitivity is 0.89±0.15, specificity is 0.85±0.13, and accuracy is 0.86±0.07, also with significance (p-value = 0.004 < 0.05).

Features	Classifier	MRI weighting	Precision	Sensitivity	Specificity	Accuracy
2D Radiomics	SVM	T1-CE	0.45±0.13	0.59±0.23	0.73±0.12	0.7±0.04
		ADC map	0.47±0.14	0.74±0.22	0.69±0.14	0.7±0.11
		T1-CE, ADC map	0.53±0.14	0.65±0.18	0.8±0.1	0.76±0.09
2D Radiomics	Logistic Regression	T1-CE	0.47±0.13	0.7±0.11	0.71±0.13	0.7±0.09
		ADC map	0.5±0.16	0.77±0.22	0.72±0.12	0.73±0.1
		T1-CE, ADC map	0.63±0.15	0.81±0.14	0.83±0.07	0.83±0.08
2D Radiomics	SVM	T1-CE	0.5±0.16	0.77±0.17	0.69±0.17	0.71±0.12
		ADC map	0.51±0.18	0.7±0.28	0.74±0.13	0.73±0.1
		T1-CE, ADC map	0.49±0.08	0.72±0.07	0.78±0.16	0.73±0.04
2D Radiomics	Logistic Regression	T1-CE	0.52±0.09	0.7±0.23	0.77±0.1	0.75±0.04
		ADC map	0.48±0.1	0.88±0.18	0.68±0.04	0.73±0.06
		T1-CE, ADC map	0.44±0.08	0.68±0.07	0.75±0.2	0.7±0.06
3D Radiomics	SVM	T1-CE	0.43±0.14	0.67±0.25	0.7±0.13	0.69±0.11
		ADC map	0.51±0.15	0.67±0.2	0.73±0.17	0.71±0.11
		T1-CE, ADC map	0.54±0.1	0.63±0.28	0.81±0.1	0.76±0.03
3D Radiomics	Logistic Regression	T1-CE	0.47±0.08	0.65±0.31	0.73±0.16	0.7±0.06
		ADC map	0.58±0.15	0.73±0.19	0.77±0.18	0.76±0.11
		T1-CE, ADC map	0.64±0.13	0.75±0.2	0.83±0.12	0.81±0.06
	CNN	T1-CE	0.27±0.17	0.46±0.32	0.63±0.11	0.59±0.05
		ADC map	0.45±0.11	0.71±0.19	0.69±0.09	0.70±0.11
Early fusion	SVM	T1-CE	0.64±0.12	0.79±0.18	0.83±0.09	0.82±0.06
		ADC map	0.74±0.24	0.82±0.2	0.86±0.13	0.85±0.07
		T1-CE, ADC map	0.73±0.17	0.81±0.18	0.87±0.1	0.86±0.05
Early fusion	Logistic Regression	T1-CE	0.6±0.07	0.85±0.08	0.79±0.09	0.81±0.06
		ADC map	0.72±0.19	0.89±0.15	0.85±0.13	0.86±0.07
		T1-CE, ADC map	0.69±0.13	**0.97±0.07**	0.83±0.1	**0.87±0.06**
Late fusion	SVM	T1-CE, ADC map	0.47±0.12	0.71±0.2	0.72±0.08	0.71±0.08
Late fusion	Logistic Regression	T1-CE, ADC map	0.44±0.1	0.63±0.11	0.72±0.08	0.7±0.05
Late fusion	Averaging	T1-CE, ADC map	0.44±0.1	0.63±0.15	0.73±0.05	0.7±0.05
Late fusion	Weighting	T1-CE, ADC map	0.54±0.07	0.75±0.18	0.78±0.05	0.77±0.06

Table 1: Performance of machine learning classifiers in 5-fold cross-validation with various settings. Precision, sensitivity, and specificity reported are scores of the positive class (high grade meningioma). Features in italic are using selected tumor slices for ROI, otherwise all tumor slices are used.

5. Discussion

This study presents the classification of low and high-grade meningioma from MR images using classical machine learning and deep learning. With the experiments that were conducted, assumptions on the results obtained can be made.

Although there has been extensive research on meningioma prediction using machine learning, there is no standardized pipeline nor exact features that are widely accepted to be the most predictive in differentiating meningioma grades, which is due to most studies being conducted in limited data and single-center studies as previously highlighted. We also mark that our machine learning approach serves as a comparison for deep learning. Therefore, we did not focus on selecting which features that are best used for meningioma grade classification. A small note on feature engineering, it has to be strictly performed only on training data to assure generalization on unseen data. Otherwise, the distribution of the hold-out data will already be encoded during training, leading to overfitting or a biased performance during validation. Several settings in our traditional machine learning pipeline were observed: ROI selection, feature type, classifiers, and MRI sequences.

In a small dataset, the common strategy to evaluate model performance is cross-validation. Since there is no independent validation dataset, with cross-validation the data is divided into training and validation set on every fold until all data have been used for validation. During training, the data is again split into training and validation to observe underfitting or overfitting. Overall, we can see results vary across validation folds. Data were randomly divided in a stratified manner, ensuring that each fold contains the same ratio of each class. Even though class distribution was similar among folds, the underlying data distribution they hold could be dif-

ferent, since high-grade meningioma include atypical or WHO Grade II and anaplastic or WHO Grade III tumors. The number of anaplastic cases was very low (only four cases available) so they were not present in all folds. This can also be seen during the feature selection step, where training data across validation folds produced different selected feature sets due to the different data distribution.

For ROI selection, two methods were used: using all tumor slices and selected three slices in the volume. The selected three slices consisted of the tumor lesion with the largest area in the volume (as the middle slice) and slices before and after it. This ROI was used for deep learning, and for comparison, it was also applied in the traditional approach. Whereas for feature extraction, two types of features were extracted: 2D (similar to deep learning) and 3D. From the results obtained, none of these settings were significant to model performance. Instead, it is interesting to note that classification results using 2D features from a limited ROI (as only three slices were used) are comparable to 3D features, which were extracted from all tumor slices therefore has more information on the tumor in the volume, showing that information from the lesion with the largest area could be sufficient.

We experimented with simple to more complex machine learning classifiers. Linear models included logistic regression and support vector machines with a linear kernel. Non-linear models were random forest and support vector machines with non-linear kernels. We found that non-linear models (radial basis function and polynomial SVM, random forest) were low in sensitivity indicating they were not able to differentiate high grade meningioma well. SVM with a linear kernel and logistic regression were classifiers with the best results in cross-validation under different settings.

The end-to-end network has comparable (for ADC maps) to lower (for T1-CE) results to hand-crafted radiomics with machine learning classifiers. This could be reasonable due to the limitation on the number of cases in our study. Deep learning is usually expected to yield higher performance than traditional machine learning when the number of cases is large. In this case, manual feature engineering may become costly in terms of time and computation, therefore making deep learning preferred. Deep learning tends to work better on a large number of data as it involves tuning a huge number of parameters to reach an optimal solution. More data is linked to better generalization ability of the network. On the contrary, when data are limited, if the network is not yet pre-trained for the given task on a large dataset, generalization can be difficult to conclude and traditional machine learning might be more useful. In our case, deep learning is comparable to the traditional approach considering high-level features that are learned by the network (and not captured in hand-crafted features) could be useful for prediction.

Transfer learning from pre-trained models on ImageNet with hyper-parameters fine-tuning was the strategy employed for deep learning. Training the network from scratch resulted in overfitting, and a similar result was observed without performing data augmentation and oversampling on the training data. Augmentation adds more diversity to the data by generating different training data on every batch, reducing the chance to overfit. The oversampling method only oversamples the minority class, which is high-grade meningioma, and samples low-grade meningioma cases only once. It also generates a balanced class data in each batch, to ensure that weights are updated in a balanced manner. Several networks were observed for experiments, ResNet-18, ResNet-50, DenseNet-121. We selected ResNet-18 as our chosen model because of its performance on validation data. For other models with a higher number of parameters (as they are deeper networks), the input data used might not be appropriate to update and adjust the weights and hence not learn well. This complies with how parameter complexity can reduce the quality of a model.

Transfer learning generally includes freezing most parts of the pre-trained model and only training certain layers, given that they are trained on similar tasks. We found that training the last convolutional layer along with the last fully connected layer performed better than training the fully connected layer alone. Training the convolutional layers enables the model to learn features that could be more relevant in the current dataset for the given task. We added a dropout layer before the fully connected layer. Dropout is a regularization technique used to reduce overfitting, by disabling random neurons (setting their weights to zero) on a single forward/backward pass on a batch data. It removes simple dependencies between neurons. Even though it is mainly used to address the overfitting problem, in our case dropout improved the performance in the validation data. Generalization was worse without dropout, even though the model could be fine-tuned to an optimal result during training. The reason could be that dropout forces the model to learn more robust and useful features along with different random subsets of other neurons and thus increases generalization performance.

In addition to training the network for end-to-end classification, we also used it as a feature extractor. For single MRI weighting, features were extracted before the fully connected layer, generating what is called high-level or deep features. Results with deep features were overall higher than radiomics features (also indicated by the statistical significance), implying that they could be more predictive in differentiating low and high-grade meningioma. For multiple MRI weightings, we combined features with two known fusion methods: early fusion and late fusion. Early fusion combines features at the feature extraction level, which generates deep features. These deep features then undergo feature

engineering similar to radiomics, followed by training machine learning classifiers. Late fusion combines features at the decision level. For this method, different classification strategies were applied: concatenation and training machine learning classifiers, averaging, weighting. In our study, early fusion had higher results than late fusion.

From our experiments, the use of different MRI weightings had variable results. In most settings, ADC maps provided higher results than T1-CE sequences. For this reason, a higher weight was given to the decision from ADC maps for weighting in the late fusion method. Combining multiple MRI weightings also showed different findings. In comparison to using T1-CE alone, using features from both weightings slightly improved the results though they did not outperform the performance from using ADC maps in most cases. Through the statistical test evaluated, using different MRI weightings was not significant to the model performance.

We compare our results to a previous study that used the dataset from the same source, although different subsets were used. Wodzinski et al. (2020) worked using T1-CE sequences in a total of 174 cases. Even though the data was highly heterogeneous, they were able to record sensitivity and accuracy at the level of 74%. The data used in this study was aimed to reduce the heterogeneity by using sequences only from 3T scanners. Similar results were difficult to be achieved using an end-to-end network with these data, with one obvious reason being a smaller number of training data. As they stated, ADC maps might be able to provide better information in the task of classifying low and high-grade meningioma, which were shown in our results. Deep features with early fusion showed improved results to previous work.

6. Conclusions

This study evaluated convolutional neural networks and traditional machine learning for the prediction of meningioma histological grading from MR images. The end-to-end classification was comparable to traditional machine learning. High-level features, which were extracted from separately trained models, showed superior performance to hand-crafted radiomics. Hand-crafted features are exact in the way of derivation, which is an advantage for feature engineering, whereas high-level features are derived from more complex models. CNNs can encode more complex patterns the deeper the layer goes. High-level features might contain information that is more useful to distinguish low and high-grade meningioma, seeing that results were significant to radiomics features. Even though the task for this study was a binary classification, it depended on tumor segmentation as meningiomas do not appear throughout the whole volume. The quality of segmentation masks thus affects the classification task, in which our study still has a drawback from using manually annotated tumor masks. For future studies, the use of multiple MR weightings should be explored more thoroughly, since routine MR sequences are usually performed for surveillance in meningioma. A larger dataset would most likely help improve the model too.

7. Acknowledgments

I would like to thank my supervisor; Henning Müller, PhD, Prof. for the support, guidance, useful suggestions and encouragement throughout this work. I would also like to thank Marek Wodziński, Tommaso Banzato DVM, PhD, and Dr. Manfredo Atzori for the discussions during the work in HES-SO Valais-Wallis. I am very grateful.

References

Aizer, A.A., Arvold, N.D., Catalano, P., Claus, E.B., Golby, A.J., Johnson, M.D., Al-Mefty, O., Wen, P.Y., Reardon, D.A., Lee, E.Q., et al., 2014. Adjuvant radiation therapy, local recurrence, and the need for salvage therapy in atypical meningioma. Neuro-oncology 16, 1547–1553. doi:10.1093/neuonc/nou098.

Aslan, K., Gunbey, H.P., Tomak, L., Incesu, L., 2018. The diagnostic value of using combined mr diffusion tensor imaging parameters to differentiate between low-and high-grade meningioma. The British journal of radiology 91, 20180088. doi:10.1259/bjr.20180088.

Banzato, T., Causin, F., Della Puppa, A., Cester, G., Mazzai, L., Zotti, A., 2019. Accuracy of deep learning to differentiate the histopathological grading of meningiomas on mr images: a preliminary study. Journal of Magnetic Resonance Imaging 50, 1152–1159. doi:10.1002/jmri.26723.

Buda, M., Maki, A., Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249–259. doi:10.1016/j.neunet.2018.07.011.

Buerki, R.A., Horbinski, C.M., Kruser, T., Horowitz, P.M., James, C.D., Lukas, R.V., 2018. An overview of meningiomas. Future Oncology 14, 2161–2177. doi:10.2217/fon-2018-0006.

Chen, C., Guo, X., Wang, J., Guo, W., Ma, X., Xu, J., 2019. The diagnostic value of radiomics-based machine learning in predicting the grade of meningiomas using conventional magnetic resonance imaging: A preliminary study. Frontiers in Oncology 9, 1338. doi:10.3389/fonc.2019.01338.

Chollet, F., 2016. Xception: Deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357 .

Claus, E.B., Bondy, M.L., Schildkraut, J.M., Wiemels, J.L., Wrensch, M., Black, P.M., 2005. Epidemiology of intracranial meningioma. Neurosurgery 57, 1088–1095. doi:10.1227/01.NEU.0000188281.91351.B9.

Coroller, T.P., Bi, W.L., Huynh, E., Abedalthagafi, M., Aizer, A.A., Greenwald, N.F., Parmar, C., Narayan, V., Wu, W.W., Miranda de Moura, S., et al., 2017. Radiographic prediction of meningioma grade by semantic and radiomic features. PLoS One 12, e0187908. doi:10.1371/journal.pone.0187908.

Gillies, R.J., Kinahan, P.E., Hricak, H., 2016. Radiomics: images are more than pictures, they are data. Radiology 278, 563–577. doi:10.1148/radiol.2015151169.

Goldbrunner, R., Minniti, G., Preusser, M., Jenkinson, M.D., Sallabanda, K., Houdart, E., von Deimling, A., Stavrinou, P., Lefranc, F., Lund-Johansen, M., et al., 2016. Eano guidelines for the diagnosis and treatment of meningiomas. The Lancet Oncology 17, e383–e391. doi:10.1016/S1470-2045(16)30321-7.

Hamerla, G., Meyer, H.J., Schob, S., Ginat, D.T., Altman, A., Lim, T., Gihr, G.A., Horvath-Rizea, D., Hoffmann, K.T., Surov, A., 2019. Comparison of machine learning classifiers for differentiation of grade 1 from higher gradings in meningioma: A multicenter radiomics study. Magnetic resonance imaging 63, 244–249. doi:10.1016/j.mri.2019.08.011.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Huang, R.Y., Bi, W.L., Griffith, B., Kaufmann, T.J., la Fougère, C., Schmidt, N.O., Tonn, J.C., Vogelbaum, M.A., Wen, P.Y., Aldape, K., et al., 2019. Imaging and diagnostic advances for intracranial meningiomas. Neuro-oncology 21, i44–i61. doi:10.1093/neuonc/noy143.

Iglesias, J.E., Liu, C.Y., Thompson, P.M., Tu, Z., 2011. Robust brain extraction across datasets and comparison with publicly available methods. IEEE transactions on medical imaging 30, 1617–1634. doi:10.1109/TMI.2011.2138152.

Kassner, A., Thornhill, R., 2010. Texture analysis: a review of neurologic mr imaging applications. American Journal of Neuroradiology 31, 809–816. doi:10.3174/ajnr.A2061.

Ke, C., Chen, H., Lv, X., Li, H., Zhang, Y., Chen, M., Hu, D., Ruan, G., Zhang, Y., Zhang, Y., et al., 2019. Differentiation between benign and nonbenign meningiomas by using texture analysis from multiparametric mri. Journal of Magnetic Resonance Imaging doi:10.1002/jmri.26976.

Komotar, R.J., Iorgulescu, J.B., Raper, D.M., Holland, E.C., Beal, K., Bilsky, M.H., Brennan, C.W., Tabar, V., Sherman, J.H., Yamada, Y., et al., 2012. The role of radiotherapy following gross-total resection of atypical meningiomas. Journal of neurosurgery 117, 679–686. doi:10.3171/2012.7.JNS112113.

Lambin, P., Leijenaar, R.T., Deist, T.M., Peerlings, J., De Jong, E.E., Van Timmeren, J., Sanduleanu, S., Larue, R.T., Even, A.J., Jochems, A., et al., 2017. Radiomics: the bridge between medical imaging and personalized medicine. Nature reviews Clinical oncology 14, 749–762. doi:10.1038/nrclinonc.2017.141.

Laukamp, K.R., Pennig, L., Thiele, F., Reimer, R., Görtz, L., Shakirin, G., Zopfs, D., Timmer, M., Perkuhn, M., Borggrefe, J., 2020. Automated meningioma segmentation in multiparametric mri. Clinical Neuroradiology , 1–10doi:10.1007/s00062-020-00884-4.

Laukamp, K.R., Shakirin, G., Baeßler, B., Thiele, F., Zopfs, D., Hokamp, N.G., Timmer, M., Kabbasch, C., Perkuhn, M., Borggrefe, J., 2019a. Accuracy of radiomics-based feature analysis on multiparametric magnetic resonance images for noninvasive meningioma grading. World neurosurgery 132, e366–e390. doi:10.1016/j.wneu.2019.08.148.

Laukamp, K.R., Thiele, F., Shakirin, G., Zopfs, D., Faymonville, A., Timmer, M., Maintz, D., Perkuhn, M., Borggrefe, J., 2019b. Fully automated detection and segmentation of meningiomas using deep learning on routine multiparametric mri. European radiology 29, 124–132. doi:10.1007/s00330-018-5595-8.

Louis, D.N., Perry, A., Reifenberger, G., Von Deimling, A., Figarella-Branger, D., Cavenee, W.K., Ohgaki, H., Wiestler, O.D., Kleihues, P., Ellison, D.W., 2016. The 2016 world health organization classification of tumors of the central nervous system: a summary. Acta neuropathologica 131, 803–820. doi:10.1007/s00401-016-1545-1.

Lu, Y., Liu, L., Luan, S., Xiong, J., Geng, D., Yin, B., 2019. The diagnostic value of texture analysis in predicting who grades of meningiomas based on adc maps: an attempt using decision tree and decision forest. European radiology 29, 1318–1328. doi:10.1007/s00330-018-5632-7.

Ostrom, Q.T., Cioffi, G., Gittleman, H., Patil, N., Waite, K., Kruchko, C., Barnholtz-Sloan, J.S., 2019. Cbtrus statistical report: Primary brain and other central nervous system tumors diagnosed in the united states in 2012–2016. Neuro-Oncology 21, v1–v100. doi:10.1093/neuonc/noz150.

Park, Y.W., Oh, J., You, S.C., Han, K., Ahn, S.S., Choi, Y.S., Chang, J.H., Kim, S.H., Lee, S.K., 2019. Radiomics and machine learning may accurately predict the grade and histological subtype in meningiomas using conventional and diffusion tensor imaging. European radiology 29, 4068–4076. doi:10.1007/s00330-018-5830-3.

Wang, C., Kaprealian, T.B., Suh, J.H., Kubicky, C.D., Ciporen, J.N., Chen, Y., Jaboin, J.J., 2017. Overall survival benefit associated with adjuvant radiotherapy in who grade ii meningioma. Neuro-oncology 19, 1263–1270. doi:10.1093/neuonc/nox007.

Watanabe, Y., Yamasaki, F., Kajiwara, Y., Takayasu, T., Nosaka, R., Akiyama, Y., Sugiyama, K., Kurisu, K., 2013. Preoperative histological grading of meningiomas using apparent diffusion coefficient at 3t mri. European journal of radiology 82, 658–663. doi:10.1016/j.ejrad.2012.11.037.

Wiemels, J., Wrensch, M., Claus, E.B., 2010. Epidemiology and etiology of meningioma. Journal of neuro-oncology 99, 307–314. doi:10.1007/s11060-010-0386-3.

Wodzinski, M., Banzato, T., Atzori, M., Andrearczyk, V., Dicente, Y., Muller, H., 2020. Training a deep neural networks for small and highly heterogeneous mri datasets for cancer grading. submitted to International Conference of the Engineering in Medicine and Biology Society .

Yan, P.F., Yan, L., Hu, T.T., Xiao, D.D., Zhang, Z., Zhao, H.Y., Feng, J., 2017. The potential value of preoperative mri texture and shape analysis in grading meningiomas: a preliminary investigation. Translational oncology 10, 570–577. doi:10.1016/j.tranon.2017.04.006.

Yin, B., Liu, L., Zhang, B.Y., Li, Y.X., Li, Y., Geng, D.Y., 2012. Correlating apparent diffusion coefficients with histopathologic findings on meningiomas. European journal of radiology 81, 4050–4056. doi:10.1016/j.ejrad.2012.06.002.

Zhu, H., Fang, Q., He, H., Hu, J., Jiang, D., Xu, K., 2019a. Automatic prediction of meningioma grade image based on data amplification and improved convolutional neural network. Computational and mathematical methods in medicine 2019. doi:10.1155/2019/7289273.

Zhu, Y., Man, C., Gong, L., Dong, D., Yu, X., Wang, S., Fang, M., Wang, S., Fang, X., Chen, X., et al., 2019b. A deep learning radiomics model for preoperative grading in meningioma. European journal of radiology 116, 128–134. doi:10.1016/j.ejrad.2019.04.022.

Appendix

Parameter	Number of cases
Scanner model	
Philips Ingenia	103
Philips Ingenia CX	1
General Electric SIGNA Pioneer	1
Pixel spacing (mm)	
T1-CE	
0.21	1
0.23	8
0.25	1
0.44	1
0.45	3
0.46	1
0.47	2
0.48	4
0.49	31
0.5	3
0.53	1
0.54	10
0.6	7
1.0	13
1.1	19
ADC maps	
0.72	6
0.82	1
0.87	1
0.9	93
0.96	3
0.98	1
Slice thickness (mm)	
T1-CE	
0.47	5
0.48	2
0.49	5
0.5	1
0.52	1
0.54	17
0.55	1
1.0	1
3.0	7
3.54	2
3.67	1
3.71	1
3.72	2
3.73	5
3.79	1
3.98	1
4.0	52
ADC maps	
3.0	21
4.0	84

Table 3: Data set details

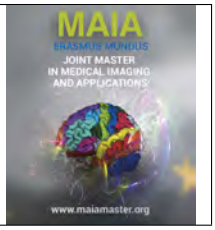
Features
T1-CE
original_glrlm_RunVariance
original_glszm_SizeZoneNonUniformity
original_glszm_SmallAreaEmphasis
exponential_glszm_ZoneVariance
logarithm_firstorder_90Percentile
logarithm_firstorder_Kurtosis
wavelet-HH_glrlm_RunLengthNonUniformityNormalized
wavelet-HH_glszm_SmallAreaHighGrayLevelEmphasis
wavelet-HL_firstorder_Kurtosis
wavelet-HL_firstorder_Median
wavelet-HL_glrlm_HighGrayLevelRunEmphasis
wavelet-LH_firstorder_Minimum
wavelet-LH_firstorder_Skewness
wavelet-LH_glcm_Imc2
ADC maps
original_glrlm_RunVariance
gradient_firstorder_Minimum
lbp-2D_firstorder_Median
wavelet-HH_glszm_HighGrayLevelZoneEmphasis
wavelet-HH_glszm_ZoneEntropy
wavelet-HL_firstorder_Minimum
wavelet-HL_firstorder_Skewness
wavelet-HL_glrlm_HighGrayLevelRunEmphasis
wavelet-LH_firstorder_Skewness
wavelet-LH_glszm_GrayLevelNonUniformityNormalized
wavelet-LH_glszm_ZoneEntropy
T1-CE
original_firstorder_10Percentile
original_glrlm_RunLengthNonUniformity
original_glszm_HighGrayLevelZoneEmphasis
exponential_glszm_SmallAreaEmphasis
lbp-2D_firstorder_InterquartileRange
lbp-2D_firstorder_Median
wavelet-HH_firstorder_Maximum
wavelet-HH_firstorder_Mean
wavelet-HH_firstorder_Skewness
wavelet-HH_glszm_SmallAreaLowGrayLevelEmphasis
wavelet-HL_firstorder_Kurtosis
wavelet-HL_firstorder_Skewness
wavelet-HL_glcm_SumEntropy
wavelet-LH_firstorder_Minimum
ADC maps
original_firstorder_Skewness
original_glcm_Imc1
original_glszm_GrayLevelNonUniformity
exponential_glszm_SmallAreaEmphasis
lbp-2D_firstorder_Median
wavelet-HH_firstorder_Mean
wavelet-HH_glszm_GrayLevelNonUniformityNormalized
wavelet-HL_firstorder_Skewness
wavelet-HL_glrlm_HighGrayLevelRunEmphasis
wavelet-HL_glszm_GrayLevelNonUniformityNormalized
wavelet-LH_gldm_DependenceEntropy
wavelet-LH_gldm_LargeDependenceHighGrayLevelEmphasis
wavelet-LH_glszm_SmallAreaHighGrayLevelEmphasis
wavelet-LH_glszm_SmallAreaLowGrayLevelEmphasis
wavelet-LH_glszm_ZoneEntropy

*T1-CE**
original_ngtdm_Busyness
exponential_glszm_ZoneVariance
gradient_firstorder_Kurtosis
wavelet-HHL_firstorder_Skewness
wavelet-HHL_firstorder_Mean
wavelet-HLH_firstorder_Kurtosis
wavelet-HLH_firstorder_Median
wavelet-HLH_glrlm_GrayLevelNonUniformityNormalized
wavelet-HLH_glszm_HighGrayLevelZoneEmphasis
wavelet-HLH_glszm_SmallAreaLowGrayLevelEmphasis
wavelet-HLL_firstorder_Kurtosis
wavelet-LHH_firstorder_Median
wavelet-LHH_glrlm_HighGrayLevelRunEmphasis
wavelet-LHL_glszm_SizeZoneNonUniformity
wavelet-LLH_firstorder_Skewness
*ADC maps**
original_glcm_Imc1
original_glszm_GrayLevelNonUniformity
wavelet-HHH_glcm_ClusterProminence
wavelet-HHL_firstorder_Mean
wavelet-HLH_firstorder_Maximum
wavelet-HLH_glszm_GrayLevelNonUniformityNormalized
wavelet-HLH_glszm_HighGrayLevelZoneEmphasis
wavelet-HLH_glszm_SizeZoneNonUniformityNormalized
wavelet-HLL_firstorder_Skewness
wavelet-HLL_glrlm_HighGrayLevelRunEmphasis
wavelet-HLL_glszm_SizeZoneNonUniformityNormalized
wavelet-LHH_glszm_GrayLevelNonUniformityNormalized
wavelet-LHL_glcm_SumEntropy
wavelet-LHL_glszm_GrayLevelNonUniformity
wavelet-LHL_glszm_GrayLevelNonUniformityNormalized
wavelet-LLH_glszm_HighGrayLevelZoneEmphasis
wavelet-LLL_glszm_SmallAreaEmphasis

Table 4: Selected radiomics features with different settings. Sequences in italic indicate ROI of all tumor slices, otherwise selected three slices are used. Asterisk represents 3D features, otherwise features are 2D.

Medical Imaging and Applications

Master Thesis, August 2020



Automated Breast Lesion Segmentation in DCE-MRI Based on Deep Learning

Roa'a Khaled, Supervisor: Robert Marti

Universitat de Girona, Girona, Spain

Abstract

Dynamic Contrast Enhanced Magnetic Resonance Imaging (DCE-MRI) is an effective tool for the diagnosis of breast lesions as it is capable of visualizing both physiological tissue characteristics and anatomical structures. Several studies have shown that features extracted from lesions in DCE-MRI are helpful for distinguishing between benign and malignant breast lesions. Therefore, the accurate segmentation of breast lesions in DCE-MRI is an essential step for automated Breast Cancer analysis, diagnosis and treatment follow-up. This task is a challenging problem and an active area of research. However, there are only few studies focusing on breast lesion segmentation in DCE-MRI so far, and most of them are using either semi-automatic methods or traditional machine learning methods.
In this work we propose an automated breast lesion segmentation method for DCE-MRI. Our method is a ROI guided, 3D patch based deep learning framework which is based on a U-Net architecture with ResNet basic blocks. First, a ROI was obtained and used to restrict a balanced patch extraction process, which was proposed in order to address both problems of the class imbalance and confounding regions. Differently from most existing works on this topic, we performed a 3D segmentation instead of 2D. Therefore, our method performs both segmentation and detection at the same time.
Additionally we propose the usage of the voxel-wise standard deviation (std) across the time dimension in order to represent time-intensity variations of each voxel in a less parametric volume. Approximately 4000 balanced Patches of size (32,32,16) were extracted from each case. AdaDelta optimizer along with the binary cross-entropy loss function were used and a 5 fold cross-validation was performed. The dataset we used consisted of 46 cases obtained from the public collection of TCGA-BRCA. Experiments were performed on all 46 cases using pre-contrast, last post-contrast and std volumes as inputs. Dice Similarity Coefficient (DSC) was used to evaluate the obtained segmentation. We obtained a mean dice of 0.645 which demonstrates the effectiveness of our method considering the complexity of the dataset we used and its incomplete annotation (ground truth).

Keywords: Breast lesions, DCE-MRI, lesion segmentation, Deep Learning, 3D U-Net

1. Introduction

Breast cancer (BC) begins with an uncontrolled change and division of cells in the breast, forming a mass (lesion) that can either grow and spread to other parts of the body (as the case of a malignant lesion), or just grow without spreading (as the case of a benign lesion) (Subbhuraam et al., 2014) (ASCO, 2019). It mostly begins in the lobules (milk glands) or in the ducts that connect the lobules to the nipple and in most cases it spreads to nearby lymph nodes, but it can also spread further through the body to areas such as the bones, lungs, liver, and brain. Once the patient is diagnosed with BC, the stage is also defined, which indicates its location, extent of growth, and whether or where it has spread (ACS, 2019-2020) (ASCO, 2019).
It is most easy to treat BC when the lesion is small, however no symptoms normally appear at that stage. A painless lump is the most common physical symptom but BC can spread to underarm lymph nodes causing a lump or swelling even before the original breast lesion grows enough to be noticed. Other signs and symptoms, such as breast pain or heaviness and persistent changes, can also happen; however they are less common. Hence, screening is very crucial for early detection (ACS, 2019-2020) (Siegel et al., 2020).

According to the World Health Organization BC is impacting 2.1 million women each year and causing the greatest number of cancer deaths among women (WHO). As stated by the American Cancer Society (ACS), BC is the most common cancer among women in US (excluding skin cancer) and the second cause of cancer deaths (after lung cancer). According to the ACS estimates of 2020 on US women, 30% of all diagnosed cancer cases will be BC cases and 15% of all cancer deaths will be due to BC (DeSantis et al., 2019) (Siegel et al., 2020). It is also estimated that in 2020 there will be 279,100 new cases of invasive BC diagnosed in both men and women (men 2,620, women 276,480) and an additional 48,530 cases of ductal carcinoma in situ (DCIS) diagnosed in women. Moreover, 42,690 BC deaths (men 520, women 42,170) will occur in 2020 (Siegel et al., 2020). Hence, the spread of BC is one of the main health challenges in the world.
Despite that, statistics have shown that the five-year survival rate in the US has increased from 75% to 91% between 1975-2015, and the five-year survival rate for early diagnosed patients (when the lesion is in the local stage) is 99%, which declines to 86% and 27% in the regional and distant diagnosed stages of BC respectively (Siegel et al., 2020). According to ACS, BC death rate is declining continuously, it has dropped by 40% from 1989 to 2017 (DeSantis et al., 2019). This is mostly due to the early detection of BC and the expanding access to high-quality prevention and treatment services.

Since imaging modalities have been playing a vital role in all phases of BC control (starting from screening and early detection to diagnosis and treatment follow-up), many imaging modalities have been continually developed in order to improve BC outcomes and survival. Each of these modalities has different clinical advantages and disadvantages as well as sensitivity and specificity. Thus, some of these modalities are used for screening purposes due to their efficiency in reaching the masses and their lower cost (such as mammography), while some (such as MRI) are used for diagnostic purposes in order to obtain more detailed evaluations after the detection of BC in screening tests. Other modalities (such as ultrasound) are used for adjunctive evaluation to assist doctors and clinicians obtaining additional confidence in their initial diagnosis (Subbhuraam et al., 2014). Moreover, using hybrid imaging techniques has been proven to improve BC detection (Iranmakani et al., 2020). Furthermore, improvements of the basic techniques used in each of these modalities have been performed throughout the years in order to improve the detection efficiency, due to the fact that cancer is a complex disease with varied pathology (Subbhuraam et al., 2014).
The choice of the modalities and techniques is also affected by the patient's state and stage, the age and the density of the breast tissue (Iranmakani et al., 2020).

Currently, there are 3 clinical breast imaging modalities used for BC detection and diagnosis: 1) Mammography, 2) Ultrasound and 3) MRI. Other breast diagnostic methods also exist, such as: tomosynthesis, elastography, photoacoustics, and optical imaging. However, they have some challenges and complexities which made them less common (Iranmakani et al., 2020).

Mammography is currently the gold standard method for BC screening and early detection. Studies have shown that it has helped increasing early diagnosis and treatment of BC and hence decreasing the mortality rate in screened women by 30% or more (Subbhuraam et al., 2014) (Alzaghal and DiPiro, 2018). However, it is known that conventional mammography is not very sensitive in detecting cancer in dense breast tissues (Subbhuraam et al., 2014) (Iranmakani et al., 2020) (Alzaghal and DiPiro, 2018). According to studies, the sensitivity (true positive rate) of this method declines from 75% to 50% in middle aged patients who has higher breast density (Iranmakani et al., 2020).

Ultrasound imaging is used as an adjunct tool to mammography to detect the location of the suspicious lesion. According to studies, the use of ultrasound as an adjunct tool to mammography improves the diagnostic yield for women with dense breasts and those at higher risk of BC, but at the expense of an increased false positive rate. Additionally, it is not possible to accurately detect lesions using just ultrasound, so it is suggested to complement mammography or other imaging techniques (Subbhuraam et al., 2014).

MRI is widely used for both the early detection and diagnosis of BC (Subbhuraam et al., 2014). It has higher sensitivity than mammography and ultrasound and it improves the yield of screening for higher risk women dramatically, so it is most widely used as a supplemental screening in high-risk women (Iranmakani et al., 2020) (Subbhuraam et al., 2014). According to the American Cancer Society (ACS), it is recommended for high risk patients to have an annual screening using MRI (Siegel et al., 2020).
Besides the higher sensitivity, MRI has higher spatial and temporal resolution and a better signal to noise ratio. It is also effective for evaluating dense breasts, it helps to evaluate inverted nipple, allows the simultaneous evaluation of both breasts, helps to determine whether lumpectomy or mastectomy is the best treatment, and it has no side effects as there is no radiation (Subbhuraam et al., 2014).
However, the widespread use of breast MRI is limited due to the following issues: the increased cost, the high false positive rates and the longer acquisition time (30 min to one hour), which also leads to patients' difficulties in maintaining proper posture. Other limitations include the poor sensitivity for diagnosing ductal carcinoma in situ (DCIS), possibility of not showing all calcifications, and the personal contraindications (including incompatible surgical implants, claustrophobia, contrast allergy, or risk of nephrogenic systemic fibrosis

in patients with renal insufficiency that receive gadolinium contrast) (Subbhuraam et al., 2014) (Iranmakani et al., 2020) (Alzaghal and DiPiro, 2018).
There are specific MRI techniques mostly used to diagnose BC, such as: Diffusion-weighted imaging (DWI) and Dynamic contrast enhanced MRI (DCE-MRI) (Iranmakani et al., 2020).

Recent research and clinical studies have shown the effectiveness of DCE-MRI for the diagnosis of BC due to its capability to visualize both physiological tissue characteristics and anatomical structures, however it is less specific (has more false positives FP) (Zhang et al., 2019a) (Subbhuraam et al., 2014).
In DCE-MR imaging the changes of T1 in tissues are measured over time after the administration of a contrast agent (gadolinium) (Tofts, 2010), so that one scan is acquired before the administration of the contrast agent and one or more scans are acquired after the administration of the contrast agent. The main purpose of DCE-MRI is to observe and quantify the contrast enhancement over time, since the degree of contrast enhancement depends on the regional blood flow, the size and number of blood vessels and their permeability, which are related to cancer tissues (Tofts, 2010). Nevertheless, it is time consuming to evaluate the large amount of information from 4D-DCE MR images for each patient, and it also requires experienced radiologists for the interpretation of those 4D-DCE MR images (Losurdo et al., 2018). Therefore, many methods have been developed to automatically extract features and interpret those DCE-MR images. Proposed features including lesion morphology, texture, and enhancement kinetics have been proved by recent studies to be useful for the identification of genomic composition of BC lesions and for patient outcomes prediction (Zhang et al., 2019a) (Iranmakani et al., 2020) (Alzaghal and DiPiro, 2018).
However, the extraction of these features requires the lesions to be accurately segmented first. Therefore, the accurate segmentation of breast lesions in DCE-MR images is a critically significant task for automated BC analysis, diagnosis and treatment follow-up (Zhang et al., 2019a).

The most straightforward way to achieve this task is to manually annotate lesion regions by radiologists, but this is time-consuming and error-prone (Zhang et al., 2019a). Therefore, automating this challenging task will help radiologists to reduce the high manual workload and obtain more accurate lesion segmentation. However, automatic breast lesion segmentation based on DCE-MRI is a challenging problem and an active area of research.

In this work we propose an automated segmentation method for breast lesions in DCE-MRI using ROI guided, 3D patch based U-Net framework. The contribution of this work is aiming to address the problems of confounding regions and class imbalance by utilizing a balanced sampling technique for patch extraction that is restricted by a ROI. Additionally, we propose a less parametric representation of the 3D+time data, that is the standard deviation along the time dimension.

The remainder of this paper is structured as follows: section 2 outlines some related works in the literature. Section 3 describes the dataset and in section 4 we introduce our proposed method. Section 5 reports and discusses the results we obtained. Finally, in section 6 we present our conclusions.

2. State of the art

Apart from the manual method, existing methods for lesion segmentation in general fall into three categories: 1) Atlas-based methods, 2) Semi-automatic methods, and 3) Learning-based methods.
Despite the promising results achieved by Atlas-based methods in other tasks (as in (Prastawa et al., 2004) and (Wang and Yushkevich, 2013)), they fail to accurately identify breast lesions. This is attributed to the fact that breast lesions mostly do not have fixed positions and regular morphological shapes. Figure 1 shows example cases from our dataset with lesions of various sizes, shapes, locations and intensities.

Since one of the challenges to automate the task of breast lesion segmentation is the difficulty of identifying them from confounding organs or vessels (as illustrated in Figure 3); semi-automatic methods have been proposed. In these methods radiologists have to define lesion regions first (bounding boxes) to make the automatic segmentation task easier (Zheng et al., 2007) (Ashraf et al., 2012) (Vignati et al., 2011). In fact, there are only few studies focusing on breast lesion segmentation in DCE-MRI, and most of them are semi-automatic methods (Zhang et al., 2019a).
Learning-based methods perform automatic lesion segmentation using supervised learning algorithms and they have achieved remarkable performance in many medical applications. There are two types of learning-based methods: 1)Traditional Machine Learning (ML) methods, and 2)Deep Learning (DL) methods (Zhang et al., 2019a).

2.1. Traditional learning-based methods

In traditional learning-based methods, feature extraction and model training are treated as two separate tasks. Several traditional learning-based methods have been proposed for breast lesion segmentation but only few of them are focusing on DCE-MRI.
Many studies have been conducted in order to identify the definitive set of features and segmentation model for DCE-MRI and in several studies similar approaches have been utilized. For instance, Gubern-Mérida et al. (2014) proposed an automated localization method for breast lesions in DCE-MR images, by first extracting

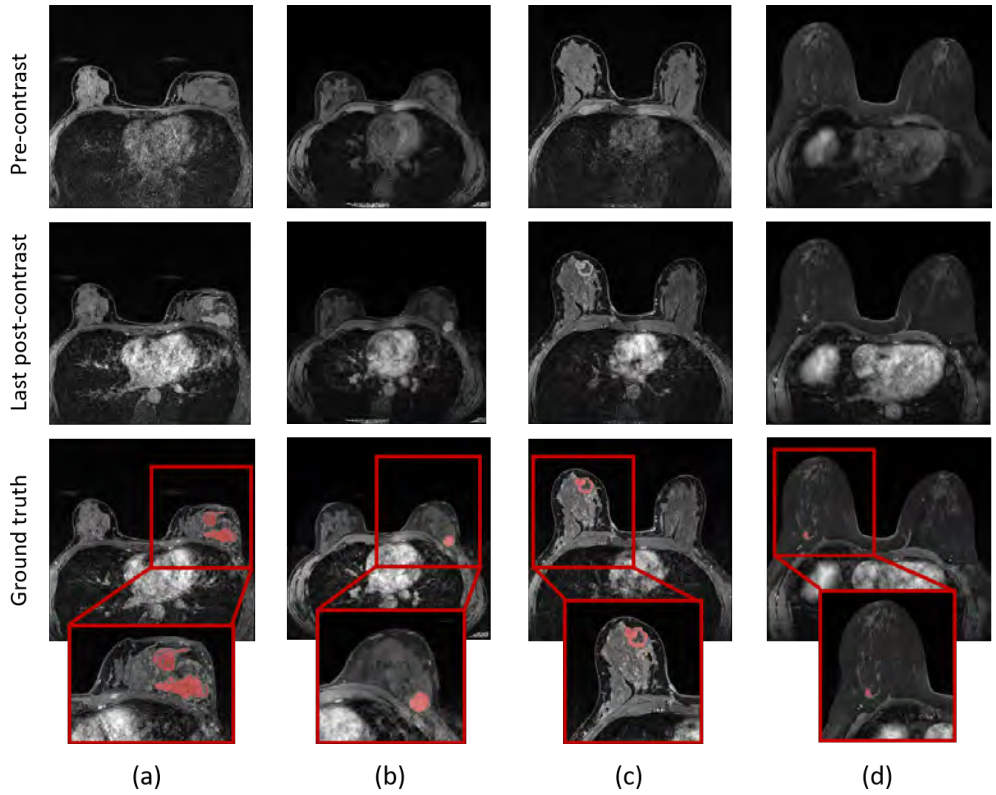


Figure 1: Breast lesions with different sizes, shapes, locations, and intensities. Cases in (a) to (d) are: A1E0, A0B6, A0DZ and A0HA respectively.

blob and relative enhancement voxel features for locating lesion candidates and then computing a malignancy score for each lesion candidate using region-based morphological and kinetic features computed on segmented lesion candidates. Kinetic features are characteristics modelling the shape of the Time Intensity Curve (TIC) and they have been proven in many papers as the most effective way to perform the lesion segmentation by means of machine learning. In addition to that, further analysis of TIC is widely used to provide several parameters which are useful for lesion diagnosis.

TIC is obtained either for each voxel or for regions of interest, and it shows the absorption and the release of the contrast agent over time according to the vascularisation characteristics of the tissue (Piantadosi et al., 2019).
Washout and plateau patterns (along with rapid up-slope in the early phase) in TIC occurring early in dynamic study are more likely to be associated with malignancy, whereas persistent pattern is usually detected with benign lesions (Dogan et al., 2006) (Cheng and Li, 2013). Figure 2 illustrates the difference between normal tissue and lesion curves.

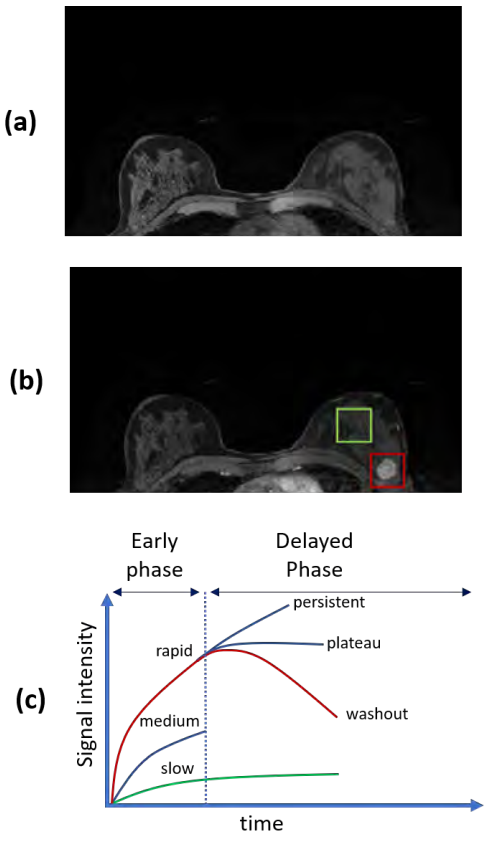


Figure 2: Illustration of the difference between Time Intensity Curves (TIC) of a normal tissue (green) and a lesion (red). (a) Pre-contrast volume. (b) Second post-contrast volume. (c) Different types of TIC from DCE-MRI including the one for normal tissue (green) and for lesion (in red).

2.2. Deep Learning Based methods

On the contrary to conventional learning-based methods, deep learning based methods (such as convolutional neural networks (CNN) and fully convolutional networks (FCN)) incorporate both feature extraction and model training into a unified learning framework so that the segmentation task is performed in an end-to-end manner. These methods have recently achieved state-of-the-art performance in medical imaging analysis.

In such methods, voxel-wise (or pixel-wise) classification models are trained, in which cubic patches centered at a particular voxel are first extracted, then the patch-wise binary classifier learns to classify voxels into either lesion or non lesion voxels (Zhang et al., 2019a).

In general, since the task of lesion segmentation can be considered as a semantic segmentation in which the input image has to be divided into Regions of Interest (ROIs) each referring to a lesion, every CNN segmentation model could be potentially used (Piantadosi et al., 2019).

One of the first works to address semantic segmentation with CNN was SegNet, developed by Badrinarayanan et al. (2017). SegNet is a deep convolutional encoder decoder architecture, followed by a pixel-wise classifier. The role of the encoder network is to learn a compact representation of the input data, while the role of the decoder network is to map the encoded features to a segmentation mask.

Similarly, the U-Net developed by Ronneberger et al. (2015) for biomedical image segmentation, exploits an encoder-decoder architecture, enhanced by the presence of skipping connections between the two sides with the aim of exploiting encoding information to improve the decoding stage and to reduce the gradient vanishing problem. In fact, the U-Net method improved the idea of a CNN by comprising regular CNN layers followed by up-sampling ones. This model has been widely used in the literature and several alterations to it have been proposed. Both SegNet and UNet architectures provide the potential to produce accurate models even with relatively small datasets.

There are several deep neural networks that have been recently proposed for breast image analysis, based on mammography, MRI, ultrasound, and whole-slide histology images. However, there are not so many works focusing on DCE-MRI. Moreover, the problems of both class imbalance and confounding regions are rarely taken into account in most of the existing deep learning based methods. These two problems are very common in breast lesion segmentation from DCE-MRI (Zhang et al., 2019a).

Chen et al. (2018) proposed a stacking of three parallel ConvLSTM (Shi et al., 2015) networks (to extract temporal and 3D features) over a 4-layer U-Net (to perform the segmentation).

Zhang et al. (2019b) investigated a DL based method to segment breast lesions in DCE-MRI scans in both 2D and 3D settings. Both 2D and 3D frameworks were proposed and evaluated. The binary cross-entropy loss function was used. The 3D U-Net performed slightly better in terms of dice coefficient and yielded less false positives. However, the dataset slices used in this study were selected only from the second-post contrast scan. Moreover, the UNet models were fed by images of bounding box regions surrounding the entire lesion instead of the full size of the breast MRI.

El Adoui et al. (2019) proposed two deep learning approaches by building two CNNs based on SegNet and U-Net. The binary cross entropy loss function was also used. In this study, U-Net outperformed SegNet which can be explained by the fact that SegNet is more adapted to the multiclassification task such as autonomous cars applications. One of the limitations of this study, is that 2D slices were used as inputs instead of 3D volumes due to the lack of data. This affects, artificially, the statistically significant differences between the outputs and the ground truth. Indeed, comparing performance by 2D slices can fail to consider the correlation in performance for same slices from the lesion. Another limitation of this study is that the ground truth labels were provided by only one radiologist. It would be better to provide the ground truth by many radiologists in order to minimize the inter-reader variability.

Piantadosi et al. (2019) proposed a 3D lesion segmentation method (3TP U-Net) using a U-Net that exploits the well-known Three Time Points approach (3TP). The 3TP approach was proposed by Degani et al. (1997) whose studies have showed that breast lesion analysis can be improved by focusing on just three well defined temporal acquisitions (t0 = pre-contrast, t1 = 2 minutes after contrast agent injection, t2 = 6 minutes after contrast agent injection). In Piantadosi et al. (2019) method, images acquired at the three specific time points were fed to the network in order to take into account the DCE-MRI fundamental characteristic. Segmentation was performed slice-by-slice, considering the three temporal acquisitions of the same slice as channels within the image. Slices were extracted along the projection with the higher resolution (the coronal projection) and to obtain a reliable and fair evaluation, the slices from the same subject were always separated across the cross-validation folds. Additionally, the Dice loss function was used .

To address the two issues of confounding regions and class imbalance, Zhang et al. (2019a) proposed a mask-guided hierarchical learning (MHL) framework for breast lesion segmentation via FCN (U-Net). First, the pre-contrast volumes were used as input to a U-Net model to generate 3D breast masks as the region of interest (ROI), so that confounding regions from input DCE-MR images are removed. Then a two-stage U-Net model to perform coarse-to-fine segmentation was used.

In the first stage the post-contrast volumes and the difference volumes (between post and pre contrast) were used along with the generated breast masks as inputs to a first stage U-Net to generate over-segmented lesion-like regions. Also to handle the class-imbalance problem, a Dice-Sensitivity-like loss function was proposed. In the second stage, an additional U-Net was used to refine the segmentation results of the previous stage, using a Dice-like loss function and a reinforcement sampling strategy.

3. Materials

3.1. Data acquisition and annotation

For this work we used a subset of the TCGA-BRCA collection, which was collected by the TCGA Breast Phenotype Research Group and made available in The Cancer Imaging Archive—TCIA (http://http://www.cancerimagingarchive.net). All patient data were obtained under IRB-approved HIPAA compliant (Clark et al., 2013) (Burnside et al., 2015).
The data subset we used consists of 46 cases all of them diagnosed with BC by performing image-guided core needle biopsy, in other words all cases had lesions. Scans were acquired at the University of Pittsburgh Medical Center (1999-2004) prior to any treatment.
MRIs were acquired using a standard double breast coil on a 1.5T GE whole body MRI system (GE Medical Systems, Milwaukee, Wisconsin, USA). Only T1-weighted dynamic contrast-enhanced MR images were used in this study. The imaging protocols included one pre- and four to six post-contrast volumes obtained using a T1-weighted 3D spoiled gradient echo sequence with a gadolinium-based contrast agent (Omniscan; Nycomed-Amersham, Princeton, NJ). Typical in-plane resolution was 0.53-0.86 mm, and typical spacing between slices was 2-3 mm.

Each breast MRI examination was independently reviewed by three expert board-certified breast radiologists blinded to outcome data. Each radiologist identified and annotated the location of the primary breast lesion in the image and measured maximal lesion diameter using a linear measurement tool. lesion location on MRI was determined using radiologist reviewer information (via a posteriori consensus). The average maximal diameter as measured by the 3 readers (called "radiologist size") was used for comparison to a computer-derived measurements. Each primary breast lesion was then automatically segmented in 3D from the surrounding parenchyma.

It is important to mention that most of the cases had multiple lesions according to the reviewer radiologists, however the Ground Truth (GT) segments only the primary lesion since the purpose of the TCGA/TCIA study was to map the radiomics (phenotypes) of the primary lesion to the corresponding clinical, histopathology, and genomic data.

3.2. Data preparation

3.2.1. Ground truth segmentations

As mentioned earlier, only primary lesion in each case is segmented in the Ground Truth (GT). Segmentations were provided as binary files, each has the following:
1. Six uint16 values for the coordinates of the lesion's cuboid (bounding box), with respect to the full volume:
y_start y_end
x_start x_end
z_start z_end
2. The N int8 0/1 values of voxels for the above specified cube, where N = (y_end y_start +1) * (x_end - x_start + 1) * (z_end - z_start + 1). Where a voxel value of 1 denotes that it is part of the lesion, while a value of zero denotes it is not.
These binary files were used to generate a full breast volume size segmentation.

3.2.2. DCE-MRI scans

Each patient DCE-MRI series was provided as a DICOM file that combines all pre- and post-contrast scans as different channels. For easier use, pre- and post-contrast scans within the series were separated and saved as Nifti files using SimpleITK library.

4. Methods

In this work we propose an automated method for segmenting breast lesions in DCE-MRI using deep learning. Our method is based on a 3D patch based modified U-Net . Moreover, a restricted patch extraction using a ROI was used. The proposed method takes into account different challenges: (1) the confounding regions, (2) the class imbalance, and (3) the large amount of 3D+time data. In order to tackle the first two problems, we performed balanced patch sampling technique restricted by a ROI to ensure that the two classes are equally distributed in the training set and to avoid having patches from confounding regions.
Regarding the third problem, we proposed the use of standard deviation (std) generated voxel-wise across the time dimension to represent the time-intensity behaviour provided by the 3D+time data and to reduce redundant information and high computational cost which are encountered if we train the network over all provided 4D data.
Moreover, different networks and parameters have been evaluated in comparison with our proposed method and the obtained results are presented and discussed in Section 5.

4.1. Pre-processing

Prior to feeding input volumes to the network, the following pre-processing steps were performed:
(1) ROI masks generation in order to exclude confounding regions. This was performed using a simple landmark detection method in which we detected the skin-air boundary between the two breasts and then excluded non-breast part of the volume that lies beyond the detected landmark.
(2) Zero padding with padding width equal to half of the patch size.
(3) Zero-mean unit-variance intensity normalization.
(4) Balanced patch extraction in order to tackle the class imbalance problem.

4.1.1. ROI masks generation

As mentioned earlier, confounding background (e.g., vessel structures and organs) in DCE-MRI makes the task of breast lesion segmentation more challenging. Therefore, it is important to generate a region of interest (ROI) that includes the breast only. Figure 3 shows examples of confounding background that can be found in DCE-MR scans.

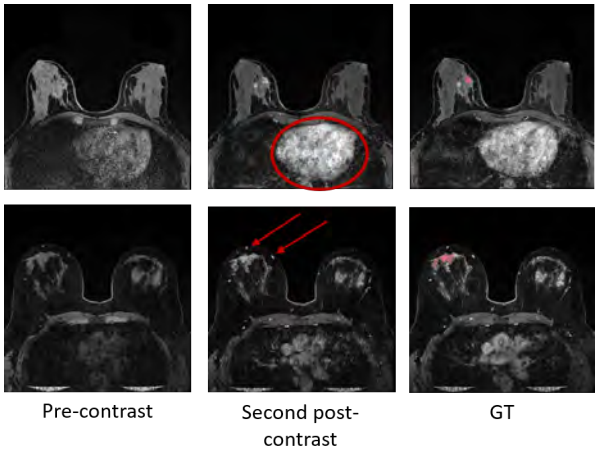


Figure 3: Example cases showing confounding regions such as vessels (arrows) and organs (circle). (top row: case A0HA, bottom row: case A1DI).

The most straightforward way is to use a breast mask as a ROI, which can remove most of the confounding organs. However, no breast masks were provided along with the dataset we are using.

Several methods have been proposed to generate breast masks. Nevertheless, in this work we implemented a simple method to generate ROI masks instead of breast masks.

Based on the fact that breasts have regular morphological shapes and relatively fixed position on MRI images, we detected a landmark at the breast-air boundary between the two breasts and then excluded parts of the volume that lie beyond the detected landmark. This was done by first obtaining the voxels at the intersection line of mid-sagittal and mid-transverse planes (located between two breasts) and then detecting the local maxima of intensities across those voxels. Then Local maxima were filtered (using empirically chosen values of height and distance between each two local maxima) such that the location of the first detected local maximum (anterior to posterior direction) corresponds to the required landmark slice. Figure 4-(a) to (c) illustrates the steps followed to detect the required landmark. Accordingly, the remaining part (coronal slices) starting from few slices (empirically chosen) after that landmark were set to zero while other parts were set to one in order to generate the 3D binary mask, figure 4-(d) shows the generated ROI mask overlaid on an example case.

4.1.2. Patch sampling

Many studies have proved that training on a patch-level performs better than on an image-level. This is attributed to the fact that CNN kernels (filters) will only process one patch at a time, rather than the whole image, which does not only lead to less computational demanding and reduced training time, but also a better features depicting and a smaller number of parameters which introduces a regularization property as well.

The common way of extracting patches from images is the uniform sampling, in which patches are extracted from all parts of an image uniformly with a certain step between each two consecutive patches. However, in the context of lesions segmentation in which the number of voxels in the lesion region (positive class) is much smaller than that in the background (negative class), this leads to a very common issue of class imbalance where only a small number of the extracted patches will be taken from the lesion class and hence yielding a poor performance of the network and misclassification of lesion voxels. This common issue has been addressed in several studies on lesion segmentation of different organs (Zhang et al., 2017) (Christ et al., 2017) (Bria et al., 2013).
Guerrero et al. (2017) proposed a method to address the class imbalance issue in which the extracted patches always contain lesion voxels and were randomly shifted so that the center of the patch does not necessarily be a lesion voxel. Another method proposed by Clèrigues et al. (2019), which utilized a balanced sampling strategy such that for each image there are equal number of patches representing both classes. Additionally, a ROI restricted technique was proposed in which negative patches extraction was restricted to be from a ROI and not backround regions.

In our work, we utilized a ROI restricted balanced sampling technique in which negative and positive patches were equally extracted. Additionally, negative patches were extracted only within the ROI we generated to avoid extracting patches from confounding

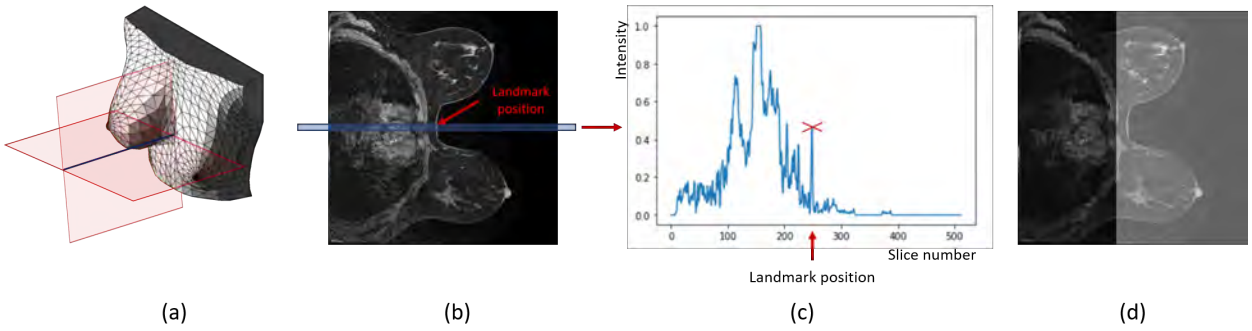


Figure 4: Illustration of ROI mask generation method. (a) 3D representation of breast with mid-sagittal and mid-transverse planes intersection. (b) Intersection line in a 2D example slice. (c) Detection of breast-air boundary landmark. (d) Obtained ROI mask overlaid on the example slice.

regions and to make sure they are located within the breast region.

4.2. Segmentation algorithm

In this section we explain our algorithm which yielded the best results among other experimented algorithms.

In this work we adopt a modified 3D U-Net architecture. U-Net is an encoder-decoder architecture originally designed for biomedical electron microscopy (EM) images multi-class pixel-wise semantic segmentation (Ronneberger et al., 2015) (Çiçek et al., 2016).

In the proposed U-Net architecture we made some alterations to the basic U-Net architecture. We used four levels (blocks) in each of the encoding and decoding paths and replaced the U-Net convolutional blocks with ResNet basic blocks. This architecture is illustrated in Figure 5. Every step in the contracting path consists of:

- ResNet basic block.
- $(2 \times 2 \times 2)$ max-pooling with stride=2 for downsampling.

Then they are followed by a latent space consisting of a ResNet basic block with rectified Linear Unit (ReLU) activation and an instance normalisation. Similarly, the expanding path consists of:

- $(2 \times 2 \times 2)$ up-convolution with stride=2.
- Concatenation with feature map from the corresponding level of the contracting path.
- ResNet basic block.

Finally, there's a $(1 \times 1 \times 1)$ output convolution layer with two output channels followed by a softmax layer which returns probabilities for each class.

Furthermore, in our proposed algorithm we used the binary cross-entropy loss function, AdaDelta optimizer and a threshold of 0.5 for generating the output segmentation. Moreover, three input volumes were fed to the U-Net: pre-contrast, last post-contrast and standard deviation (std) volumes. Finally, 4000 balanced patches of the size (32,32,16) were extracted with a sampling step of (16,16,16). Our obtained results are reported in Section 5.

5. Results and Discussion

Different experiments were performed throughout the development process of our framework in order to improve the performance. First we investigated different patch sizes. Then we investigated the usage of different optimizers and loss functions.

Additionally, we investigated different combinations of input scans among the provided DCE-MRI time series volumes (a pre-contrast and four post-contrast volumes) as well as other volumes that we generated (such as the subtraction volumes) as an attempt to find the combination that yields better results. Besides that, we obtained different volumes (such as mean and standard deviation) to represent the time-intensity information in a way that reduces the several time point volumes to a less parametric volume.

Finally, we investigated the performance of our proposed architecture explained in section 4.2 and another two different U-Net based architectures: (1) Basic U-Net, (2) Two hierarchical basic U-Nets.

All experiments were performed using 5 fold cross-validation across the provided 46 cases. In each fold 9 cases were used for testing (10 cases in the last fold) and the remaining cases were shuffled and divided into 80% for training and 20% for evaluation. By doing so, we obtained lesion segmentation results for each of the 46 cases.

As evaluation criteria, we used the Dice Similarity Coefficient (DSC) described in equation 1, since it is one of the most used metrics in the state of the art of medical image segmentation based on neural networks.

$$g(x) = \frac{2TP}{2TP + FP + FN} \tag{1}$$

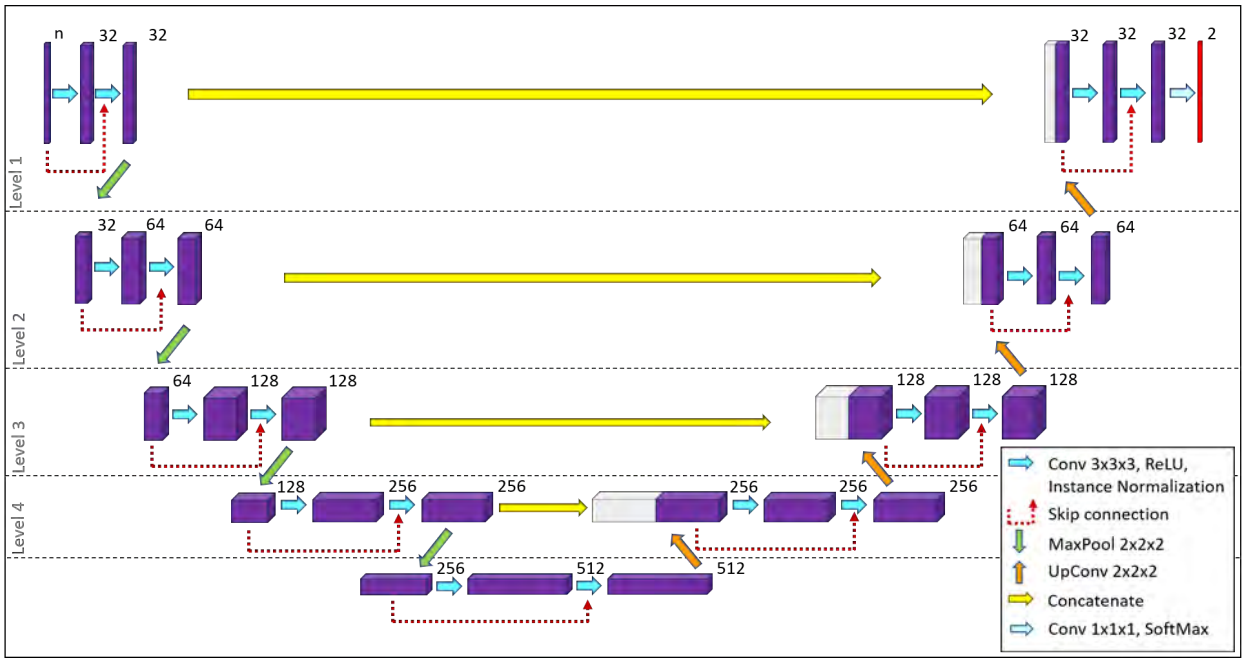


Figure 5: Our proposed U-Net with ResNet basic blocks architecture.

where TP, FP and FN refer to True Positive, False Positive and False Negative respectively.
However, the provided GT of our dataset has only the main lesion segmented for each case and most cases were denoted by radiologists as having multiple lesions (multi-centric or multi-focal). Hence, the obtained DSC values might not adequately evaluate the segmentation performance of our algorithm because other (non-primary) lesions will be considered as FPs (even though they are correctly detected by our algorithm). Therefore we obtained a second dice after post-processing the obtained segmentation in order to approximately evaluate how good the main lesion was segmented, regardless of the present of any other lesions (secondary lesions) .
To do this we first detected the the largest three connected components (CCs) in the obtained segmentation. Then we obtained the distances between their centroids and centroid of the lesion in the GT. Finally we kept only the nearest CC to the segmented lesion in GT.
However, it is important to mention that GT is never used for performing post-processing. This was used just to obtain a better indication of the performance considering the problematic (incomplete) GT we had and that secondary lesions will be miss-classified as FPs.

In this section, we first introduce our Implementation details and then we present the different experiments implemented and the obtained results along with the discussion.

5.1. Implementation details

The proposed architecture and all other experimented architectures were implemented in Python 3.7.4 using Pytorch 1.4.0 machine learning framework.

All python scripts were executed on Ubuntu on a physical server hosted in our university equipped with 256GB RAM, and Nvidia GeForce RTX 2080 GPU with 11016 MB memory.

5.2. Experiment 1: Threshold

Thresholding is needed in order to binarize the output probability map and generate a segmented volume. We studied the usage of different thresholds and obtained the results which are shown in Table 1.

Table 1: Mean dice and standard deviation (std) obtained using different thresholds.

Threshold	mean	std
0.9	0.490	0.273
0.5	**0.508**	0.275
0.4	0.500	0.271
0.3	0.499	0.271
0.2	0.032	0.120

As we can see from Table 1, the obtained probabilities were high so that using lower threshold (but not too low) did not affect the results.
Therefore, a threshold value of 0.5 was used in our proposed algorithm as well as in all experiments discussed in the following subsections.

5.3. Experiment 2: Patch size

Several studies have been conducted to investigate the effect of the patch size in patch based neural networks and some of them have showed that using bigger patches may improve the segmentation results, as the

network can capture more contextual information (Farabet et al., 2013) (Li et al., 2014). However, the bigger the patches are, the more computationally demanding it is and the longer it will take for training. Moreover, very big patches may also affect the results. Hence, it is very important to choose the optimal patch size.
Accordingly, we compared the performance with four different patch sizes: (32,32,32), (32,32,16), (16,16,16), and (16,16,8). The obtained results are reported in Table 2.
In these experiments the basic U-Net was used with sampling step equals to the patch size (i.e. non-overlapping patches). Moreover, pre-contrast volume, last post-contrast volume and the subtraction between them were used as inputs.

Table 2: DSC values (mean± std) for different patch sizes. DSC_1 is the normal dice (without post-processing) and DSC_2 is the dice of main lesions only.

Patch size	DSC_1	DSC_2
(32,32,32)	0.508±0.275	0.576±0.291
(32,32,16)	**0.511±0.266**	**0.608±0.273**
(16,16,16)	0.517±0.267	0.630±0.279
(16,16,8)	0.398±0.250	0.643±0.276

Several observations can be made from Table 2. Patch sizes of (32,32,32) and (32,32,16) performed better if to consider both values of DSC_1 and DSC_2, with average dice values of 0.508 and 0.511 respectively (0.576 and 0.608 considering main lesions only).
The slightly better performance when using patch size of (32,32,16) than (32,32,32) can be attributed to the size of our scans, which have less axial slices than coronal and sagittal. All cases has 512x512 coronal-sagital size whereas the number of axial slices was different from case to another (in the range of 58 to 112 slices).
We can also observe that in terms of segmenting the main lesion only (i.e. values of DSC_2), the patch size (16,16,8) performed better. However the normal dice (DSC_1) was much lower which indicates that more FPs were detected. This might be explained by the fact that many cases had small lesions which were better segmented using smaller patch size but this caused more FPs to be also detected.

5.3.1. Experiment 3: Optimizer

Experiments were performed to compare the performance of AdaDelta optimizer to another optimizer (Adam). Table 3 shows the obtained results. In these two experiments we used the basic U-Net with patch size of (32,32,32) and sampling step equals to the patch size. Moreover, the pre-contrast volume, last post-contrast volume and the subtraction between them were used as inputs.

As we can see from Table 3, better results were obtained using AdaDelta optimizer.

Table 3: DSC values (mean± std) for two different optimizers. DSC_1 is the normal dice (without post-processing) and DSC_2 is the dice of main lesions only.

Optimizer	DSC_1	DSC_2
AdaDelta (lr=1)	**0.508±0.275**	**0.576±0.291**
Adam (lr=0.0001)	0.367±0.289	0.552±0.302

5.4. Experiment 4: Loss function

In existing works, different loss functions were proposed for lesion segmentation tasks.
Binary cross-entropy loss is a commonly used loss function in the literature.
However, since it evaluates the class predictions for each voxel individually and then averages over all voxels, it accounts for all voxels equally. This leads to equal learning to each voxel in the volume. However, in the case of unbalanced classes in the volume this can be a problem since training can be dominated by the most prevalent class. This problem is very common in the task of lesion segmentation as voxels in the background (negative class) outnumber voxels in lesion regions (positive class), which causes the prediction of learned networks to be biased towards the background regions, and the lesion regions will only be partially detected or missing. Therefore, other loss functions have been proposed mainly to overcome this issue. Long et al. (2015) have proposed a weighted cross entropy loss by weighting each class in order to avoid the class imbalance issue. Meanwhile, Ronneberger et al. (2015) have proposed a loss weighting scheme for each pixel such that higher weight is assigned to pixels at the contour of segmented objects.
Another loss function is the Dice-Sensitivity-like loss proposed by Zhang et al. (2019a) which is a combination of dice and sensitivity. Dice coefficient, unlike other measurements (such as the traditional overall accuracy, mean squared error, or cross-entropy) highly focuses on the lesion class and penalizes the missed voxels as well as false positives. Sensitivity adds an additional bias toward detection of lesion voxels and therefore addresses the issue of imbalance by shifting the focus toward the minority class (lesion voxels). However, this loss function aims to segment lesion regions, as well as keep those lesion like regions as much as possible, which yields a rough segmentation, in other words many false positive voxels will be included. Therefore, the obtained rough segmentation was refined using a second network with a dice-like loss function.

In this work we compare the performance of the cross-entropy loss and other loss functions:

1- Dice loss function, which = 1 - DSC.

2- Combination (summation) of both dice loss and binary cross-entropy loss functions.

3- Dice-sensitivity loss function, proposed by Zhang et al. (2019a), which is given by: 2 - DSC - Sensitivity.

Table 4 shows the obtained results.
In these experiments the basic U-Net was used with patch size of (32,32,32) and sampling step equals to the patch size. Moreover, pre-contrast volume, last post-contrast volume and the subtraction between them were used as inputs.

Table 4: DSC values (mean± std) for different loss functions. DSC_1 is the normal dice (without post-processing) and DSC_2 is the dice of main lesion only.

Loss Function	DSC_1	DSC_2
Cross Entropy	**0.508±0.275**	**0.576±0.291**
Dice	0.470±0.294	0.505±0.309
Dice+Cross Entropy	0.472±0.309	0.521±0.313
Dice Sensitivity	0.483±0.320	0.524±0.336

As shown in Table 4, the best results were obtained by using the cross-entropy loss. Although one would expect segmentation related loss functions (e.g. dice loss) to obtain better results, our experiments show that more generic losses (i.e cross-entropy) obtain better results. However, this behaviour is to be further investigated in the future.

5.5. Experiment 5: Input scans

As mentioned earlier in Section 2, one of the limitations in most of the existing works was that only one temporal acquisition among the time series was used. Therefore, it is interesting to use several time acquisitions since the 3D+time data of DCE-MRI involves important information about the Time Intensity Curve for each voxel, which if utilized, might improve the performance of lesion segmentation task using DCE-MRI.
The method of utilizing two temporal acquisitions (pre and post-contrast) along with the subtraction between them was proposed by Zhang et al. (2019a). Another method of utilizing Three Time Point acquisitions (3TP) was proposed by Piantadosi et al. (2019).
In this subsection we compare the performance of using the inputs combination presented in our proposed method (i.e. utilizing the std along with pre and last post-contrast) to other inputs combinations including the ones proposed by Zhang et al. (2019a) (i.e. pre, post and difference) and Piantadosi et al. (2019) (i.e. 3TP). Table 5 reports the obtained results.

In these experiments the basic U-Net was used with sampling step equals to the patch size.
Two groups of experiments were performed, for the first group we used patches of size (32, 32, 32) whereas for the second group patch size of (32, 32, 16) was used.

As we can see from Table 5, utilizing the std as proposed in our method achieved a mean dice of 0.573 (and 0.654 if to consider main lesions only). This result outperforms those obtained by utilizing the difference volume, 3TP and even utilizing many acquisitions (pre, first 3 posts and last post).
Both std and the subtraction volumes emphasize the difference in time-intensity behavior of normal tissues and lesions, in other words lesions supposed to be more enhanced than a normal tissue. However, the better performance of utilizing std volume over the subtraction volume might be explained by the fact that std volume includes information from all temporal acquisitions in the series, whereas the subtraction is generated using only two acquisitions. Additionally, the lesion might not be very clear in the subtraction volume in cases with low changes in lesion enhancement or dense breast tissues.
The better performance of utilizing the std compared to using several temporal acquisitions (pre, first 3 posts and last post-contrast) might be attributed to the fact that when using several scans there will be redundant information affecting the results. Moreover, using many input scans is very computationally demanding and requires long training time.

Qualitatively speaking, results of utilizing the std volume showed improved segmentation of small lesions, irregular lesions and lesions with lower enhancement. Additionally, less FPs caused by confounding regions (especially organs) were observed in several cases. Some qualitative improvements in segmentation results are shown in Figures 6 and 7.

As we can see from Figure 6-(a) which shows a case with an irregular lesion, larger overlapping with the GT was achieved after utilizing std volume. Similarly in Figure 6-(b) which shows a case with a small lesion, we can see that the output segmentation improved dramatically after utilizing std volume.
In figure 7-(a) we can see that the lesion was segmented well in both experiments (using difference and using std), however as observed in (b) which shows another slice of the same case shown in (a), less FPs were detected when utilizing std and hence the obtained dice increased significantly.

5.6. Experiment 6: U-Net architecture

In this subsection we compare results obtained using our proposed architecture (which we described in Section 4.2) with another two U-Net based architectures. The first is a basic U-Net with three levels instead of four along with cross-entropy loss function. The second one is the two hierarchical basic U-Nets approach proposed by Zhang et al. (2019a), in which we used dice-sensitivity-like loss in the first stage and dice-like loss in the second stage, as proposed by Zhang et al. (2019a) . U-Nets of both stages had three levels only.
The obtained results are reported in Table 6.

As observed from Table 6, our proposed architecture outperformed the other architectures achieving a mean dice of 0.645 (and 0.708 if to consider only main lesions).

The better performance of our architecture is attributed to the higher depth compared to other experimented networks, since it is known that deeper networks

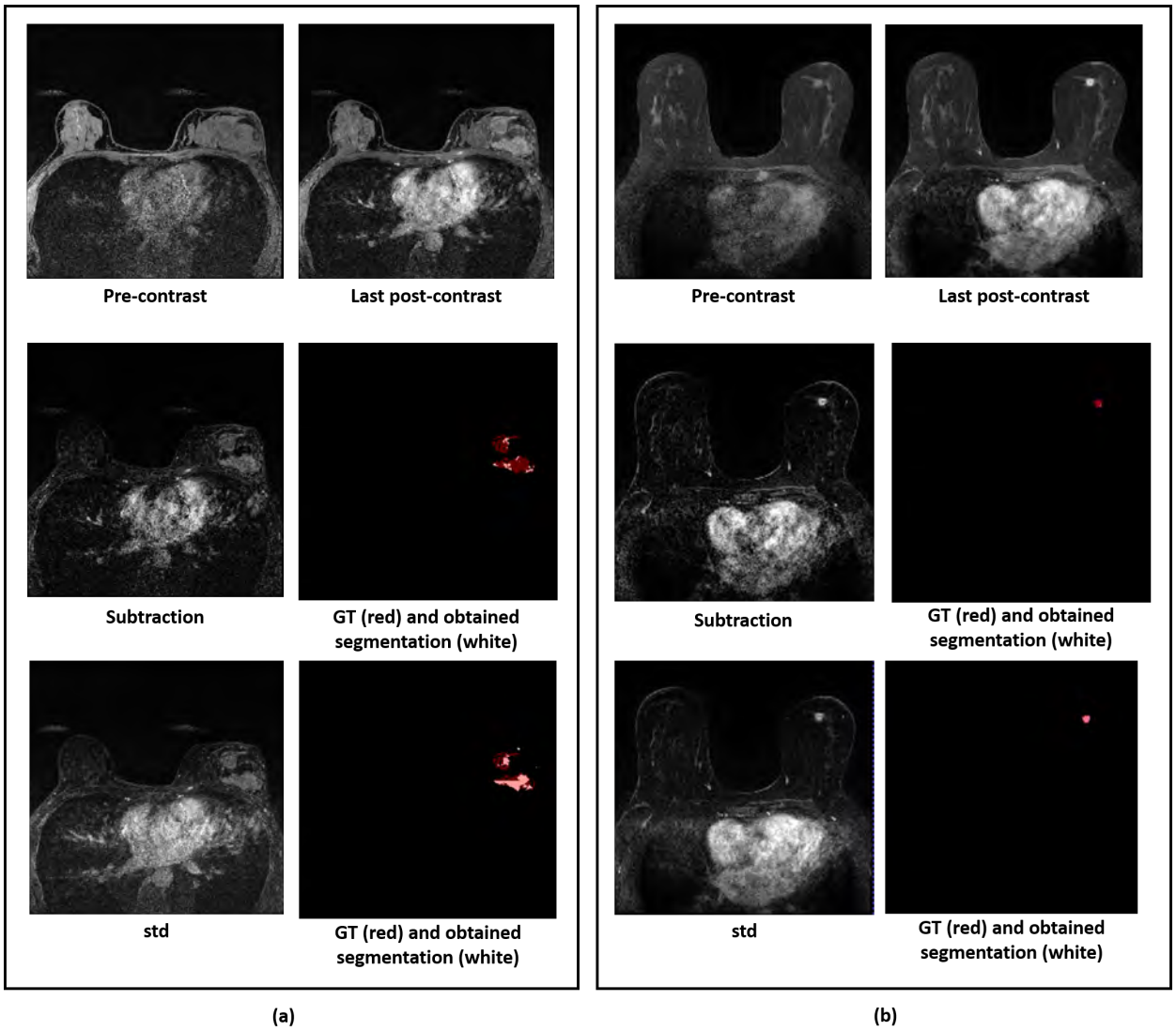


Figure 6: Example cases of improved segmentation of small, irregular and low enhanced lesions when std volume was used compared to when subtraction volume was used. On the right is cese A0E0 and on the left is case A201. Middle row shows obtained subtraction and corresponding output segmentation, bottom row shows obtained std and corresponding output segmentation. GT is represented in red and output segmentation in white.

Table 5: DSC values (mean± std) for different input scans. DSC_1 is the normal dice (without post-processing) and DSC_2 is the dice of main lesions only.

Sampling Step	Inputs	DSC_1	DSC_2
32,32,32	pre, last post	0.485±0.291	0.557±0.292
	pre, last post, subtraction (pre - last post)	0.508±0.275	0.576±0.291
	pre, last post, subtraction(pre - last post, signed image)	0.460±0.290	0.525±0.317
	last post, subtraction (pre - last post)	0.477±0.297	0.550±0.320
	pre, post 2, subtraction (pre - post2)	0.514±0.275	0.599±0.285
32,32,16	pre, posts (1 to 3), last post	0.548±0.277	0.614±0.279
	pre, post2, last post (3TP)	0.527±0.289	0.641±0.284
	pre, last post, std	**0.573±0.269**	**0.654±0.273**
	pre, last post, mean	0.558±0.284	0.649±0.283
	pre, post 2, std	0.555±0.278	0.644±0.285

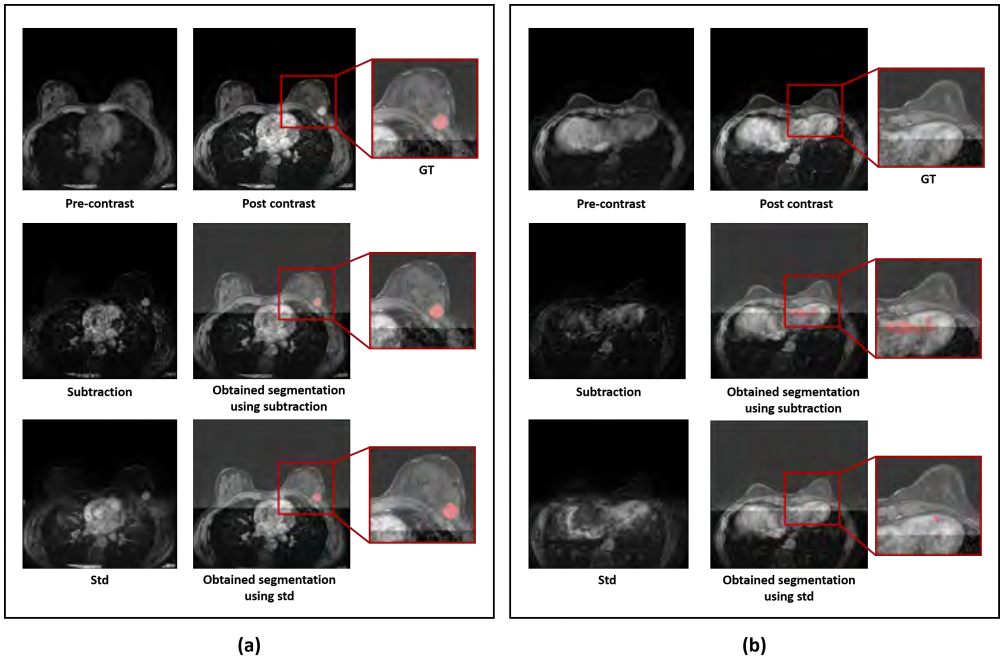


Figure 7: Example case (A0B6) of improved segmentation in terms of less FPs using std compared to using subtraction. Axial slices 47 and 14 are shown in (a) and (b) respectively. Middle row shows obtained subtraction and corresponding output segmentation, bottom row shows obtained std and corresponding output segmentation. All segmentations are overlaid along with the ROI.

Table 6: DSC values (mean± std) for different U-Net architectures. DSC_1 is the normal dice (without post-processing) and DSC_2 is the dice of main lesions only.

Network	Inputs	Patch Size	Sampling Step	DSC_1	DSC_2
Basic U-Net	pre, last post, std	32,32,16	32,32,16	0.573±0.269	0.654±0.273
Two hierarchical U-Nets	pre, last post, subtraction	32,32,32	32,32,32	0.482±0.294	0.581±0.301
U-Net with ResNet blocks	pre, last post, std	32,32,32	16,16,16	**0.645±0.248**	**0.708±0.230**

can lead to better performance. However, this is not the only reason, as deeper networks are more prone to overfitting due to the vanishing and exploding gradients problem. Utilizing the ResNet blocks make it possible to train deeper network and avoiding overfitting due to the skip connections which allows the output of some earlier layers to be fed directly to deeper layers.
Observed improvements included better segmentation of small lesions and less FPs. Figures 9 and 8 show some qualitative improvements in segmentation results using our network compared to the basic U-Net.

As expected, obtained dices for multiple lesion cases were affected by the incomplete GT issue since our network either segmented all lesions (not only primary ones) or missed primary lesions while segmenting secondary lesions, which might happen if the primary le-

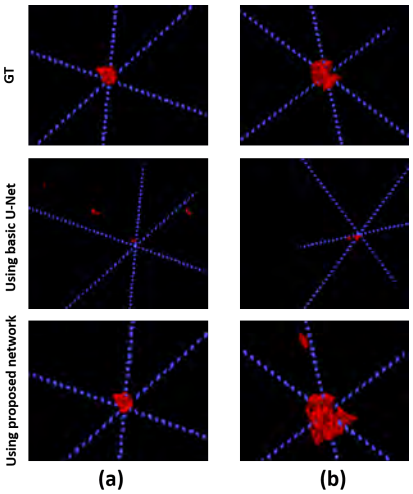


Figure 8: 3D rendering of obtained segmentation using our proposed network compared to basic U-Net and GT. (a) Case A0GZ and (b) Case A0RX.

sion is more difficult to be detected (e.g. smaller or less enhanced). Figure 10 shows an example case with multiple lesions. For this case our network did not segment the primary lesion but on the other hand it segmented the other lesion (which is not in GT) very well. Therefore, we believe results could be improved with a better complete GT segmentation.

Another observation is that our network also segmented axillary lymphadenopathy (which appears in several cases among our dataset), even though it is not necessarily a lesion, it is often a sign associated with breast cancer. Axillary lymphadenopathy is defined as changes in the size and consistency of lymph nodes in the armpit (axilla) and it is a symptom associated with a range of diseases and conditions from mild infections to breast cancer (Gupta et al., 2017) (Samiei et al., 2019). Figure 11 shows an example case diagnosed with unicentric lesion and an axillary lymphadenopathy. As we can see in Figure 11, the lesion was segmented well by our algorithm, however the dice was affected due to segmenting the axillary lymphadenopathy which also had bigger size than the lesion.

As for the limitations of our study, the dataset we used had small number of cases. Having larger dataset is believed to improve the performance. Additionally, all cases in our dataset contained at least one lesion which might cause issues when dealing with normal MRIs (i.e. healthy cases). Although MRI studies are usually acquired for high risk women or in cases of suspicious findings (with a higher incidence than screening population), our work assumes that at least a lesion is present in the volume, which may not always be the case.
Another limitation related to our dataset as well is that we did not test with volumes acquired from different scanners. All volumes used in this study were acquired with the same scanner. It is likely to obtain worse results when training with one scanner and testing with another.

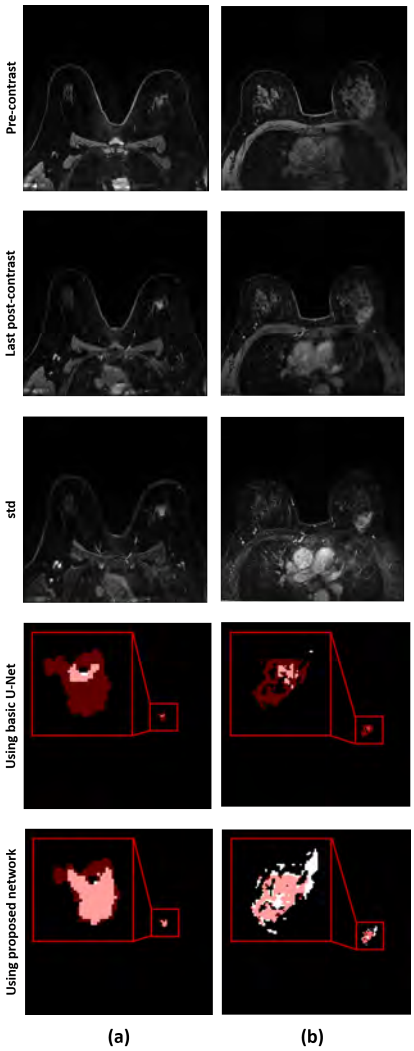


Figure 9: Example cases of improved segmentation achieved using our proposed architecture compared to a basic U-Net. Top row: Case A0GZ and bottom row: Case A0RX.

However, comparing our results to other existing works (mentioned in Section 2.2) including state-of-the-art, shows that our proposed method is promising, since we performed 3D segmentation of full-sized 4D data on the contrary to most of the existing works in which 2D segmentation was performed or segmenting only bounding boxes around the lesion or even utilizing one temporal acquisition only. In addition to that, we obtained good mean dice considering the incomplete GT annotation issue which indeed has affected the obtained dices and considering the small and complex dataset we used which has lesions of small sizes, irregular shapes, low enhancement, difficult locations near to confounding organs and, Axillary lymphadenopathy.

Table 7 shows a comparison between the results obtained by our method and results obtained in other works. It is important to mention that comparing our

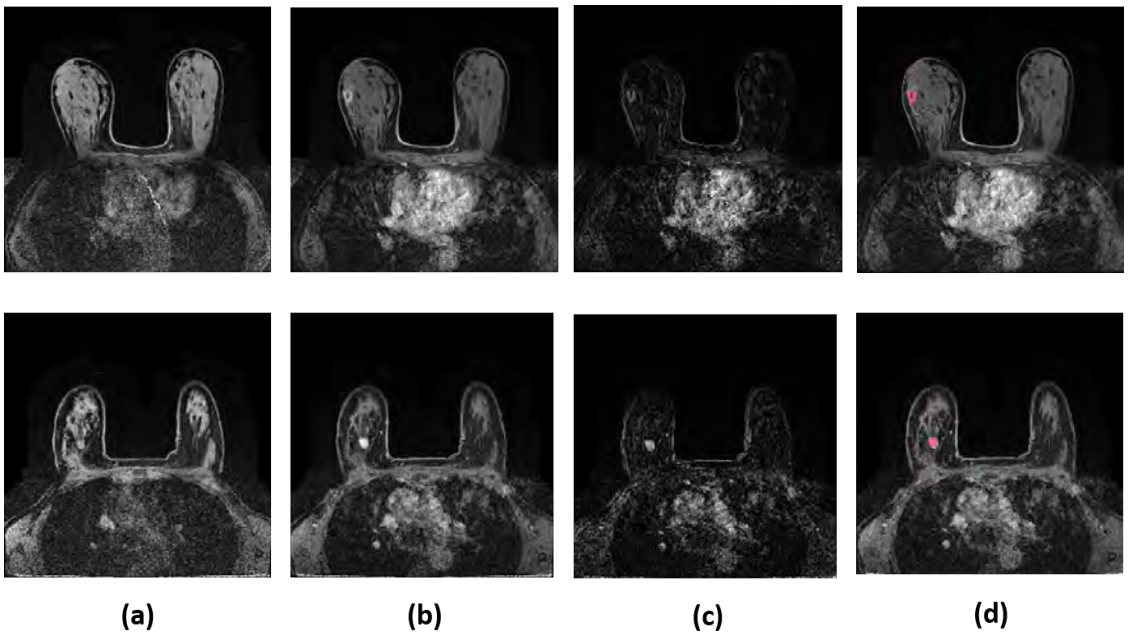


Figure 10: Example case (A0H7) with multiple lesions. Top row (slice 26) shows primary lesion segmented in GT but not segmented by our algorithm. Bottom row (slice 43) shows secondary lesion (not in GT) segmented by our network. (a) Pre-contrast volume. (b) Last post-contrast volume. (c) Std volume. (d) GT (lower row) or our segmentation (upper row) overlaid in red.

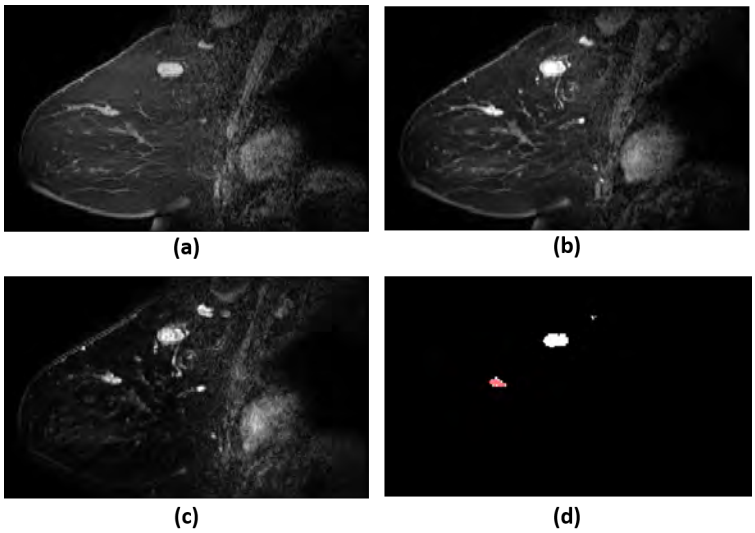


Figure 11: Example case (A18H) with an axillary lymphadenopathy. (a) pre-contrast. (b) last post-contrast. (c) std volume. (d) Obtained segmentation (white) and GT (red).

method with the existing approaches is difficult as they have been using different datasets and input information (3D/2D, whole images or ROIs).

6. Conclusions and future works

In this master thesis an automated breast lesion segmentation method was proposed for DCE-MRI. Our proposed method is a 3D patch based modified U-Net framework. In our modified U-Net we introduced ResNet basic blocks instead of basic U-Net blocks. Additionally, we utilized a ROI restricted balanced patch extraction in order to address both the class imbalance and confounding regions problems. Differently from most existing works on this topic, 3D segmentation was performed instead of 2D. Therefore, our method performs both segmentation and detection at the same time. Additionally we utilized not only one temporal acquisition (as in most existing works) but different temporal scans instead. The aim was to take advantage of the several acquisitions from the original 4D volume in a simple way by introducing a representative volume via

Table 7: Comparison of our proposed method and other existing methods

	Architecture	2D/3D	Number of cases (public/private)	Inputs	Loss function	Evaluation criteria and score	Scanner
Zhang et al. 2019b	U-Net	2D	1246 slices (private)	2nd post-contrast (lesion bounding boxes)	Cross-entropy	DSC = 0.91	-
	U-Net	3D	158 cases (private)	2nd post-contrast (lesion bounding boxes)	Cross-entropy	DSC = 0.92	
El Adoui et al. 2019	U-Net	2D	5452 slices (private)	Post-contrast	Cross-entropy	IoU = 0.7614	1.5T Siemens
	SegNet	2D	5452 slices (private)	Post-contrast	Cross-entropy	IoU = 0.6888	
Piantadosi et al. 2019	U-Net	2D	35 case (256x128x80) (private)	Pre-contrast, 2 min. post-contrast, and 6 min. post contrast	Dice	DSC = 0.6124	1.5T Siemens
Zhang et al. 2019a	Two hierarchical U-Nets	3D	272 cases (private)	Pre-contrast, post-contrast, and subtraction (breast mask guided)	First stage: Dice-sensitivity-like Second stage: Dice-like	DSC = 0.72	1.5T GE and 3.0T Siemens
Our proposed work	U-Net with ResNet basic blocks	3D	46 cases (public)	Pre-contrast, post-contrast, and std (ROI mask guided)	Cross-entropy	DSC = 0.645 (0.708 for primary lesions only)	1.5T GE

obtaining voxel-wise standard deviation across the time dimension. Moreover, different patch sizes, optimizers, loss functions were investigated.

Experiments were performed on 46 cases and DSC was used to evaluate the obtained segmentation. We obtained a mean dice of 0.645 (0.708 for main lesions only) which is promising considering the various issues encountered with the incomplete GT and the complicated dataset that included very small, irregular, low enhanced lesions as well as side lesions and confounding background.

Further improvements could be achieved by incorporating larger dataset with a complete annotation for those cases with multiple lesions. Moreover, using a breast mask instead of a simple ROI could also potentially alleviate the issue of confounding regions as it will help excluding confounding regions of the organs without excluding side lesions.

Finally, the deployment of deeper architecture and the deployment of an alternative way to represent the 4D volumes in a reduced parametric volume that better captures the TIC of each voxel could also improve the results.

7. Acknowledgments

I would like to thank my supervisor Dr. Robert Martí for the support, guidance and reviewing this manuscript. My thanks also to Joel Vidal for providing help with dataset preparation and Zohaib Salahuddin for his support and suggestions.

I would also like to thank the TCGA Breast Phenotype Research Group for providing the computer-extracted lesion segmentation data used in this study, which comes from the University of Chicago lab of Maryellen Giger.

References

ACS, 2019-2020. Breast cancer facts figures. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2019-2020.pdf. Accessed: June 2020.

Alzaghal, A.A., DiPiro, P.J., 2018. Applications of advanced breast imaging modalities. Current Oncology Reports 20, 57. doi:10.1007/s11912-018-0700-3.

ASCO, 2019. Guide to breast cancer. cancer.net.

Ashraf, A., Gavenonis, S., Daye, D., Mies, C., Rosen, M., Kontos, D., 2012. A multichannel markov random field framework for tumor segmentation with an application to classification of gene expression-based breast cancer recurrence risk. IEEE transactions on medical imaging 32. doi:10.1109/TMI.2012.2219589.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2481–2495.

Bria, A., Karssemeijer, N., Tortorella, F., 2013. Learning from unbalanced data: A cascade-based approach for detecting clustered microcalcifications. Medical image analysis 18, 241–252. doi:10.1016/j.media.2013.10.014.

Burnside, E., Drukker, K., Li, H., Bonaccio, E., Zuley, M., Ganott, M., Net, J., Sutton, E., Brandt, K., Whitman, G., Conzen, S., Lan, L., Ji, Y., Zhu, Y., Jaffe, C., Huang, E., Freymann, J., Kirby, J., Morris, E., Giger, M., 2015. Using computer-extracted image phenotypes from tumors on breast magnetic resonance imaging to predict breast cancer pathologic stage. Cancer 122. doi:10.1002/cncr.29791.

Chen, M., Zheng, H., Lu, C., Tu, E., Yang, J., Kasabov, N., 2018. A Spatio-Temporal Fully Convolutional Network for Breast Le-

sion Segmentation in DCE-MRI: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VII. pp. 358–368. doi:10.1007/978-3-030-04239-4_32.

Cheng, L., Li, X., 2013. Breast magnetic resonance imaging: kinetic curve assessment. Gland Surgery 2. doi:10.3978/j.issn.2227-684X.2013.02.04.

Christ, P., Ettlinger, F., Grün, F., Elshaera, M., Lipkova, J., Schlecht, S., Ahmaddy, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Hofmann, F., D'Anastasi, M., Ahmadi, S.A., Kaissis, G., Holch, J., Sommer, W., Braren, R., Heinemann, V., Menze, B., 2017. Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks .

Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., Tarbox, L., Prior, F., 2013. The cancer imaging archive (tcia): Maintaining and operating a public information repository. Journal of digital imaging 26. doi:10.1007/s10278-013-9622-7.

Clèrigues, A., Valverde, S., Bernal, J., Freixenet, J., Oliver, A., Lladó, X., 2019. Acute ischemic stroke lesion core segmentation in ct perfusion images using fully convolutional neural networks. Computers in Biology and Medicine 115, 103487. doi:https://doi.org/10.1016/j.compbiomed.2019.103487.

Degani, H., Gusis, V., Weinstein, D., Fields, S., Strano, S., 1997. Mapping pathophysiological features of breast tumors by mri at high spatial resolution. Nature Medicine 3, 780–782.

DeSantis, C.E., Ma, J., Gaudet, M.M., Newman, L.A., Miller, K.D., Goding Sauer, A., Jemal, A., Siegel, R.L., 2019. Breast cancer statistics, 2019. CA: A Cancer Journal for Clinicians 69, 438–451. doi:10.3322/caac.21583.

Dogan, B., Whitman, G., Kushwaha, A., Phelps, M., Dempsey, P., 2006. Bi-rads-mri: a primer. AJR. American journal of roentgenology 187, W152–60. doi:10.2214/AJR.05.0572.

El Adoui, M., Mahmoudi, S., Larhmam, A., Benjelloun, M., 2019. Mri breast tumor segmentation using different encoder and decoder cnn architectures. Journal of Computers 8, 52. doi:10.3390/computers8030052.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1915–1929.

Gubern-Mérida, A., Martí, R., Melendez, J., Hauth, J., Mann, R., Karssemeijer, N., Platel, B., 2014. Automated localization of breast cancer in dce-mri. Medical image analysis 20. doi:10.1016/j.media.2014.12.001.

Guerrero, R., Qin, C., Oktay, O., Bowles, C., Chen, L., Joules, R., Wolz, R., Valdes-Hernandez, M., Dickie, D., Wardlaw, J., Rueckert, D., 2017. White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks. NeuroImage: Clinical 17. doi:10.1016/j.nicl.2017.12.022.

Gupta, A., Metcalf, C., Taylor, D., 2017. Review of axillary lesions, emphasising some distinctive imaging and pathology findings. Journal of Medical Imaging and Radiation Oncology 61, 571–581. doi:10.1111/1754-9485.12579.

Iranmakani, S., Mortezazadeh, T., Sajadian, F., Ghaziani, M.F., Ghafari, A., Khezerloo, D., Musa, A.E., 2020. A review of various modalities in breast imaging: technical aspects and clinical outcomes. Egyptian Journal of Radiology and Nuclear Medicine 51, 57. doi:10.1186/s43055-020-00175-5.

Li, H., Zhao, R., Wang, X., 2014. Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification .

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440.

Losurdo, L., Basile, T., Fanizzi, A., Bellotti, R., Bottigli, U., Carbonara, R., Dentamaro, R., Diacono, D., Didonna, V., Lombardi, A., Giotta, F., Guaragnella, C., Mangia, A., Massafra, R., Tamborra, P., Tangaro, S., La Forgia, D., 2018. A gradient-based approach for breast dce-mri analysis. BioMed research international 2018, 9032408. doi:10.1155/2018/9032408.

Piantadosi, G., Marrone, S., Galli, A., Sansone, M., Sansone, C., 2019. Dce-mri breast lesions segmentation with a 3tp u-net deep convolutional neural network, in: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 628–633. doi:10.1109/CBMS.2019.00130.

Prastawa, M., Bullitt, E., Moon, N., Van Leemput, K., Gerig, G., 2004. Automatic brain tumor segmentation by subject specific modification of atlas priors. Academic radiology 10, 1341–8. doi:10.1016/S1076-6332(03)00506-3.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: MICCAI.

Samiei, S., Nijnatten, T., Beek, H., Polak, M., Maaskant-Braat, A., Heuts, E., Kuijk, S., Schipper, R.J., Lobbes, M., Smidt, M., 2019. Diagnostic performance of axillary ultrasound and standard breast mri for differentiation between limited and advanced axillary nodal disease in clinically node-positive breast cancer patients. Scientific Reports 9. doi:10.1038/s41598-019-54017-0.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., WOO, W.c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting .

Siegel, R.L., Miller, K.D., Jemal, A., 2020. Cancer statistics, 2020. CA: A Cancer Journal for Clinicians 70, 7–30. doi:10.3322/caac.21590.

Subbhuraam, V.S., Ng, E., Acharya, U.R., Faust, O., 2014. Breast imaging: A survey. World journal of clinical oncology 2, 171–178. doi:10.5306/wjco.v2.i4.171.

Tofts, P.S., 2010. T 1-weighted dce imaging concepts : Modelling , acquisition and analysis. Proceeding paper.

Vignati, A., Giannini, V., Luca, M., Morra, L., Persano, D., Carbonaro, L., Bertotto, I., Martincich, L., Regge, D., Bert, A., Sardanelli, F., 2011. Performance of a fully automatic lesion detection system for breast dce-mri. Journal of magnetic resonance imaging : JMRI 34, 1341–51. doi:10.1002/jmri.22680.

Wang, H., Yushkevich, P.A., 2013. Multi-atlas segmentation without registration: A supervoxel-based approach, in: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 535–542.

WHO, . Breast cancer. https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/. Accessed: June 2020.

Zhang, J., Gao, Y., Park, S.H., Zong, X., Lin, W., Shen, D., 2017. Structured learning for 3-d perivascular space segmentation using vascular features. IEEE Transactions on Biomedical Engineering 64, 2803–2812.

Zhang, J., Saha, A., Zhu, Z., Mazurowski, M.A., 2019a. Hierarchical convolutional neural networks for segmentation of breast tumors in mri with application to radiogenomics. IEEE Transactions on Medical Imaging 38, 435–447.

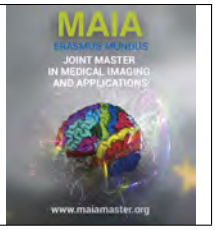
Zhang, L., Luo, Z., Chai, R., Arefan, D., Sumkin, J., Wu, S., 2019b. Deep-learning method for tumor segmentation in breast DCE-MRI , in: Chen, P.H., Bak, P.R. (Eds.), Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications, International Society for Optics and Photonics. SPIE. pp. 97 – 102. doi:10.1117/12.2513090.

Zheng, Y., Baloch, S., Englander, S., Schnall, M., Shen, D., 2007. Segmentation and classification of breast tumor using dynamic contrast-enhanced mr images, pp. 393–401. doi:10.1007/978-3-540-75759-7_48.

Çiçek, , Abdulkadir, A., Lienkamp, S., Brox, T., Ronneberger, O., 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation .

Medical Imaging and Applications

Master Thesis, August 2020



Survival Time Prediction of Metastatic Melanoma Patients by Computed Tomography using Convolutional Neural Networks

Zakia Khatun[1], Adrien Bartoli[2], Benoit Magnin[2]

[1] *Universite de Bourgogne (France), UNICLAM (Italy), and Universitat de Girona (Spain)*
[2] *EnCoV-TGI, Institut Pascal, Universite Clermont Auvergne, Clermont Ferrand, France*

Abstract

Metastatic melanoma is a fatal disease with a poor prognosis and rapid systemic spread. Follow-up and study are imperative, especially within the early periods after diagnosis as the expected cure is seldom obtained after surgical excision and with adjuvant therapy.
Typically, computed tomography (CT) scan contains a huge sum of data that ought to be completely analyzed and assessed by the radiologist or other healthcare proficient in a brief time. In this case, the CAD framework can be of extraordinary offer assistance as a moment supposition for experts. In spite of the fact that the CAD framework may play an imperative part in the analysis, small research has been published on the survival time prediction of metastatic melanoma based on the CAD system.
Our objective is to study the prediction of the survival time of patients with metastatic melanoma in terms of 1-year survival as a binary classification. Dataset used in this study contains CTs of 71 patients with metastatic melanoma who are studied at Universite Clermont Auvergne Hospital. The number of lesions per patient varies from 1 to 11.
To reach the objective, the survival time is anticipated using the accessible CT data as input of a 3D CNN. In this category, survival time is anticipated using full CT volumes and also from extracted 3D CT patches containing lesion regions. Here, the patches are extracted given the ground truth masks of the lesions. Moreover, segmentation of lesions coming from different organs is performed using two different 3D CNNs to examine the prediction of survival time based on newly extracted 3D CT patches. These new patches are extracted using our anticipated segmentation predicted masks of lesions. As the final test, it is also inspected whether aggregated deep segmentation feature map can help to predict survival time being an extra input channel to the CT data for 3D CNN or not.
Our study shows that using aggregated deep segmentation feature map as an extra input channel to CT data comes about in superior performance in survival time prediction compared to using as it were only 3D CT patches as input. In expansion, the prediction of survival time based on newly extracted 3D CT patches coming from our segmentation predicted masks is similar to the survival prediction using the 3D CT patches coming from ground truth masks.
Further investigation of this study can be the addition of radiomic features with aggregated deep segmentation feature map as additional input to CT data other than experimenting on bigger dataset. Including clinical data such as age, sex, etc. can as well play a vital role.

Keywords: Survival Time Prediction, Metastatic Melanoma, Computed Tomography, Lesion Segmentation, Deep Segmentation Feature Maps, Convolutional Neural Networks

1. Introduction

1.1. Metastatic Melanoma

Melanoma is considered to be the most aggressive skin cancer, with a worldwide increasing incidence (Rotaru et al., 2019). It is the fifth most common cancer among men and the sixth most common cancer among women. According to Cancer.Net Editorial Board, in 2020 an estimated 100,350 adults (60,190 men and 40,160 women) in the United States will be diagnosed with invasive melanoma of the skin.

It begins within the melanocytes. It occurs due to

a mutation in melanin producing skin cells. Once it spreads, or metastasizes, the disease is known as metastatic melanoma. Common sites for metastases incorporate the lymph nodes, lungs, liver, bones, and brain.

Metastatic melanoma occurs when the melanoma isn't identified and treated early. This sort of melanoma may typically happen during stage III or stage IV (Enninga et al., 2017).

1.2. Imaging Modalities

Common tests to find out if the cancer has spread to other parts of the body are X-rays, CT, MRI, PET, Blood tests, etc. The imaging modality used in this study is CT scan. Where images show soft tissue differentiated with anatomic detail, facilitating phenomenal demonstrative precision. Unlike ordinary x-rays, the detectors of the CT scanner don't create an image. The image of that segment is taken from different angles. This permits to recover the data on the profundity coming about 3D images with better subtle elements. In CT scan, Hounsfield Unit (HU) is proportional to the degree of x-ray attenuation and it is allocated to each pixel to show the image that represents the density of the tissue.

To evaluate the extension of the metastatic disease, CT is one of the most used modalities which gives the sites of lesion present and corresponding estimate, shape, and thickness information.

Figure 1 shows a 55 years old patient's liver metastases from metastatic melanoma (Ozaki et al. (2017)) which shows early pseudo-progression during treatment.

Figure 1: a) Liver metastases before the start of treatment (target lesion: 31 mm), b) CT at the 3-month assessment (target lesion: 63 mm), c) CT at the 5-month assessment (target lesion: 31 mm), associated with a change of the density, & d) CT at the 8-month assessment (target lesion: 20 mm)

1.3. Survival Analysis

In numerous cancer studies, the most outcome assessed is the estimation of the life expectancy of a specific study populace. It is also known as the 'Time to an Event of interest' analysis. This is often called survival time or event time. The objective is to estimate the time for an individual or a group of individuals to experience an event of interest. However, censorship is common, which implies that incomplete data is accessible on the survival time of some individuals. Censorship is a key feature that distinguishes survival analysis from other areas of statistics (Leung et al., 1997).

In addition, survival data are rarely distributed normally, but are biased and generally include many early events and relatively few late events. It is these characteristics of the data that make special methods called survival analysis necessary (Clark et al., 2003).

In figure 2, according to Melanoma Research Alliance, we can see that the survival rate of metastatic melanoma is really low compared to early stages of melanoma.

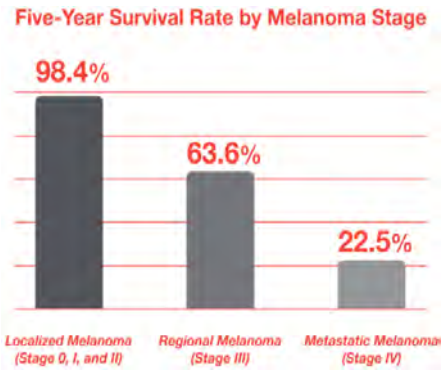


Figure 2: Survival rate of metastatic melanoma. (2020)

So, within the case of metastatic melanoma, survival analysis is significant which includes the modeling of time to event data.

Survival analysis endeavors to answer questions such as: what is the proportion of a populace that will survive past a certain time? Of those that survive, at what rate will they pass on or fail? Can different causes of passing or failure be taken into consideration? How do specific circumstances or characteristics increment or diminish the probability of survival?

To address the above-specified questions, there are researches where the baseline strategies incorporate the use of deep features, radiomic features, the combination of both features applying machine learning algorithms (e.g. Haarburger et al. (2018)). Where machine learning has seen progressions in numerous divisions like industry, scholarly, and extraordinarily medical domain.

1.4. Our Baseline

In our work, we aim to predict survival time in terms of 1-year survival. Where short survival incorporates

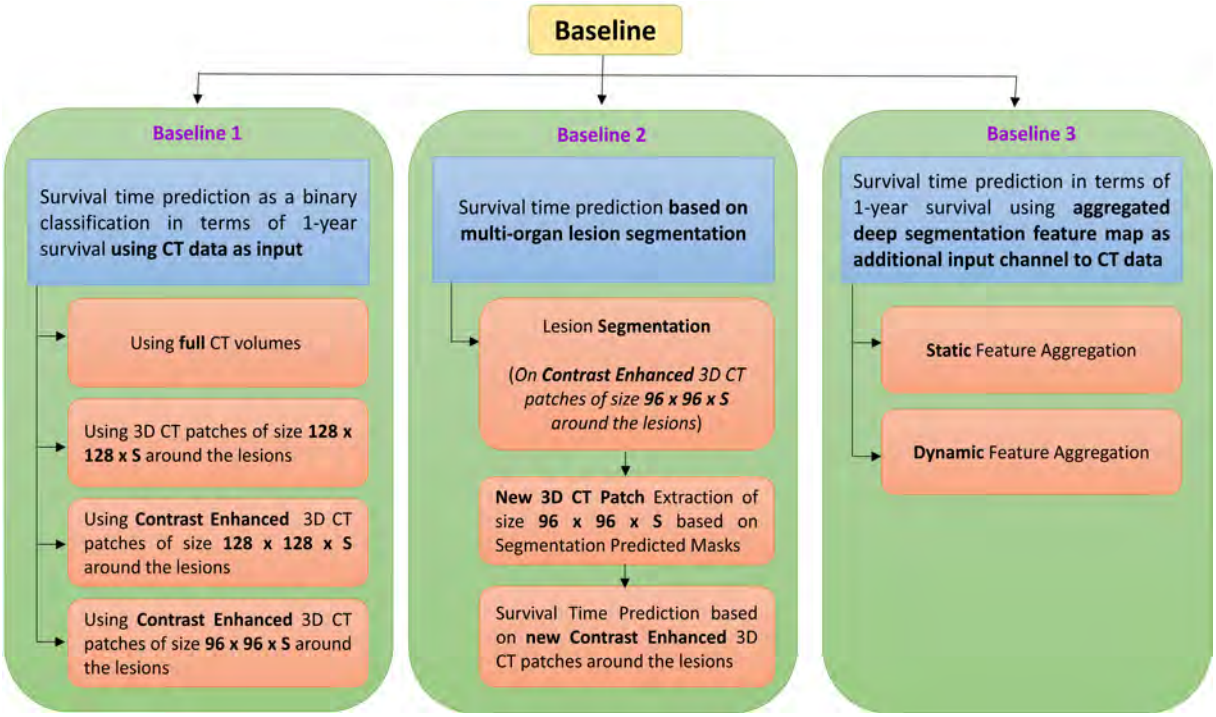


Figure 3: Baseline overview

the patients who survived less than 1 year and the patients who survived more than 1 year have a place to long survive. Indeed in spite of the fact that there is research for survival analysis utilizing machine learning but to our best knowledge, there's a really small work where deep learning has been addressed to anticipate the survival time within the shape of binary classification without using any radiomic feature.

For classification tasks, deep learning is getting to be a powerful tool for CAD frameworks day by day. Increased number of images, classification, and regression models in medical image analysis make deep learning amazingly prevalent. Deep learning is a course of machine learning algorithms propelled by the structure of a human brain. Deep learning algorithms utilize complex multi-layered neural networks, where the level of deliberation increments slowly by non-linear transformations of input data.

The main objective of our study is accomplished through three different baselines which are as follows:

i Survival time prediction as a binary classification in terms of 1-year survival using CT data as input

ii Survival time prediction based on multi-organ lesion segmentation

iii Survival time prediction in terms of 1-year survival using aggregated deep segmentation feature map as additional input channel to CT data

The remaining segment of this paper is organized as follows:

Section 2 is dedicated to the state of the art articles focusing on classification, segmentation, and classification using segmentation feature maps aggregation. Section 3 presents the data set and the description of the methodologies used. The key outcomes of different experiments are presented in section 4. Section 5 explains the discussion based on our investigation alongside the challenges and future work. And the work is concluded in section 6.

2. State of the art

Computer-aided diagnosis (CADx) / computer-aided detection (CADe) is rapidly entering the mainstream of medical imaging and diagnostic radiology. The main goal of CAD systems is to assist radiologists in image interpretations to identify abnormal signs at an earliest. The computer outputs serve as a second opinion to the radiologists to make a final decision. A CAD system may incorporate steps like image processing, image feature analysis, segmentation and data classification. To perform these tasks, deep learning has become an active area. In spite of being an active area, to our best knowledge, for survival time prediction not many researches have been conducted where the prediction is made as binary classification solely using deep learning without taking into consideration of any radiomics features or machine learning classifiers. So we have got inspiration from other close relevant state of the art methods.

This literature review is organized in three different sections which are responsible for classification task,

multi-organ lesion segmentation and classification using the aggregation of deep segmented feature maps.

2.1. Classification

Burgh et al. (2016) performed deep learning based prediction of survival on MRI in amyotrophic lateral sclerosis. Where 135 patients were classified into three survival duration subgroups. Total of four deep learning networks have been experimented in which clinical and imaging data are combined using layered deep learning. Their study concerns only one organ which is brain so their approach of adding brain morphology as input based on the cortical thickness and sub-cortical volume measurements is coherent. In our case, our CT data comes from different parts of the body and the morphology of different organs are not comparable.

Nie et al. (2019) performed multi-channel 3D deep feature learning for survival time prediction of brain tumor patients using multi-modal neuroimages for 68 patients. Here, their baseline consists of two parts, 1) 3D CNN-based deep learning to conduct feature learning, and 2) a SVM for final prediction (long or short OS). Their 3D CNN architecture is implemented by the widely used deep learning framework Caffe, to extract features from multi-modal brain images and their multi-channel metric maps in a supervised manner. These features are expected to classify individual image patches according to the survival of the patient. Then, the high-level features of all patches of the patient, as well as the important limited demographic and tumor-related features, are integrated to train the SVM classifier for the survival time prediction of the patients.

Han et al. (2019) studied 178 patients with High-Grade Gliomas for the prediction of survival. In their study, the interesting section is the deep feature extraction not from the whole input rather from the lesion regions only. Later, they combined the deep features with radiomics features.

Alexander et al. (2019) performed deep survival regression for metastatic colorectal cancer on three different CT datasets. They carried out two experiments. One of them is Cox regression and another one is deep survival regression. For the deep survival regression, they used a deep convolutional neural network based on ResNet.

Jiang et al. (2019) proposed breast cancer histopathological image classification using convolutional neural network with small SE-ResNet module. They proposed a novel CNN architecture for the classification of breast cancer histopathology images using the small SE-ResNet module.

Chen et al. (2019b) developed and validated a prognostic nomogram for recurrence-free survival after complete surgical resection of local primary gastrointestinal stromal tumors based on deep learning. In their study, a ResNet model was developed based on contrast-enhanced computed tomography (CE-CT) in a training cohort consisted of 80 patients. The patients were pathologically diagnosed with gastrointestinal sromal tumors (GISTs) and validated in internal and external validation cohort respectively.

2.2. Segmentation

This section is dedicated to the literature review for the segmentation of lesion coming from different organs of the body.

In both BraTS 2017 and 2018, besides the segmentation of tumor sub-structure another goal was to predict overall survival. For the task of patient overall survival (OS) prediction, once the participants produced their segmentation labels in the pre-operative scans, they were called to use these labels in combination with the provided mpMRI data to extract imaging/radiomic features. The features which were considered as appropriate, those were asked to analyze through machine learning algorithms, to predict patient overall survival.

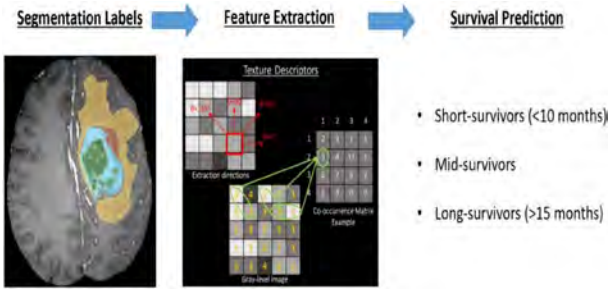


Figure 4: Illustrative pipeline example for predicting patient overall survival (Bakas et al., 2018)

Isensee et al. (2018) conducted brain tumor Segmentation and radiomics survival prediction as a contribution to the BRATS 2017 challenge. Their work was inspired by the U-Net architecture. They designed the network to process large 3D input blocks of 128x128x128 voxels. Just like the original U-Net, their architecture comprises a context aggregation pathway that encodes increasingly abstract representations of the input as they progress deeper into the network. And that is followed by a localization pathway which recombines these representations with shallower features to precisely localize the structures of interest.

Myronenko (2018) experimented 3D MRI brain tumor segmentation using auto encoder regularization. In his approach, a larger encoder part is in charge of the image features extraction using the ResNet (He et al. (2016)) blocks with normalization and ReLu as activation function. The decoder was dedicated to reconstruct the segmentation mask having a single block per spatial level.

Skourt et al. (2018) performed lung CT image segmentation using U-net architecture which performed really well which inspired them to apply for lung nodule segmentation as well.

In the work of Adoui et al. (2019), they studied two different fully CNN encoder–decoder type architectures, namely U-Net and SegNet to perform breast tumor segmentation. 86 volumes of DCE-MRI data were used. According to the radiologist, the predicted segmentation using U-Net showed better accuracy than the segmentation done by humans in some cases. However, SegNet's qualitative results were not very close to the ground truth.

Hosseinzadeh et al. (2019) proved that adding probabilistic zonal prior can be helpful for automatic detection of clinically significant prostate cancer (csPCa) in bi-parametric magnetic resonance imaging (bpMRI). In their work, using a 3D-UNet prostate probabilistic and deterministic zonal segmentations were generated which were later fed into a 2D-UNet as additional input channel which worked as weight map.

Ben-Cohen et al. (2016) studied liver segmentation and detection of liver metastases in CT using automated fully convolutional network. Their approach of using FCN with data augmentation, addition of neighbour slices, and appropriate class weights provided the best results compared to other state of the art sparse dictionary classification techniques and patch based CNN.

2.3. Aggregation of Deep Segmentation Features Maps

Saha et al. (2020) performed weakly supervised 3D classification of chest CT using aggregated multi-resolution deep segmentation feature maps. In their work, they used dual-stage convolutional neural network (CNN) to perform organ segmentation and binary classification of four representative diseases in lungs. For the final classification using 3D-ResNet, they added the aggregated deep segmentation feature maps as an additional input channel to the CT data. Their study concluded that adding segmentation feature maps as additional input channel improved their overall classification performance.

Wong et al. (2018) showed that segmentation network is helpful to build medical classifier even with a very limited amount of data. In their work, they concluded that by using a segmentation network pre-trained on similar data as the classification task, the machine can first learn the simpler shape and structural concepts before tackling the actual classification problem which usually involves more complicated concepts resulting better performance.

2.4. Ideas Implemented

From the above specified state of the art methods, one point to note down from the study of Dong et al. (2019), Han. et al (2019) and some other studies is the use of 3D small patches containing lesion regions rather than using the complete volume. Because the total volume does not contain similarly valuable data just like the lesion regions besides the issue of computational cost and not being viable as well. So, in our classification task for survival prediction, 3D CT patches containing lesion regions are used. And as classification network, ResNet is used as a reference of Alexander et al. (2019), Jiang et al. (2019) and Chen et al. (2019b) whose basic network was ResNet.

Studies like Bakas et al. (2018), Isensee et al. (2018) and Saha et al. (2020), it is clear that classification can take assistance of segmentation. So in our work apart from classification we focus on segmenting lesion as well which lesions come from different parts of the body. And as segmentation network, U-Net is used as other studies like Skourt et al. (2018), Adoui et al. (2019), Hosseinzadeh et al. (2019), Ben-Cohen et al. (2016) did for their segmentation cases. Besides, a FCN is studied as well.

Another interesting point of the literature review is the success of using segmentation features to guide classification studied by Saha et al. (2020), Wong et al. (2018) and also Figure 4 shows. In our research, once we get our best segmentation model, features are extracted from that to feed in the classification model as additional channel to CT input.

3. Material and Methods

3.1. Material

Our dataset contains CT scans of 71 patients with metastatic melanoma who has lived either more than a year or less than a year and this time is counted based on time between two follow ups. Their ages range from 27 to 89 years. These patients were studied in Universite Clermont Auvergne Hospital. They were treated with anti PD1 immunotherapy (nivolumabor/pembrolizumab). The size of each CT scan is 512 x 512 x S, where S represents a varying number of slices. The pixel spacing between slices is X x Y x 2 mm^3, where X and Y represent a varying number less than 1.

To identify metastases, images from the last CT scan before immunotherapy were visually assessed by radiologists. And each lesion was manually segmented by two radiologists with 4 and 8 years experience which resulted for each lesion in one volumic (3D) mask for the whole lesion and one 2D mask which is the slice containing the largest lesion. The number of lesions per patient varies from 1 to 11. And these lesions are mostly from abdomen and some from brain, neck and thorax.

Figure 5 shows the axial, coronal, and sagittal view of a patient data with metastatic melanoma.

3.2. Data Pre-processing

The format of raw CT data is converted from DICOM to NIFTI format which is a very simple to use and minimalistic data format. Based on experimental outcomes, different pre-processing techniques have been applied which are mentioned in separate sections as per experiment.

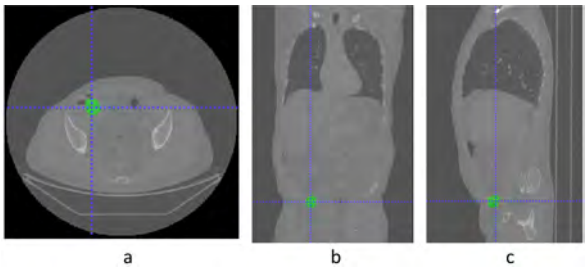


Figure 5: a) Axial, b) Coronal, & c) Sagittal view of a patient data

3.2.1. *Pre-processing 1*

As the pixel spacing between slices of each full CT volume is not constant so data resampling is performed using B-spline interpolation to voxels of size 1 x 1 x 1 mm^3. And these resampled CT datas are next normalized.

3.2.2. *Pre-processing 2*

From the full volume CT, 3D patches of size 128 x 128 x S are extracted which contain lesion regions. Here, S is a varying number as each lesion contains different number of slices. The process of 3D patch extraction is summarized in algorithm 1. And, the pixel spacing between slices is resampled to 1 x 1 x 1 mm^3 and normalization is performed.

Algorithm 1 summarizes the steps followed to extract 128 x 128 x S sized single patch from a full volume CT. The size of the full CT scan and 3D mask of it's each lesion is 512 x 512 x S. Across all the slices of that mask it is checked where the lesion (binary object) is present. Only those slices are taken into consideration. Among those slices, the slice containing largest lesion area is identified. The centroid of that particular mask slice is computed. From that centroid, a patch is cropped down from full CT of height and width of 128 (2 x 64) and number of slices depends on in how many slices the lesion is present.

As each patient has multiple lesions, so following the above mentioned algorithm in a loop, for each lesion individual 3D patch is extracted. As an example, If patient 1 has 5 different lesions (1 full volume CT but 5 different 3D masks of same size), it indicates that we have 5 different 3D CT patches for that particular patient.

3.2.3. *Pre-processing 3*

The default intensity values of the most of original CTs are in range of [-3024, 3071] HU. This intensity range is clipped to [-1000, 800] HU following a linear transformation which ensures a well covered spectrum of CT intensity values over CTs within the dataset. Few CTs specially covering the neck and thorax are in range of [-1024, 3071] HU and the CTs of brain are in range of [-3024, 2218] HU. These CTs are clipped to have an intensity in range of [-600, 800] HU. And these CTs are normalized as usual to ensure spatial uniformity.

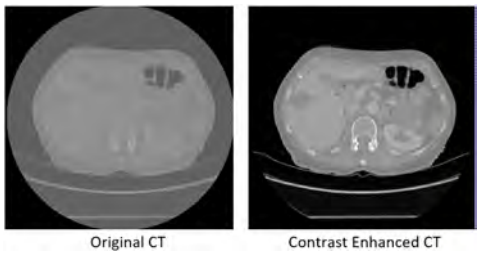


Figure 6: Intensity Clipping Effect

Figure 6 shows the effect of intensity clipping of a CT scan from [-3024, 3071] HU to [-1000, 800] HU.

input : 1 Full volume CT and Corresponding 1 3D mask containing single lesion (512 ×512×S)
output: 1 3D CT Patch (128 ×128×S) containing lesion

```
Finding the total number of slices
Finding binary object:
for i ← 1 to NumberOfSlices do
    ObjectArea = bwarea ( mask );
    if ObjectArea > 0 then
        Save the lesion areas per slice;
    end
end
Finding the maximum area among the saved lesion areas/slice
Identifying the centroid of the largest binary object
measurements = regionprops ( LargestBlob, 'Centroid' );
centroids = [ measurements.Centroid ];
Fetching x and y coordinates from the Centroid
Creating a cuboid object by setting Xlimits, Ylimts & Zlimits (First & Last slices containing lesion)
cuboid = images.spatialref.Cuboid ( [ (x-64) (x+64) ], [ (y-64) (y+64) ], [ S1 S2 ] );
Cropping and saving the cubiod from full CT
```

Algorithm 1: 3D CT patch extraction

After performing intensity clipping and normalization, 3D patches of size 96 x 96 x S are extracted following the same algorithm mentioned in algorithm 1.

Figure 7: Samples of extracted 3D CT patches

Figure 7 shows some of the 3D patches which are extracted using algorithm 1 mentioned earlier.

3.3. Methods Overview

3.3.1. Baseline 1 (Survival Time Prediction as a Binary Classification in terms of 1-year Survival using CT Data as Input)

In this baseline, survival time is anticipated through four different experiments which are explained below.

i Using Full CT Volumes

ii Using 3D CT Patches of size 128 x 128 x S around the lesions

iii Using Contrast Enhanced 3D CT Patches of size 128 x 128 x S around the lesions

iv Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions

3.3.1.1 *Using Full CT Volumes*

Data:

Pre-processing 1 mentioned in subsection 3.2.1 is used in this experiment which contains normalized and resampled full volume CTs of 71 patients.

As the volume of each CT is not constant, therefore to provide a constant volume at the network input, 20 random patches of size 112 x 96 x 96 (z, x, y) are taken from each full volume CT. Among 71 patients, we train on 80% of the data and test on the rest.

And as an increase in data, horizontal flip, vertical flip and random rotation of 45 degrees are applied in the training data.

Network Architecture:

The 3D CNN used in this experiment is based on a flexible Resnet architecture (He et al. (2015a)) using residual units proposed in the work of He et al. (2016). Figure 9 is an illustration of the proposed network architecture.

In the initial step, a 3D convolution is performed on the input data. Later in three different resolution scales, features are learned. In each resolution, 2 R-blocks (residual units) are used being inspired by the proposed

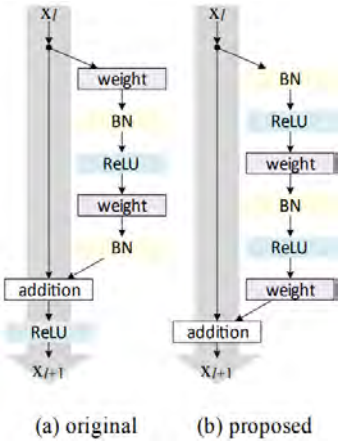


Figure 8: Residual Unit (R-Block), (He et al., 2016)

residual unit of (He et al., 2016). Figure 8 shows that proposed residual unit.

Keeping similarity with their proposed residual unit, each R-block (residual unit) used in our study consists of the following:

- Batch normalization:

  It is a mechanism that aims to stabilize the distribution of inputs to a given network layer during training. This is achieved by augmenting the network with additional layers that set the first two moments (mean and variance) of the distribution of each activation to be zero and one respectively. Then, the batch normalized inputs are also typically scaled and shifted based on trainable parameters to preserve model expressivity. This normalization is applied before the non-linearity of the previous layer (Santurkar et al., 2018).

- Activation:

  General rectified linear unit (ReLU) expedites convergence of the training procedure and leads to better solutions than conventional sigmoid like units (Nair and Hinton (2010), Maas et al. (2013)).

  Whereas parametric rectified linear Unit (PReLU) generalizes the traditional rectified unit. PReLU improves model fitting with nearly zero extra computational cost (He et al., 2015b).

$$f(x) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i y_i, , & \text{if } y_i \leq 0 \end{cases}$$

  Here y_i is the input of the nonlinear activation f on the i^{th} channel, and a_i is a coefficient controlling the slope of the negative part. The subscript i in a_i indicates that we allow the nonlinear activation to vary on different channels. When $a_i = 0$, it becomes ReLU; when a_i is a learnable parameter, we refer to the above mentioned equation as Parametric ReLU (PReLU).

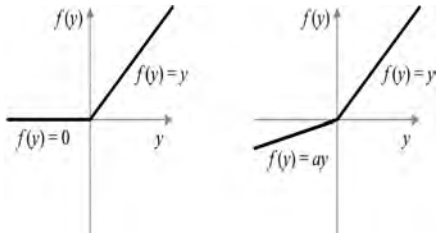


Figure 10: ReLU Vs. PReLU. For PReLU, the coefficient of the negative part is not constant and is adaptively learned ((He et al., 2015b)

Figure 10 shows the shapes of ReLU and PReLU. If a_i is a small and fixed value, PReLU becomes the Leaky ReLU (LReLU) (Maas et al. (2013)) (a_i = 0.01). The motivation behind using LReLU is to avoid zero gradients. In our study, as activation function LReLU is used.

- 3D Convolution:
  To extract spatial features on three dimensions, 3D convolution is performed. Due to computational expense and large number of trainable parameters, 3D kernels are not used rather 3 x 3 x 3 kernels are used which contain fewer weights and easier to convolve.

To preserve the time complexity of each layer / resolution, the size of the feature maps is halved by using 3D max pooling and the number of filters is doubled.

The total number of features at the last R-block of third resolution scale is 14 x 12 x 12 x 128. These features are later passed through a batch normalization layer along with Leaky ReLU activation function following a global average pooling layer. Lastly, a softmax activation is used for the final prediction of survival time.

Training:

20 random patches of size 112 x 96 x 96 (z, x, y) from each full volume CT training data are taken to train. As an adaptive learning rate optimization algorithm, adam optimizer is used which uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Sparse balanced cross entropy loss is used as a loss function because our data set is unbalanced. Depending on the frequency of the two classes available within the training and validation data, the weight is calculated accordingly to calculate the loss.

Batch size used to update weight is 4. To initialize weight, uniform transformation is used. The training is performed for 50,000 iterations using cyclic learning rate where the rate is in range of [0.00001, 0.00025].

The survival time prediction of each patient is computed by the majority voting of the predictions of all the random patches of that particular individual full volume CT.

3.3.1.2 *Using 3D CT Patches of size 128 x 128 x S around the lesions*

Data:

This experiment follows the pre-processing 2 mentioned in subsection 3.2.2. Here, instead of using several random patches from full volume CT like the previous experiment, 3D patches are extracted from each full volume CT according to the region of the lesions. This gave rise to 460 3D patches of size 128 x 128 x S for a total of 71 patients. Here, S is a varying number which represents the number of slices.

While training, to provide fixed size input to the network, 100 random patches of size 2 x 96 x 96 (z, x, y)

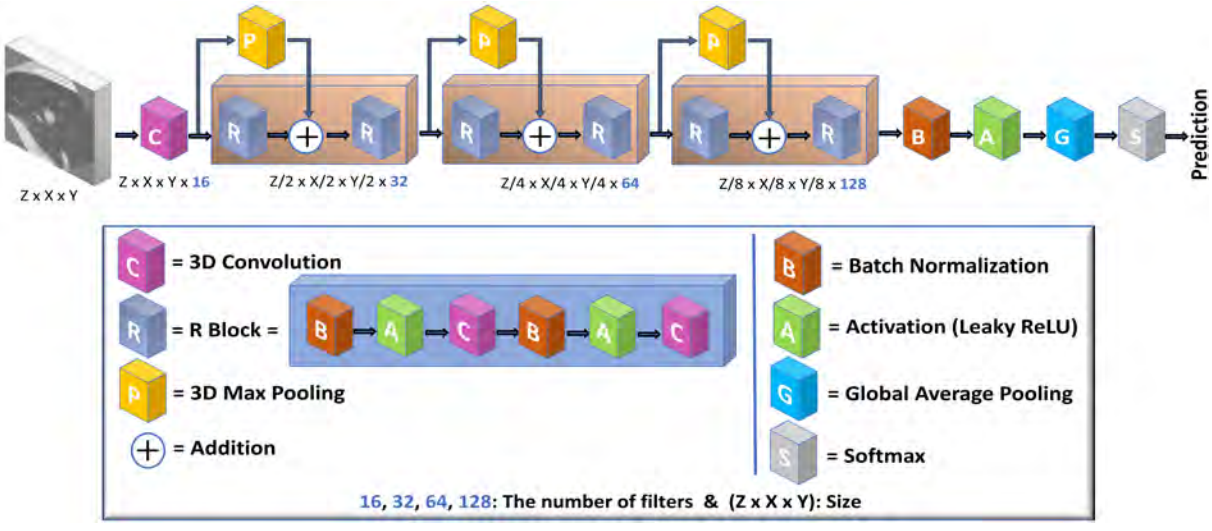


Figure 9: Baseline ResNet Architecture with 2 R-blocks in each resolution

are used from each extracted 3D CT patch of size 128 x 128 x S. The reason behind choosing only 2 slices along z is that the 3D patches contain slices between 2 and 45 slices. A few 3D patches contain only a single slice which have been removed from the data because they are single sliced while other 3D patches are volumic.

To increase the data, such as horizontal flip, vertical flip and random rotation of 45 degrees are used for training.

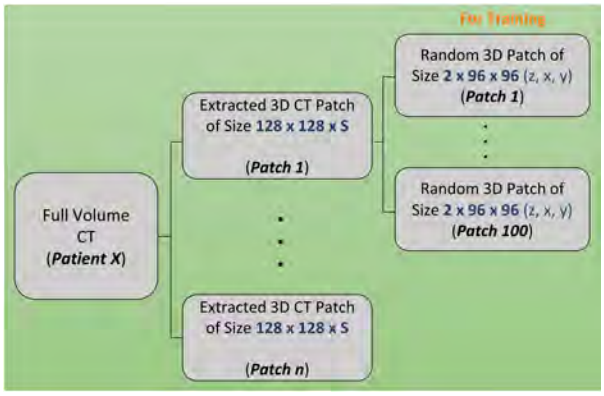


Figure 11: Patch Extraction Order

Figure 11 represents the patch extraction order from a single patient data. Here, 3D CT patches of size 128 x 128 x S are extracted following algorithm 1. Later, during training, from each physical cropped patch of size 128 x 128 x S, 100 random patches of size 2 x 96 x 96 (z, x, y) are used as network input.

Network Architecture:

The same ResNet architecture mentioned above is used for this experiment. Where a 3D convolution is initially applied in the input of shape 2 x 96 x 96 (z, x, y). Mode of padding used for convolution is *same*. And the features are learned in three different resolutions where each resolution scale contains 2 R-blocks (residual units). After the last R-block of third resolution, there are 1 x 12 x 12 x 128 features. Which are passed through a batch normalization layer and Leaky ReLU activation followed by a global average pooling layer. And the final survival time prediction comes from a softmax activation.

Training:

Similar training steps are followed for this experiment where 100 random patches of size 2 x 96 x 96 (z, x, y) are taken as fixed size input from 128 x 128 x S sized physical extracted 3D patches. Adam is used as an optimizer. Sparse balanced cross entropy as a loss function and weight initialization by uniform transformation. The training is performed for 50,000 iterations using cyclic learning rate in range of [0.00001, 0.00025].

The survival time prediction of each patient is computed by the majority voting of the predictions of all the corresponding 3D patches of size 128 x 128 x S. And the prediction of each 128 x 128 x S sized 3D CT patch comes from the majority voting of the predictions of the random patches of size 2 x 96 x 96 (z, x, y). This process can be easily understood from figure 11.

3.3.1.3 *Using Contrast Enhanced 3D CT Patches of size 128 x 128 x S around the lesions*

This experiment follows exactly the same network architecture and training steps as the previous experiment mentioned in subsection 3.3.1.2. Except this time, the extracted 3D CT patches of size 128 x 128 x S are enhanced in contrast using the intensity clipping method mentioned in pre-processing 3 (subsection 3.2.3).

The purpose of this experiment is to observe if there is any change in performance due to contrast enhancement.

3.3.1.4 *Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions*

Data:

Data used in this experiment follow pre-processing 3 mentioned in subsection 3.2.3. Each 3D patch is 96 x 96 x S in size which is smaller than previous experiments where the size of the extracted patches was larger. The reason behind this is discussed in the discussion section. As before, S is a variable number because each lesion contains a different number of slices.

To provide constant size input to the network, the variable slice numbers are padded to be a constant number using reflection mode of padding, which is 32 slices. After padding, each input size of the network becomes 32 x 96 x 96 (z, x, y).

As an increase in data, horizontal flip, vertical flip and random 45 degrees rotation are applied.

Network Architecture:

This experiment also follows similar ResNet architecture illustrated in figure 9. The input size of 32 x 96 x 96 (z, x, y) is initially passed through a 3D convolution and in three different resolutions, the features are learned. Here, at the end of the last R-block of the third resolution, the number of features is 4 x 12 x 12 x 128. These features are also transmitted through a batch normalization layer, activation Leaky ReLU following the global average pooling layer. And as before, a softmax activation is applied for the final prediction of the survival time.

Training:

Normalized, padded and contrast enhanced 3D patches of size 32 x 96 x 96 (z, x, y) are used for train-

ing. As an optimizer, adam is used. The loss of cross entropy function is used and the batch size used is 4. As before, training is performed for 50,000 iterations where the cyclic learning rate in the range [0.00001, 0.00025] is used. And the prediction of the survival time for each patient is based on the majority vote of the associated 3D patches which contain lesions for that particular patient.

3.3.2. Baseline 2 (Survival Time Prediction based on Multi-Organ Lesion Segmentation)

The motivation behind this experiment is to predict survival time based on newly extracted 3D CT patches which are extracted using our segmentation predicted masks and to compare the prediction outcome with the original 3D patches which are extracted using ground truth masks.

This experiment is divided into two main sections. The first section comprises the segmentation of lesions and in the second section new 3D CT patches are extracted based on our segmentation prediction masks on which survival is predicted.

3.3.2.1 *Lesion Segmentation using modified U-Net*

Data:

The data used in this experiment follow the pre-processing 3 mentioned in the subsection 3.2.3. Among the previous four experiments of baseline 1 mentioned above (prediction of survival time without any segmentation), the experiment using data of pre-processing 3 (subsection 3.3.1.4) performs better. This is why in this experiment, similar data will be used to compare performance using segmentation.

Pre-processed 3D patches are 96 x 96 x S in size. To provide constant size input to the network, the variable slice numbers are padded to be a constant number which is 32. After padding, each input size of the network is 32 x 96 x 96 (z, x, y).

As data augmentation, horizontal flip, vertical flip and random rotation of 45 degrees are applied.

Network Architecture:

In regular U-Net, there are a large number of feature channels in the up-sampling part, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting part, and yields a u-shaped architecture.

Figure 12 illustrates the U-Net architecture used in this study. This network is based on a flexible U-NET architecture (Ronneberger et al. (2015)) using residual units (He et al. (2016)) as feature extractors. Here too, the features are treated in three different resolutions where each resolution scale contains 2 R-blocks. Which basically sums up, each encoder and decoder unit contains 2 R-blocks where each R-block contains batch normalization, Leaky ReLU activation function and 3D convolution.

After sequential up-sampling and concatenation, the size of the features of the last R-block of the last decoder is 32 x 96 x 96 x 16. At the end, another 3D convolution is performed and for the final prediction of

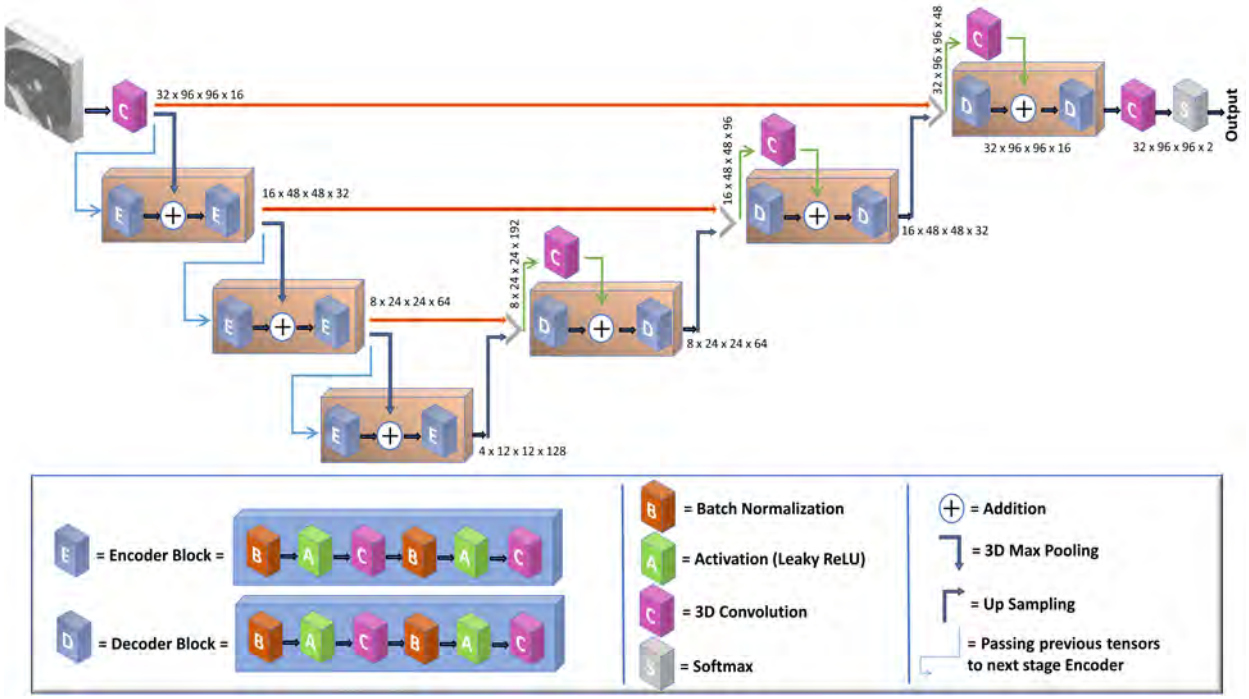


Figure 12: Baseline U-Net Architecture with 2 R-blocks in each resolution

the survival time, a softmax activation is used.

Training:

Normalized, contrast enhanced and padded data of size 32 x 96 x 96 is used for training. As optimizer, adam is used, which combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can manage sparse gradients on noisy problems. Considering different loss functions, two different experiments have been performed to compare the performance. One experiment uses the dice loss only as a loss function which calculates the average similarity of the class dice and another is the loss of dice with a loss of cross entropy. The batch size used to update the weight is 4. A uniform transformation is used to initialize the weight. Training is performed for 50,000 iterations using a cyclic learning rate where the rate is in the range of [0.00001, 0.00025]. This is how we perform lesions segmentation using modified UNet architecture.

3.3.2.2 *Lesion Segmentation using FCN*

Data:

Similar data (Normalized, contrast enhanced and padded of size 32 x 96 x 96) is used for lesion segmentation.

Network Architecture:

Figure 13 illustrates the image segmentation network used in this experiment. This network is based on a FCN architecture (Long et al., 2015) using residual units (He et al. (2016)) as feature extractors. Features are learned through three different resolutions where each resolution contains 2 R-blocks. At the end of the last R-block of the third resolution, number of features are 4 x 12 x 12 x 128. These features are up-sampled and added with the features learned in other resolutions which are passed through a 3D convolution and batch normalization layer. After third up-sampling, the available features are then transmitted through a 3D convolution and the final prediction comes from a softmax activation.

Training:

Similar training parameters like U-Net are performed but using the FCN network mentioned above.

Once the best segmentation network is identified, using the predicted masks, new 3D CT patches are extracted. Then, these new 3D CT patches are used to predict survival using the best classification model coming from baseline 1 (subsection 3.3.1). Figure 14 is a representation of the baseline.

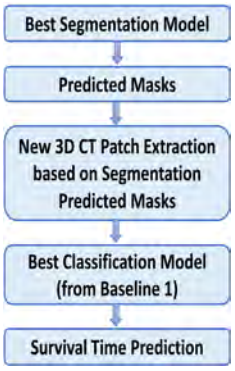


Figure 14: Baseline 2

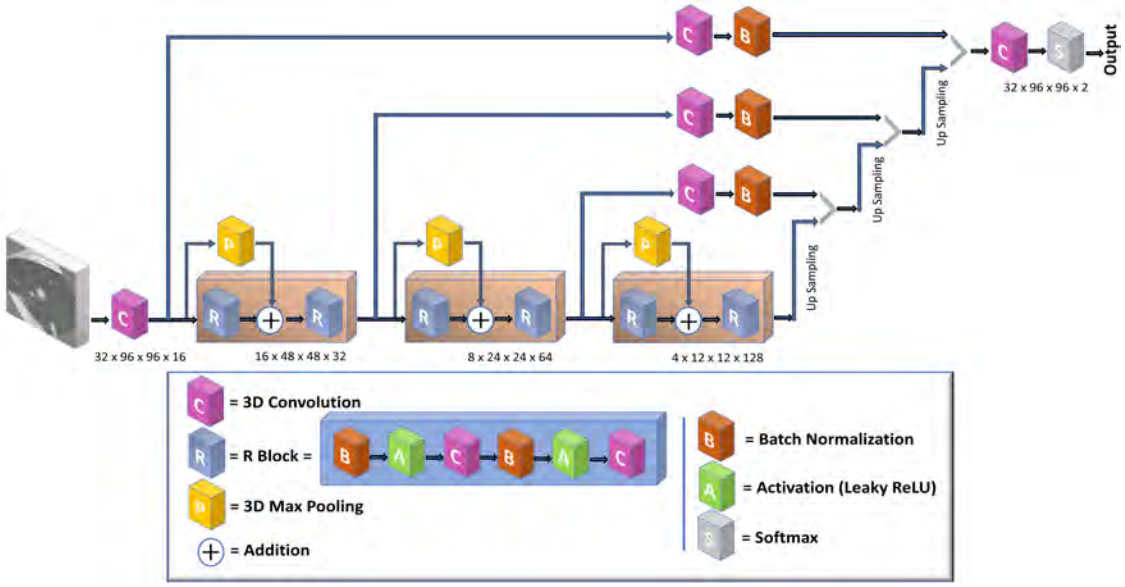


Figure 13: FCN Architecture with Identity Mapping and 2 R-blocks in each resolution

3.3.3. Baseline 3 (Survival Time Prediction in terms of 1-year Survival using Aggregated Deep Segmentation Feature Map as additional input channel to CT data)

In this experiment, survival time is predicted using similar ResNet architecture mentioned in figure 9. Here, as an additional input channel to the CT data, segmentation features extracted from the third resolution of the U-Net decoder block (last decoder) are used. These extracted segmentation feature maps are considered as an additional input channel for the CT data.

Data:

The data following pre-processing 3 mentioned in subsection 3.2.3 are used for this experiment. Each normalized and contrast enhanced 96 x 96 x S sized data are padded to be constant size of 32 x 96 x 96 (z, x, y).

As data augmentation, horizontal flip, vertical flip and random 45 degrees rotation are applied.

Network Architecture:

For this experiment, two different networks are used which is shown in figure 15. The first network is the U-Net network from where segmentation feature maps are extracted from the last decoder unit. Where the number of features is 32 x 96 x 96 x 16 which means the total number of feature maps is 16 and the size of these each feature map is 32 x 96 x 96 which is similar to the input CT size. As these feature maps will be used as additional input channel for CT so it is required to have the feature maps having same size of input CT.

Figure 16 shows an example case where the input and corresponding 16 segmentation feature maps are shown.

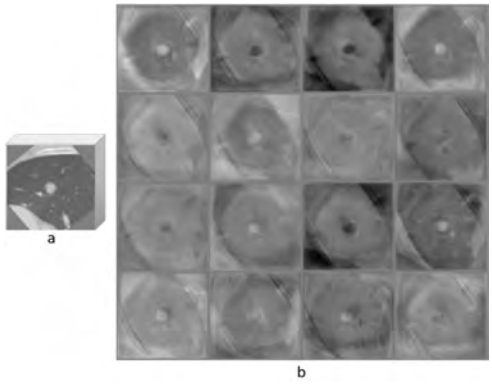


Figure 16: a) Input 3D CT Patch, & b)Segmentation Feature Maps

All these feature maps are passed on to the feature aggregator, which combines them to a single fused volume. Here, as feature aggregation, two different techniques are followed which are the following.

1. Static Feature Aggregation
   In this feature aggregation technique, in each feature map equal weight is applied, thereby numer-

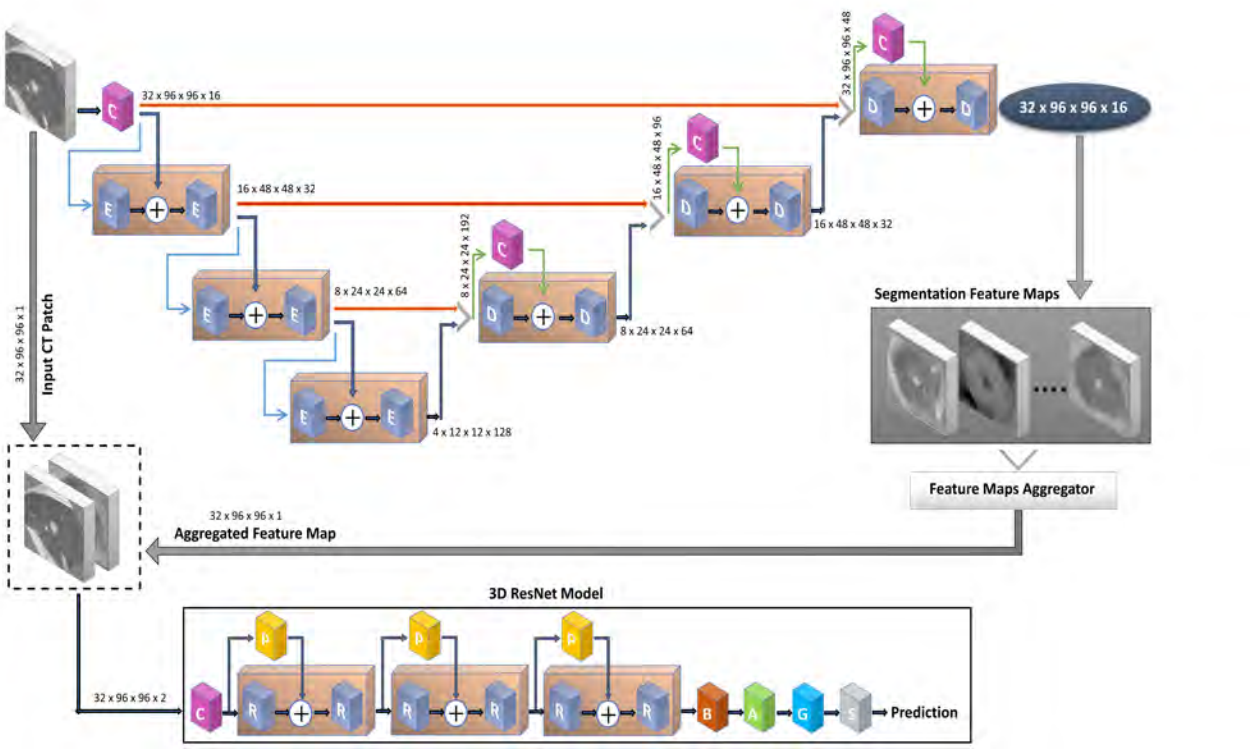


Figure 15: Dual Stage 3D CNN architecture with aggregated deep segmentation feature map as additional input channel

ically averaging all feature maps along equivalent voxels.

2. Dynamic Feature Aggregation
   In the case of dynamic aggregation, instead of averaging the feature maps all 16 feature maps are passed through a 1 x 1 3D convolutional layer. For this reason, the weight of each feature map is no longer averaged, but the weight is now learnable.

After aggregating the feature maps, this results in a single volume which is then concatenated with the input CT as an additional input channel. This input combined with 2 channels is next injected to a ResNet network (figure 9) for the survival time prediction.

Training:
Normalized, contrast enhanced and padded data of size 32 x 96 x 96 are trained for 50,000 iterations using adam optimizer with cyclic learning rate where Leaky Relu is used as activation function with sparse balanced cross entropy loss. Batch size used is 4. And the survival is predicted by the majority vote of 3D patches.

4. Results

In this section, the performances of the survival time prediction are presented in three separate sections according to the baselines. Experiments which are based on 3D CT patches (containing regions of lesions), their predictions are presented in two categories. One prediction is based on the lesions (prediction for each 3D patch). Another prediction is for each patient which comes from the majority vote of the 3D patch predictions.

Figure 17 illustrates lesion-wise vs. patient-wise survival time prediction.

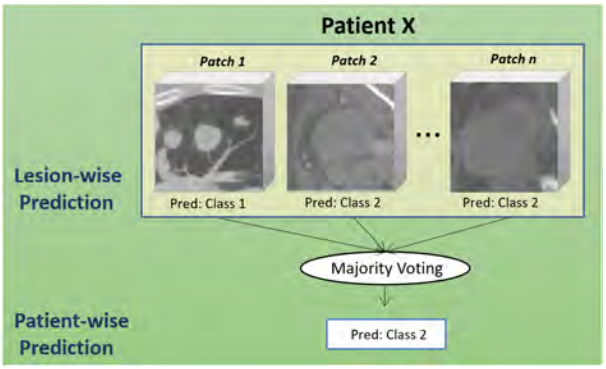


Figure 17: Lesion-wise Vs Patient-wise Prediction

4.1. Baseline 1 (Survival Time Prediction as a Binary Classification in terms of 1-year Survival using CT Data as Input)

This sub-section highlights the performances of baseline 1 survival time predictions. Here, the survival time prediction performances of four different experiments of baseline 1 are presented separately.

4.1.1. Using Full CT Volumes

As mentioned in sub-section 3.3.1.1, in this experiment 20 random patches of size 112 x 96 x 96 (z, x, y) are taken from each full volume CT as fixed size input. Table 1 shows the performance of survival time prediction in terms of accuracy, sensitivity and specificity. Here, the performance is assessed on 20 % of the data (test data) out of 71 available patients.

Table 1: Survival Time Prediction from Full CT Volumes

Accuracy	Sensitivity	Specificity
0.8000	1.0000	0.2500

4.1.2. Using 3D CT Patches of size 128 x 128 x S around the lesions

In this experiment, 100 random 3D patches of size 2 x 96 x 96 (z, x, y) are taken from each physical 3D patch of size 128 x 128 x S (contains lesion). Table 2 is the representation of the performance of predicting survival time per lesion and per patient. Here as well, survival is assessed on 20 % test data.

Table 2: Survival Time Prediction Using 3D CT Patches of size 128 x 128 x S around the lesions

Prediction	Accuracy	Sensitivity	Specificity
Lesion-wise	0.7078	0.7576	0.5652
Patient-wise	0.7894	0.8667	0.5000

4.1.3. Using Contrast Enhanced 3D CT Patches of size 128 x 128 x S around the lesions

Also in this experiment, 100 random 3D patches of size 2 x 96 x 96 (z, x, y) are taken from each physical 3D patch of size 128 x 128 x S (contains lesions) but this time the patches are contrast enhanced. Table 3 represents the performance of predicting survival time per lesion and per patient on 20% test data.

Table 3: Survival Time Prediction Using Contrast Enhanced 3D CT Patches of size 128 x 128 x S around the lesions

Prediction	Accuracy	Sensitivity	Specificity
Lesion-wise	0.7283	0.8030	0.5384
Patient-wise	0.7894	0.8667	0.5000

4.1.4. Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions

Here, 3D patches of size 32 x 96 x 96 (mentioned in 3.3.1.4, contain lesion regions) are used which are normalized like the other experiments but further improved

by contrast and is padded to have a similar spatial size for all inputs. Table 5 summarizes the performance of predicting survival time on 20% test data among 71 patients.

Table 5: Survival Time Prediction Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions

Prediction	Accuracy	Sensitivity	Specificity
Lesion-wise	0.8512	0.8788	0.7333
Patient-wise	0.8422	0.9333	0.5000

Table 4 is a summary of the prediction of survival time per patient of the four experiments belonging to baseline 1 where only CT data is used as input.

4.1.5. Average Performance of the Best Performing Experiment of Baseline 1

Table 4 clearly indicates that the experiment using contrast enhanced 3D patches of size 96 x 96 x S around the lesions performs the best.

This best performing experiment of this baseline is further tested in another **two** new data folds (Each fold contains 80 % data for training and 20 % data for prediction). Table 6 shows the average performance of **three** different folds.

Table 6: Survival Time Prediction Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions (Average performance of 3 different folds)

Prediction	Accuracy	Sensitivity	Specificity
Patient-wise	0.8771	0.9333	0.6667

4.2. Baseline 2 (Survival Time Prediction based on Multi-Organ Lesion Segmentation)

This area highlights the survival time prediction on the new 3D CT patches which are extracted using our best segmentation model predicted masks.

Data which performed the best in previous baseline (Fold 1 data used in the 4^{th} experiment of table 4) is used here to see if newly extracted 3D CT patches can perform similar to the previous 3D CT patches being extracted using ground truth masks. The performances of both segmentation and survival time prediction based on our segmentation are shown below.

4.2.1. Lesion Segmentation Performance

Lesions coming from different organs like brain, neck, thorax and abdomen are segmented by three different experiments. To observe the segmentation performance, the dice score is used which is shown below.

Here, the performance is assessed on 20% test data among 71 patients which contains 81 3D CT patches of size 32 x 96 x 96 (z, x, y).

Table 7: Lesion Segmentation Performance (CE: Cross Entropy)

Network	Loss Function	Avg Dice
Modified UNet	Dice Loss	0.7989
Modified UNet	Dice Loss + CE Loss	**0.8262**
FCN	Dice Loss + CE Loss	0.7761

Figure 18 represents a histogram of the dice scores of test 3D CT patches using the best segmentation model presented in table 7 (UNet with loss of dice and loss of cross entropy).

4.2.2. Survival Time Prediction based on New 3D CT Patches being extracted using our Segmentation Predicted Masks

Here, survival time is predicted using the best classification model from Table 4 (Using Contrast Enhanced 3D Patch of 96 x 96 x S around the lesion). And for prediction, the newly extracted 3D CT patches are used which are extracted using our predicted masks of the best segmentation model from Table 7 (modified UNet with loss of dice and loss of cross entropy).

Table 8: Survival Time Prediction based on New 3D CT Patches being extracted using our Segmentation Predicted Masks

Prediction	Accuracy	Sensitivity	Specificity
Patient-wise	0.8422	0.9333	0.5000

Table 8 shows that the prediction of survival time from new 3D patches predicted by our segmentation predicted masks is similar to the prediction on previous 3D CT patches being extracted using ground truth masks. The reason behind this similarity is discussed in the discussion section.

Table 4: Survival Time Prediction (**Patient-wise**) of **Baseline 1**

Approaches	Accuracy	Sensitivity	Specificity
Using Full CT Volumes	0.8000	1.0000	0.2500
Using 3D CT Patches of size 128 x 128 x S around the lesions	0.7894	0.8667	0.5000
Using Contrast Enhanced 3D CT patches of 128x128xS around the lesions	0.7894	0.8667	0.5000
Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions	**0.8422**	**0.9333**	**0.5000**

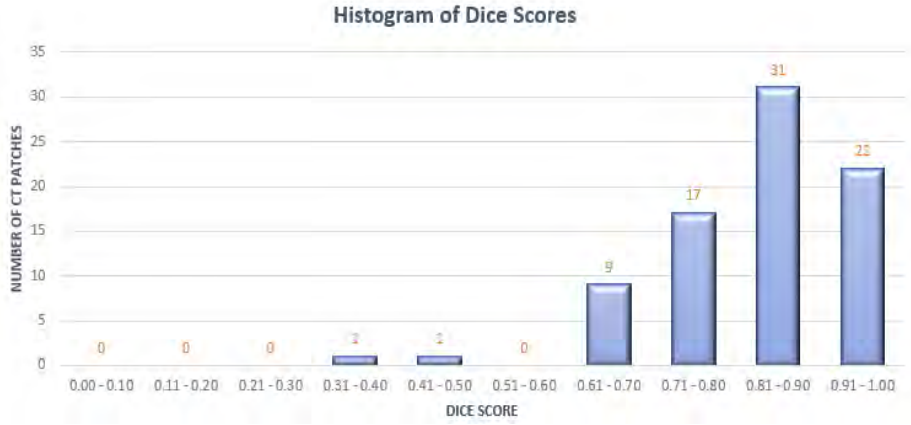


Figure 18: Dice Histogram

4.3. *Baseline 3 (Survival Time Prediction in terms of 1-year Survival using Aggregated Deep Segmentation Feature Map as additional input channel to CT data)*

This sub-section includes the performance where survival time is predicted using the volume of the aggregated deep segmentation feature map as an additional channel to the original input CT patch. Here, the result is displayed in two different sections depending on the type of feature aggregation.

4.3.1. *Static Feature Aggregation*

Table 9 is the representation of the survival time prediction using aggregated segmentation feature map as an additional input channel to the original CT patch. Here feature aggregation follows static manner mentioned in subsection 3.3.3.

Here, performance is assessed on 20% test data (Fold 1 data used in the 4^{th} experiment of table 4).

Table 9: Survival Time Prediction using Aggregated Deep Segmentation Feature Maps as Additional Input Channel

Prediction	Accuracy	Sensitivity	Specificity
Lesion-wise	0.8765	0.8667	0.8787
Patient-wise	0.8422	0.8667	0.7500

4.3.2. *Dynamic Feature Aggregation*

Here, survival time is predicted where the aggregation of segmentation feature maps follows the dynamic manner mentioned in the subsection 3.3.3.

Here as well, performance is assessed on 20% test data (Fold 1 data used in the 4^{th} experiment of table 4).

Table 10: Survival Time Prediction using Aggregated Deep Segmentation Feature Maps as Additional Input Channel

Prediction	Accuracy	Sensitivity	Specificity
Lesion-wise	0.8765	0.8000	0.8939
Patient-wise	0.8947	0.9333	0.7500

Table 11 is the summary result of the patient-wise prediction of the two types of feature aggregation methods (Static & Dynamic).

Table 11: Survival Time Prediction (Patient-wise) of **Baseline 3**

Approaches	Accuracy	Sensitivity	Specificity
Static F.A.	0.8422	0.8667	0.7500
Dynamic F.A.	**0.8947**	**0.9333**	**0.7500**

4.3.3. *Average Performance of the Best Performing Experiment of Baseline 3*

The best performing experiment of baseline 3 is the approach where feature maps are dynamically aggregated.

This best performing approach is tested on another **two** new folds (Each fold contains 80 % data for training and rest for prediction). And table 12 shows the average performance of **three** different folds.

Table 12: Survival Time Prediction Using Contrast Enhanced 3D CT Patches of size 96 x 96 x S around the lesions using Dynamic Feature Aggregation (Average performance of 3 different folds)

Prediction	Accuracy	Sensitivity	Specificity
Patient-wise	0.9122	0.9333	0.8334

All these above mentioned predictions are side by side presented in Appendix A.

5. Discussion

To achieve the objective of our study, three different baselines are followed.

In each baseline, different experiments are performed to find out the best performing experiment on that particular baseline. And these experiments are performed on 1 fold of data only (80% data for training and rest for prediction). Once the best performing experiment / approach for survival prediction of any baseline is identified, that particular experiment is tested on two additional new folds. Next, we find the average performance of those three different folds.

5.1. Baseline 1 (Survival Time Prediction as a Binary Classification in terms of 1-year Survival using CT Data as Input)

In this baseline, survival time is predicted when only CT data is used as input. In this category, the first experiment carried out uses the full CT volume. Initially, our goal was to check if we can identify survival from the entire CT volume without the aid of any mask. But since the sizes of input CTs of patients are not the same, several fixed size random patches are taken for training and predicting. From Table 4, we can see that by taking random patches from the full CT volumes, it isn't great to distinguish the negative class resulting in a specificity equal to 0.25.

Since not all slices contain a lesion region, it is not advisable to use the entire CT volumes. Thus, the next experiment is carried out by the extraction of 3D CT patches of size 128 x 128 x S which contain lesion areas. The reason for choosing the height and width of 128 x 128 is that it is the size that covers the largest lesion, even though the number of this size lesion is not much. Here, the number of slices (S) containing a lesion varies from 2 to 45 slices. In this experiment as well, since the sizes of the input volumes are not fixed so 100 random patches of size 2 x 96 x 96 (z, x, y) are used from each 128 x 128 x S sized extracted/cropped patch to feed in the network input. Here, the specificity improves by double but the sensitivity decreases.

In the next experiment, the 3D CT patches of size 128 x 128 x S are contrast enhanced by intensity clipping method. In a similar manner, 100 random patches of size 2 x 96 x 96 (z, x, y) are used from each contrast enhanced patch of size 128 x 128 x S. The purpose of this experiment is to observe the effect of contrast enhancement. From table 2 and table 3, we can see the performance improvement using the contrast enhancement in the lesion prediction even though the per patient prediction resulted same. It is well known that contrast enhancement can emphasize the overall or local characteristics of the images, clear the unclear image, emphasize certain features of interest, suppress features that are not of interest, enlarge the difference between the features of different objects in the images. It can improve image quality, enrich information, enhance image interpretation and recognition (Maini and Aggarwal, 2010). In the following experiments, contrast enhancement in data is used due to its enhanced performance.

In the previous experiment, as a fixed input size, 2 slices are taken along the z axis. In this case, lesions that contain a greater number of slices, even for them, we have always taken partial cuts of 2 slices. For example, when a lesion contains 30 slices, among these slices, it contains it's specific volumic contextual information. The network can get an idea of the axial, coronal and sagittal information better from a complete volumic angle. But when each time only 2 slices are chosen, the network gets closer to the information at the slice level and gets deprived of taking advantage of the volumetric nature of data. Thus, in the next experiment, instead of taking 2 slices only from the extracted 3D CT patches each time, the whole 3D CT patches are used.

In the fourth experiment, the patches of size 96 x 96 x S are extracted from the full CT volume using ground truth masks which contain lesion areas. The reason behind extracting patches of comparatively smaller size (96 x 96) than the previous experiments is that most lesions are small even though there are very few large lesions. In this experiment, the 3D CT patches are improved in contrast. This time also, the sizes of the input volumes are not fixed so all the volumes are padded to have a same size of 32 x 96 x 96 (z, y, x). If we see the performances, we can see that the prediction of the survival time of this experiment is better than the previous three experiments. The reason behind this is the effect of the contrast enhancement (Figure 6) and also this time the CT patches which contain a higher number of slices have not been cut to be a number of lower slices (2 slices) like the previous experiment.

Thus, among the four above mentioned experiments of baseline 1, the most successful experiment is the experiment using contrast enhanced 3D CT patches of size 96 x 96 x S (containing lesion areas). So, for the next experiments, this dataset is used. This experiment is also tested on two additional folds of data and the average performances of three individual folds are indicated in the table 6.

5.2. Baseline 2 (Survival Time Prediction based on Multi-Organ Lesion Segmentation)

In this baseline, the survival prediction is based on the segmentation of the lesions. And our intention is to observe whether performance guided by our segmentation can function similarly to performance guided by ground truth masks created by radiologists.

Unlike other human organs (e.g. brain, lung etc.), lesions from different organs are different in nature. Most of the lesions are pulmonary lesions and some of them are brain lesions and lesions from neck and thorax. A pulmonary lesion is a much smaller microstructure. In addition, pulmonary nodules are charac-

terized by blurred boundaries, large shape deformation, and rich texture information that is very different from that of normal human organs (Chen et al., 2019a). It is therefore more challenging to segment them compared to other larger organs and in addition the lesions come from different organs of different nature.

From Table 7, we can see the performance of the segmentation of lesions using two different networks. First, a UNet architecture mentioned in figure 12 is studied using the loss of dice only where the average dice score is 0.7989. Later, as a loss function, the combination of dice loss and cross entropy loss are used which gives better result, as we can see from Table 7. So, this proves that adding cross entropy loss with dice loss works better in segmentation. With this combination of loss function, a FCN network is also experimented to compare performance. It turns out that the modified UNet works better than the FCN network mentioned in figure 13.

Thus, the best performer of the three experiments is the modified UNet with loss of dice and loss of cross entropy. Figure 19 shows some of the predictions using the best segmentation network.

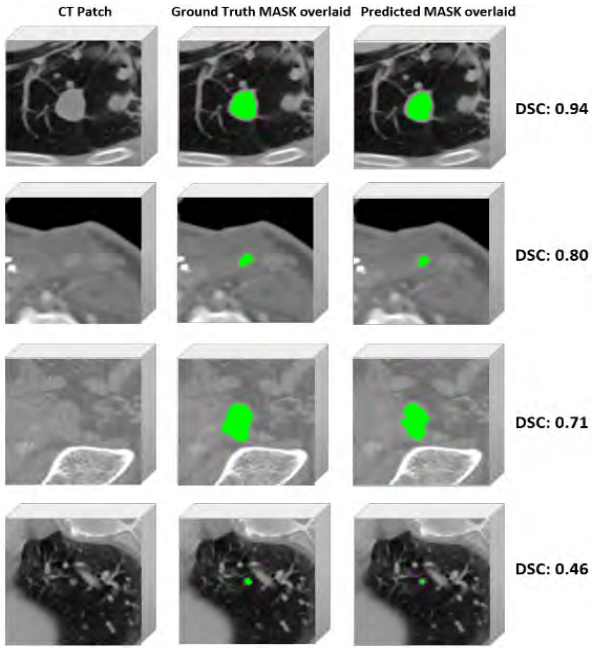


Figure 19: Lesion Segmentation Performance

From figure 19, we can see that even for a very low dice score of 0.46, the segmentation network can identify the real lesion. And from the histogram shown (figure 18), we can see that there is no case where the segmentation network fails to predict the lesion. So basically, for all lesions the network can identify the lesion. And which is very close to the truth.

Now our goal is to extract new 3D CT patches from the original CT volume based on our predicted segmentation masks instead of using the ground truth masks this time. Once the patches predicted by segmentation masks are extracted, the survival time is predicted using the best classification model from the baseline 1.

After extracting new 3D CT patches, it turns out that the patches predicted by our segmentation are very very close to the original 3D CT patches. The reason for this similarity is explained below.

Let's consider a case of poor dice score of 0.46 shown in Figure 19. From the segmentation predicted mask, we go back to the original full volume mask space to calculate the centroid of the binary object. And from this new centroid, we extract a new 3D CT patch from full CT according to algorithm 1. If we look very closely at the centroid of the original mask and our predicted segmentation mask, we can see that they are very close to each other. Also in other cases, the new centroids are very close to the centroids of the original lesions. This results in new CT patches very similar to the one extracted using ground truth mask. Even though the newly extracted patches are very little different (slightly shifted), but all of the new 3D CT patches contain the entire true lesion. So basically the 3D CT patches extracted using the ground truth mask and our segmentation predicted mask contain the entire area of the lesions. And since the final prediction actually comes from the region of the lesion, the prediction of survival time is therefore similar in both cases.

5.3. Baseline 3 (Survival Time Prediction using Aggregated Deep Segmentation Feature Maps as Additional Input Channel)

In this baseline, survival time is predicted using the aggregated segmentation feature map as an additional input channel for the CT input patch. And feature aggregation is done in two different ways which are static aggregation and dynamic aggregation. At the start, using only single fold data (80% for training and rest data for prediction), two of these experiments are performed. Later, the best performing experiment is further tested on two additional new folds.

In the case of static aggregation, each feature map receives a similar weight thus ensures the overall network stability. Being an additional input channel to CT patches, survival time prediction performance improves comparatively. From Table 9, we can see that the accuracy of the patient-wise prediction remains the same for the best performing network coming from baseline 1 (Table 4, experiment 4). But the specificity improves even if the sensitivity becomes a little lower.

The convolution layer in dynamic aggregation allows each feature map to receive a learnable weight. By convolution, it calculates the low level feature representations which has a property to preserve the spatial or positional relationships between input data points. The feature maps obtaining a learnable weight is better than the simple average to obtain an equal weight. Table 11 reflects this. In this experiment of dynamic aggregation, the sensitivity improves and the accuracy becomes

higher compared to all the other experiments discussed previously.

As the experiment of using dynamic feature aggregation outperforms static aggregation, this experiment is further tested on two additional folds. The average performance of the three folds is presented in the table 12 which even surpasses the best performance of baseline 1 presented in table 6.

Thus, we can see that among all the experiments of three different baselines, the dynamic aggregation of the feature maps as an additional input channel to 3D CT patch (containing lesion regions) is the best performing. In this approach, the network not only obtains the input CT patch like baseline 1 and baseline 2, but also obtains the mapping information from the segmentation. This appears as a valuable demeanor in predicting survival time.

5.4. Challenges

One of the main challenges of this study is that some of the available data on patients is censored, which means that their events are not known. In our study, it is assumed that those patients who are censored have the same survival perspective as uncensored patients. In addition, all patients are carriers of the same disease and the number of patients is small. Another challenge is the segmentation of lesions where the lesions come from different organs, their types and shapes are also different and some of them are very tiny.

5.5. Future Work

In the context of future work, it would be interesting to study the extraction of deep segmentation features not only from the last decoder block of UNet, but from different resolution levels. DenseVNet may be an option for this purpose.

In addition, besides studying deep features only, it will be worth studying the addition of clinical data (e.g. age, sex) to predict survival time.

Another experiment may also be the combination of clinical data, deep features and radiomic data which are the quantitative or semi-quantitative features from medical images.

Last but not least, more data will be a plus for this study as we know that the amount of data affects the deep learning approach.

6. Conclusions

In this study, survival time of patients with metastatic melanoma is predicted using different 3D CNNs where a baseline is based on CT data as conventional classification problem where two other baselines are based on segmentation. This study demonstrate that aggregated deep segmentation feature map being an additional input to CT data can play a critical role in predicting survival time. In our study, the patients are grouped into two different classes according to their survivals. One group belongs to the short survival class whose survival is less than 1-year and the patients of long survival class survived for more than 1-year.

To experiment, in addition of using CT input data, an aggregated segmentation feature map is added as an additional input channel to the CT data. At the first baseline, survival time is predicted using only CT data where no segmentation is involved. Among the four experiments tested according to the concept of using CT data only, the experiment where 3D contrast enhanced CT patches of size 96 x 96 x S are used (later padded to be 32 x 96 x 96) provides the best result. It performs best over the experiments using the entire CT volume or using a small fixed number of sliced CT patches. In this experiment, the improvement in contrast and the use of full lesion areas (3D) made this experiment better in the baseline 1. In the next baseline, survival time is predicted based on newly extracted 3D CT patches predicted by our segmentation. And for the segmentation of lesions, different models (3D UNet, FCN) are tested with different loss functions such as loss of dice only and the combination of loss of dice with loss of cross entropy. Our study shows that in our lesion segmentation case, the combination of dice loss and cross-entropy loss performs the best. And since no case has failed in the segmentation of the lesions and the CT patches predicted by the new segmentation are very close to the original 3D CT patches and cover the real lesion regions, the prediction is similar to the prediction using 3D CT patches extracted using the ground truth masks. In the third baseline, our two-stage CNN architecture makes full use of the segmentation model by taking advantage of it's deep segmentation feature maps. And the dynamically aggregated segmentation feature maps being additional input channel surpasses the performance of regular classification network.

7. Acknowledgments

I would like to thank my supervisors, Professor Dr. Adrien Bartoli and Dr. Benoit Magnin, as well as the EnCoV research team for providing me with all of the materials, infrastructure and continuous support to conduct this study. Gratitude goes to NVIDIA as well to support our study by GeForce GTX 1080. Special thanks go to Gulnur Semahat Ungan and Fakrul Islam Tushar (MAIA 2^{nd} batch) for their continued support and motivation. I would also like to thank Anindo Saha (MAIA 3 rd batch) for our discussion. I want to thank all members of the EnCoV research team for their support and friendliness. I am grateful to my colleague Yamid Espaniel (Ph.D) for having always been so united in all aspects. I would also like to thank the MAIA family and the European Union for this wonderful journey. Last but not least, I want to thank my creator and my two family

members for permitting me to come overseas and to realize one of my dreams through the MAIA team and for trusting me as always.

References

, 2020. Melanoma: Statistics. https://www.cancer.net/cancer-types/melanoma/statistics. Accessed: 2020.07.10.

, 2020. Melanoma survival rates. https://www.curemelanoma.org/about-melanoma/melanoma-staging/melanoma-survival-rates. Accessed: 2020.07.10.

Adoui, M.E., Mahmoudi, S.A., Larhmam, M.A., Benjelloun, M., 2019. Mri breast tumor segmentation using different encoder and decoder cnn architectures. Computers doi:10.3390/computers8030052.

Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R.T., Berger, C., Ha1, S.M., Rozycki, M., et al., 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge doi:10.17863/CAM.38755.

Ben-Cohen, A., Diamant, I., Klang, E., Amitai, M., Greenspan, H., 2016. Fully convolutional network for liver segmentation and lesions detection. MICCAI 2016 - Deep Learning and Data Labeling for Medical Applications doi:10.1007/978-3-319-46976-8_9.

Burgh, H.K.D., Schmidt, R., Henk-JanWesteneng, Reus, M.A., den Berg, L.H., den Heuve, M.P., 2016. Deep learning predictions of survival based on mri in amyotrophic lateral sclerosis. NeuroImage: Clinical 13. doi:10.1016/j.nicl.2016.10.008.

Chen, S., Ma, K., Zheng, Y., 2019a. Med3d: Transfer learning for 3d medical image analysis, arxiv:1904.00625v4, URL: https://arxiv.org/abs/1904.00625.

Chen, T., Liu, S., Li, Y., Feng, X., Xiong, W., Zhao, X., Yang, Y., Hu, C.Z.Y., Chen, H., Lin, T., Zhao, M., Liu, H., Yu, J., Xu, Y., Zhang, Y., Lia, G., 2019b. Developed and validated a prognostic nomogram for recurrence-free survival after complete surgical resection of local primary gastrointestinal stromal tumors based on deep learning. EBioMedicine 39, 272–279. doi:10.1016/j.ebiom.2018.12.028.

Clark, T.G., Bradburn, M.J., Love, S.B., Altman, D.G., 2003. Survival analysis part i: Basic concepts and first analyses. British Journal of Cancer 89, 232–238. doi:10.1038/sj.bjc.6601118.

Enninga, E.A.L., Moser, J.C., Weaver, A.L., Markovic, S.N., Brewer, J.D., Leontovich, A.A., Hieken, T.J., Shuster, L., Kottschade, L.A., Olariu, A., Mansfield, A.S., Dronca, R.S., 2017. Survival of cutaneous melanoma based on sex, age, and stage in the united states, 1992–2011. Cancer Medicine 6, 2203–2212. doi:10.1002/cam4.1152.

Haarburger, C., Weitz, P., Rippel, O., Merhof, D., 2018. Image-based survival analysis for lung cancer patients using cnns, arxiv:1808.09679v2, URL: https://arxiv.org/abs/1808.09679, doi:10.1109/ISBI.2019.8759499.

Han, W., Qin, L., Bay, C., Chen, X., Yu, K.H., Miskin, N., Li, A., Xu, X., Young, G., 2019. Deep transfer learning and radiomics feature prediction of survival of patients with high-grade gliomas. American Journal of Neuroradiology doi:10.3174/ajnr.A6365.

He, K., Zhang, X., Ren, S., Sun, J., 2015a. Deep residual learning for image recognition, arxiv:1512.03385v1, URL: https://arxiv.org/abs/1512.03385.

He, K., Zhang, X., Ren, S., Sun, J., 2015b. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, arxiv:1502.01852v1, URL: https://arxiv.org/abs/1502.01852.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks, arxiv:1603.05027v3, URL: https://arxiv.org/abs/1603.05027.

Hosseinzadeh, M., Brand, P., Huisman, H., 2019. Effect of adding probabilistic zonal prior in deep learning-based prostate cancer detection, arxiv:1907.12382v1, URL: https://arxiv.org/abs/1907.12382.

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H., 2018. Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge, arxiv :1802.10508v1, URL: https://arxiv.org/abs/1802.10508.

Jiang, Y., Chen, L., Zhang, H., Xiao, X., 2019. Breast cancer histopathological image classification using convolutional neural networks with small se-resnet module. PLOS One doi:10.1371/journal.pone.0214587.

Leung, K.M., Elashoff, R.M., Afifi, A.A., 1997. Censoring issues in survival analysis. Annual Review of Public Health 18, 83–104. doi:https://doi.org/10.1146/annurev.publhealth.18.1.83.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. Proc. icml 30.

Maini, R., Aggarwal, H., 2010. A comprehensive review of image enhancement techniques. JOURNAL OF COMPUTING 2. URL: https://arxiv.org/abs/1003.4053.

Myronenko, A., 2018. 3d mri brain tumor segmentation using autoencoder regularization, arxiv:1810.11654v3, URL: https://arxiv.org/abs/1810.11654.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on International Conference on Machine Learning , 807–814.

Nie, D., Lu, J., Zhang, H., Adeli, E., Wang, J., Yu, Z., Liu, L., Wang, Q., Wu, J., Shen, D., 2019. Multi-channel 3d deep feature learning for survival time prediction of brain tumor patients using multi-modal neuroimages. Scientific Reports 9. doi:10.1038/s41598-018-37387-9.

Ozaki, Y., Shindoh, J., Miura, Y., Nakajima, H., Oki, R., Uchiyama, M., Masuda, J., Kinowaki, K., Kondoh, C., Tanabe, Y., Tanaka, T., Haruta, S., Ueno, M., Kitano, S., Fujii, T., Udagawa, H., Takano, T., 2017. Serial pseudoprogression of metastatic malignant melanoma in a patient treated with nivolumab: a case report. BMC Cancer 17. doi:10.1186/s12885-017-3785-4.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, arxiv:1505.04597v1, URL: https://arxiv.org/abs/1505.04597.

Rotaru, M., Jitian, C.R., Iancu, G.M., 2019. A 10-year retrospective study of melanoma stage at diagnosis in the academic emergency hospital of sibiu county. Oncology Letters 17, 4145–4148. doi:10.3892/ol.2019.10098.

Saha, A., Tushar, F.I., Faryna, K., D'Anniballe, V.M., Hou, R., Mazurowski, M.A., M.D., G.D.R., Lo, J.Y., 2020. Weakly supervised 3d classification of chest ct using aggregated multiresolution deep segmentation features. SPIE Medical Imaging doi:10.1117/12.2550857.

Santurkar, S., Tsipras, D., Ilyas, A., Madry, A., 2018. How does batch normalization help optimization? arxiv:1805.11604v5, URL: https://arxiv.org/abs/1805.11604.

Skourt, B.A., Hassani, A.E., Majda, A., 2018. Lung ct image segmentation using deep neural networks. Procedia Computer Science 127, 109–113. doi:10.1016/j.procs.2018.01.104.

Wong, K.C., Syeda-Mahmood, T., Moradi, M., 2018. Building medical image classifiers with very limited data using segmentation networks. Medical Image Analysis 49, 105–116. doi:10.1016/j.media.2018.07.010.

Appendix A. Survival Time Predictions

The table represented in figure 20 is a side by side representation of all the predictions mentioned earlier. And this clearly demonstrates that survival time prediction based on aggregated deep segmentation feature map of our two-stage CNN architecture as additional input channel outperforms conventional classification network.

Baseline No	Experiment	Tested on	Type	Accuracy	Sensitivity	Specificity
1	1. Using full CT volume	1-fold	Patient-wise	0.8000	1.0000	0.2500
	2. Using 3D CT patches of size **128 x 128 x S** around the lesion	1-fold	Lesion-wise	0.7078	0.7576	0.5652
			Patient-wise	0.7894	0.8667	0.5000
	3. Using **Contrast Enhanced** 3D CT patches of size **128 x 128 x S** around the lesion	1-fold	Lesion-wise	0.7283	0.8030	0.5384
			Patient-wise	0.7894	0.8667	0.5000
	4. Using 3D CT patches of size **96 x 96 x S** around the lesion	1-fold	Lesion-wise	0.8512	0.8788	0.7333
			Patient-wise	**0.8422**	**0.9333**	**0.5000**
	Performance of Experiment 4 (Avg. of 3-folds)	**3-folds**	**Patient-wise**	**0.8771**	**0.9333**	**0.6667**
2	Prediction based on **newly** extracted 3D CT patches of size **96 x 96 x S** around the lesion using **our segmentation prediction masks**	1-fold	Patient-wise	**0.8422**	**0.9333**	**0.5000**
3	1. Static Aggregation	1-fold	Lesion-wise	0.8765	0.8667	0.8787
			Patient-wise	0.8422	0.8667	0.7500
	2. Dynamic Aggregation	1-fold	Lesion-wise	0.8765	0.8000	0.8939
			Patient-wise	**0.8947**	**0.9333**	**0.7500**
	Performance of Experiment 2 (Avg. of 3-folds)	**3-folds**	**Patient-wise**	**0.9122**	**0.9333**	**0.8334**

Each fold: 80% data for training & 20% for test

= 2nd best performing exp of that baseline (1-fold)

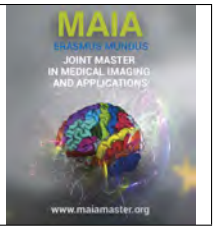
= Best performing exp of that baseline (1-fold)

= Best performing exp of that baseline (Avg of 3-folds)

Figure 20: Survival time predictions (Side by side)

Medical Imaging and Applications

Master's Thesis, August 2020



Prediction of clinical status, ADAS-Cog13 score and ventricles' volume using an ensemble of regression models

Isaac Llorente Saguer*, Sergi Valverde, Arnau Oliver, Xavier Lladó
for the Alzheimer's Disease Neuroimaging Initiative**

** illorentes at gmail . com*

***Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf*

Abstract

Background. Alzheimer's disease is the most prevalent cause of all the dementias, yet no cure or drug succeeded in stopping or slowing its progression. Accurate prediction of the disease could not only be useful per se but could also help in subject selection and stratification for clinical trials at an early stage of the disease.

The Alzheimer's Disease Prediction Of Longitudinal Evolution (TADPOLE) Challenge compares the performance of algorithms at predicting the future evolution of individuals from the ADNI database at risk of Alzheimer's disease. The three tasks of the challenge consisted in giving monthly predictions of three different variables: the clinical diagnosis (Normally cognitive, Mild Cognitive Impairment, Alzheimer's Disease), the Alzheimer's Disease Assessment Scale Cognitive Subdomain (ADAS-Cog13) score, and the volume of the ventricles.

Material and Methods. The data from ADNI consisted of a variety of biomarkers from different modalities: age, sex, education level, ethnicity, race, MRI (volumes, cortical thickness, surface area), PET (FDG, AV45 and AV1451), DTI (regional means of standard indices), CSF measurements (Amyloid-beta, Tau and P-Tau), and cognitive tests (CDR Sum of Boxes, ADAS11, ADAS13, MMSE, RAVLT, Moca, Ecog). The training, validation and test sets consisted of multiple entries from 1667, 896 and 219 subjects, respectively. An entry is defined as a set of biomarkers taken in the same examination period.

In this work, a highly adaptive framework is proposed that includes the expansion of features to include longitudinal information, a set of nine regression models, and two ensemble methods to combine the predictions of the trained models, using a genetic algorithm to find a more optimal linear combination of model weights than the direct mean of all models would give. Predictions from longitudinal data are compared with those using cross-sectional data.

Results. Our proposed framework, trained with the same data, was able to achieve results comparable to the best ones reported in the challenge platform in all of the tasks, which were based on independent strategies. A mAUC of 0.936 was obtained for the clinical status task to classify between Cognitively Normal (CN), Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD). A Mean Absolute Error of 3.44 was obtained for the ADAS-Cog13 task, while a Mean Absolute Error of 0.397% of intracranial volume was obtained for the prediction of the ventricles' volume task. The experimental results showed that longitudinal data had more significant predictive power than cross-sectional data.

Conclusions. On one side, the obtained results look promising as they pushed the state-of-the-art, and could be used for a future estimated evaluation of a patient's state, allowing in turn a better subject selection for clinical trials. On the other, the generic and adaptive nature of the framework presented in this work proved successful for different tasks, and could also be used for other problems like classification or regression.

Keywords: Alzheimer, prediction, TADPOLE challenge, longitudinal, neurodegenerative, atrophy

1. Introduction

Alzheimer's Disease is a progressive brain disorder that slowly hinders the ability to recall memories and to carry out the simplest tasks for self-sufficiency. Worldwide, at least 50 million people were believed to be living with Alzheimer's disease or other dementias in 2018 (Patterson et al., 2018); if breakthroughs are not discovered, rates could exceed 152 million by 2050. Apart from the high costs involved as a society, the personal cost both emotionally and lifestyle-wise that affects the patient, as well as familiars and friends around, is nothing less than unfathomable. According to the Alzheimer's Association [3] , one in three seniors dies with Alzheimer's or another dementia, killing more people than breast cancer and prostate cancer do combined.

Early detection of the disease will potentially accelerate the development of new therapies by ensuring that appropriate people are enrolled in clinical trials. The Alzheimer's Association commissioned a study of the potential cost savings of early diagnosis (Mebane-Sims, 2018), assuming that 88% of individuals who will develop Alzheimer's disease would be diagnosed in the MCI phase rather than the dementia phase or not at all. Approximately 7 trillion dollars could be saved in medical and long-term care costs for individuals who were alive in 2018 and will develop Alzheimer's disease.

Different data sets have been proposed in order to facilitate the discovery of new methods for the early detection and tracking of the Alzheimer's Disease (ADNI, OASIS, AMP-AD, Parelsnoer Neurodegenerative Diseases study). In addition, several challenges were proposed for classification (CAD Dementia[1], Alzheimer's Syndrome Prediction Challenge[2]). In contrast, the TADPOLE challenge aims to predict the evolution of Alzheimer's disease.

TADPOLE challenge. Organized by the EuroPOND initiative in collaboration with ADNI, the TADPOLE challenge (Marinescu et al., 2018) was born with the purpose of identifying algorithms and features that could best predict the evolution of Alzheimer's disease. Given a set of biomarkers from up to 1737 different subjects, the challenge participants were asked to provide three different variable predictions:

Task 1: Clinical status. Following the guidelines proposed by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Hyman et al., 2012), this first task required the participants to predict the future clinical status out of three different classifications: Cognitively Normal (CN), Mild Cognitive Impairment (MCI) or Alzheimer's Disease (AD).

Task 2: ADAS-Cog13. The Alzheimer's Disease Assessment Scale-cognition sub-scale (ADAS-Cog) (Rosen et al., 1984) is the most widely used general cognitive measure in clinical trials of AD (Connor and Sabbagh (2008); Ihl et al. (2012)). The ADAS-Cog was developed as an outcome measure for dementia interventions; its primary purpose was to become an index of global cognition in response to anti-dementia therapies. The ADAS-Cog assesses multiple cognitive domains including memory, language, praxis, and orientation. This second task's objective is to predict this score.

Task 3: Ventricles volume. The variable that had to be predicted in this last task was the volume of the ventricles, divided by intracranial volume (ICV), as estimated via the standard ADNI image processing pipeline (ADNI, 2020), which uses the FreeSurfer software (Reuter et al., 2012).

Goals of this project. The objectives for this Master's Thesis are twofold: first, to better understand the nature of the disease as well as the available predictive models through the analysis of the TADPOLE challenge and literature review of similar tasks. Secondly, following the initial research phase, we aim to develop a novel approach based on a well structured automatic feature expansion and data enhancing architecture followed by getting predictions from that enhanced data set with multiple models, and finishing with an ensemble of the different models able to participate in all three tasks of the challenge. Our results show that our proposed approach improves on the state-of-the-art results by means of using the longitudinal data of ADNI subjects with a set of different regressions models, ensembled to smoothen the response. The same architecture is used to predict the three challenge tasks reaching the first position in all of them, whereas, in the challenge, the best result for each task corresponded to a different and specific approach.

2. Challenge details

This section will provide some insight into the Data available within the Challenge and the evaluation metrics used to rank the methods for the three different tasks.

2.1. Data

Data used in the challenge, and therefore also in this Master's Thesis (Figure 1), was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [3], using the ADNIMERGE spreadsheet, to which the following was added: regional MRI (volumes, cortical thickness, surface area), PET (FDG,

[1] https://caddementia.grand-challenge.org
[2] http://challenge.xfyun.cn/2019/gamedetail?type=detail/alzheimer
[3] adni.loni.usc.edu

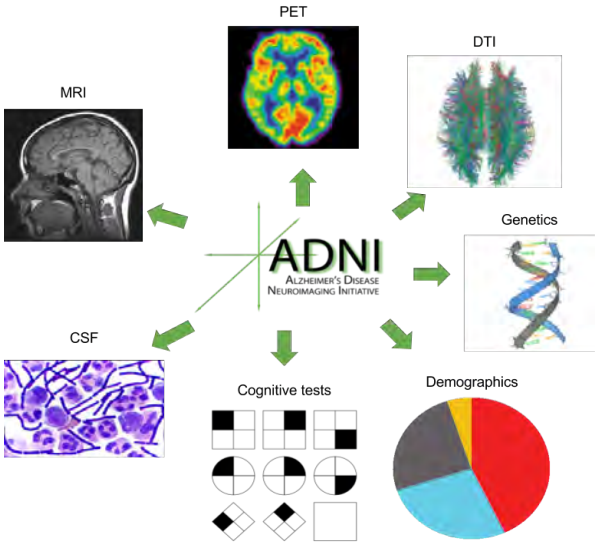


Figure 1: Representation of the different biomarkers in the challenge. *Figure from tadpole.grand-challenge.org.*

AV45 and AV1451), DTI (regional means of standard indices) and CSF measurements (Amyloid-beta, Tau and P-Tau).

The MRI measurements included were FreeSurfer processed ROI volumes, cortical thicknesses, and cortical surface areas from the UCSFFSL (longitudinal pipeline) and UCSFFSX (cross-sectional pipeline) tables.

The organizers describe four different data sets, namely a proposed training set, validation sets (longitudinal and cross-sectional) and a test set. They are hereby described:.

D1. TADPOLE Standard training set
The TADPOLE standard training set draws on longitudinal data from the entire ADNI history. The data set contains measurements for every individual that has provided data to ADNI in at least two separate visits (different dates) across three phases of the study: ADNI1, ADNI GO, and ADNI2.

D2. TADPOLE Standard prediction set
The set of D2 entries contained all available longitudinal data for prospective ADNI-3 subjects that were rollovers from earlier ADNI studies.

D3. TADPOLE Cross-sectional prediction set
The TADPOLE cross-sectional prediction set contains a single (most recent) time point and a limited set of variables from each rollover individual in D2.

D4. TADPOLE test set
The TADPOLE test set contains visits from ADNI rollover subjects that occurred after 1 Jan 2018 and contain at least one of the three outcome measures: diagnostic status, ADAS-Cog13 score, or ventricle volume.

It is important to note that there is not a homogeneous time-point distance neither for the same or different patients, as well as a different number of total events per patient (ranging from one single entry up to 19). Although the data from D1 and D2 consists of 12.742 entries for 1737 different subjects, not all biomarkers or even target task ground truths are available (see Table 1). We can have a broad view of the demographics of the different data sets in the Appendix A at the end of this document.

Table 1: Available data in number of visits (% of total visits in parenthesis) of each of the challenge data sets.

	Data sets			
	D1	D2	D3	D4
Features	Number of visits with available data (as % of total visits)			
Cognitive	8862 (69.9%)	5218 (68.1%)	753 (84.0%)	223 (95.3%)
MRI	7884 (62.2%)	4497 (58.7%)	224 (25.0%)	150 (64.1%)
PET (FDG)	2119 (16.7%)	1544 (20.2%)	0 (0.0%)	0 (0.0%)
PET (AV45)	2098 (16.6%)	1758 (23.0%)	0 (0.0%)	0 (0.0%)
PET (AV1451)	89 (0.7%)	89 (1.2%)	0 (0.0%)	0 (0.0%)
DTI	779 (6.1%)	636 (8.3%)	0 (0.0%)	0 (0.0%)
CSF	2347 (18.5%)	1458 (19.0%)	0 (0.0%)	0 (0.0%)

2.2. *Evaluation metrics*

The metric used to evaluate the fist task (clinical diagnosis) is the Multi-class area under the receiver operating curve (mAUC), an extension of the classical ROC curve for non-binary classification problems. From the challenge instructions, the AUC $\hat{A}(c_i|c_j)$ for classification of a class c_i against another class c_j, is:

$$\hat{A}(c_i|c_j) = \frac{S_i - n_i(n_i+1)/2}{n_i n_j} \tag{1}$$

where n_i and n_j are the number of points belonging to classes i and j respectively, while S_i is the sum of the ranks of the class i test points after ranking all the class i and j data points in increasing likelihood of belonging to class i (Hand and Till, 2001). For situations with three or more classes, $\hat{A}(c_i|c_j) \neq \hat{A}(c_j|c_i)$. Therefore, in the challenge the average was used:

$$\hat{A}(c_i, c_j) = \frac{\hat{A}(c_i|c_j) + \hat{A}(c_j|c_i)}{2} \tag{2}$$

The overall **mAUC** is obtained by averaging equation (2) over all pairs of classes. For L classes, the number of pairs of classes is $L(L-1)/2$, so that:

$$\text{mAUC} = \frac{2}{L(L-1)} \sum_{i=2}^{L} \sum_{j=1}^{i} \hat{A}(c_i, c_j) \tag{3}$$

The class probabilities that go into the calculation of S_i in equation [1] are p_{CN}, p_{MCI} and p_{AD}, which are derived from the likelihoods L_{CN}, L_{MCI} and L_{AD} provided by the participants by normalising by their sum so that, for example:

$$p_{CN} = L_{CN}/(L_{CN} + L_{MCI} + L_{AD}) \tag{4}$$

For tasks 2 and 3 the metric used for the evaluation of the predictions is the *Mean Absolute Error*:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \quad (5)$$

where n is the total number of entries to predict, $\hat{y}_i$ is the i-th prediction, and y_i is the ground truth of the i-th entry.

For the ADAS-Cog13 task as well as the ventricles' volume prediction, the participants had to provide a best-guess value as well as a 50% confidence interval for each individual.

3. State-of-the-art approaches

On July 2019 the challenge results were made public, with a different method scoring the best for each of the tasks. There were a total of 93 submissions from 33 different international teams. The platform remained open for additional submissions after the challenge deadline, with some teams further improving the results they had obtained as well as some others participating for the first time.

The remaining of this section is structured as follows: first, the most relevant and recent state-of-the-art methods will be reviewed for each of the tasks, and then some general figures of all the methods submitted in the challenge will be given.

Task 1: Clinical status. An extensive literature review by Weiner et al. (2017) summarized a set of methods (manual prediction by a clinical expert, statistical prediction using regression, machine learning, data-driven disease progression models) and biomarkers (hippocampal atrophy, β-amyloid and tau protein deposition) that could be useful for this task. In the test set of the challenge, the best result by a good margin was obtained by the *Frog* team using a combination of feature engineering in order to capture longitudinal information (70 direct features + 420 augmented ones) and a gradient boosting tree-based method (XGBoost), achieving a mAUC of 0.931. The second best mAUC score was 0.921 (Moore et al., 2019) with an approach using 16 features and two Random Forest models to predict NC to MCI and MCI to AD, respectively. The third best score within the competition timeline was 0.907 (Venkatraghavan et al., 2019) using up to 338 features and an SVM-based classifier. The latest public update on the results (01-16-2020) showed a Recurrent Neural Network (LSTM) achieving a mAUC score of 0.909 (Nguyen et al., 2018). The best ensemble that the organisers calculated from all the participants' entries (*ConsensusMedian*) achieved a mAUC of 0.925. A comparison of 15 studies presented by (Moradi et al., 2015) reported lower performance (maximum AUC of 0.902) for the simpler two-class classification problem of separating MCI-stable from MCI-converters in ADNI. A more recent work by Varatharajah et al. (2019) proposed a model which included PET, MRI, and CSF variables in addition to age and expression of CR1 (complement receptor 1) of a total of 135 subjects and was able to predict MCI-to-AD progression with an AUC of 0.92.

Task 2: ADAS-Cog13. To the best knowledge of the challenge organisers, no previous similar studies forecasting future ADAS-Cog13 or ventricle volume existed, so TADPOLE set a new benchmark to evaluate the performance on these important prediction tasks.

ADAS-Cog13 scores were more difficult to forecast than clinical diagnosis or ventricle volume. The only single method able to forecast ADAS-Cog13 better than informed random guessing (*RandomisedBest*) was the *BenchmarkMixedEffects* (Marinescu et al., 2020), a simple mixed effects model with no covariates and age as a regressor, achieving MAE in the test set of 4.19. The best ensemble (*ConsensusMean*) that the organisers calculated from the participants' entries achieved a MAE of 3.75.

Task 3: Ventricles volume. The best participant result for this task was 0.41% ICV (Inter-cranial volume) (Venkatraghavan et al., 2019) using up to 338 features and a Machine learning and Data-driven disease progression model. In this case, the best results of the different competitors were all very close to each other. The best ensemble (*ConsensusMedian*) that the organisers calculated from the participants' entries achieved a MAE of 0.38% ICV.

3.1. Features

The number of used features for all the submitted methods and benchmarks range from 3 to all of them (1907 initial data set columns). Guessing from the last visit only requires one feature per task, hence the 3 minimum features.

Some of the approaches expanded the initially chosen features by up to 20 times more. Feature expansion included: feature ratios (relative brain structure volumes), historical measures (maximum, minimum, mean, standard deviation), moving average, current age, cubic root of volumes, square root of surfaces, etc.

Some of the teams then fed those features to their models or applied further manual or automatic feature selection.

3.2. Prediction models

There was a large variety of models, which we can summarize in the following categories:

- Regression/Proportional hazards models.
- Neural networks (RNN, LSTM).
- Disease progression models.

- Machine learning (Random forest, gradient boosting, SVM/SVR).
- Other (ex. Clinician decision tree).

When comparing results we are not comparing models, but a more complex strategy that involves feature selection, model tuning, training and evaluation strategy, etc. Nevertheless, the top scores for the clinical status are well above the rest, and they use Gradient boosting and random forest methods, respectively. On the other hand, the top results for the ventricle volume prediction come from either hazard models or disease progression models. For the prediction of the ADAS-13 score, no model came close to beating the *Mixed Effects Benchmark*. It is worth observing that not only the three challenge tasks were won by different teams, but also with different models, completely different in nature.

4. Methodology

Our main objective is to design a single strategy that can be applied to the three challenges. The proposed framework consists of the following steps:

1. Data preparation - Feature engineering.
2. Individual models' selection and tuning.
3. Individual models' training and evaluation.
4. Model ensembles' building and validation.

4.1. Feature engineering

Selection. Out of all the features available, different subsets of them were selected in order to run the experiments. The different choices were guided by the documentation in the same challenge as well as by published papers on the topic cited in this presented work. The features were: cognitive tests (Skinner et al., 2012), hippocampus related features (De Leon et al. (1993), Devanand et al. (2007), Li et al. (2019)) from different image modalities, CSF measures (tau and beta-amyloid) (Gamblin et al., 2003), age (Podcasy and Epperson, 2016), sex (Podcasy and Epperson, 2016), education (Mortel et al., 1995), the APOE4 allele (Burke and Roses, 1991), marital status, ethnicity and race. The 36 ventricle and 59 hippocampus features extracted from image modalities (MRI, PET) were added separately so that we had a baseline and an extended data set.

Cleaning. After having selected the features to work with, an exploratory data analysis was performed in order to first check if the data needed to be cleaned, for example taking care of outliers (looking at the distribution and with priors about each of the features' range), input types (for example removing texts in numerical features), repeated features (automatically done with correlation evaluation) and uniform features (counting the unique values), and secondly to see if we could infer some information.

Expansion. The clean selected features were then expanded in various ways listed hereby:

- Clustering of the data with hdbscan (McInnes et al., 2017) and mini batch k-means with the intention to help the models identify potential groups with similar responses.
- One-hot-encoding categorical features to avoid the models suggesting a spurious correlation among the different labels (such as within different ethnicities).
- Historical: max and minimum, range (max-min), ratio (max/min), growth (max-min divided by their time difference), and time since the historical max, min and clinical status change.
- Days to predict, instead of months, to account for the extra precision that the dates from the subject evaluations provide.

All features and regression labels were transformed into 0-1 range. Depending on the model, this transformation can impact both the performance and the training time. Furthermore, we can later on trim the predictions to adjust to the capped reality (maximum ADAS-Cog13 score is 85, for example).

Time-warping. In this step, time-warping (self-coined term, explained in the next lines) is performed in order to make the models predict all future events from any given past entry (and its historical information if we are working with longitudinal data). This allows us to have many more training points, and to expand the prediction range, as it can be seen in Figure 2. The intuition behind this is that not only we will have more points to train, but also, since the same features will be used to predict so many different future points, the importance of the prediction time will be easily picked up by the models, which will have to understand the neurodegenerative nature of the problem at hand.

Reduction. The final step involving direct feature selection is an automatic fitting of a gradient boosting algorithm in order to assess the importance of each of the features and to eliminate the ones that do not seem to provide information to the model. CatBoost (Prokhorenkova et al., 2017) is used for this purpose, although the modular architecture would allow any alternative method (recursive feature elimination, variance threshold, univariate selection, etc.) to be analysed.

A graphical scheme of all the required steps for the feature engineering process can be seen in Figure 3.

4.2. Model selection and tuning

Since two of the tasks were regression problems, and the classification one had a natural order regard-

Original data set

Entry number	Months from baseline	Diagnosis
E1	0	label 1
E2	3	label 2
E3	6	label 3
E4	12	label 4

Predict Next

Entry number	Time distance to predict	Diagnosis to predict
E1	3	label 2
E2	3	label 3
E3	6	label 4

Time Warped

Entry number	Time distance to predict	Diagnosis to predict
E1	3	label 2
E1	6	label 3
E1	12	label 4
E2	3	label 3
E2	9	label 4
E3	6	label 4

Figure 2: Time warping to predict all future events from each entry. Given a set of original entries (and their associated biomarkers) from a subject, we can try to predict only the next diagnosis from each entry, or all future ones. By predicting all future entries instead of only the next one, the data set is vastly augmented, and the time-to-predict is expanded as well.

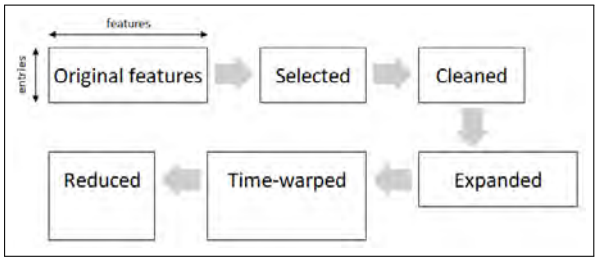


Figure 3: Feature engineering. This simplistic diagram makes it easier to see how the data set is expanded either feature-wise (horizontally) or entry-wise (vertically) after each step.

ing severity or disease stage, regression models were selected for our prediction purposes. The following models were selected either because of their promising validation score, or to add diversity to the ensemble:

- Scikit-learn models (Pedregosa et al., 2011): Ridge, SGDRegressor, SVR, AdaBoostRegressor, ExtraTreesRegressor).
- XGBoost (Chen and Guestrin, 2016).
- CatBoost (Prokhorenkova et al., 2017).

Out of the three different tasks, the ventricle volume prediction was the one that felt more rich and reliable, since the clinical status consists only of 3 classes (so the neurodegenerative process is heavily discretized), and the second task is the output score of the ADAS-Cog13 cognitive test, which is known to suffer from low reliability across consecutive visits (Grochowalski et al., 2016). The previously mentioned regression models from scikit-learn were tuned to predict the ventricle volumes using the gridsearch available from the same library, while the rest used the hyperopt library (Bergstra et al., 2013). In the given case that the top results of the same model were close to each other, and of different nature structurally, two versions of such models were kept (XGBoost, SVR). The actual parameters of all 9 chosen regression models can be seen in Appendix D.

In addition, TabNet (Arik and Pfister, 2019), a deep learning approach released in late 2019 by Google Research was also tested. TabNet uses sequential attention to choose which features to reason from at each decision step, which the authors claim it enables interpretability and more efficient learning as the learning capacity is used for the most salient features.

4.3. Training and evaluation of single models

Ensemble methods usually provide a stronger generalization response (Marinescu et al., 2020), which is why many winning solutions on Kaggle competitions are ensemble models (Bansal, 2018). In order to pursue this goal, different strategies were used throughout the design of the experiments, that fall into the ensembling category:

- **Out-of-fold prediction**: The evaluation of the models will be made using a set of out-of-fold predictions in a cross-validation loop fashion (see Figure 4).
- **Bagging**: This strategy can be seen throughout different parts of the whole process. Bagging-based models are used (CatBoost, XGBoost, ExtraTrees). Models will be trained with different initialization settings and their predictions will be averaged. After a model has been trained in a fold, it will provide a prediction for the test set, which will ultimately be averaged for the same model, across all folds.
- **Boosting**: Boosting models (based on an ensemble of decision trees) were used (CatBoost, XGBoost).
- **Average**: As an extra final prediction, the average across multiple models' predictions will be computed, both simple and weighted.
- **Stacking**: First level predictions (from models that train directly on the original data set) will be combined with or without the original data set to create a new training set for second level meta-models.

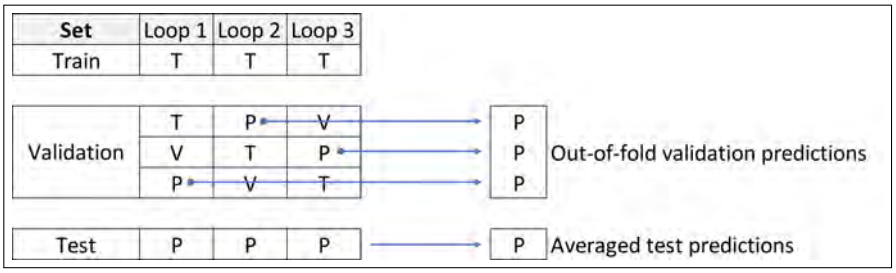


Figure 4: Training and evaluation strategy example with 3 folds, instead of the 10 used in the proposed work, for simplification. As we can observe, at each loop or fold, the model predicts a section of the validation set as well as the test set. After all folds are looped through, we are left with the whole out-of-fold validation predictions as well as with a test set prediction, that is the average of all the model predictions that made throughout the loops.

Training the models to the whole data set without any evaluation could unknowingly lead to overfitting; the same could happen if we have a validation set that is too similar to the training set, but not to the testing one: the validation of our methods could be inflated with respect to their evaluation in the test set. The benefit of using out-of-fold predictions is that it allows us to both have a better evaluation of our models, since we evaluate it in slightly different scenarios, and with a different set, while it allows us to eventually train with all the data set (aggregating the different folds). The two aforementioned positive aspects of this strategy are lost by using a single holdout validation set.

For the purpose of avoiding the aforementioned issues two validation strategies are proposed, namely Last and Blind:

Last mode. The Last mode will evaluate the models on the last entry of the subjects in the D2 data set. This will give the most information possible to the models, which should translate in a better prediction. However, it might also lead to an inflated validation score, because of the extra information on the subject (and the subject's historical features), since all entries except the last ones will be considered during training.

Blind mode The Blind validation will evaluate the models on subjects that are not present during the training. Although in this mode the conditions for the models are much harsher than the real task (predicting a future entry of a known subject), it should exhibit a good generalization behaviour, avoiding overfitting to the validation set, which could tarnish the ensemble performance.

We divide the subjects in the validation data set into 10 folds stratified by the number of entries per subject. Each loop during training, models that had the possibility of accepting a selected data set to be used for early stopping trained on 80% of the validation subjects, used 10% of them for early stopping and predicted the outcomes from the remaining 10%. The models that did not have early stopping used 90% of the subjects for training, and 10% for testing. This was done 10 times in order to loop through all 10 folds of the whole validation data set.

The structure of the architecture for training one model can be seen in Figure 4, while a whole overview of the experiments is shown in Figure 5.

The same metrics of the challenge will be used to compare the obtained results or our work to the state-of-the-art. However, since the number of data entries per subject has a wide range (2 to 19), and even wider after time-warping (1 to 171), by evaluating the MAE directly we would be giving considerably more weight to subjects with a high number of entries. This statistic can still be useful as a measure of how the model is able to perform on average for a wider range of prediction time lengths, but since the entries are not at all independent, we propose an alternative to the direct MAE, which is the mean of the subject-wise mean of the absolute errors. This can be seen as a stratified MAE, as seen in Figure 6. The equation is as follows:

$$MAE_{strat} = \frac{1}{m}\sum_{s=1}^{m}\left(\frac{1}{n_s}\sum_{i=1}^{n_s}|\hat{y}_i - y_i|\right) \tag{6}$$

expanding on the variables of the equation (5) with m being the total number of subjects and n_s the total number of training entries for subject s.

4.4. Ensemble predictions

For the ensemble (to provide full predictions of the test set), various approaches are proposed: the mean of all model predictions, and the weighted version of the mean. Using a linear combination of the models and evaluating on the out-of-fold predictions is a manoeuvre that has stronger smoothing characteristics than using a meta-model, just by assessing their complexity level. Weighting the different models, thus, is a great resource to allow us to combine the different models in order to achieve a better score, but even so, its performance is heavily dependant on how much the validation and test set are alike. That is why the simple mean of predictions is considered as well.

In order to perform the weighted ensemble, this initial path was explored:

- Removing highly correlated models. Diversity is a key point to be able to get different information

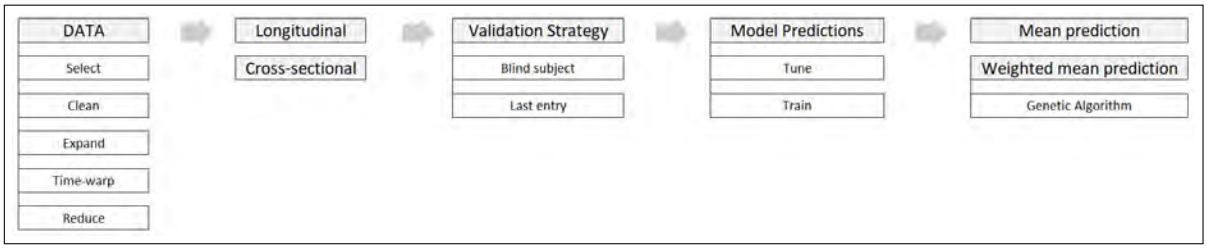


Figure 5: Global overview of the main strategy used in the proposed work.

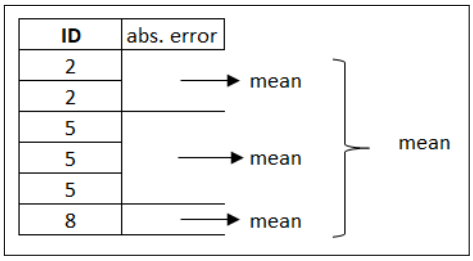


Figure 6: MAE stratification per subject. To account for the vast difference in subject entries, and therefore prediction per subjects, we propose an evaluation that compensates the imbalance by averaging subject-wise metrics.

from different sources. Spearman's rank correlation coefficient was used for the analysis.

- Removing "bad" performing models (according to the evaluation scores). If we add models with very bad scores, they might make it more difficult to achieve better ensemble scores.
- Weighting the models according to their evaluation score, with different transformation functions.

However, the best performing ensemble is not necessarily the union of the best models (evaluated individually), and if it is, how many models should be taken? How should the weights be decided? The above strategy relied on many untested questionable assumptions. Therefore, a second strategy was chosen over the first one: All the models' predictions would be fed into a Genetic Algorithm[4] that would explore different weight vectors (as DNA) in order to find a good working "team" (ensemble) out of all the available "players" (models). The genetic algorithm quickly provided a set of weights that had a better response in the validation set. This experiment was run several times (10) in order to have a more reliable answer, since we would not know if the algorithm was stuck on a local minimum. Another advantage of this method is that it can automatically filter out models by giving them low weights. The parameters used for the genetic algorithm were as follows:

```
'max_num_iteration': 3000,
'population_size':100,
'mutation_probability':0.1,
'elite_ratio': 0.02,
'crossover_probability': 0.5,
'parents_portion': 0.3,
'crossover_type':'uniform',
'max_iteration_without_improvement':100
```

[4] github.com/ahmedfgad/GeneticAlgorithmPython

5. Results

As mentioned earlier, the metrics used to evaluate the different models in the validation set were the same used in the challenge (see section 2.2), with the addition of the stratified MAE. Confidence intervals were calculated empirically with the percentile bootstrap method (Carpenter and Bithell, 2000), since the distribution from where the statistic was taken did not follow a Gaussian distribution (see Appendix C).

The presented results are structured in three subsections: first the results on the validation set (D2) are shown, then the results on the test set (D4) are presented, finishing with a comment on the TabNet deep learning approach.

5.1. Validation results

For simplification, only the most relevant results of the experiments will be described here, since by combining 46 models, 2 data sets, 2 data modalities (longitudinal and cross-sectional), multiple validation sets (Blind, Last entry/ies), bagging and automatic feature reduction, the total number of experiments amounts to a very high number (over three thousand). To be noticed that this estimation is without involving multi-level stacking. Even so, the code was prepared to execute any of the combinations. The decisions of what strategy to use were made by testing such strategy variations, looking at the best single model score evaluated in the validation set, since it would set up a baseline for the ensemble to improve upon. These decisions will be listed in the Discussions section as they reflect on the interpretation of the results. Statistical significance of the results was made to the experiments that were chosen as final ones, described further below for each of the tasks.

Task 1: Clinical diagnosis. As observed in Figure 7, most of the models alone were able to reach high results, with the highest being 0.918 (0.915-0.921 95%CI). On top of that, by performing a weighted average of all the models' probabilities, we are able to increase the performance to 0.931 (0.929-0.934 95% CI), with a p-value <0.05 vs all the other models. The simple average also provided good results, at 0.923 (0.920-0.926 95% CI). Figure 8 shows the receiver operating characteristic curve for the two ensembles in the validation set, which is widely extended in the literature. To better perceive the selected strategy differences, Figure 9 shows the mAUC (with bars for the 95% CI), comparing the best single model and the different ensemble strategies with the longitudinal and cross-sectional data set.

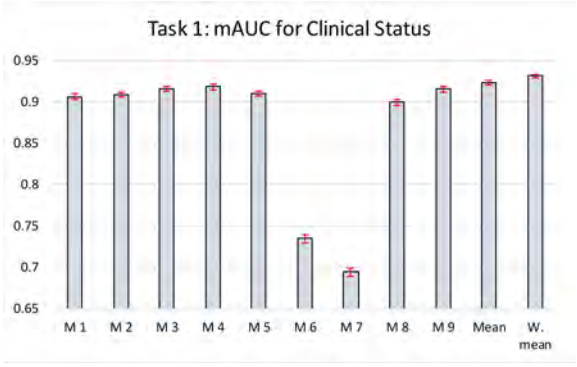


Figure 7: Task 1 validation mAUC scores for the 9 models + ensembles (mean and weighted mean), with 95% CI as per percentile bootstrapping. Longitudinal information was used. Models listed in Appendix D.

Task 2: ADAS-Cog13. In this task, the weighted mean ensemble managed to break the barrier of 4, obtaining a MAE of 3.92 (3.86-3.97 95% CI), while the simple average reached 4.32 (4.27-4.37 95% CI). Figure 10 shows the distribution of the validation scores. In this task as well, to better perceive the selected strategy differences, Figure 11 shows the MAE (with bars for the 95% CI), comparing the best single model and the different ensemble strategies with the longitudinal and cross-sectional data set.

Task 3: Ventricles' volume. We can observe good results for all the models (Figure 12), since the best challenge result was 0.41. The weighted mean provides promising results, with a % ICV MAE of 0.133 (0.130-0.136 95% CI and <0.05 p-value wrt the rest of the individual models), while the simple mean provides a worse validation score at 0.181 (0.178-0.184 95% CI). For this task as well, to better perceive the selected strategy differences, Figure 13 shows the MAE (with bars for the 95% CI), comparing the best single model and the different ensemble strategies with the longitudinal and cross-sectional data set.

5.2. Test results

The results mentioned in the previous sub-section would serve to compare to other results obtained by testing on the D2 data set, and helped us decide on the final proposed methods. However, the challenge only made public the metrics from the D4 test set. Table 2 shows our obtained results in the D4 test set and also the ones reported[5] by the best three state-of-the-art methods in each of the challenge tasks. Note that our approaches were able to improve results in all the three tasks.

5.3. TabNet

In the original paper, considering 10K to 10M samples, the range of $N_{steps} \in [3, 10]$ is said to be optimal. We tested a number of them to tune the net, and the results can be seen in Table 3.

6. Discussion

In this section we will discuss the findings of the experiments for the different strategies that lead to the final ones presented at the end. All statistical analysis is done on the validation set.

Validation modality. The results from evaluating only the last entry for each subject were very good. Actually, they were so good metric-wise that they were probably overfitting, with the best single model scoring over 99% mAUC for clinical diagnosis, 2.12 MAE for ADAS-Cog13 and just under 0.06% ICV for the ventricles' volume. This strategy was discarded as a precaution, and the subsequent tests were done by the Blind validation mode. The alleged overfitting is assumed to be due a disparity between the test set and the validation set, even though the results were valid on their own.

Automatic feature cleaning. By reducing the number of features fed to the models we reduce the overall complexity, and if the removed features did not contribute much, the models might even have an easier time converging to a good solution. The computation was much faster (2.6, 6.6 and 9.0 times faster for tasks 1, 2 and 3, respectively), albeit the results were slightly worse. Comparing the weighted ensemble predictions we find that the raw version is statistically higher (p-value <0.0001 in all three tasks). Since results were a primary objective, the non-cleaning method was chosen.

Data set. The smaller data set provided faster and slightly better results, so the choice was straightforward.

[5] https://tadpole.grand-challenge.org/D4_Leaderboard/

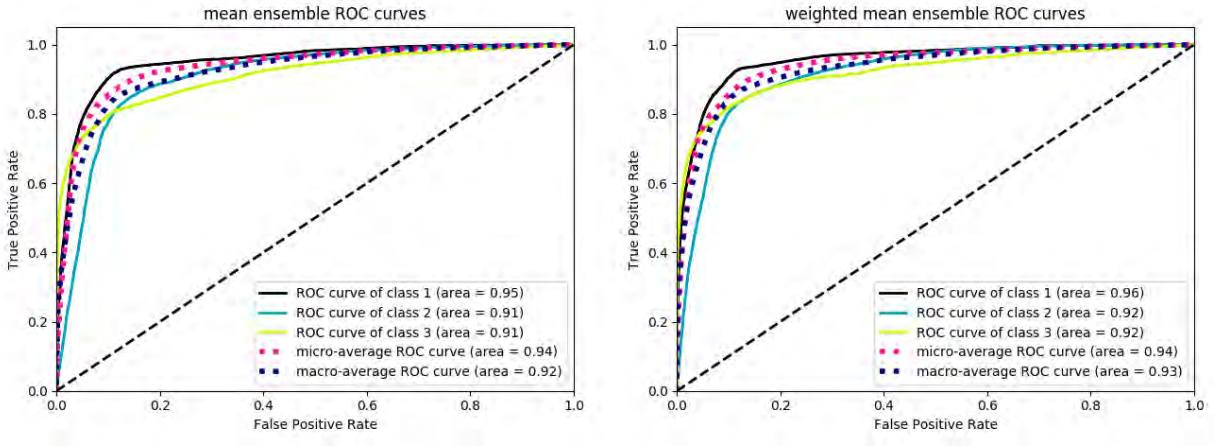


Figure 8: Task 1 ROC curves for the ensembles (mean and weighted mean). Classes 1, 2 and 3 are NC, MCI and AD, respectively. Data used was longitudinal. We can observe the one-versus-all AUC scores for the three classes.

Table 2: Comparison of the results in the challenge test set. The three best results of the live leaderboard are shown along with the proposed two ensembles and the two best models (according to the validation evaluation). We also show the results of TabNet and of the ensemble of 35 classification models.

	Method	Diagnosis [mAUC]	Method	ADAS-Cog13 [MAE]	Method	Ventricles [%ICV MAE]
TADPOLE challenge	Frog	0.931	BenchmarkMixedEffects	4.19	EMC1-Std	0.41
	Threedays	0.921	FortuneTellerFish-Control	4.70	EMC1-Custom	0.41
	CBIL-MinMFa	0.909	BenchmarkMixedEffectsAPOE	4.75	ImaUCL-Covariates	0.42
Methods studied in this work	Our w. mean	**0.936**		5.06		0.402
	Our mean	0.935		**3.44**		**0.397**
	SGD (model 5)	**0.936**		5.13		0.458
	XGBoost (model 4)	**0.936**		5.13		0.407
	TabNet	0.926		5.09		0.464

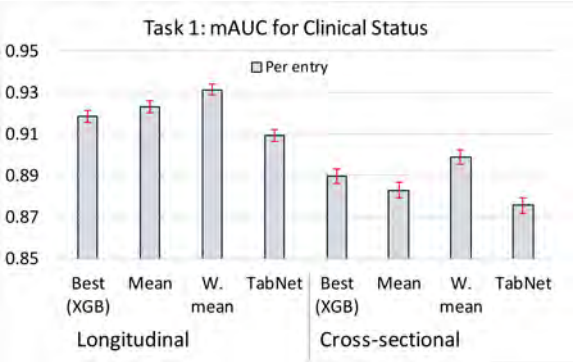


Figure 9: Task 1 validation mAUC scores for the best model, ensembles (mean and weighted mean) and TabNet, with 95% CI as per percentile bootstrapping. Longitudinal data and cross-sectional are compared. The mAUC is calculated across all entries.

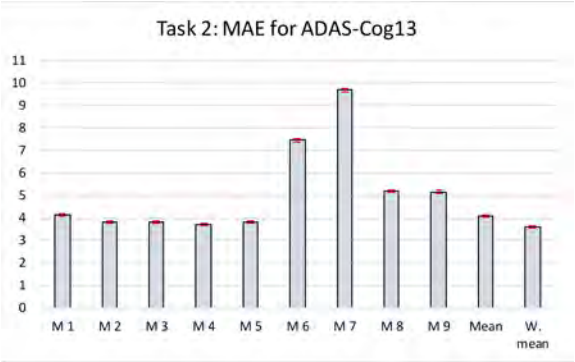


Figure 10: Task 2 validation MAE scores for the 9 models + ensembles (mean and weighted mean), with 95% CI as per percentile bootstrapping. Models listed in Appendix D.

Bagging (seeding). Having fixed the previous experiment settings, this rather simple strategy of averaging the same model with different initialization settings managed to improve a bit the results of the first task. However, for simplification, we present all the following results without this bagging strategy. Moreover, the way that the models ensemble their bagged predictions would lead to extra experiments as well. We used the mean, but the median could have been used as well.

Stacking. By using the first level models predictions as features, with or without the original features, meta-models were trained. The results were suspiciously

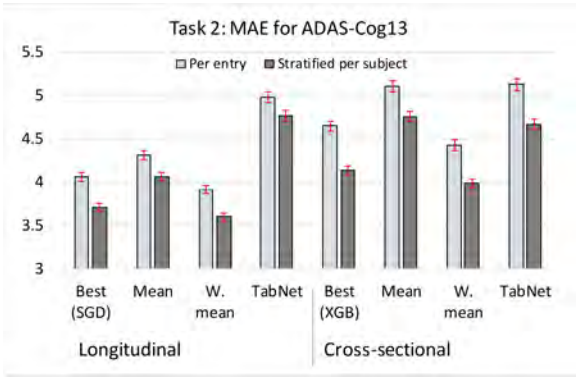


Figure 11: Task 2 validation MAE and stratified MAE scores for the best model, ensembles (mean and weighted mean) and TabNet, with 95% CI as per percentile bootstrapping. Longitudinal data and cross-sectional are compared

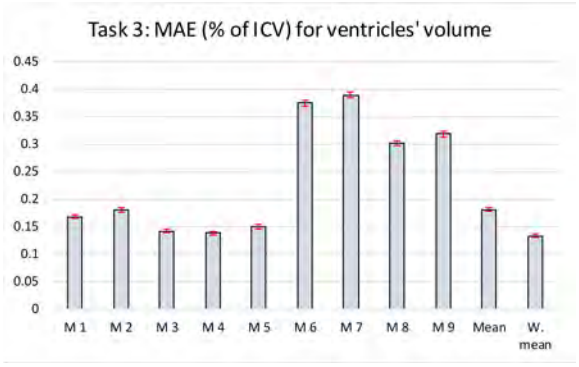


Figure 12: Task 3 validation MAE scores of % ICV for the 9 models + ensembles (mean and weighted mean), with 95% CI as per percentile bootstrapping. Models listed in Appendix D.

good as well, so in fear of overfitting, this strategy was discarded at the current configuration.

Weighted ensemble. By allowing our architecture to optimise a better combination of model predictions we saw how it constantly got better results (p-value <0.0001 in all three tasks against the best single method). The high adaptive flexibility of this strategy can be seen in Figure 14, where, for example, the second model has a lot of weight for the first task, but barely any weight for the third task. We can also observe how the third and fourth models (both XGBoost with different parameters) were consistently picked as valuable models. The downside of trying to optimize results is that you inevitably fit to the set you are evaluating your models with, and as stated earlier in this document, if the test set does not have the same representative population as the validation one, difference in performance is to be expected.

Mean ensemble. The direct mean method is a powerful strategy in the sense that it listens to all models equally, meaning it is less prone to being sensitive to a single

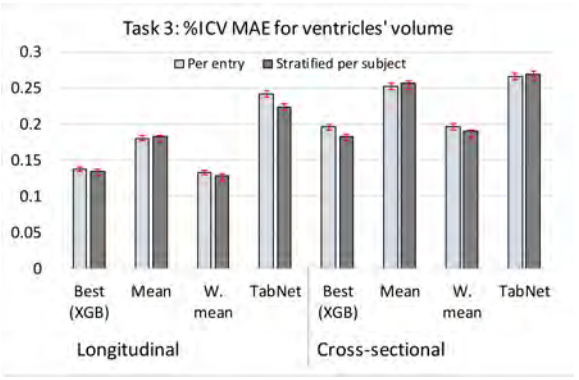


Figure 13: Task 3 validation MAE and stratified MAE scores for the best model, ensembles (mean and weighted mean) and TabNet, with 95% CI as per percentile bootstrapping. Longitudinal data and cross-sectional are compared

Table 3: Performance of TabNet with respect to the number of steps. The metrics for the three tasks are mAUC, MAE and MAE, respectively. We can observe how increasing the number of steps quickly leads to overfitting.

Steps #	Task 1	Task 2	Task 3
1	0.905	4.365	0.226%
2	0.908	4.443	0.222%
3	0.899	4.474	0.269%
5	0.893	4.985	0.238%
8	0.875	4.825	0.361%

model overfitting in the validation set, and although behind in performance to the weighted mean in the validation set, it outperformed all published challenge methods in the test set. A key factor is without doubt the variety in the models, since it will benefit both the direct mean and the weighted mean ensembles. We can observe the correlation coefficient of the models' predictions using Spearman's method in Appendix B.

Results. Given the predictive power of the proposed work relative to the current entries for the same challenge, we conclude that it sets up a new benchmark for the early prediction of neurodegenerative evolution of Alzheimer's Disease for all three of the challenge tasks. As seen in Table 2, although XGBoost really stands out in clinical diagnosis and ventricles' volume, the mean and weighted mean are more stable as a method, especially if we take the validation scores into account, where the models were tested by predicting multiple entries (27712, 27896 and 15700 for tasks 1, 2 and 3, respectively) for 896 subjects, instead of the 219 subjects seen in the testing set.

Deep learning. As we could appreciate in the Results section, the performance of TabNet that resulted from our experiments falls behind our proposed method. Most likely, this novel deep learning method could ben-

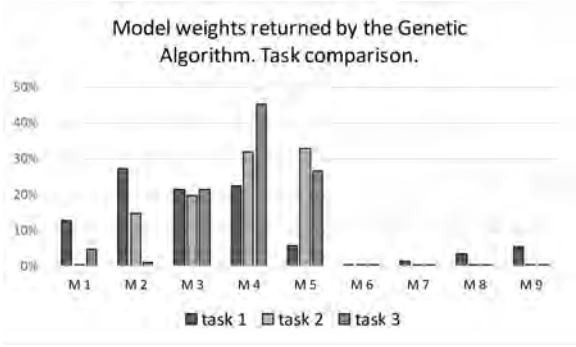


Figure 14: Percentage of total weights given to the 9 regression models, for the three tasks. It can be observed how different models are better suited for different tasks. If the bar is not seen it means that its weight is approximately zero. Models listed in Appendix D.

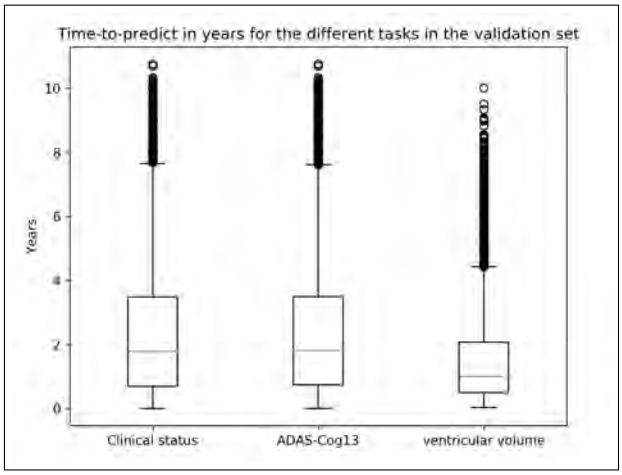


Figure 15: Time-to-predict (in years) for the different tasks in the validation set. We can see the wide spread of the data. The difference between the tasks is related to the availability of their respective labels.

efit from more data, as we observed that results from increasing its number of steps (within the range proposed in the original paper and beyond) got worse (see Figure 3). As more information is gathered and existing data sets expand, TabNet's performance could see an improvement. Another deep learning approach with minimalRNN proposed by Nguyen et al. (2020), who participated in the challenge and just recently published their latest results having improved on the method after the challenge was over, is also underperforming compared to our proposed approach. Lastly, it is worth mentioning that a total of five teams used recurrent neural networks in their submissions, and another one used a deep fully connected network. None reached the performance of our proposed method.

Interpretability. Unsurprisingly, the "*time to predict*" is in the top 2 of feature importance for all tasks, since the model will adapt the prediction to the point in time that we ask it to. The other strong feature that makes the top 2 (as per the CatBoost ranking) is a historical measure of the variable to be predicted, as it is the last direct reference point. Another point to be made in terms of interpretability is to ask what is it that the models can predict exactly in time: since we evaluate on all possible time ranges, we lose the possibility of discovering that perhaps a model can have an extraordinary prediction power up until a certain period. Undoubtedly, data availability limits the amount of stratification we can make in order to study different prediction ranges. On the other hand, the current method is a one-fits-all solution that can predict all known time-points within the data set (from a few months till years ahead, as we can observe in Figure 15), with its own advantages over a more restricted method.

7. Limitations

Three main limitations are discussed: data, target variables and method.

Data. Data can be a limitation in terms of comparability or global assessment. While it is safe to compare the results with other methods that use the same data, it can be tricky to compare with approaches that use other data sets, or to infer actual real-life performance. The ADNI database was reportedly promoted as a solution to overcome the limitations of the clinical and neuropsychological tests available for monitoring disease progression at that time (Mueller et al., 2006).

The test set was derived from an early stage of the ADNI3 batch, which currently has over 700 rollover subjects. This higher number can be more informative than the 219 subjects used in the challenge, which might suffer from a small sample size. Future evaluations of the predictions on multiple entries of multiple subjects could provide a better estimation of the predictive power of different strategies.

As we expand in more detail in the *Future work* section, while it is necessary to acquire data for experiments and methods comparison, old data can lack the newest biomarkers or acquisition protocols. While it can still serve as a baseline for comparable results, neglecting the use of the most up to date methods could be a barrier for breaking the ceiling of the current state-of-the-art.

Target variables. As stated by (Marinescu et al., 2020), *clinical diagnosis* has only moderate agreement with gold-standard neuropathological post-mortem diagnosis (Beach et al., 2012). With the advent of post-mortem confirmation in ADNI, future challenges might address this by evaluating the algorithms on subjects with pathological confirmation. Similarly, *ADAS-Cog13* is known to suffer from low reliability across consecutive visits (Grochowalski et al., 2016), and TADPOLE algorithms fail to forecast it reliably. However, this might be related

to the short time-window (1.4 years), and more accurate predictions might be possible over longer time-windows when there is more significant cognitive decline. *Ventricle volume* measurements depend on MRI scanner factors such as field strength, manufacturer and pulse sequences (Han et al., 2006), although these effects have been removed to some extent by ADNI through data preprocessing and protocol harmonization. Moreover, scans from subjects with later-stage dementia are more difficult to segment, potentially due to increased motion and lower grey-white contrast (Henschel et al., 2020). Specifically, the increase in ventricle volume, subsequent shrinkage of GM and/or increased white matter lesion load can have a profound effect and are frequently difficult to segment with traditional neuroimaging pipelines such as FreeSurfer, which is the tool used in the pipeline to extract MRI regional volumes, including the ventricles.

Method. Since the modelling of the disease progression is left up to the ensemble, and by extension to the different models used, no clear boundaries are set up which could prevent possible egregious errors (if any). On the other hand, our general method has the potential to skip possible limitations of using such guided strategies (Event Based Modelling, Disease Progression Modelling, etc.), like the oversimplification that occurs when underestimating the number of pattern groups or the choice of biomarkers, which could be missing on possible confounders.

8. Future work

Further research on this work could explore different paths, that we divided into two categories: Method and Alzheimer's Disease.

Method. Given the highly automation and flexibility of the proposed framework, it could be very easily applied to other problems, be them neurodegenerative disease-based or not. As a matter of fact, even though the data was the same, the whole framework was used to predict three different variables, which achieved results comparable to those of the state-of-the-art. Some selected expansion points on this line could be:

- To automatically expand both the training and test set with noise (small variations to feature values).
- To try different strategies for missing values imputation .
- To explore different feature selection methods.
- To automatise interpretability of the results, and to be able to trace back the most useful models, strategies and features.

Alzheimer's Disease. Having achieved the aforementioned results, it could be interesting to see if we could model the disease progression for differently detected groups. Combining a carefully designed disease progression curve with the capability of machine learning models to extract information out of features that are most likely related could produce a cleaner output, and easier to interpret.

Exploring other aspects of the Alzheimer's Disease along with the known biomarkers could open a door for better understanding the disease and needs of the person, which could not only help the subject but also the caregivers. Furthermore, specially designed clinical trials could have more tools to monitor the effectiveness, and as a first instance, to more correctly choose the participants.

Research on new biomarkers and tests is ever advancing. While it is very handy to have a database as big as ADNI, older data is not necessarily up to the current knowledge. For example, (Palmqvist et al., 2020) just recently proposed to examine plasma tau phosphorylated at threonine 217 (P-tau217) as a diagnostic biomarker for AD, and while "further research is needed to validate the findings", it is a promising avenue, especially if it can end not only in more precise tests but also faster cheaper and safer. As a clarification, ADNI is adapting the protocols and introducing new biomarkers as research advances.

Additional exploration paths that were not covered specifically in *Methods* above:

- Changing the current "prediction at a given time" to "time to prediction" might be more useful from a clinical perspective.
- Discuss with clinical experts on the field as well as with patients what unanswered questions could have a bigger impact on different aspects (health, quality of life, cost, potential clinical trials for drug tests, etc).
- Tuning models specifically for the task they have to predict, focusing on one task at a time.
- A better understanding of the progression of the variable to predict can be used as a prior to the models to make sure we avoid impossible situations, such as the ventricles' volume abruptly shrinking in size, or someone scoring a number out of the table for ADAS-Cog13.

9. Conclusions

In this master's thesis we proposed an architecture to extract information out of a set of features from biomarkers together with different ensembles of regression models. The developed modular and adaptive architecture was tested against the best results (with

the most recent update being from June 2020) of the TADPOLE challenge (for Alzheimer's Disease prediction) using the same data. The proposed method had a predictive power on the three challenge tasks (prediction of clinical status, ADAS-Cog13 and ventricles' volume) comparable to the current state-of-the-art, achieving the first ranking position in all of them.

Statistical analysis was made to evaluate and better understand the models and ensembles. The experiments showed that training and predicting from longitudinal data proved to perform better than predicting from a single time-point (cross-sectional data).

The obtained results suggest that they could be used for a future estimated evaluation of a patient's state, allowing in turn a better subject selection for clinical trials. Furthermore, the generic and adaptive nature of the framework presented in this work proved successful for different tasks, and could also be used for other problems.

10. Acknowledgments

I am grateful to the VICOROB institute, and the neuro-imaging group NIC-VICOROB for providing me resources to carry on this research.

All this work would have been much more difficult without the invaluable work from the organizers of the TADPOLE Challenge, and I owe a lot of knowledge to their resources, especially to the synthesized data sets and documentation gathered from ADNI.

I am also thankful to my family and close friends, for enduring my long hours of research and coding.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies;Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging;Servier; Takeda Pharmaceutical Company; and Transition Therapeutics.The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

ADNI, 2020. Methods and Tools: Acquizition, pre-processing and quality control. adni.loni.usc.edu/methods/mri-tool/mri-analysis. [Online; accessed 15-05-2020].

Arik, S.O., Pfister, T., 2019. Tabnet: Attentive interpretable tabular learning. arXiv preprint arXiv:1908.07442 .

Association, A., et al., 2020. 2020 alzheimer's disease facts and figures. Alzheimer's & dementia 16, 391–460.

Bansal, S., 2018. Historical Data Science Trends on Kaggle. kaggle.com/shivamb/data-science-trends-on-kaggle. [Online; accessed 26-02-2020].

Beach, T.G., Monsell, S.E., Phillips, L.E., Kukull, W., 2012. Accuracy of the clinical diagnosis of alzheimer disease at national institute on aging alzheimer disease centers, 2005–2010. Journal of neuropathology and experimental neurology 71, 266–273.

Bergstra, J., Yamins, D., Cox, D.D., 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. Proceedings of the 30th International Conference on Machine Learning .

Burke, J., Roses, A., 1991. Genetics of alzheimer's disease. International journal of neurology 25, 41–51.

Carpenter, J., Bithell, J., 2000. Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. Statistics in medicine 19, 1141–1164.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 785–794. URL: https://doi.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

Connor, D.J., Sabbagh, M.N., 2008. Administration and scoring variance on the adas-cog. Journal of Alzheimer's Disease 15, 461–464.

De Leon, M., Golomb, J., George, A., Convit, A., Tarshish, C., McRae, T., De Santi, S., Smith, G., Ferris, S., Noz, M., 1993. The radiologic prediction of alzheimer disease: the atrophic hippocampal formation. American Journal of Neuroradiology 14, 897–906.

Devanand, D., Pradhaban, G., Liu, X., Khandji, A., De Santi, S., Segal, S., Rusinek, H., Pelton, G., Honig, L., Mayeux, R., et al., 2007. Hippocampal and entorhinal atrophy in mild cognitive impairment: prediction of alzheimer disease. Neurology 68, 828–836.

Gamblin, T.C., Chen, F., Zambrano, A., Abraha, A., Lagalwar, S., Guillozet, A.L., Lu, M., Fu, Y., Garcia-Sierra, F., LaPointe, N., et al., 2003. Caspase cleavage of tau: linking amyloid and neurofibrillary tangles in alzheimer's disease. Proceedings of the national academy of sciences 100, 10032–10037.

Grochowalski, J.H., Liu, Y., Siedlecki, K.L., 2016. Examining the reliability of adas-cog change scores. Aging, Neuropsychology, and Cognition 23, 513–529.

Han, X., Jovicich, J., Salat, D., van der Kouwe, A., Quinn, B., Czanner, S., Busa, E., Pacheco, J., Albert, M., Killiany, R., et al., 2006. Reliability of mri-derived measurements of human cerebral cortical thickness: the effects of field strength, scanner upgrade and manufacturer. Neuroimage 32, 180–194.

Hand, D.J., Till, R.J., 2001. A simple generalisation of the area under the roc curve for multiple class classification problems. Machine learning 45, 171–186.

Henschel, L., Conjeti, S., Estrada, S., Diers, K., Fischl, B., Reuter, M., 2020. Fastsurfer-a fast and accurate deep learning based neuroimaging pipeline. NeuroImage , 117012.

Hyman, B.T., Phelps, C.H., Beach, T.G., Bigio, E.H., Cairns, N.J., Carrillo, M.C., Dickson, D.W., Duyckaerts, C., Frosch, M.P., Masliah, E., et al., 2012. National institute on aging–alzheimer's association guidelines for the neuropathologic assessment of alzheimer's disease. Alzheimer's & dementia 8, 1–13.

Ihl, R., Ferris, S., Robert, P., Winblad, B., Gauthier, S., Tennigkeit, F., 2012. Detecting treatment effects with combinations of the adas-cog items in patients with mild and moderate alzheimer's disease. International journal of geriatric psychiatry 27, 15–21.

Li, H., Habes, M., Wolk, D.A., Fan, Y., Initiative, A.D.N., et al., 2019. A deep learning model for early prediction of alzheimer's disease dementia based on hippocampal magnetic resonance imaging data. Alzheimer's & dementia 15, 1059–1070.

Marinescu, R.V., Oxtoby, N.P., Young, A.L., Bron, E.E., Toga, A.W., Weiner, M.W., Barkhof, F., Fox, N.C., Eshaghi, A., Toni, T., et al., 2020. The alzheimer's disease prediction of longitudinal evolution (tadpole) challenge: Results after 1 year follow-up. arXiv preprint arXiv:2002.03419 .

Marinescu, R.V., Oxtoby, N.P., Young, A.L., Bron, E.E., Toga, A.W., Weiner, M.W., Barkhof, F., Fox, N.C., Klein, S., Alexander, D.C., et al., 2018. Tadpole challenge: Prediction of longitudinal evolution in alzheimer's disease. arXiv preprint arXiv:1805.03909 .

McInnes, L., Healy, J., Astels, S., 2017. hdbscan: Hierarchical density based clustering. The Journal of Open Source Software 2. URL: https://doi.org/10.21105%2Fjoss.00205, doi:10.21105/joss.00205.

Mebane-Sims, I., 2018. Alzheimer's association, 2018 alzheimer's disease facts and figures. Alzheimers Dement 14, 367–429.

Moore, P., Lyons, T., Gallacher, J., Initiative, A.D.N., 2019. Random forest prediction of alzheimer's disease using pairwise selection from time series data. PloS one 14, e0211558.

Moradi, E., Pepe, A., Gaser, C., Huttunen, H., Tohka, J., Initiative, A.D.N., et al., 2015. Machine learning framework for early mri-based alzheimer's conversion prediction in mci subjects. Neuroimage 104, 398–412.

Mortel, K.F., Meyer, J.S., Herod, B., Thornby, J., 1995. Education and occupation as risk factors for dementias of the alzheimer and ischemic vascular types. Dementia and Geriatric Cognitive Disorders 6, 55–62.

Mueller, S., Weiner, M., Thal, L., Petersen, R., Jack, C., Jagust, W., Trojanowski, J., Toga, A., Beckett, L., 2006. Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative. Cogn Dement 5, 56–62.

Nguyen, M., He, T., An, L., Alexander, D.C., Feng, J., Initiative, A.D.N., et al., 2020. Predicting alzheimer's disease progression using deep recurrent neural networks. NeuroImage , 117203.

Nguyen, M., Sun, N., Alexander, D.C., Feng, J., Yeo, B.T., 2018. Modeling alzheimer's disease progression using deep recurrent neural networks, in: 2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI), IEEE. pp. 1–4.

Palmqvist, S., Janelidze, S., Quiroz, Y.T., Zetterberg, H., Lopera, F., Stomrud, E., Su, Y., Chen, Y., Serrano, G.E., Leuzy, A., et al., 2020. Discriminative accuracy of plasma phospho-tau217 for alzheimer disease vs other neurodegenerative disorders. JAMA .

Patterson, C., et al., 2018. World alzheimer report 2018: the state of the art of dementia research: new frontiers. Alzheimer's Disease International (ADI): London, UK , 32–36.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Podcasy, J.L., Epperson, C.N., 2016. Considering sex and gender in alzheimer disease and other dementias. Dialogues in clinical neuroscience 18, 437.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., 2017. Catboost: unbiased boosting with categorical features. arXiv:1706.09516.

Reuter, M., Schmansky, N.J., Rosas, H.D., Fischl, B., 2012. Within-subject template estimation for unbiased longitudinal image analysis. Neuroimage 61, 1402–1418.

Rosen, W.G., Mohs, R.C., Davis, K.L., 1984. A new rating scale for alzheimer's disease. The American journal of psychiatry .

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52, 591–611. URL: https://doi.org/10.1093/biomet/52.3-4.591, doi:10.1093/biomet/52.3-4.591.

Skinner, J., Carvalho, J.O., Potter, G.G., Thames, A., Zelinski, E., Crane, P.K., Gibbons, L.E., Initiative, A.D.N., et al., 2012. The alzheimer's disease assessment scale-cognitive-plus (adas-cog-plus): an expansion of the adas-cog to improve responsiveness in mci. Brain imaging and behavior 6, 489–501.

Varatharajah, Y., Ramanan, V.K., Iyer, R., Vemuri, P., 2019. Predicting short-term mci-to-ad progression using imaging, csf, genetic factors, cognitive resilience, and demographics. Scientific reports 9, 1–15.

Venkatraghavan, V., Bron, E.E., Niessen, W.J., Klein, S., Initiative, A.D.N., et al., 2019. Disease progression timeline estimation for alzheimer's disease using discriminative event based modeling. NeuroImage 186, 518–532.

Weiner, M.W., Veitch, D.P., Aisen, P.S., Beckett, L.A., Cairns, N.J., Green, R.C., Harvey, D., Jack Jr, C.R., Jagust, W., Morris, J.C., et al., 2017. Recent publications from the alzheimer's disease neuroimaging initiative: Reviewing progress toward improved ad clinical trials. Alzheimer's & Dementia 13, e1–e85.

Appendix A. Data

Demographics of the different data sets, to be read vertically for each data set, or horizontally to compare different data sets among them.

Table A.4: Demographics

	Data sets			
	D1	D2	D3	D4
Subjects	1667	896	896	219
Measure	**Cognitively Normal**			
Number (%)	508 (30.5%)	369 (41.2%)	299 (33.4%)	94 (42.9%)
Visits per subject	8.3 (4.5)	8.5 (4.9)	1.0 (0.0)	1.0 (0.2)
Age	74.3 (5.8)	73.6 (5.7)	72.3 (6.2)	78.4 (7.0)
Gender (% male)	48.60%	47.20%	43.50%	47.90%
MMSE	29.1 (1.1)	29.0 (1.2)	28.9 (1.4)	29.1 (1.1)
Converters	18 (3.5%)	9 (2.4%)		
Measure	**Mild Cognitive Impairment**			
Number (%)	841 (50.4%)	458 (51.1%)	269 (30.0%)	90 (41.1%)
Visits per subject	8.2 (3.7)	9.1 (3.6)	1.0 (0.0)	1.1 (0.3)
Age	73.0 (7.5)	71.6 (7.2)	71.9 (7.1)	79.4 (7.0)
Gender (% male)	59.30%	56.30%	58.00%	64.40%
MMSE	27.6 (1.8)	28.0 (1.7)	27.6 (2.2)	28.1 (2.1)
Converters	117 (13.9%)	37 (8.1%)		9 (10.0%)
Measure	**Alzheimer's Disease**			
Number (%)	318 (19.1%)	69 (7.7%)	136 (15.2%)	29 (13.2%)
Visits per subject	4.9 (1.6)	5.2 (2.6)	1.0 (0.0)	1.1 (0.3)
Age	74.8 (7.7)	75.1 (8.4)	72.8 (7.1)	82.2 (7.6)
Gender (% male)	55.30%	68.10%	55.90%	51.70%
MMSE	23.3 (2.0)	23.1 (2.0)	20.5 (5.9)	19.4 (7.2)
Converters				9 (31.0%)

Appendix B. Correlation of regression models' predictions

Using the spearman method we display the correlation of the predictions among all of the regression models used in the ensemble.

Table B.5: Correlation of models' predictions

M2	M3	M4	M5	M6	M7	M8	M9	
86.8%	90.4%	91.6%	96.9%	59.2%	48.8%	90.1%	92.2%	M1
	87.8%	88.3%	87.7%	58.0%	48.1%	87.0%	89.3%	M2
		95.5%	89.6%	58.7%	49.0%	92.7%	92.4%	M3
			90.8%	58.4%	49.0%	94.4%	93.7%	M4
				59.6%	48.9%	89.4%	92.6%	M5
Scale					42.2%	56.5%	58.1%	M6
99.0%			Task 1			47.1%	47.2%	M7
95.0%							93.9%	M8

M2	M3	M4	M5	M6	M7	M8	M9	
91.1%	91.5%	92.6%	96.9%	73.6%	64.6%	84.8%	83.1%	M1
	94.7%	95.5%	94.2%	70.1%	62.6%	90.5%	90.0%	M2
		97.5%	93.7%	71.8%	62.6%	90.0%	87.5%	M3
			94.8%	71.8%	63.0%	90.3%	87.8%	M4
				73.5%	65.1%	88.6%	88.3%	M5
Scale					64.9%	62.9%	62.8%	M6
99.0%			Task 2			55.7%	58.6%	M7
95.0%							92.6%	M8

M2	M3	M4	M5	M6	M7	M8	M9	
96.6%	98.8%	98.8%	99.3%	93.1%	92.6%	95.3%	94.1%	M1
	97.7%	97.8%	97.8%	90.3%	90.0%	95.4%	96.5%	M2
		99.6%	99.1%	91.8%	91.4%	96.4%	95.6%	M3
			99.1%	91.9%	91.5%	96.5%	95.7%	M4
				92.4%	91.9%	96.4%	96.1%	M5
Scale					98.9%	87.8%	87.2%	M6
99.0%			Task 3			87.4%	86.9%	M7
95.0%							97.0%	M8

Appendix C. Test of normality

The Shapiro-Wilk test of normality (Shapiro and Wilk, 1965) returned scores of 0.74 and 0.70 for the absolute errors of tasks 2 and 3 predictions, respectively. The shape of the histograms (not shown) are positively skewed, as expected from the absolute of an error centered around zero. In addition, the following figure shows the probability plot for the normal distribution against the distribution from which the average should be computed to provide the MAE for tasks 2 and 3.

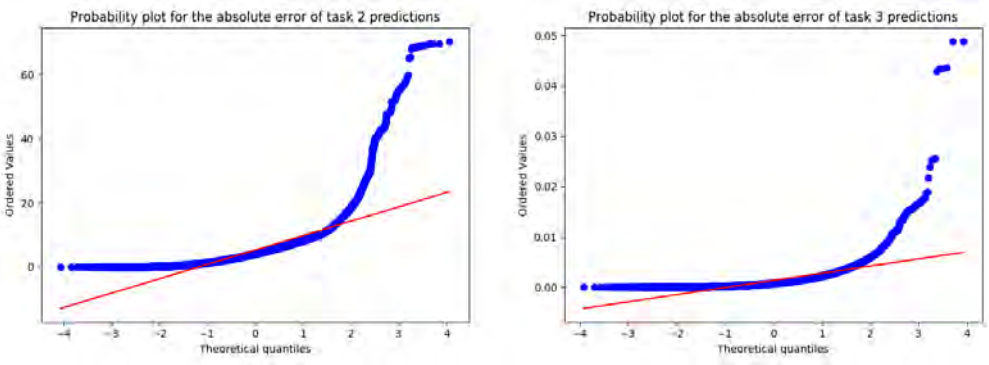


Figure C.16: Probability plot for normal distribution against the absolute error of the validation predictions in the ADAS-Cog13 task

Appendix D. Regression models' parameters

These are the parameters used for the 9 regression models used in the ensemble of this work, as returned by automatic optimization to the Task 3 using hyperopt and cross-validated gridsearch from sklearn.

M 1	**Ridge**
	solver = sparse_cg
	alpha: 0.0109
	max_iter: 707
	tol: 6.95e-06

M 2	**CatBoostRegressor**
	early_stopping_rounds = 200
	eval_metric = MAE
	loss_function = MAE,
	task_type = GPU
	border_count: 31
	l2_leaf_reg: 26.0
	learning_rate: 0.0614
	max_depth: 5.0
	n_estimators: 1400.0

M 3	**XGBRegressor**
	eval_metric = mae
	tree_method = gpu_hist
	colsample_bytree: 0.70
	gamma: 7.526
	learning_rate: 0.0977
	max_depth: 8
	min_child_weight: 2.0
	n_estimators: 1135
	subsample: 0.826
	objective: reg:squarederror
	early_stopping_rounds = 100

M 4	**XGBRegressor**
	eval_metric = mae
	tree_method = gpu_hist
	colsample_bytree: 0.70
	gamma: 9.355
	learning_rate: 0.114
	max_depth: 5
	min_child_weight: 16.0
	n_estimators: 924
	subsample: 0.796
	objective: reg:squarederror
	early_stopping_rounds = 100

M 5	**SGDRegressor**
	validation_fraction = 0.2
	alpha: 8.377e-05
	epsilon: 0.00918
	eta0: 0.0816
	l1_ratio: 0.393
	learning_rate: adaptive
	loss: epsilon_insensitive
	max_iter: 482
	n_iter_no_change: 36
	penalty: l1
	tol: 0.000169

M 6	**SVR**
	cache_size = 4000
	C: 49
	epsilon: 0.101
	gamma: 0.00184
	kernel: rbf
	max_iter: 1529

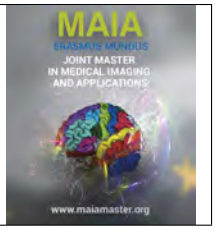
M 7	**SVR**
	cache_size = 4000
	C: 534
	epsilon: 0.101
	gamma: 0.00393
	kernel: rbf
	max_iter: 1876

M 8	**AdaBoostRegressor**
	learning_rate: 0.316
	loss: linear
	n_estimators: 29

M 9	**ExtraTreesRegressor**
	bootstrap: True
	ccp_alpha: 0.000960
	max_features: 0.856
	max_samples: 0.526
	min_samples_leaf: 1
	min_samples_split: 5
	n_estimators: 64

Medical Imaging and Applications

Master Thesis, August 2020



Multi-Organ Multi-Label Classification of 3D CT using Chained 3D Squeeze and Excitation

Prem Prasad, **Supervisor:** Joseph Lo, PhD

Department of Radiology, Duke University School of Medicine, Durham, North Carolina

Abstract

Multi-label classification of large 3D CT volumes can pose many challenges that can affect the performance of a model. Such factors include the sheer large sizes of the 3D volumes, inter-dependencies within classes, presence of unwanted features, as well as similarities that exist between numerous slices. Although Residual Networks have been proven to have positive impact in multiple deep learning tasks, their full potential is limited with the presence of the above-mentioned challenges in the task at hand. In recent years, to help prioritize useful features, Squeeze and Excitation modules have been introduced whose goal is to utilize and propagate channel inter-dependencies with an enhanced spatial encoding achieved by adaptive reconstruction of features learned, thereby prioritizing and advancing relevant features and subduing weak one. This strategy has been used extensively in other natural imaging tasks such as object detection in robotics, human action recognition, as well as in medical imaging tasks such as pulmonary node detection in lungs and brain tumor segmentation but in a limited and reserved manner. In this paper, this strategy is taken to the next level with multiple, directly linked Squeeze and Excitation modules inspired by the need for larger image volumes requiring recurring feature recalibration. We aim to enhance and take advantage of the previously boosted features in the network immediately rather than in later stages, and to do so with almost no additional computational cost. This strategy is applied to a multi-organ, multi-label disease classification task for body CT, specifically for normal vs. four diseases each in three selected organ systems: lungs/pleura, liver/gallbladder, and kidneys. Results show a steady increase in receiver operating characteristic (ROC) area under the curve (AUC) performance across almost all the organs and disease labels up until an optimum point and a gradual decrease thereafter.

Keywords: Computed Tomography, deep learning, multi-label classification, Chained Squeeze and Excitation, 3D CNN,

1. Introduction

Computed Tomography (CT) is one of several medical examination modalities used by physicians in procedures. Examples include detecting and monitoring of severe diseases such as cancers and stones, abnormalities, locating and monitoring tumors or any other infections, and even diagnosing the presence of vascular diseases that may prove to be fatal through strokes or kidney failure to name a few. CT scans are based on X-ray projections taken at various angles that are reconstructed to yield cross-sectional images of the area being monitored that include many anatomical structures such as bones, soft tissues of organs and even blood vessels. Since CTs scans are composed of many stacked 2D images, they have larger storage requirements which have a direct impact on training strategies of deep learning models.

As a non-invasive procedure, CT has revolutionized modern healthcare, and thus resulted in exponential increase in its use over the past few decades, therefore CT has become one of the most common imaging modalities used for diagnosis and treatment of diseases alongside MRI, Ultrasound, and Nuclear Imaging to name a few. A study done by Rebecca *et al.* was aimed at understanding radiation dose associated with common computed tomography examinations which included a study of the widespread use of CT in medical institutions [25] . A study from 2007 [6] and 2018 [9] show

the number of CT examinations performed in the United States alone. Roughly 72 million scans were done in 2007 and about 80 million in 2018, which goes to show the increase in scans being performed. In another study performed by Ingrid *et al.* the authors conduct a study to investigate radiologist's attitudes, activities and usage of CT imaging examinations, which concluded that a large percentage of radiologists in the United States not only perform CT examinations, but also read studies on CT screening, mostly towards lungs CT and CACS (Coronary Artery Calcium Scoring) for heart warnings against potential heart disease risk [14]. This goes to show the importance and rapid growth of computed tomography.

With such high volumes of scans being performed and with increasing number of patients taking them, it is vital that doctors are able to detect and isolate areas of interest within the scan much quicker, which is a very tedious task. This is why having the right Computer Aided Design (CAD) system, which are computer systems that aid in analysis of medical data, to assist doctors in their diagnosis is key to keep up with the ever-increasing load. To overcome this challenge of assisting doctors for quicker localization, several computer-aided-detection (CAD) methods have been developed over the past decade. Many researchers have proposed and developed such CAD systems that help in detection, localization, segmentation and also classification using neural networks, one such example being convolutional neural networks. Wentao Zhu *et al.* in their study propose DeepLung, an automated lungs CT cancer diagnosis system that utilized CNNs in a dual path mechanism, where the first task is to detect and localize nodules in lungs, and then followed by a subnetwork for classification [3].

Convolutional Neural Networks (CNN) have made their mark with their superior performances in computer vision tasks [26][23][16]. The core idea behind the convolution operation is to create filters that express local patterns within an image and eventually a representation of the image at various hierarchical depths. The question that many researchers are trying to answer is how do we know which features are most representative and important for the given task such that performance can be improved? To answer this, many neural network architectures have been proposed that target smart use of features. One example is by C. Szegedy *et al.* where the authors introduced Inception, a well-known deep learning architecture that uses a mutli-scale approach to feature creation with different kernel sizes [7]. Another example of unique feature reuse is by Sean Beal *et al.* where they introduce Inside-Out-Net (ION) which is an architecture innovation that relies on not just information present in the region of interest, but also outside of it [24]. They make use of information extracted at different levels of abstraction and concatenate them. Both these papers being publish in CVPR 2015 which is a top journal indicates the importance and potential that feature engineering can bring, the former being cited a little under 10,000 times while the later over 600 times.

In my study, I explore an architectural unit called Squeeze and Excitation as proposed by [17] which is another creative architectural innovation that aims to exploit the relationship between channels. The main objective of using this unit is to aid adaptive feature recalibration from the channels generated after convolution operations, such that discriminative features are given preference over non-essential ones. There are many added benefits in using this unit. Firstly, they can be inserted into any architectural design at any depth, thus highlighting its versatility. Second, they come with little-to-none additional computational cost with a small increment in model complexity but bring about consistent performance gains as seen later in this study. Thirdly, the roles they play when placed in different depths of the network vary, making them adaptive. Placement in earlier layers focus on significant lower level features while later use focuses on amplifying higher level features.

In a previous study, single Squeeze and Excitation (SE) blocks were applied on 2D images from ImageNet with dimension of 256×256 [17]. The amount of information present in these images is significantly lower than the data being dealt with in our study since CTs are three-dimensional volumes with large space requirements. Additionally, since SE blocks aim to propagate relevant features, I hypothesize that larger image volumes require more than one of these units to extract useful features. Therefore in this study, I propose a network that uses a chain of SE blocks motivated by the belief that with large amounts of information present in 3D CT volumes, single SE blocks are insufficient to meet the demands of the generalizing network built on this dataset.

2. State of the art

2.1. Single vs Multi-Label

The difference between a multi-label and multi-class problem should be understood before moving forward. A multi-class classification assumes that a given image can fall within only one class out of multiple, while a multi-label classification allows a certain image to be labelled with more than one tag, where each tag can resemble the presence of certain target objects. Due to the relatively new field and to its complexity, multi-label approaches have been lesser experimented with than single class problems such as nodules in lung, lesions in liver and cysts in kidneys.

Since this study is focussed on a multi-label study, it is important to know the advancements in both single as well as multi-label problems. In the area of lung CT, sufficient research has been done seeking to solve single label problems. For example, Y. Xie *et al.* aim to classify malignant and benign lung nodules on Chest CTs

[4]. In this particular example, they too had to overcome the challenge of lack in large training data by designing and developing a multi-view approach to take advantage of the limited Chest CT data available. To salvage the little data available, they developed a model that breaks down a 3D nodule into nine different views and for each view, a sub-model collects "knowledge" or information about the nodule from that view such as the general appearance and shape. This collection of information from each view is termed "knowledge based-collaboration" (KBC) sub-model by the authors, which allows for a multi view KBC (MV-KBC) network to be trained once all nine sub-models have collected their respective information, and the final model used to classify the lung nodules based on the adaptive weighing scheme learned during training. In a study [12], the authors rely on another technique to tackle the lung nodule detection problem. The general idea behind their work was to make use of a combination of two groups of images where the first consisted of original lung CT patches, while the second had binary images made after complex preprocessing enhancements of their respective patches in the first group. Finally, the combined original and preprocessed images were grouped and fed to a deep learning network.

Similarly, for Liver CT, following are single label studies done. F. P. Romero *et al.* , use a straightforward approach to discriminate between cancerous and non-cancerous liver lesions in abdominal CT images [2]. They used existing architectures such as Inception for feature extraction with residual connections to reinforce features. By simple use of existing architectures and advancements to produce accuracies of over 0.96. In another study [8], the authors propose a network that focusses on feature engineering and reuse to tackle the problem of liver lesion segmentation. Their network is a solution to the problem of 2D methods not performing well on 3D data, and 2.5D and 3D methods having too high levels of complexity. Here, they propose a "feature fusion" method with attention mechanism which combines features from varying levels of complexity (high level and low level) to get competitive results in the MICCAI 2017 Liver Tumor Segmentation (LiTS) Challenge.

Similarly, for Kidney CT, here are related tasks. Authors X. Yan *et al.* segmented kidney tumors in CT images by using a hybrid model combining two networks, where the first network provides an approximate segmentation of the kidney along with the tumor, while the second network refines the initial segmentation [29]. This method is shown to be memory efficient by attaining state-of-the-art results with reduced computational demand. In another study, Q. Yu *et al.* propose a new architectural structure for tumor segmentation in CT images called the "Crossbar-Net". Here, the concept involves using a horizontal and vertical patch to extract information and train two sub-models with a cascading fashion which are aimed to complement each other's errors in segmentation until convergence [30]. This was extended beyond the use in kidney, but also in the segmentation of cardiac MR images and X-ray breast masses with comparable results with state-of-the-art

To the best of my knowledge, there are very few works that have experimented with multi-labelling tasks due to the assigned complexity that comes with it. In a study, the authors have made use of multi-label learning as a way to quantify emphysema distribution by differential diagnosis of lung pathologies of five lung tissue patterns [21]. The method proposed not only diagnoses, but also attempts to quantify the severity and spread of emphysema in the volumetric 3D CT scans. Their approach made use a two-step approach similar to this study where volumetric segmentation of the lung is performed to differentiate is from the rest in the CT image, after which a multi-label models are used for the classification. The results obtained from their pipeline is compared to the diagnosis done by the radiologists to understand how well their proposed models have performed.

The need for labelled data to feed the data-hungry deep-learning models of today has been a problem that researchers have been working on for a long period time. Fortunately, with the improvements in Natural Language Processing technology, the ability to extract relevant information from reports have made great progress. Vast amounts of radiological studies performed on patients along with their respective radiological reports are stored in the Picture Archiving and Communications System (PACS) and electronic medical records systems in medical institutions. Researchers have made several attempts to extract the knowledge stored in these systems. X. Wang *et al.* introduced a new database called ChestX-ray8 which is collection of chest X-ray examinations, and have used natural language processing to extract a maximum of eight disease labels from the text-only reports associated with these scans, where cases may have more than one associated disease label attached to them [28]. The goal of their work was to make available large sets of publicly available labelled cases that can spark advancements and keep up with the demand of labelled data of many deep-leaning models of today. Other work was done by fellow researchers that have labelled CT cases using rule based models which will be described in more detail later in this study [27].

To summarize, current deep learning systems require a lot of training data that are sufficient for the models to learn the relevant features for good performance on unseen data, but due to insufficiency of annotated examples, radiologists manually provide the annotations, which is a time-consuming process and is an unreasonable long term solution. Unfortunately, with the increase in complexity of today's deep learning models

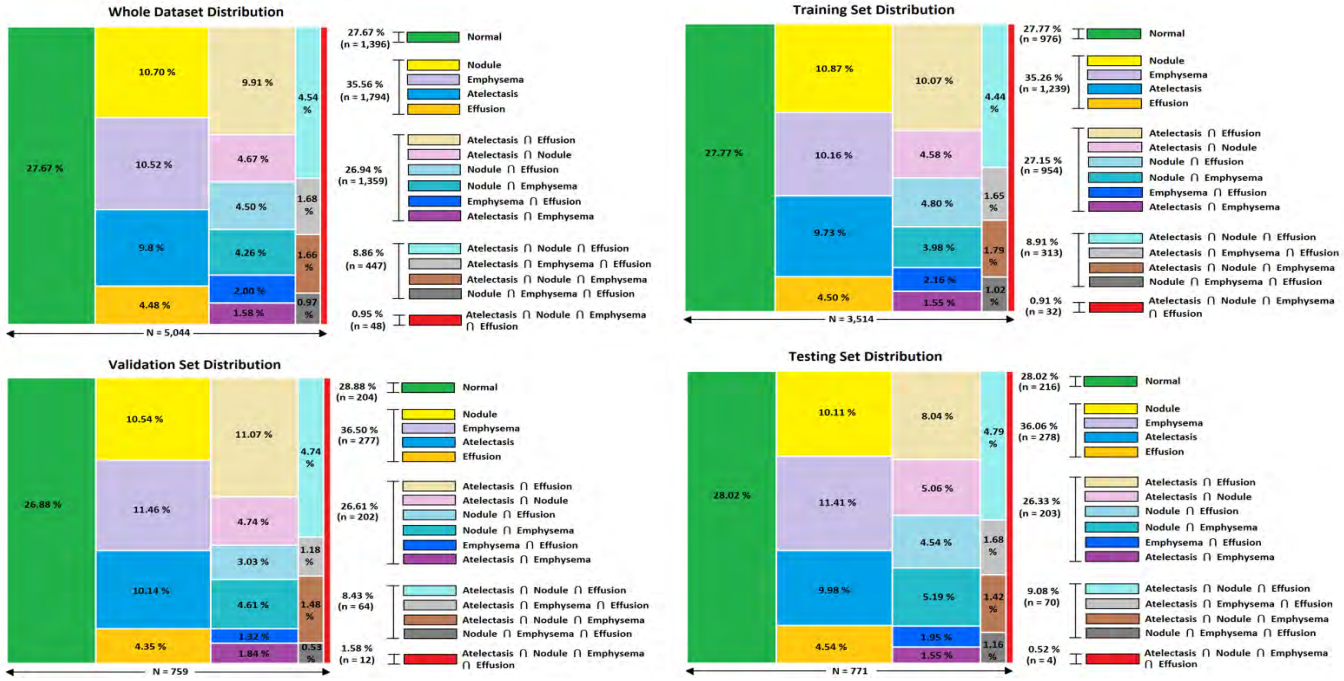


Figure 1: **Lung:** Data distribution between Training, Validation, and Testing Sets

which are ever more data-hungry, and with the slow pace of annotated samples, there lies a bottleneck.

2.2. Squeeze and Excitation

Convolutional Neural Networks (CNN) have made tremendous advancements in recent times and have proven to be real effective when used in various visual tasks [26][23][16], where the underlying idea behind convolutions is to create filters that learn certain representations of the images in both the spatial and channel-wise domains. From other works [15][7], we understand that integration of learning mechanisms that can help identify and extract correlations spatially, and with no additional supervision, can have great positive impacts in performance of deep learning models. Deep learning researchers have understood the potential that learned features have and thus attempted to engineer solutions by numerous architectures that focus on feature engineering, and creative ways to capture relevant features with proper reuse. Wide Residual Networks (WRN) use wider networks rather than deeper ones to yield better performances. Wide networks imply the increased use of kernels, thus creating more features in the network, but this brings about another challenge: are all channels relevant? Not every channel of the many created contribute to the overall result, and with additional channel information present in the network there is an inherent increase in the cost of computation. Beside the computation cost, there is a point at which increase in channel information can negatively impact the model performance. To overcome this issue, a new unit called Squeeze and Excitation [17] aims to find relationships within channels that can help identify and propagate features that are more relevant, and thus improve the representational quality of networks. The advantage of using these units is that they have very low computational requirements and add only slight increments in model complexities.

Squeeze and Excitation have been investigated in various studies. In [19], the authors have published work this year that aims to improve pulmonary nodule detection in lung cancer screening. They propose "DeepSEED" which is a 3D convolutional neural network with the use of Squeeze and Excitation. They tackle the class-imbalance problem in this architecture, by modifying the cross-entropy loss to decrease the false positive rates that are frequent when dealing with detections of nodules. The impact of Squeeze and Excitation in their work has been shown to outperform state-of-the-art detection models by a large margin. In another study [20], the authors try to deal with the class imbalance problem, but they propose a network that uses a "Squeeze-and-Excitation Pyramidal Residual Network" (SE-PyramidNet) with a following Generative Adversarial Network (GAN) to generate less represented classes, which had beat comparable state-of-art deep CNNs. The use of Squeeze and Excitation improved the identification accuracy thus increasing the likelihood of generated examples being correct by the GAN network. In another study done by P. Ghosal *et al.* [10], the authors use these units to help improve the classification performance of brain tumors of MRI images. They made use of the a simple ResNet-101 with

attached Squeeze-and-Excitation blocks to get improvements in sensitivity and specificity from other state-of-the-art performances.

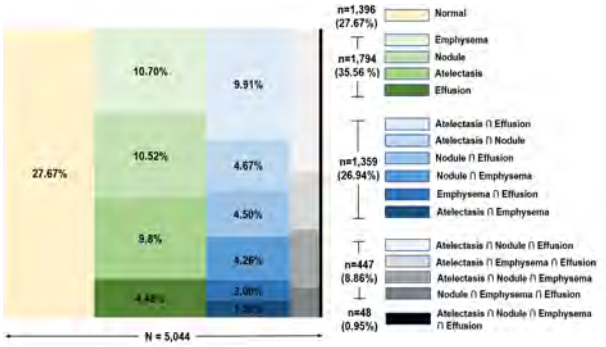


Figure 2: Data Distribution of Lung

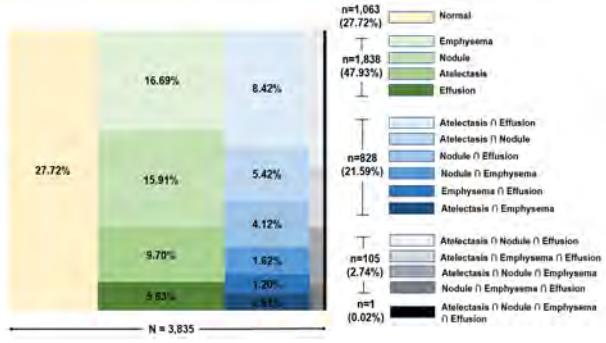


Figure 3: Data Distribution of Liver

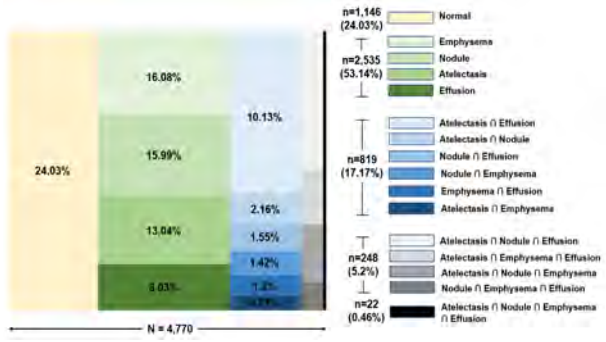


Figure 4: Data Distribution of Kidney

To summarize, the above-mentioned examples prove the use of Squeeze-and-Excitation in the medical domain, but they really have no bounds in their integration with works in other fields. In the absence of advancements that focus on feature engineering, it may be difficult to design and develop new CNN architectures which sometimes require extensive trial and error of hyperparameters, but by the use of units that focus on the selective importance of features, great improvements in model performances can be realized.

3. Material and methods

3.1. Dataset

The dataset used in this study are 3D Chest Abdomen Pelvis (CAP) CT scans taken at Duke University Health System from the 2012 to 2017. The downloaded dataset of cases for each organ type contained a total of 5,044 scans for lung/pleura, 3,835 scans for liver/gallbladder and 4,770 scans for the kidneys.

In each organ class (lung/pleura, liver/gallbladder, and kidney), we aim to assign multiple labels to cases. In lung, the existing labels are emphysema, nodule, atelectasis, effusion as the diseased labels, and a normal label when cases lack all abnormalities. As for liver, the labels are liver lesion, gallstone, fatty liver, dilation, and normal. For kidneys, the labels are kidney stones, kidney lesions, cyst, atrophy, and normal. It should be noted that each case can have multiple labels assigned to it, for example a case can not only have emphysema in the lungs but also presence of atelectasis and traces of effusion. An extensive visualization of the dataset showing the percentage of cases having no diseases, exclusively one abnormality, and the overlap in the presence of multiple abnormalities are shown in figures 2, 3, 4. For all three organs, there are many instances of cases having a single abnormality as well as multiple, concurrent abnormalities. This is true across all three organ types being studied.

The available data is divided into 70% for training, 15% for validation, and 15% for testing. Figure 1 shows the percentage of exclusive and overlapped labelled abnormalities in each of the training, validating, and testing sets for lung. It was important that the proportions are kept approximately similar across all sets. Ensuring this proper division of cases is a key step in ensuring we maximize our generalization capacity of models being developed.

3.2. Case labelling

Similar to previous studies using automated labelling of images, we took advantage of the information present in their corresponding radiological reports. To generate the labels for scans, a rule-based algorithm (RBA) was constructed that utilized text from over 300,000 reports of CTs [27]. Radiological reports contain four sections: scan indication, imaging technique, findings, and impression, but not all sections provide useful information that can lead to identifying a label. The section that had relevant information was the 'findings' sections since this was where radiologists note observations in the scan and can have detailed information about abnormalities in a patient. Other sections were not included since they provide information on past history, imaging techniques, and other data that can hinder the finding of the most appropriate labels.

In that study, a list of keywords were extracted that can positively identify an organ and also the potential

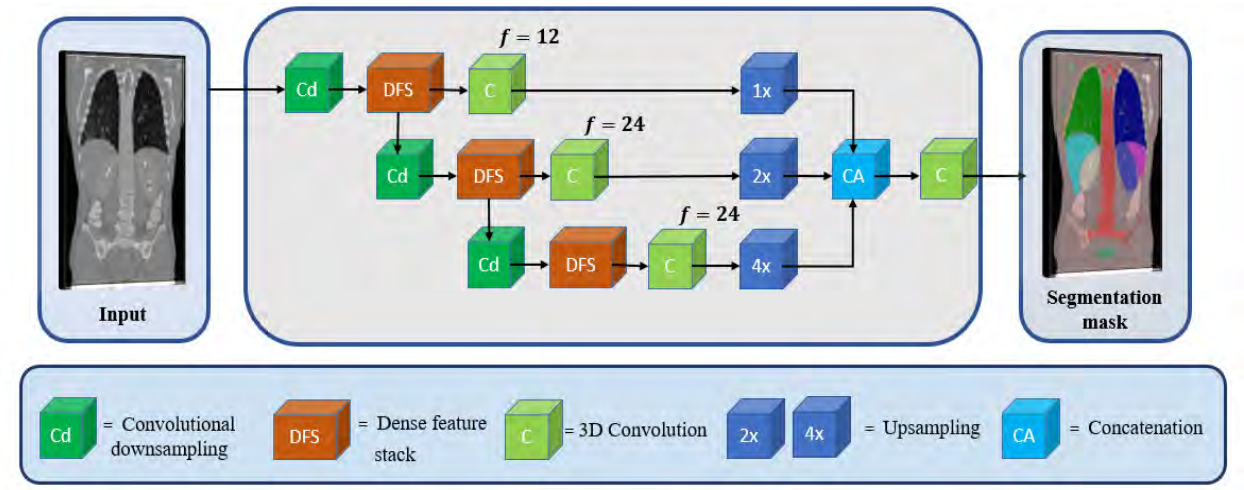


Figure 5: Segmentation Network Diagram

abnormalities present in that scan. Radiologists tend to use a set of common words to describe certain abnormalities in specific organs, and by creating a list of these keywords, it will be easier to discern the likelihood of the reports belonging to one over another. As much as there are common words, reports can contain text that are irrelevant to describing a disease or an organ. To overcome this challenge, the term frequency inverse document frequency (TFIDF) is used to filter out relevant words associated with a particular organ. Finally, the chosen reports taken forward were the ones where the RBA could positively identify a normal scan from a scan containing any of the four abnormalities. A more extensive study can be found here [5].

3.3. Pre-processing

Hounsfield units quantify density of materials in CT scans. Air having the least attenuation is assigned the lowest value of -1000 HU (Hounsfield Unit), water is assigned the value of zero, and the remaining scale up to a maximum of +3000 HU for metals having highest attenuation coefficients. In our study, since our focus is on chest abdomen and pelvic regions, the range of the HU scale chosen is [-1000, +800] for all lung CT volumes, and [-200, +500] for liver and kidney CT volumes because these ranges capture the necessary contrast of the anatomy in question. Once these ranges of HU units have been clipped, they are followed by a linear transformation for normalization.

Since the volumes are of high resolution and varying voxel spacing of the CT images from patient to patient, a resampling step is necessary to ensure smaller memory requirements but keeping relevant information and maintaining constant spacing among all 3D CT volumes. All CT volumes were re-sampled to a resolution with voxel sizes of $2mm \times 2mm \times 2mm$ via B-spline interpolations. To ensure segmentation maps are consistent with modified volumes, they too underwent the same transformation to maintain consistency.

3.4. Segmentation

The segmentation module is the fist module in the overall multi-label classification pipeline. 50 labelled CT volumes were used in its training and testing, where 44 random volumes were used for training and the remaining 6 volumes for validation. Here, three well known CNN architectures were experimented with: 3D U-Net [22], 3D FCN [18] and DenseVNet [11], and was later found that DenseVNet was the best performing, leading to its use as the final chosen segmentation model in the pipeline. It should be noted that the segmentation model was trained with normal cases only, therefore the performance dropped with the inclusion of diseased cases. To overcome this drawback, the base model's incorrect segmentation masks of the diseased examples were manually corrected and fed back to the network to retrain with the combined normal as well as corrected diseased cases to get the final fine-tuned segmentation model. More on this can be found here [27].

Figure 5 shows the network structure of the segmentation module. As seen in the above diagram, the segmentation network utilizes information at three resolutions. At each resolution, a convolutional downsampling operation is used to reduce the feature maps by half creating a dense feature stack followed by a 3D convolution at different levels with varying number of filters f. Since the dimensions of the created feature maps vary at each resolution, they need to be modified to the same dimensions, which is why upsampling blocks are used to reverse the action of downsampling of the convolutional downsampling block. After needed dimension matching at all levels, the feature maps are concatenated, and likelihood logits are generated by the last convolution operation.

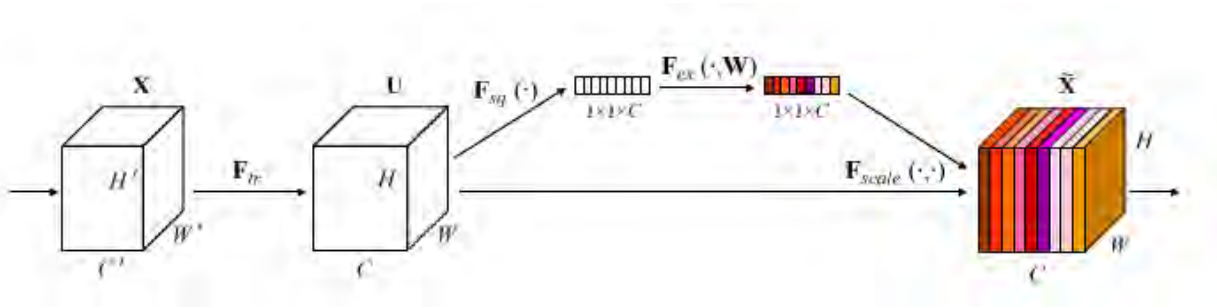


Figure 6: Squeeze and Excitation Block

3.5. Squeeze and Excitation Blocks

The structure and positioning of the Squeeze-and-Excitation block as illustrated by [17] is shown in figure 6. The **X** block represents an input image having dimensions $H' \times W' \times C'$ (height,width,channels) being sent through a transformation F_{tr} that can apply a certain transform (such as a convolution) the input **X** to get the features as represented by block **U** having dimension $(H \times W \times C)$. Since the block **U** represents the features generated, it is passed through a Squeeze and Excitation module which produces modulation weights for every channel which are later combined with the original feature maps of block **U** to generate feature maps with altered weights, which can be taken forward in the network.

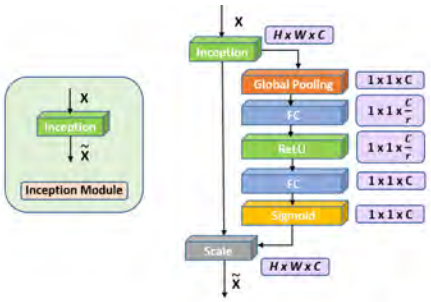


Figure 7: Inception Module *(left)* with its SE counterpart integration *(right)*

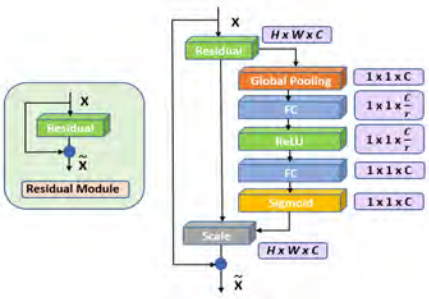


Figure 8: Residual Module *(left)* with its SE counterpart integration *(right)*

The Squeeze-and-Excitation has two parts as the name suggest: a squeeze followed by an excitation. The Squeeze operation's main objective is to generate per-channel representations obtained from the its spatial domain $(H \times W)$ by means of global averaging pooling or other mechanisms. Followed by this operation is the excitation phase that takes as input the representations of all channels made in the previous step and generate channel-wise attention weights that modify the weights of the initial weights of features U.

As stated earlier, one of the advantages of using SE blocks is its flexibility to work well when placed in any network architecture. To get a clear picture of the integration of the SE block into any existing framework, figures 7 and 8 inspired from the original paper are shown where a comparison can be drawn by the representation of the original Inception and Residual modules alongside their Squeeze-and-Excitation module counterparts. Figures shown are just an example, but the SE units can be used in any modern deep learning architectures.

Following is an explanation and break-down of the two operations: Squeeze and Excitation

3.5.1. Squeeze

Output features learned after an operation such as convolution, are receptive only to certain local receptive fields, and is the case with all points in the feature space. Adjacent points may share overlaps in the receptive fields, but in general all points of the features are generated from different receptive fields. Thus, each unit is unable to take advantage of the contextual information present outside its local receptive field. To overcome this problem, the information present outside this region needs to be exploited and this is where the squeeze operation comes in. This operation condenses the spatial information of all units in a given channel to a statistic that can be understood as a general description of its corresponding channel. This operation can be achieved through various aggregation techniques, but in this study a global average pooling is used. To understand the impact of other techniques, the original paper [17] has more information. Following equation 1 is the mathematical representation of the squeeze operation.

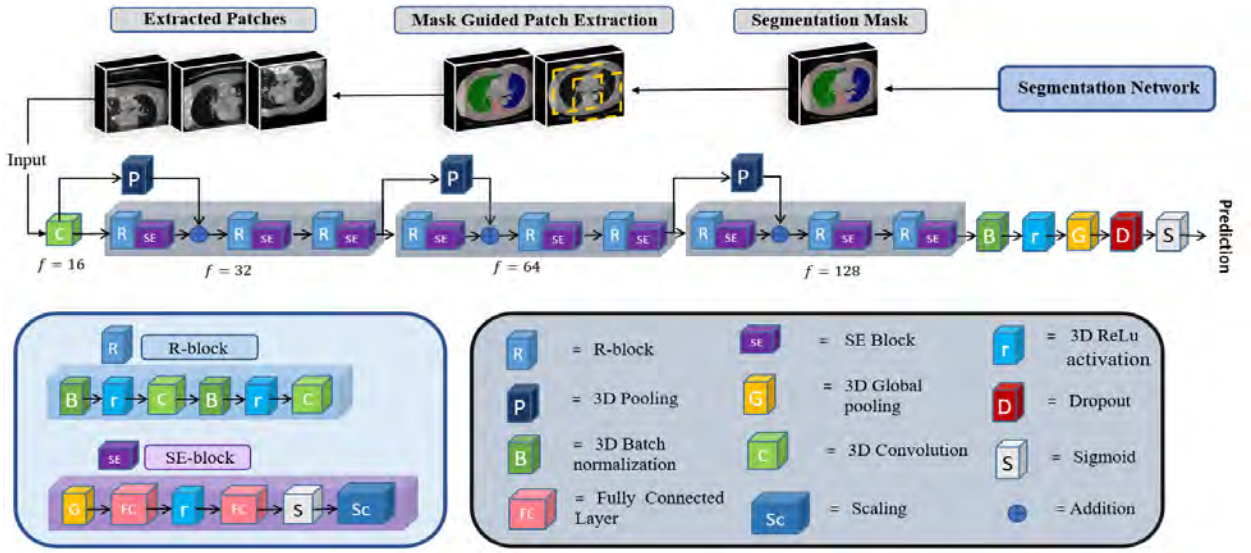


Figure 9: Classification Network Diagram with integrated SE blocks

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \qquad (1)$$

A statistic $\mathbf{z} \in \mathrm{R}^C$ is obtained after reducing $\mathbf{U}$ through its spatial dimensions $H \times W$, such that the c-th element of the statistic $\mathbf{z}$ shown above is computed.

3.5.2. Excitation

After the squeeze operation has been completed and channel-wise descriptors are generated, the excitation phase's objective is to find relationships between those descriptors. In order to find hidden dependencies between the channel descriptors, two criteria have to be met for the operation to find meaningful relationships. Firstly, it should have the capacity to learn non-linear relationships to allow flexibility, and secondly, ability to capture non-mutually exclusive relationships to ensure more than one channel to be emphasized rather than only a single channel.

Use of these blocks have their advantages, but another parameter exists that controls their complexity to improve generalizability: dimensionality reduction layer with reduction ratio r. This parameter controls the fraction of original channel descriptors that get emphasized by the excitation phase, which in turn has a direct effect on the number of parameters getting passed onto later stages in the network. To make this happen structurally, the channel descriptors generated by the squeeze operation are first passed through a fully connected layer parameterized by r (and thus reducing the number of channels), followed by the activations (excitation phase) and later reinstating the original number of channels by another fully connected layer with the initial number of channels. These transformations can be seen in figures 7 and 8.

3.5.3. Chained Squeeze and Excitation

From the previous study [17], it is evident that the use of Squeeze and Excitation has performance gains with the smart reuse and recalibration of feature maps, but these experiments have been done using a single SE block per layer. Depending on the image size being dealt with, usage of a single SE block can have varied results. In the experiments performed in the original paper, the images used are from ImageNet with dimensions 256×256, which are far smaller in comparison to the task at hand. In our case, dealing with 3D CT volumes have a lot of redundant and often similar information present between slices, thus using only one SE block per layer may be insufficient for the network to find relevant features on which to perform recalibration. With higher volumes of data come greater need for better feature recalibration capabilities, which is the inspiration behind linking multiple SE blocks.

3.6. Image Classification

3.6.1. Classification Network Design

In an ideal case of classification, the whole image is taken in as input and is trained on through a series of steps, but in this study, due the large volumes of 3D CT images that can be 512x512x1000 or more in size, a different approach had to be taken that does not exceed the memory limitations of the computing systems. Thus, a patch-based approach had been taken. To accomplish this, the segmentation masks are used as guides to extract the patches for each of the organ classes. For example, the segmentation mask of the lungs generated by the segmentation module is used in finding the lung region within the CT volume within which patches were then extracted. Thus, this process can be understood as segmentation-guided patch extraction strategy.

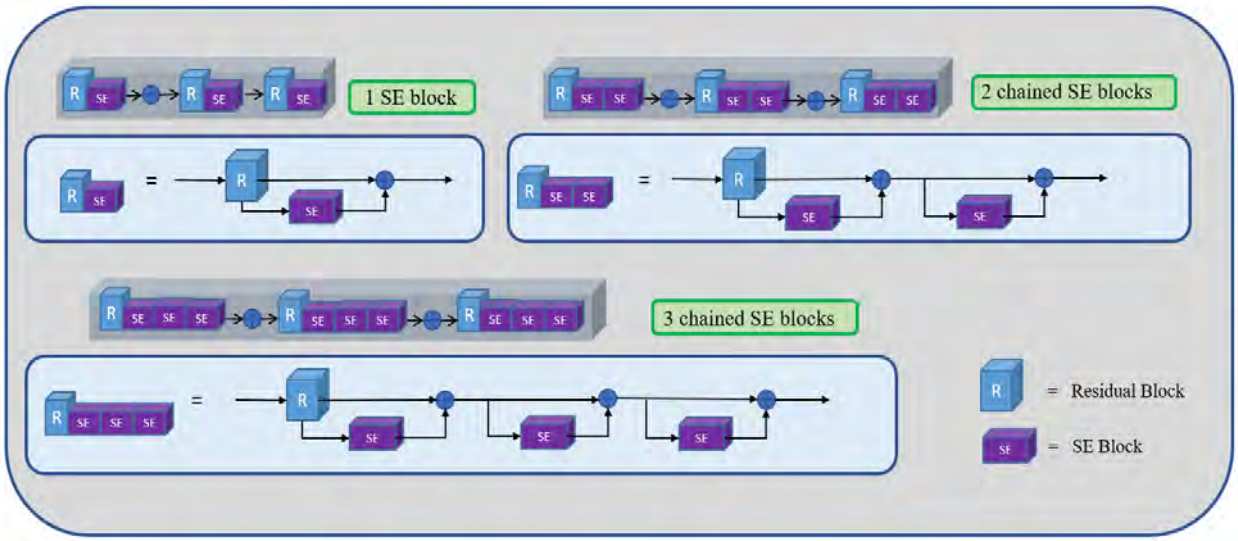


Figure 10: Structure of chained SE blocks

This had been performed for all organ classes namely lungs/pleura, liver/gallbladder, and kidneys. An added advantage of this strategy is that is allows for patches to have less of unwanted background information in the patches while focussing more on the region of the organ.

The baseline for this study is work done by a fellow researcher [Tushar et al] under the department of radiology at Duke University Health Systems of whose work was in turn inspired by the 3D CNN ResNet [13]. Following is a diagram that visualizes the overall structure of the classification network with integrated Squeeze and Excitation Blocks. As seen in the above figure, the input to the network are patches containing parts of the original 3D CT scan for the respective organ. The patch size selected for the lungs is $(128 \times 160 \times 160)$ while for liver and kidneys a patch size of $(96 \times 128 \times 128)$ was taken so that each organ could be accommodated within a patch and avoid including background information which would have adverse effects in our model performance.

The network consists of three major sections, where each one resembles learning at a lower resolution. Each resolution consists of pairs of Residual and SE blocks(one or more) occurring three times. After the first Residual and SE block at beginning of each resolution, the output of the previous layer is concatenated so as to reinforce learned features from the previous layer. It should be noted that the number of filters is doubled every layer to increase learned features. Each Residual block at the different layers consist of two repetitions of batch normalization, Rectified Linear Unit (ReLU) activation, and 3D convolutions. On the other hand, an SE block is composed of global average pooling, a fully connected (FC) layer, followed by ReLU activation with an additional FC layer, a sigmoid and completed by a scaling operation to get the channels back to original dimensions for concatenation.

The chained system of SE blocks is shown in a graphical representation in Figure XXX:

The output of the final resolution having 128 channels is passed through a batch-normalization and a ReLU followed by a global max-pooling, dropout of 0.25, and finally a softmax classification layer to generate the final prediction.

3.6.2. Network Training

The first task was to train the segmentation network to generate the segmentation masks of the chest-abdomen organs. As mentioned earlier, the use of DenseVNet was chosen due to its superior performance in the validation set compared to other tested architectures. After fine-tuning this network with the use of both normal cases and ones with abnormalities, it was used on the original 3D CT volumes to get the final segmentation masks of the chest-abdomen organs. Patches of dimensions $(128 \times 160 \times 160)$ for lungs and $(96 \times 128 \times 128)$ for both liver and kidneys were generated guided by the segmentation masks. Therefore, the segmentation masks not only helped in distinguishing the relevant organ types (lungs/pleura, liver/gallbladder, and kidney), but also in facilitating the extraction of patches that met memory restriction of our computing resources that would later be fed into the classification network.

Other training parameters included a kernel initialization of uniform distribution, a weighted cross-entropy loss function with an Adam optimizer for weights, with roughly 70 epochs of training. As for the computing resources, four Nvidia TITAN RTX GPUs were used with a memory of 11GB each for all computations.

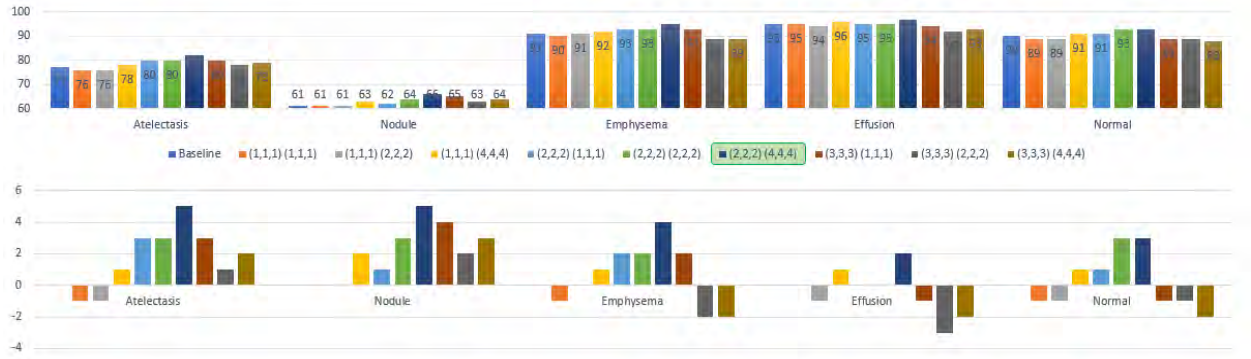


Figure 11: **Lung:** Absolute *(top)* and Relative *(bottom)* Multi-label AUC performance

Figure 12: **Liver:** Absolute *(top)* and Relative *(below)* Multi-label AUC performance

3.6.3. Experiments and Results

Having the optimal batch-size can help models get better at generalization. Experiments were first done to find the optimal batch size for this study. We ran tests with batch size of 6, 9, and 12 for lungs to observe the performance change. The figure 13 represents the training and validation learning curves taken when the mentioned batch sizes were used.

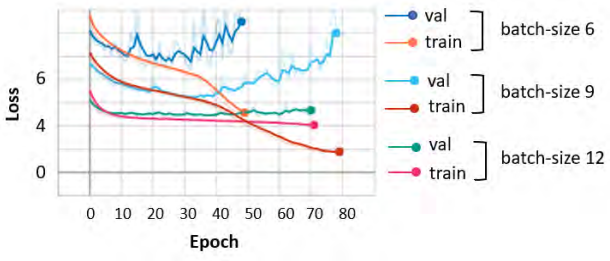


Figure 13: Learning curves

As seen from the curves in figure 13, it is evident that using a batch size of 9 yielded a better trade-off among the ones experimented with, which was the parameter of choice in the all experiments that followed. The learning curves when using a batch-size of 6 yielded lower error rates, but there is no indication of network learning since both training and validation curves do not diverge, which can be associated to a case of an under-fit model.

The effect of using SE blocks has been shown to be very promising in studies discussed earlier, but to investigate its impact in our dataset we ran an initial test applying a single Squeeze-and-Excitation block coupled into a residual unit for the lungs. AUC performance being our primary metric for comparison, we monitor the change in AUC scores in each category of labels from the experimental results when compared to the baseline having no SE blocks in the network. To test the hypothesis that larger images (such as the data being worked with in this study) require multiple SE blocks to identify and propagate useful features, experiments were performed using a range of SE blocks and reduction ratios. We ran experiments with varying number of squeeze-and-excitation units with constant reductions, and later with varying reductions while number of squeeze-and-excitation blocks were kept constant. Figures 11, 12, and 14 show the results obtained.

To clarify the notations used, it should be reiterated that the classification network has features learned at three resolutions, thus the representation *(x1,x2,x3) (y1,y2,y3)* in the figure implies that *x1* number of blocks with a corresponding *y1* reduction ratio was used in all SE blocks in the first stage, *x2* number of blocks with a *y2* reduction ratio used in all SE blocks in the second

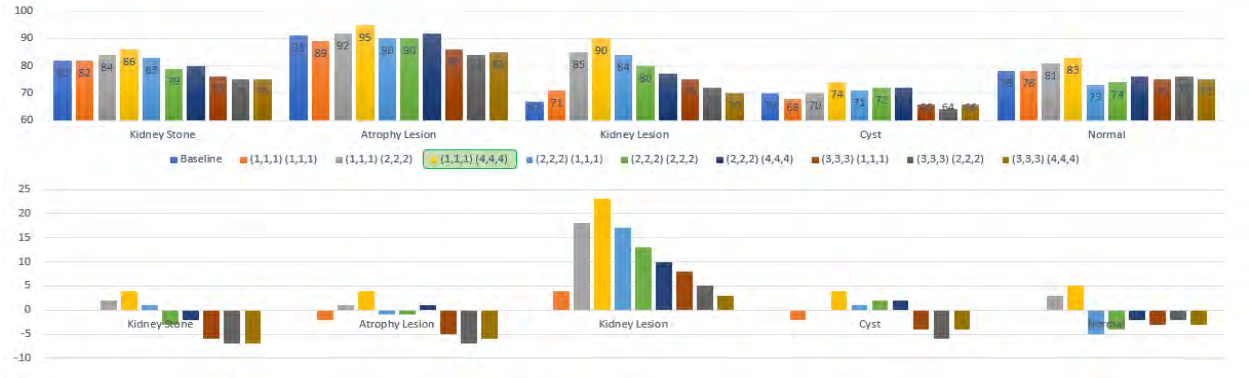


Figure 14: **Kidney:** Absolute *(top)* and Relative *(below)* Multi-label AUC performance

stage, and finally *x3* number of blocks with a *y3* reduction ratio was used in all SE blocks in the third stage.

Note there is a consistent pattern in each abnormality category, where AUC gradually increases up until a certain combination of parameters (SE blocks and reduction ratios), then steadily decreases. The point of best performance is different for each organ as highlighted in green, but the pattern seems to hold for all. This pattern is best shown in the bottom row where AUCs are shown relative to the baseline model, and occurred independently of the absolute AUC performance (top row) in each organ. When taking into consideration a fixed reduction ratio, the number of SE blocks that reach best performance is two for Lungs and Liver while a single block is sufficient for Kidney, while on the other hand when keeping the number of blocks constant and varying the ratios, an increase in reductions results in better performance until the optimal configuration, and has a reverse effect thereafter.

4. Discussion

Previous studies in medical imaging have often dealt with single label problems such as the presence of a disease, but there have not been as many multilabel problems being studied. Having CAD systems that can label multiple abnormalities for a given case can help physicians make faster decisions resulting in better healthcare to patients. Current CAD systems utilize techniques that can have significant positive impact on the task being handled, and one such structure that has been explored in this study is the impact Squeeze and Excitations have on network performance. Although many prior researchers made use of these units, they have been done so in a reserved manner by limiting the extent to which Squeeze-and-Excitation is used, such as a single SE block. Since the core idea of using these units is to prioritize relevant features among many other non-essential ones, the inspiration to use multiple chained units stems from the fact that the volumes used in this study are substantially larger which demanded more attention to feature selection.

In this study, we evaluated the effects of using different numbers of SE blocks and reduction ratios, and found that performance would peak for a certain optimal combination. Increasing the SE blocks by a larger degree can have negative impact on the model. Adding blocks into the network increases the complexity by a small degree, but too many can make it harder for generalization and result in performance worse than the baseline. It can be concluded that having a right balance of SE blocks and reduction ratios can bring about the best performance in deep learning models using them.

The hyperparameter combination delivering the best performance within each organ shifted from more to less complexity when moving from Lungs to Liver to Kidney. One possible explanation for this shift could be the size of the organ being experimented with. Anatomically, lungs/pleura is the largest organ by volume among the three with liver being second, and finally kidney being significantly smaller. It could be argued that since larger organs contain more information, it may require more complex squeeze-and-excitation structures to condense the information and find useful discriminative features.

When taking kidney patches into consideration, since the volume is considerably smaller than lungs and liver, the need for number of repetitions of feature recalibration is lower. Remarkably, the improvement in performance was observed for all of the diseases with different appearances occurring in three separate organs. The one particular abnormality that benefited the most from our proposed method was kidney lesions, suggesting that even a small organ like the kidney can benefit from this feature reduction and boosting strategy.

From this study we find that added use of these units to a certain extent can help improve performance, but further aggressive use can be harmful.

There are a few limitations in this this work. The results suggest that extent of use of these units are data dependant and thus should be adapted for the problem

being solved. Additionally, since many possible combinations exist that can yield different results, performing a grid search technique is not feasible since each set of training can take more than ten GPU-days. Thus, a more systematic and hypothesis driven approach should be taken when selecting the parameters to experiment with.

In future works, a few types of experiments can be performed for this task. In the results shown, each resolution makes use of the same set of parameters across the layers, but a more dynamic approach can be taken with gradual increase or decrease in SE blocks and/or reductions. Since this study dealt with smaller number of blocks and reductions with a bell-shaped improvement, it may be possible that this pattern can emerge when dealing with higher parameters. Besides optimizing parameters of this unit, it is also possible to experiment with other placement strategies of the unit within the network. As stated in the original paper [17], preplacement, post-placement and Identity placements of these units can be tried. It can also be possible to try other Squeeze and Excitation techniques as suggested in [1]. Here they propose spatial squeezing with channel excitation, channel squeezing with spacial excitation, and finally a concurrent spatial and channel squeeze with channel excitation.

5. Conclusions

Squeeze-and-Excitation units have great potential in improving model performance when used effectively. The challenge would be to first understand the problem being dealt with and finding the appropriate use to these blocks to salvage more relevant features. In this study, since the problem was in dealing with larger volumes, it inspired the need to use these feature-recalibration units more aggressively, resulting in consistent improvement in performance across multiple diagnostic tasks. Therefore, understanding the fine balance between these parameters and adjusting them to suit the needs and requirements of the problem is something that needs to be given extra thought.

6. Acknowledgments

I would firstly like to acknowledge Fakrul Islam Tushar for his initial work on this problem before I took it over. He helped provide a smooth transition that allowed me to perform my study and answer any questions I had with regards to the project. I would also like to thank Brian Harrawood for his assistance in allowing access to computing resources and ensuring system downtime was kept minimal. Having weekly check-ins with my supervisor Dr. Joseph Lo and occasional suggestions from Tushar is greatly appreciated since their feedback were of great help in completing this study.

References

[1] Christian Wachinger Abhijit Guha Roy Nassir Navab. "Concurrent Spatial and Channel Squeeze Excitation in Fully Convolutional Networks". In: *MICCA* (2018).

[2] F. P. Romero et al. "End-To-End Discriminative Deep Network For Liver Lesion Classification". In: *IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (2019), pp. 1243–1246.

[3] Wentao Zhu et al. "Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification". In: *IEEE WACV* (2018), pp. 673–681.

[4] Y. Xie et al. "Knowledge-based Collaborative Deep Learning for Benign-Malignant Lung Nodule Classification on Chest CT". In: *IEEE Transactions on Medical Imaging* 38.4 (2019), pp. 991–1004.

[5] Fakrul I. Tushar Anindo Saha Khrystina Faryna. "Weakly supervised 3D classification of chest CT using aggregated multi-resolution deep segmentation features". In: *SPIE* (2020).

[6] Bhargavan M Berrington de González A Mahesh M. Kim KP. "Projected cancer risks from computed tomographic scans performed in the United States in 2007". In: *Arch. Intern. Med.* 169.22 (2007), pp. 2071–2077.

[7] Y. Jia et al. C. Szegedy W. Liu. "Going deeper with convolutions". In: *CVPR* (2015).

[8] X. Chen, R. Zhang, and P. Yan. "Feature Fusion Encoder Decoder Network for Automatic Liver Lesion Segmentation". In: (2019), pp. 430–433.

[9] "Dangers of CT Scans and X-Rays - Consumer Reports". In: (2018).

[10] P. Ghosal and L. Nandanwar et.al. "Brain Tumor Classification Using ResNet-101 Based Squeeze and Excitation Deep Neural Network". In: (2019), pp. 1–6.

[11] E. Gibson et al. "Automatic Multi-Organ Segmentation on Abdominal CT With Dense V-Networks". In: *IEEE Transactions on Medical Imaging* 37.8 (2018), pp. 1822–1834.

[12] W. Qian H. Jiang H. Ma. "An Automatic Detection System of Lung Nodule Based on Multigroup Patch-Based Deep Learning Network," in: *IEEE Journal of Biomedical and Health Informatics* 22.4 (2018), pp. 1227–1237.

[13] K. He et al. "Deep Residual Learning for Image Recognition". In: (2016), pp. 770–778.

[14] Jonathan H. Sunshine Ingrid M. Burger Nancy E. Kass. "The Use of CT for Screening: A National Survey of Radiologists' Activities and Attitudes". In: *RSNA* (2008).

[15] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *ICML* (2015).

[16] E. Shelhamer J. Long and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *CVPR* (2015).

[17] Li Shen Jie Hu and Gang Sun. "Squeeze-and-excitation networks". In: *Conference on Computer Vision and Pattern Recognition* (2018), pp. 7132–7141.

[18] Trevor Darrell Jonathan Long Evan Shelhamer. "Fully Convolutional Networks for Semantic Segmentation". In: *CVPR* (2015).

[19] Y. Li and Y. Fan. "DeepSEED: 3D Squeeze-and-Excitation Encoder-Decoder Convolutional Neural Networks for Pulmonary Nodule Detection". In: (2020), pp. 1866–1869.

[20] J. Liu et al. "Teaching Squeeze-and-Excitation PyramidNet for Imbalanced Image Classification with GAN-based Curriculum Learning". In: (2018), pp. 2444–2449.

[21] M. Negahdar, A. Coy, and D. Beymer. "An End-to-End Deep Learning Pipeline for Emphysema Quantification Using Multi-label Learning". In: (2019), pp. 929–932.

[22] Thomas Brox Olaf Ronneberger Philipp Fischer. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *MICCAI* (2015).

[23] R. Girshick S. Ren K. He and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *NIPS* (2015).

[24] Kavita Bala Sean Bell C. Lawrence Zitnick. "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks". In: *Computer Vision and Pattern Recognition* (2015).

[25] Marcus R Smith-Bindman R Lipson J. "Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer". In: *Arch. Intern. Med* 169.22 (2009), pp. 2078–2086.

[26] A. Toshev and C. Szegedy. "DeepPose: Human pose estimation via deep neural networks". In: *CVPR* (2014).

[27] Fakrul Islam Tushar et al. *Weakly Supervised Multi-Organ Multi-Disease Classification of Body CT Scans*. 2020. eprint: `arXiv:2008.01158`.

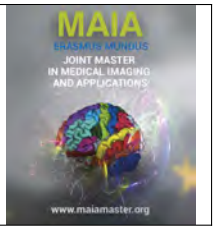
[28] X. Wang et al. "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". In: (2017), pp. 3462–3471.

[29] X. Yan et al. "An Efficient Hybrid Model for Kidney Tumor Segmentation in CT Images". In: (2020), pp. 333–336.

[30] Q. Yu et al. "Crossbar-Net: A Novel Convolutional Neural Network for Kidney Tumor Segmentation in CT Images". In: *IEEE Transactions on Image Processing* 28.8 (2019), pp. 4060–4074.

Medical Imaging and Applications

Master Thesis, August 2020



Multi-Resolution 3D Convolutional Neural Networks for Automatic Coronary Centerline Extraction in Cardiac CT Angiography Scans

Zohaib Salahuddin, Hannes Nickisch and Matthias Lenga

Philips Research Hamburg, Germany

Abstract

Heart Disease is the leading cause of deaths in the United States (Heron, 2020). Among heart diseases, Coronary Artery Disease (CAD) is the most common type of heart disease responsible for the death of 365,914 people in the US in 2017 (Benjamin et al., 2019). Cardiac Computed Tomography Angiography (CCTA) provides a non-invasive way for the rapid visualization of the heart in order to aid in the diagnosis of coronary artery disease. The analysis of the tubular structure of coronary arteries in 3D CCTA scans is a highly intricate, difficult and time consuming task. Coronary centerline extraction in the CCTA scans is a prerequisite for the evaluation of stenoses and atherosclerotic plaque (Hampe et al., 2019).

We propose a novel deep learning-based fully automatic coronary artery centerline extraction method. A dual pathway Convolutional Neural Network (CNN) operating on multi-scale 3D local input patches is used to predict the direction towards the centerlines of the coronary arteries from the center of the patch as well as the presence of a bifurcation simultaneously. Two or more continuation directions are derived based on the result of bifurcation detection. This iterative tracking scheme is initialized from a model based segmentation of the heart which places distinct landmarks at the left and right ostium points. The tracker detects the entire left and right coronary tree based on these two seed points, taking steps in accordance with the predicted directions and the patch type prediction. A similar multi-scale dual pathway 3D CNN is trained to identify coronary artery endpoints for terminating the tracking process.

The 3D CNNs were trained using a Philips proprietary dataset consisting of 43 images obtained from nine different sites. Four-fold cross validation was performed on the dataset. An average sensitivity of 87.1% and clinically relevant overlap of 89.1% was obtained on the philips dataset. In addition, the MICCAI 2008 Coronary Artery Tracking Challenge (CAT08) training and test dataset was then used as a test set in order to evaluate the generalization and benchmark the performance of the algorithm. An average overlap of 93.6% and clinically relevant overlap of 96.4% was obtained. The proposed method achieved better performance in terms of overlap metrics than the current state-of-the-art automatic centerline extraction techniques on CAT08 dataset with a vessel detection rate of 95%. In case the vessel detection by the automatic method fails, the vessel can be retrieved by specifying one point on the coronary artery. This proposed algorithm can also be used to obtain centerlines related to other tubular structures, e.g. rib centerlines in thorax CT images.

Keywords: CCTA, Coronary Artery Disease, Centerline Extraction, Multi Resolution CNN, Bifurcation Detection, Tracking

1. Introduction

Coronary artery disease is one of the leading causes of deaths worldwide. It was responsible for 9.43 million deaths in 2016 (WHO, 2018). Coronary arteries are responsible for supplying oxygenated blood to the heart muscles. Two main arteries branch off the aorta namely Left Main Coronary Artery (LCA) and Right Coronary Artery (RCA) which supply blood to left and right parts of the heart respectively. These two main arteries then divide into a network of smaller coronary

arteries which wrap themselves around the heart. Coronary artery disease is the narrowing or blockage of these coronary arteries due to the build up of cholesterol and fatty deposits called plaque on the inner lining of the arterial wall. This constriction can result in an inadequate supply of blood to the heart muscles which can be fatal (Malakar et al., 2019). Hence, there is a need for timely diagnosis and detection of this constriction in the arteries.

Coronary Angiography (CA) is an invasive procedure for coronary artery disease evaluation which provides information only related to the coronary lumen. Coronary angiography requires a contrast agent and is often performed along with cardiac catheterization. Complications due to the invasive nature of coronary angiography occur in less than 2% of the cases, with mortality of less than 0.08% (Tavakol et al., 2012). Hence, there is a non-negligible risk associated with coronary angiography. Computed Tomography Angiography (CCTA) is a non-invasive alternative which provides information on the extent and type of plaque present (Paech and Weston, 2011). CCTA images have a high spatial resolution consisting of hundreds of slices. However, CCTA acquisitions expose the patient to a higher dosage of radiation. Manual reading of volumetric CCTA images is a time consuming task even for trained experts due to the size and diversity of the arteries. Due to increasing number of CCTA scans, automatic analysis of CCTA images and improved 3D visualization is desirable.

There are many techniques to visualize the coronary arteries in the CCTA images such as maximum intensity projection (MIP), volume rendering techniques (VRT), multi planar reformatting (MPR) and curved multi planar reformatting (cMPR) (Cademartiri et al., 2007). Such advanced visualization techniques facilitate image reading and are, for example, used to guide stenosis and plaque detection (Stimpel et al., 2018). The computation of MPRs and cMPRs typically relies on centerlines of the coronary arteries. Hence, an important building block in the diagnosis of coronary artery disease is the extraction of coronary artery centerlines.

Manual extraction of coronary artery centerlines is time consuming, error prone and has a large inter-operator variability. In order to support the radiographer in the extraction of coronary artery centerlines, many interactive, semi-automatic and automatic methods for coronary centerline extraction have been proposed. The reformatted images obtained using centerlines can also be used for other purposes such as lumen segmentation of coronary arteries (Huang et al., 2018). Deep learning and machine learning-based methods typically use coronary artery centerline extraction as a preprocessing step for the plaque identification and stenosis analysis (Hampe et al., 2019). A recurrent neural network was used by Zreik et al. (2019) to detect stenosis from multi-planar reformatted (MPR) images which were reconstructed using extracted coronary centerlines. Hence, an automatic coronary artery centerline extraction algorithm which provides consistent performance on CCTA images with variable image quality and calcium scores in a few seconds is desirable.

We propose a fully automatic coronary centerline extraction pipeline based on dual pathway multi-scale 3D convolutional neural network. This pipeline comprises of three modules. The first module called Direction and Bifurcation Classification network (**DBC-Net**), is a multi-scale 3D CNN for a local patch to determine the direction towards the center of the coronary artery with respect to the center of the patch as well as the patch type (normal or bifurcation). The second module namely Stop Patch Classification network (**STC-Net**), consists of another multi-scale 3D CNN to determine if the patch contains the artery or not. The third module called **Tracker**, orchestrates the centerline extraction. The tracking is initialized at two ostium points obtained automatically. The tracker obtains predictions for directions and patch type from the DBC-Net for each patch. The tracker then takes steps in order to determine the centerline of the arteries. The tracker terminates based on the output of the STC-Net.

We propose an Automatic Coronary Tracking (**AuCoTrack**) method which was evaluated using four-fold cross validation on a Philips dataset. In order to compare this approach to state-of-the-art methods, an evaluation was conducted on the MICCAI 2008 training and test dataset of Coronary Artery Centerline Extraction Challenge. Additionally, an analysis of the algorithm was performed to correlate the qualitative and quantitative analysis as well as the failure cases with research findings.

2. State of the art

In order to extract the coronary artery centerlines, three types of approaches can be adopted: *automatic*, *semi-automatic* and *interactive*. According to MICCAI 2008 Coronary Artery Centerline Extraction Challenge guidelines as specified in Schaap et al. (2009), an approach may be defined as fully automatic if it utilizes no manually placed initialization points to track the entire coronary tree. If one point per vessel is provided

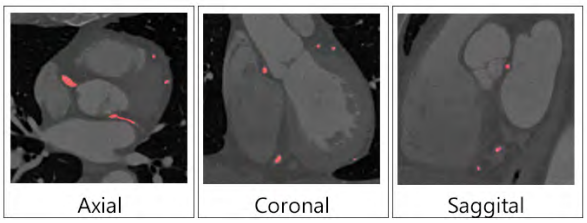


Figure 1: Axial, coronal and saggital slices of a CCTA image. The coronary arteries are overlayed as red. The presence of vessel like structures around the heart make the extraction of coronary centerlines complicated.

to extract the coronary tree, the approach is said to be semi-automatic. If more than two points per vessel are required to obtain the coronary tree, it is labelled as an interactive approach. Extraction of the entire coronary tree based on interactive and semi-automatic approaches requires anatomical knowledge and manual inspection of the CCTA image to place points for each coronary artery individually. Since there are a lot of coronary arteries present in each coronary tree, these approaches increase the processing time. Hence, there is a need to establish a robust and automatic coronary artery centerline tracking algorithm which requires minimal user interaction.

A multiple hypothesis tracking approach based on mathematical template vessel model combined with standard minimal paths method was used by Friman et al. (2020) to extract coronary artery centerlines. Standard minimal path-based methods experience shortcut issues and may require a lot of interaction to extract the entire vessel tree. At the time of this publication, Friman et al. (2020)'s method ranked first on MICCAI 2008 Coronary Artery Centerline Extraction Challenge as an interactive method. It requires 2.6 points on average per vessel and takes 6 min to extract four coronary arteries per CCTA image. Schaap et al. (2009) used multivariate linear regression on image intensities to estimate an initial vessel boundary followed by a subsequent refinement of this result using non-linear regression. This method requires 2.2 points per vessel and takes 22 min to extract four coronary arteries per CCTA image. High processing times along with repeated user interaction per vessel is not desirable in clinical practice.

Krissian et al. (2008) used morphological operations and denoising filters to obtain a region of interest. The probability of belonging to the coronary artery class for each voxel was then determined using a fuzzy classifier. The start points were automatically determined and the end points were provided manually for each vessel. A minimal path between these two points was traced based on voxel probability map generated by the classifier to obtain the centerlines. This semi-automatic method takes 7 h to extract four coronary arteries per CCTA image. Cetin Karayumak et al. (2012) used a second order tensor constructed from directional intensity measurements to track the entire coronary tree from a single seed point placed at the center of the cross-section of one of the vessels. This method utilizes an automatic branch detection based on K-means clustering of the intensity values. As a pre-processing step, a calcification filter is applied which requires annotations by an expert on the training CCTA scans. This methods takes 8 to 10 min on a 2.67 GHz dual processor to detect coronary arteries per CCTA scan. Cetin Karayumak and Unal (2015) also proposed an extension of this method to utilize cylindrical flux-based higher order tensor (HOT) in 4D which also solves the problem of branch detection. This method takes 30 s to detect coronary arteries per CCTA scan on a Intel Processor Xeon X560 @ 2.67 GHz CPU computer of 64 GB memory.

State-of-the-art performance for automatic coronary centerline extraction was achieved by Zheng et al. (2013). This method utilizes a segmentation mask to define a vessel specific region of interest (ROI) in order to constrain the centerline refinement by their model driven algorithm for extracting the main branches. The side branches are then traced by using region growing based on lumen segmentation. It was trained on 108 images of their proprietary dataset and takes 60 s to extract coronary arteries per CCTA scan. Kitamura et al. (2012) constructed a shape model of the coronary vessels and an Adaboost classifier in order to differentiate between normal and abnormal vessels for automatic centerline extraction. This method was trained on a proprietary dataset and the entire coronary tree centerline extraction takes 160 s per CCTA scan. Frangi et al. (1998) introduced a multiscale vessel enhancement filtering which obtained a vesselness measure based on eigen values of a Hessian. Yang et al. (2011) employed an improved version of Frangi's multiscale vessel enhancement filtering to obtain an initial tree which was further refined by branch searching automatically. This method takes 120 s on a standard desktop computer to track the entire coronary tree in a CCTA image.

Some methods utilize various handcrafted features such as virtual contrast and morphological operations. These handcrafted features are based on certain assumptions and they require explicit modelling in cases when the underlying assumptions do not hold e.g. bifurcations (Cetin Karayumak et al., 2012; Cetin Karayumak and Unal, 2015; Frangi et al., 2000; Krissian et al., 2008; Wang and Smedby, 2008).

Recently, an iterative CNN tracker was proposed in order to extract centerlines (Wolterink et al., 2019). This method does not require any handcrafted features. They proposed a serial tracker that utilizes the direction and step-size predicted by the CNN in order to obtain the centerlines. They were able to achieve near state-of-the-art performance as an interactive method. This method requires at-least one seed point per vessel in order to extract its centerline. Some vessels require more than one points due to premature termination of the tracking algorithm. An additional CNN to extract seed points for the vessels was also proposed in order to make the algorithm automatic. However, a limitation of this algorithm is that the seed identification CNN requires training images in which all the coronary arteries have been annotated (Wolterink et al., 2019). Hence, this method requires 10 s to extract 4 coronary arteries per CCTA scan. Bifurcation detection in coronary arteries is a challenging task. Wolterink et al. (2019)'s CNN tracker extracts the coronary arteries in two directions without taking bifurcations into account.

We propose a novel 3D CNN-based algorithm that

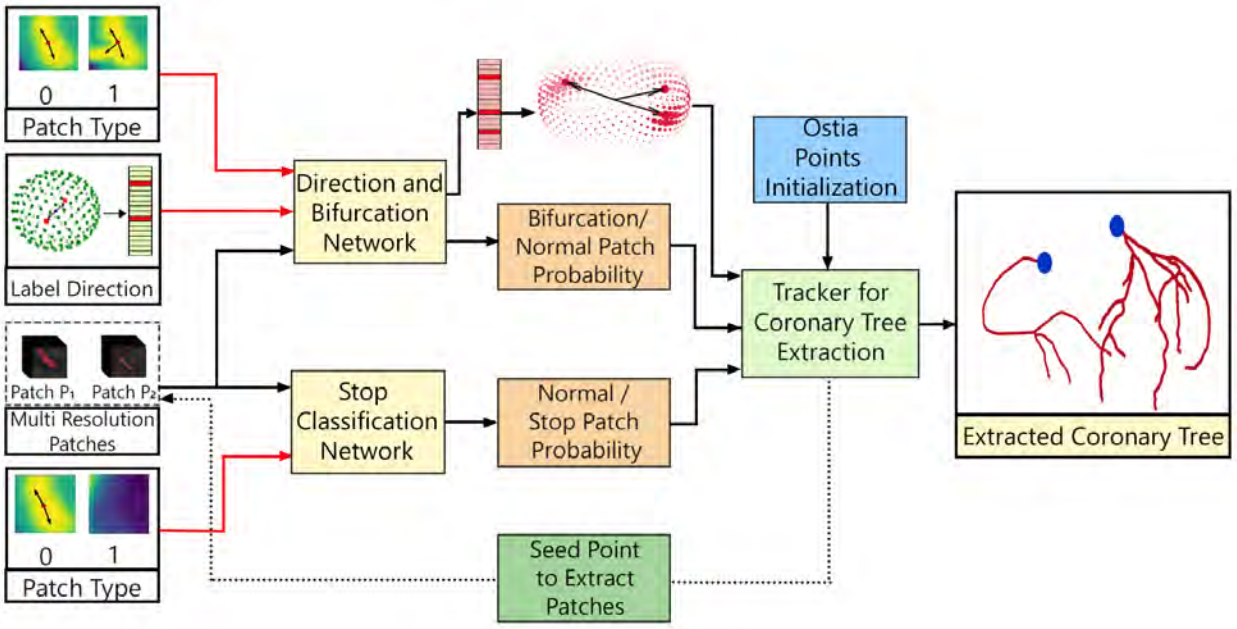


Figure 2: Overview of the proposed method. The red lines represent the inputs given to the direction and bifurcation classification model (DBC-Net) and the stop patch classification model (STC-Net) during the training phase along with the multi resolution patches P_1 and P_2.

is able to extract the entire coronary tree automatically. This approach does not require any pre-processing step or handcrafted filters. No user interaction is required to obtain the entire coronary tree. In contrast to the CNN approach by Wolterink et al. (2019), bifurcations are also detected by the CNN and consequently the resulting directions are predicted by the CNN which make it possible for the entire coronary tree to be extracted by a seed point placed automatically anywhere on the coronary tree instead of requiring one seed point per vessel. The termination of the tracking in our proposed method is guided by another 3D CNN which prevents premature termination. Figure 2 shows overview of the entire pipeline of the proposed algorithm.

3. Dataset

3.1. Dataset

3.1.1. Philips Dataset

The Philips dataset consists of 43 images acquired from 9 different clinical sites which were annotated by clinical experts. Philips dataset contains images from 64-slice Philips Brilliance CT, 256-slice Philips Brilliance iCT, Philips Ingenuity CT, Philips IQon Spectral CT scanners and a few images from Siemens SOMATOM Force CT scanners. The CCTA images in the dataset have a resolution ranging from $0.25 \times 0.25 \times 0.33\ mm^3$ to $0.48 \times 0.48 \times 0.80\ mm^3$ with a mean resolution of $0.40 \times 0.40 \times 0.43\ mm^3$. There is considerable variability in the coronary arteries labelled for each case. The number of annotated coronary arteries per CCTA scan varies from 4 to 20. The mean number of annotated coronary arteries per CCTA scan in this dataset is 9. Depending on the number of annotated coronary arteries, the number of centerline points per case varies from 933 to 3200 with a mean of 1737. The Philips dataset in total contains 428 annotated coronary arteries. Four-fold cross validation has been performed in order to evaluate the proposed algorithm.

3.1.2. CAT08 Dataset

The MICCAI 2008 Coronary Artery Centerline Extraction Challenge (CAT08) dataset consists of 32 publically available CCTA images comprising of 8 training and 24 test CCTA images.[1] The centerline annotations for test dataset are not available and the extracted centerlines can be evaluated only once on the evaluation framework. CAT08 dataset contains images from 64-slice CT Siemens Scanner and dual source CT Siemens Scanner reconstructed to a resolution of $0.32 \times 0.32 \times 0.4\ mm^3$. Both the training and test set images were utilized as a test set for evaluating the performance of our algorithm on different scanners. Each image contains annotations for four arteries. The three fixed arteries present in all the CAT08 CCTA images include Left Anterior Descending Artery (LAD), Left Circumflex Artery (LCX) and Right Coronary Artery (RCA). However, the fourth artery in each case has arbitrarily been chosen. Since, fully automatic algorithms extract the entire coronary tree, there is a need to do a vessel by vessel evaluation (Schaap et al., 2009). Hence, the CAT08 challenge provides with points in the distal end

[1] `http://coronary.bigr.nl/centerlines/about.php/`

of the arteries that can be used to select the artery and evaluate metrics. If the entire coronary artery centerline extraction has not been successful, another point is also provided in the proximal end of the artery. Only one of these points may be utilized to select the artery.

4. Method

The training dataset for the direction classification and bifrucation detection model (**DBC-Net**) and stop patch classification model (**STC-Net**) consists of 3D isotropic patches P_1 of resolution $0.5\times0.5\times0.5\ mm^3$ and P_2 of resolution $1\times1\times1\ mm^3$. These training patches are centered at a location $\mathbf{x}$ in a CCTA image in the vicinity of an annotated centerline point. The radius of the annotated coronary arteries ranges from 0.179 mm to nearly 3.55 mm near the ostium in the Philips dataset. As a rule of thumb, the size of the patches should be small enough to not lose the local context in the smaller arteries but it should be also be big enough to be able to determine the direction in the broader portion of the arteries. This implies the approximate minimum patch size to be $\frac{3.55\times2}{0.5}$ = 14. In order to cover artery sections with a large diameter, we choose the patch size of 19. It is small enough to allow fast forward and backward pass as well as sufficiently large to encapsulate the whole context of the coronary artery information.

The direction vectors from the center of the patch c_p to the adjacent centerline points need to be determined to guide the tracking algorithm. The label direction vectors are obtained by placing a sphere of radius R at the center of the patch as shown by Figure 3. The annotated centerline points within the sphere are designated as positive and those outside the sphere as negative. We determine the exit points of the arteries contained in the sphere by observing the sign changes associated with each artery. If there is a bifurcation, there will be three

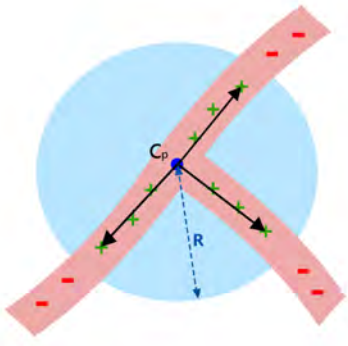


Figure 3: 2D projection of the 3D sphere of radius R placed at center of the patch c_p for getting label direction vectors. The annotated centerline points inside the sphere are indicated by + sign and the ones outside are indicated by a - sign. The label direction vectors are obtained by detecting the sign changes.

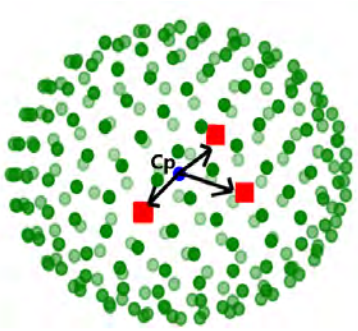


Figure 4: The green dots on the sphere $\boldsymbol{S_d}$ correspond to the $\boldsymbol{N_d}$ admissible movement directions. The center of the patch $\boldsymbol{c_p}$ is denoted by a blue dot and the red squares indicate the closest points on the sphere grid associated with the direction vectors which are assigned the value 1. The remaining grid points are assigned the value 0.

exits (sign changes) from the sphere and there will be only two exits in the normal case. The sphere radius should be large enough in order to detect three sign changes associated with the occurrence of the bifurcation. However, if R is made too large, the bifurcations will be detected well before the patch center c_p is at the bifurcation point. This would allow tracked centerlines to branch prematurely before only to be joined later. The radius R was fixed to 1.5 mm.

The direction vectors obtained are then discretized on a unit sphere S_d placed at the center of the patch. Approximate equidistant discrete grid on this sphere is obtained using Spherical Fibonacci Mapping (Keinert et al., 2015). Each grid point corresponds to an admissible movement direction. Given a set of direction vectors related to a specific center point, the point on the discrete sphere grid which makes the minimum angle with the corresponding direction vector is assigned the value 1. All other points on the sphere grid which do not have any direction vector associated with them are assigned the value 0. Figure 4 shows how the label direction vectors are associated with discrete locations on the sphere. Finally, the vector encoding the movement directions is normalized to unit length. The problem of determining the direction vectors is then simply reduced to the classification of discrete locations on the unit sphere. The number of discrete locations N_d on the unit sphere S_d is fixed to 1000.

4.1. Augmentation

We use various augmentation strategies during training in order to improve the overall robustness of our tracker. Firstly, augmentation by randomly generated translations was introduced to teach the tracker how to recover from centerline deviations. Translation augmentation is introduced by adding a small deviation Δ_t to the center of the patch c_p and extracting the patch at this new translated center $C_t = c_p + \Delta_t$. This deviation should not be so large that the artery is no longer

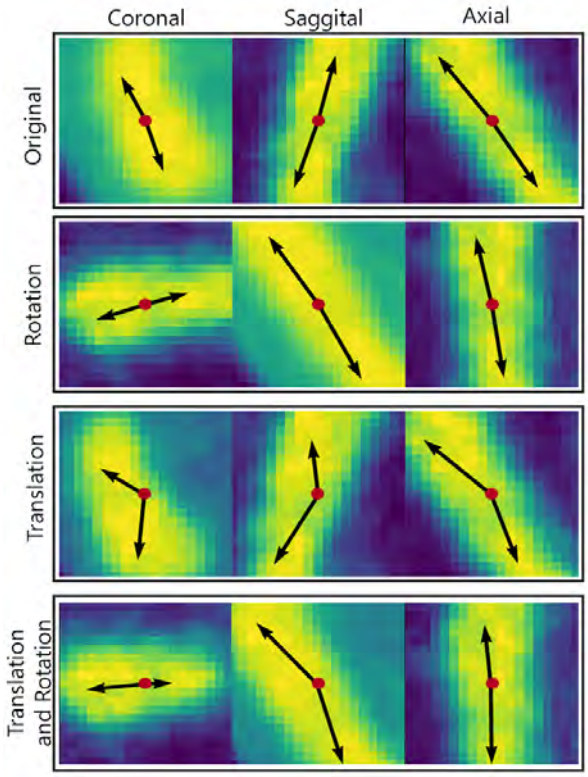


Figure 5: Maximum intensity projections (MIP) in coronal, axial and saggital views of the 3D patch in order to visualize augmentation. The direction vectors are shown after applying 3D rotation and 3D translation augmentation.

within the field of view. It should be ensured that atleast half of the artery is visible in the patch. Hence, the translation augmentation is applied with respect the radius of the artery at the annotated centerline point. The amount of applied deviation is $\Delta_t = \lambda_t \times radius$ where λ_t is uniformly sampled from the interval [0,1]. A small translation may result in a drastic change in direction vectors. The direction vectors are highly sensitive to the center of the patch c_p. Consequently, the label direction vectors are determined with respect to a pseudo center C_{pseudo} which is closer to the original center in order to dampen the effect of the translation on the direction.

$$C_{pseudo} = 0.8 \cdot c_p + 0.2 \cdot \Delta_t \quad (1)$$

Patches P_1 and P_2 are given as input to the DBC-Net as shown in Figure 7. These patches are extracted at the translated center C_t and the label direction vectors are determined with respect to the pseudo center C_{pseudo}.

Rotational augmentation is also introduced by rotating the 3D patches around the center randomly around the three axis (ϕ_x, ϕ_y, ϕ_z). The label direction vectors are obtained before applying the rotational augmentation. The rotation of the patches is incorporated in the corresponding labels by applying the same rotation to the label direction vectors. The rotated direction vectors are then associated with discrete directions on the sphere S_d. Figure 5 shows how the label direction vectors are transformed after applying augmentation.

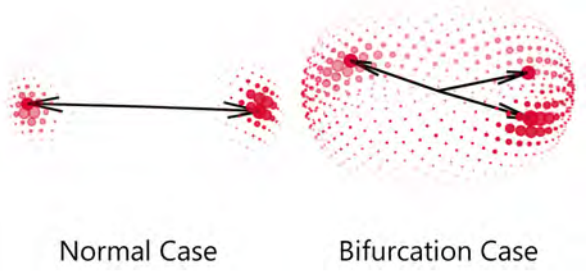


Figure 6: The output response obtained from the direction layer L_D of DBC-Net. Two or three peaks are observed in the response according to the patch type. The resulting direction vectors obtained are also shown in black.

4.2. Ostia Points for Algorithm Initialization

In order to initialize the fully automatic centerline extraction of the coronary arteries, two seed points corresponding to the left and right coronary trees need to be obtained automatically. The algorithm can be initialized by selecting a point which may be located anywhere on the coronary artery tree. An algorithm based on Model based Segmentation (MBS) was used to determine the left and right ostium origin points from the aorta. These points were used for initialization of the tracking algorithm. The spatial location of the ostia landmarks is derived from the mesh topology (Ecabert et al., 2008).

4.3. Bifurcation Prediction

The entire left and right coronary tree can be traced by using one seed point each placed anywhere if all of the bifurcations are correctly detected by an algorithm. The accurate classification of patch type as bifurcation or normal is essential to the tracking of entire coronary tree based on a single seed point. Depending on this prediction, the number of direction vectors obtained will be two or three respectively. Hence, the subsequent network will also predict the bifurcation type in order to facilitate the tracking procedure.

In our training set, uniformly sampling center points from the coronary trees resulted in a rare occurrence of patches containing bifurcations. We utilized the strategy of **Importance Sampling** in order to assure that 20% of the patches in a mini-batch include bifurcation.

4.4. Direction and Bifurcation Classification Network (DBC-Net)

We propose a combined approach for classifying directions to the neighboring centerline points from patch center c_p on a unit sphere S_d having N_d discrete directions, as well as patch type classification P_c (Normal or Bifurcation). The employed CNN network consists

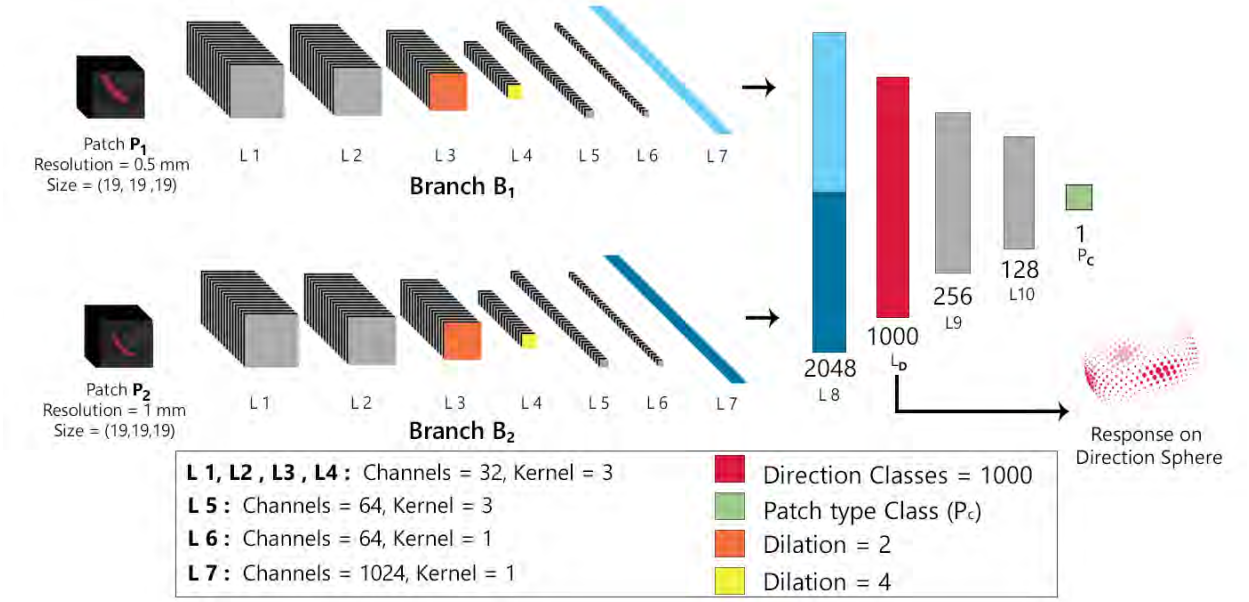


Figure 7: The proposed dual pathway multi-resolution architecture proposed for simultaneous direction and patch type classification. Patches P_1 and P_2 of size $19 \times 19 \times 19$ with resolution $0.5 \times 0.5 \times 0.5$ mm^3 and $1 \times 1 \times 1$ mm^3 respectively are fed to the network to get direction class predictions on the direction sphere S_d. The direction layer L_D is then followed by a series of linear layers to get a single patch type prediction P_C (normal or bifurcation).

of a Deep Medic-based architecture (Kamnitsas et al., 2016). Figure 7 shows that the proposed architecture has two branches B_1 and B_2 that take 3D patches P_1 and P_2 with resolutions $0.5 \times 0.5 \times 0.5$ mm^3 and $1 \times 1 \times 1$ mm^3 as input respectively. Each branch consists of 7 3D convolutional layers with kernel size of 3. Layers L_3 and L_4 use dilated convolutions with the spacing of 2 and 4 between the kernel points respectively. After each convolutional layer, 3D batch normalization was employed. At the end of these 7 layers, the output of the branches B_1 and B_2 is concatenated in Layer L_8. L_8 is then reduced to the number of direction classes N_D to form the direction class layer L_D. The Layer L_D is then subjected to two linear layers L_9 and L_{10} to get patch type classification P_C.

ReLU activation function is used after all the layers except the layers L_D and P_c. Patch type class layer P_c uses sigmoid activation function and the direction class layer L_D uses softmax activation function. **Binary cross entropy loss** (BCE_{patch}) is used for the patch type classification and **categorical cross entropy loss** ($CE_{direction}$) is used for the direction classification. The combined loss function used to train the network is as follows:

$$Total\ Loss = CE_{direction} + \lambda_b \times BCE_{patch} \qquad (2)$$

λ_b is fixed at 5. The other hyper-parameters tuned for this set up include learning rate of 0.0001 with Adam optimizer and mini-batch size of 64.

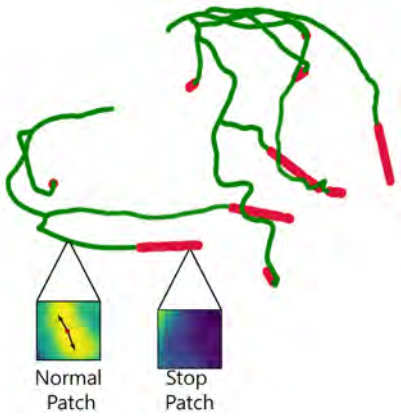


Figure 8: The patch labels used for training the stop patch classification model. The "Normal" class patches correspond to the green portion in the figure. The "Stop" class patches correspond to red portion beyond the last annotated centerline point.

4.5. Stop Patch Classification Network (STC-Net)

In order to terminate the tracking algorithm once the end point of a coronary artery has been reached, we propose to train a separate 3D CNN. The same architecture as shown in Figure 7 is used to train the stop patch classification model (**STC-Net**). The endpoints of coronary arteries can be quite ambiguous. In order to get patches corresponding to the stopping criteria, we sample points beyond the end of the coronary arteries. This is achieved by using the direction vector obtained by subtracting the penultimate centerline endpoint from centerline endpoint. Points beyond the endpoint up to

```
stop_counter_max = 3  # stopping criteria constant for stopping network
dir_entropy_max = 0.8  # stopping criteria constant for direction entropy
stop_prob_max = 0.3  # stop probability threshold for classifying endpoint patches
# active: (3d point, segment_id, parent_id, stop counter, 3d previous dir vector)
active,centerline = empty_queue,empty_list  # centerline: (3d point, segment_id, parent_id)
next_segment_id = 0 # next possible segment index

for all ostia: # iterate over inlets
    active.add(next_point_id++, next_segment_id++,None,0,None)  # no parent - seed points

while len(active)>0:
    point,segment_id,parent_id,stop_counter,prev_dir = active.pop()
    if stop_counter > stop_counter_max: continue # termination criterion
    centerline.append(point,segment_id,parent_id)  # adding to the set of centerline pts
    patch = sample(point) # patch sampler

    # get direction, patch_type (bifurcation or normal)  predictions from DBC-Net
    dir_response,patch_type_bifur,dir_entropy = DBC_Net(patch)
    # get direction vectors depending on patch_type and dir_response
    dir_vect = get_direction_vectors(patch_type_bifur, dir_response)
    candidates = (dir_vect * step size ) + point # get candidates from direction vectors
    # get patch_type (endpoint or not) predictions from STC-Net
    stop_prob = STC_Net(patch)

    if stop_prob > stop_prob_max or dir_entropy > dir_entropy_max: stop_counter++
    else: stop_counter = 0 # reset

    # nearest neighbor distance check if each candidate point is an active point
    candidates = [c for c in candidates if distance(c,centerline) > step_size/2]
    dir_vector = (candidates - point)/step_size   # get direction vectors back
    if len(candidates) == 1:   # continue segment
        active.add(candidates, segment_id, point_id, stop_counter, dir_vect)
    else:
        for ind, cand in enumerate(candidates): # start segments from bifurcation at cand
            active.add(cand, next_seg_id++, None, stop_counter, dir_vect[ind])
```

Listing 1: Pseudo code of the tracking algorithm. *get_direction_vectors* function returns the direction vectors to the neighboring centerline points taking the direction response and bifurcation prediction from the DBC-Net as input. The *centerline* list contains the tracked coronary tree at the termination of this algorithm.

5 mm are sampled and labelled as stop patch type. All the other centerline points are labelled as normal patch type. Only binary cross entropy loss for stop patch type classification is employed for training the network. The overall stopping criteria is based on the predictions by the STC-Net and the entropy of the direction prediction response by the DBC-Net. Wolterink et al. (2019)'s stopping criterion is solely based on moving average entropy which results in premature termination as well as leakage in some of the cases. Our combined stopping heuristic tries to solve the issue of premature termination in the presence of plaque and stenosis.

4.6. Tracking Implementation

The tracking starts by obtaining the seed points, one for each coronary tree, from the ostium initialization module. These seed points are added to an active queue. We continue the tracking until there are no points in the active queue. We obtain two patches P_1 and P_2 of resolution $0.5 \times 0.5 \times 0.5$ mm^3 and $1 \times 1 \times 1$ mm^3 respectively centered at the point popped from the queue. These patches are fed to the DBC-Net and STC-Net. The STC-Net outputs the probability of the patch being a stop patch or normal patch.

The DBC-Net determines the direction predictions on the unit sphere S_d as well as classifies if the given patch contains a bifurcation or not. The DBC-Net learns to predict some relatively high values near the correct direction class as, for example, shown by Figure 6. We observed that high probabilities were assigned to the neighbors of the correct direction class as well. Smoothing with a gaussian kernel of size 16 was applied in order to get rid of the noise. Once the predictions are smoothed out, we detect two or three peaks depending on the patch type classification.

Depending on the direction response prediction, the direction D_1 is obtained by taking into account the previously tracked centerline point. If this is the first point being extracted, we take the maximum of the direction response as D_1. If a centerline point has been previously extracted, we take into account the previous direction D_{prev} used to obtain this patch. The angle between D_1 and D_{prev} should be less than 60° in order to make sure that the tracker always proceeds forward. The maximum response obtained in this constrained field of view is labelled as D_1. The second direction D_2 should be at

least 110° farther from D_1 in order to make sure that the opposite direction is correctly tracked. The third direction D_3 should be at least 40° farther from the first and second responses. In case the patch type is normal, only D_1 and D_2 are determined. The candidate points S_{cand_i} are obtained from the direction vectors D_i as follows:

$$S_{cand_i} = S_{point} + \Delta_{step} \times D_i \tag{3}$$

i vares from 1 to 2 in normal case and 1 to 3 in case a bifurcation has been detected. S_{point} represents the current patch center and Δ_{step} is the step size. The distance between all the candidate points and already finalized centerline points is determined. Candidate points with a distance > $(\Delta_{step})/2$ are added to the active queue.

A combined criterion based on the predictions of the STC-Net and entropy determined from the direction prediction response of the DBC-Net is used to terminate the tracking. If the entropy exceeds a threshold of 0.8 or the stop patch probability goes above 0.5, the stop counter is updated by one. If the none of these two conditions are satisfied, counter is reset. If the stop counter exceeds 3, the active point is not included into the list of the tracked list of centerline points and tracking terminates.

Information related to the previous direction and stop counter is kept in the active queue along with the 3D point coordinates. It is also important to keep track of the separate segments and their parents in the queue. The segment terminates at each bifurcation point. Listing 1 shows the simplified pseudo code for the tracker implementation. Figure 9 shows how individual vessels can be obtained making use of the segment information stored during tracking.

5. Evaluation Measures

A prerequisite to the evaluation of all the metrics is the conformity in the spacing between the tracked centerline points and the ground truth centerline points. The ground truth annotations of the coronary arteries

Figure 9: Different vessels in the coronary tree obtained from the tracked result. Each color represents a different coronary artery.

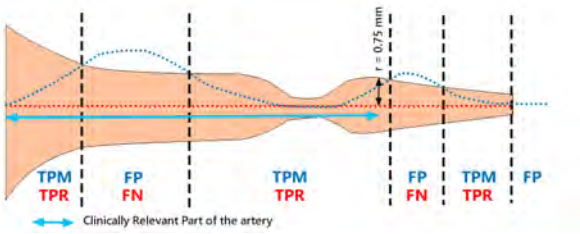


Figure 10: True Positive Reference (TPR), True Positive Measured (TPM), False Positive (FP) and False Negative (FN) associated with different parts of the tracked (blue dotted line) and reference centerline (red dotted line). Clinically relevant part of the vessel is also shown in the figure. More details about metric calculation can be found in Schaap et al. (2009).

and the tracked arteries are resampled uniformly to obtain same spacing between consecutive points (Schaap et al., 2009).

A point on the ground truth centerline is labelled as True Positive Reference **(TPR)** if a tracked centerline point is present within the corresponding annotated radius and it is labelled as False Negative **(FN)** otherwise. A point on the tracked centerline is labelled as True Positive Measured **(TPM)** if a ground truth centerline point is present within the corresponding annotated radius and it is labelled as False Positive **(FP)** otherwise. Figure 10 shows how TPR, TPM, FP and FN are obtained in terms of tracked and reference centerlines.

The points towards the distal end of the coronary arteries may be ambiguous and not clinically relevant.The endpoint of the clinically relevant part of each artery is defined as the most distal point of the vessel with an radius greater than 0.75 mm.

Sensitivity determines how much of the ground truth coronary tree has been correctly tracked by the algorithm. A sensitivity value of 1 indicates that the entire coronary tree has been covered by the centerline extraction algorithm.

$$\text{Sensitivity} = \frac{TPR}{TPR + FN} \tag{4}$$

The number of annotated coronary arteries varies from 4 to 20 in the CCTA images in the Philips dataset. In an effort to obtain the entire coronary tree, the algorithm will also track the arteries that have not been annotated. However, a check should be maintained to see that the algorithm doesn't detect many spurious vessels. Hence, the deviation from the coronary reference tree is kept in check in terms of **False Positive Rate** (FPR). For calculating the Sensitivity and False Positive Rate, we set the threshold radius to 1 mm.

$$\text{FPR} = \frac{FP}{TPM + FP} \tag{5}$$

Overlap measure as defined in equation 6 similar to dice in segmentation. **Average Overlap** (OV) takes the

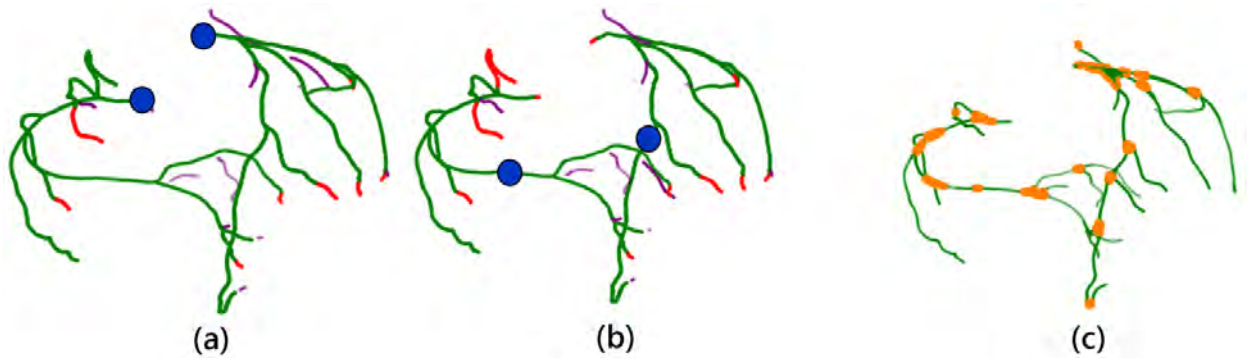


Figure 11: Qualitative Representation of the Tracked Result. (a) shows the coronary artery centerline extraction result when the algorithm is initialized by placing the seed points at the ostium. (b) shows the coronary artery centerline extraction result when the algorithm is initialized by placing the seed points in the middle of LAD and RCA. The green part of the extracted coronary tree indicates the portion tracked within the radius threshold of the annotated centerline points. The red part corresponds to the part missed by the tracking algorithm. The purple part corresponds to extra tracked arteries that are not present in the ground truth annotation. In (c), the bifurcation detection in orange is shown overlayed on the tracked coronary artery tree.

entire reference and extracted coronary artery into consideration. **Clinically Relevant Overlap** (OT) calculates the overlap only for the clinically relevant part of the artery. **Overlap until First Error** (OF) calculates the portion of the overlap accurately tracked until the first error occurs (Schaap et al., 2009).

$$\text{Overlap} = \frac{TPR + TPM}{TPR + TPM + FN + FP} \tag{6}$$

The deviation of the extracted points from the reference centerline points is determined only for regions of the reference tree which are labelled as True Positive Reference. The average of the Euclidean distance between the reference centerline points and the nearest tracked centerline point determines the **Accuracy Inside** (AI).

6. Results

Fully automatic coronary centerline extraction methods extract the entire coronary tree without requiring any manually placed seed point to be provided for the vessels. The quantitative analysis of the extracted centerlines is performed individually for each coronary artery. The MICCAI CAT08 dataset provides for each case a point in the distal end of the coronary artery for selection. In case the coronary artery centerline is not present in the distal end, another point is provided in the proximal part of the artery which can be utilized for coronary artery selection.[2] The evaluation guidelines of the challenge only allow the usage of one of these points. In order to keep the evaluation consistent, the quantitative analysis in Philips dataset is also performed by utilizing a point in the distal or proximal end of the coronary artery for selection.

[2] `http://coronary.bigr.nl/centerlines/about.php/rules.php`

6.1. Philips Dataset

The Philips dataset comprising 43 CCTA scans was used to train the DBC-Net for simultaneous direction classification and bifurcation detection as well as the model for the detection of stop patches. The dataset was randomly shuffled and 33 CCTA images were used for training. The remaining 10 CCTA images were used for validation. Four-fold cross validation was performed for the final model. The seed point for the initialization of the tracker in order to obtain the centerlines for left or right coronary tree can be given anywhere on the coronary tree. However, the seed point for all the experiments was given near the ostium as this point can be obtained automatically from the model based segmentation.

No. of Resolutions	S	OV	OT	AI
1	82.3	76.4	85.9	0.37
2	**88.9**	**81.2**	**87.4**	**0.32**
3	87.4	78.6	86.5	0.34

Table 1: The effect of varying number of resolutions levels in the DBC-Net in terms of total sensitivity (S, in %), overlap (OV, in %), clinically relevant overlap (OT, in %) and accuracy inside (AI, in mm).

Table 1 shows the overlap metrics and accuracy inside on the validation set when the number of input resolution levels and consequently the pathways in the architecture are varied. In case of a single pathway, only 1 resolution of 0.5 x 0.5 x 0.5 mm^3 is used. In case of a dual pathway, 2 resolutions of 0.5 x 0.5 x 0.5 mm^3 and 1 x 1 x 1 mm^3 are utilized. In case of three pathways, 3 resolutions of 0.5 x 0.5 x 0.5 mm^3, 1 x 1 x 1 mm^3 and 1.5 x 1.5 x 1.5 mm^3 are employed. The dual pathway architecture with only two resolutions performs better as indicated by overlap and accuracy metrics.

Figure 11 (a) shows that the result of automatic coronary centerline extraction when the seed points for

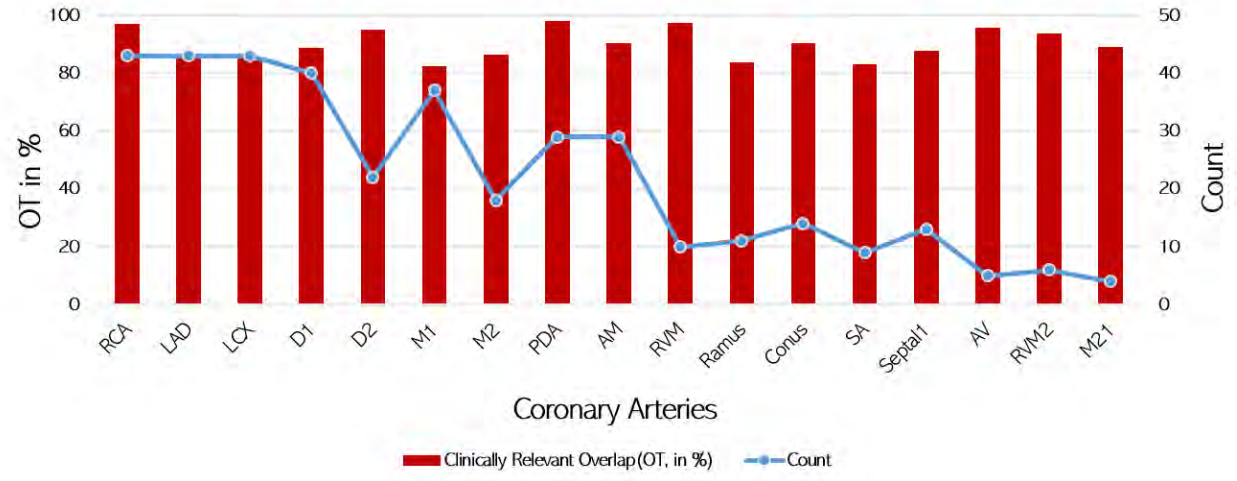


Figure 12: Clinically relevant overlap of all the arteries present the Philips dataset which occur more than 3 times. The number of occurrences of each artery is also shown in the plot. The horizontal axis shows the names of different coronary arteries, the left vertical axis shows their corresponding mean clinically relevant overlap obtained and the right vertical axis shows the number of times each artery is encountered in Philips dataset.

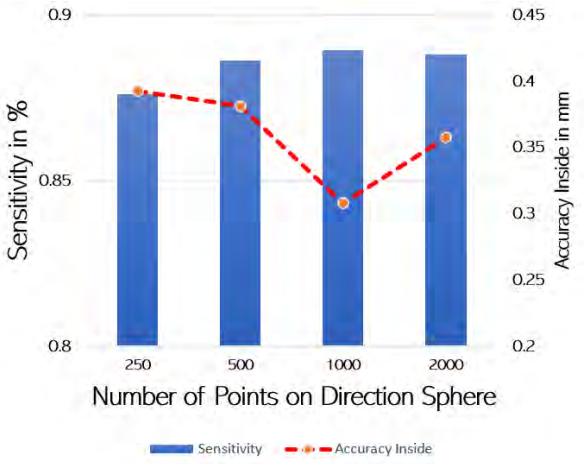


Figure 13: The effect of number of discrete direction points N_d on the unit sphere S_d for direction classification in terms of sensitivity and accuracy inside.

tracker initialization are placed at the left and right coronary ostium. 11 (b) shows that when the seed points are placed in the middle of LAD and RCA coronary arteries. The extracted coronary tree is almost similar in both the cases. The sensitivity obtained when the seed points are placed at the coronary ostia for fold 3 is 88.9% and it is 87.3% when the seed points are placed in the middle of LAD and RCA. This shows that the seed point can essentially be placed anywhere on the coronary tree. Figure 11 (c) shows the bifurcation detection overlayed on the tracking result. The orange color on the coronary tree indicates that a bifurcation has been detected at that centerline point by the DBC-Net which means that three direction vectors will be obtained to generate the candidate points.

Figure 13 shows the effect of increasing the number of sphere direction points. As the number of the direction points increase, average accuracy inside and sensitivity drops after a maximum. Based on this observation, we fixed the number of direction points on the sphere at 1000 for further analysis

There is a data imbalance in the patches extracted from the CCTA images. Due to low number of patches with bifurcations, importance sampling is applied in order to ascertain that a percentage of the mini-batch during training contains bifurcation patches. Figure 14 shows the effect of increasing the importance sampling factor while using a fixed mini-batch size of 64. As the importance sampling parameter increases, the sensitivity increases due to better detection of bifurcations. However, the detection of vessels not annotated in the dataset also increases. The importance sampling parameter is fixed at 10 in order to keep the false positive rate

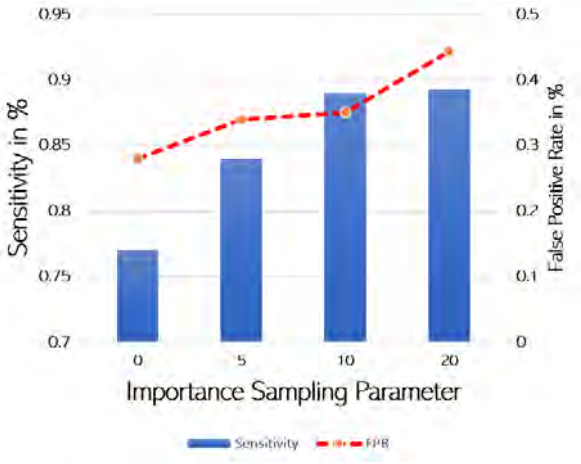


Figure 14: The effect of importance sampling parameter while using a mini-batch of size 64 on sensitivity and false positive rate.

Loss Function	S	OV	OT	AI
Softmax and Categorical Cross Entropy	**88.9**	**81.2**	**87.4**	**0.32**
Sigmoid and Binary Cross Entropy	88.4	78.6	86.7	0.34

Table 2: The comparison between two different types of loss functions for direction classification in terms of senstivity (S, in %), overlap (OV, in %), clinically relevant overlap (OT, in %) and accuracy inside (AI, in mm).

in check so that spurious vessels are not detected.

The final model uses softmax activation function for the direction classification layer (D) in the model and categorical cross entropy loss. Table 2 shows that this choice performs slightly better than sigmoid activation for the direction layer (D) and binary cross entropy loss. For the final hyper-parameter choice, we conducted a four-fold cross validation on randomly generated splits of the Philips dataset. The quantitative measures were averaged across the folds and we obtained an average sensitivity of 87.1%, clinically relevant overlap of 89.1% and overlap of 80.4% was obtained. An average accuracy inside of 0.34 mm was obtained which is within the average voxel size of $0.40 \times 0.40 \times 0.43 mm^3$.

6.2. CAT08 Training Dataset

The training dataset of CAT08 challenge was used as a test set in order to determine the performance of the proposed algorithm. The best model from the cross validation of the Philips dataset was used to extract centerlines for CAT08 training dataset. This model was not re-trained on CAT08 dataset. The tracker was initialized using ostium points derived from the Model Based Segmentation of the heart. The training dataset of CAT08 challenge consists of 8 CCTA images containing 32 annotated vessels. This dataset contains images of varying quality and calcium score.

Table 3 shows that an average overlap of **93.4%**, clinically relevant overlap of **95.9%** and overlap until first error of **76.5%** was obtained for these 8 CCTA scans. All these CCTA scans have an image resolution of $0.32 \times 0.32 \times 0.4 mm^3$.The average accuracy obtained was 0.36 *mm* which is approximately within the dimension of the one voxel. The average time taken to extract the entire coronary tree on a GTX 1080 GPU is 41 s. For all cases, 15 out of 16 vessels were automatically detected. One vessel from case 3 which was missed due to failure in corresponding bifurcation detection required an additional seed point in order to be detected. This is a good test of generalization of the algorithm as the model was trained on CCTA scans from Philips scanners and CCTA scans in CAT08 dataset come from different types of Siemens scanners.

Figure 15 shows the results of automatic coronary centerline extraction for case 4 from CAT08 training set and case 11 from CAT08 test set. The blue dots correspond to the ostia locations obtained from model based segmentation for tracker initialization. The proposed algorithm extracts the entire coronary tree while annotations for only 4 coronary arteries were provided in the ground truth.

No.	Image Quality	Calcium Score	OV	OF	OT	AI	T
0	Moderate	Moderate	94.2	77.7	95.1	0.4	55
1	Moderate	Moderate	97.3	99.4	99.6	0.32	39
2	Good	Low	98.3	99.7	100	0.31	43
3	Poor	Moderate	86.3	63	89.1	0.4	41
4	Moderate	Low	92.9	57.3	97.9	0.33	31
5	Poor	Moderate	97.6	77.5	99.7	0.43	33
6	Good	Low	96.7	87.2	99.6	0.3	36
7	Good	Severe	83.9	49.1	86.3	0.38	48
Avg			**93.4**	**76.5**	**95.9**	**0.36**	**41**

Table 3: Results of our method on CAT08 training set which was used as a test set. For each case, overlap (OV, in %), overlap until first error (OF, in %) and clinically relevant overlap (OT, in %) ,average accuracy inside (AI, in *mm*), time taken for coronary tree extraction (T, in s) along with subjective image quality and calcium score is shown.

6.3. CAT08 Testing Dataset

The CAT08 test dataset comprises of 24 CCTA scans of varying image quality and calcium scores. We tested our algorithm on these 24 CCTA images containing 96 vessels in order to benchmark the performance of our algorithm against methods available on the leaderboard of CAT08 challenge. Both the DBC-Net and the STC-Net were now trained on 43 cases of Philips dataset and 8 cases of CAT08 training dataset. These models were used to extract centerlines of the coronary arteries in the CAT08 test set which were then submitted to the evaluation framework online.

Table 4 shows the performance of the algorithm on the testing set of MICCAI 2008 challenge. An average overlap of **93.6%**, clinically relevant overlap of **96.4%** and overlap until first error of **76.3%** was obtained for these 24 CCTA scans. Cases 8, 10 and 27 required one additional seed point due to failure in the detection of bifurcations for one of the vessels. There are significant

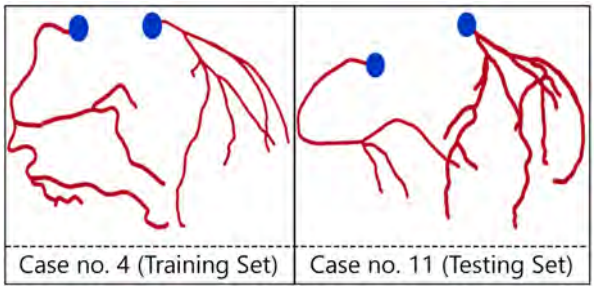


Figure 15: The left and right coronary trees extracted without any user intervention for two cases in training and testing set of CAT08 Challenge. The blue dots correspond to the ostia locations used for tracker initialization obtained automatically.

motion artifacts present case 26 which hamper the bifurcation detection. Hence, additional seed points are provided for 3 of the vessels in this CCTA image. There are 132 vessels present in the training and test set of CAT08 dataset. Overall, 125 vessels were automatically detected without requiring any seed points.

Table 5 shows the comparison of the performance of our algorithm **AuCoTrack** against the current automatic coronary centerline extraction techniques and the state-of-the-art CNN-based technique which requires at-least one point per vessel for the centerline extration. Our proposed method achieves better overlap rank of **9.87** than other automatic techniques of Zheng **10.43**, Kitamura **13.81** and Yang **10.55**. Wolterink's CNN-based approach requires atleast one seed point per vessels. AuCoTrack successfully detected approximately 95% percent of the vessels in CAT08 dataset requiring no user interaction. However, Wolterink's approach gives OV, OF and OT values of 93.7% , 81.5% and 96% respectively. Our method achieves almost the same performance as the Wolterink's CNN approach while reducing the need of interaction to almost zero. Our approach also requires an average time of **42.6 s** using a GTX 1080 GPU to extract the entire coronary tree. Our method is faster than all other automatic approaches but, the comparison is difficult as the computational resources of all the methods are not same.

No.	Image Quality	Calcium Scores	OV	OF	OT	AI	T
8	Poor	Low	84.6	48.7	91.1	0.46	31
9	Good	Low	95.4	70.2	98.5	0.34	37
10	Moderate	Moderate	94.8	91.4	97.3	0.36	36
11	Good	Moderate	91	53.7	91.8	0.39	49
12	Good	Moderate	89.1	29.4	94.2	0.38	55
13	Moderate	Low	97.8	96.4	97.8	0.37	39
14	Moderate	Severe	97.7	67.2	98.7	0.4	57
15	Moderate	Moderate	96.1	100	100	0.34	65
16	Good	Low	97.4	92.4	100	0.4	45
17	Poor	Severe	91.6	68.3	95.8	0.41	33
18	Good	Moderate	96.6	95.9	98.9	0.3	38
19	Moderate	Moderate	95	100	100	0.34	35
20	Moderate	Moderate	90.1	46.2	90.5	0.45	43
21	Good	Low	96.7	97.2	99.3	0.36	36
22	Good	Low	96.8	99.6	99.7	0.35	33
23	Moderate	Moderate	98.3	97.4	99.2	0.36	47
24	Moderate	Severe	93.3	52.2	95.3	0.3	35
25	Good	Moderate	95.1	69.5	98.2	0.39	51
26	Poor	Low	76.7	34.8	86.6	0.5	47
27	Good	Moderate	85	55.3	85.6	0.42	47
28	Good	Low	95.9	94.4	97.4	0.32	38
29	Poor	Moderate	98.4	95.1	99.7	0.32	44
30	Good	Low	95.1	77.5	98.5	0.33	42
31	Good	Moderate	98.8	99.2	100	0.31	40
Avg			**93.6**	**76.3**	**96.4**	**0.37**	**42.6**

Table 4: Results of our method on CAT08 test set. For each test case, overlap (OV, in %), overlap until first error (OF, in %) and clinically relevant overlap (OT, in %) ,average accuracy inside (AI, in *mm*), time taken for coronary tree extraction (T, in s) along with subjective image quality and calcium score is shown.

Method	OV	OF	OT	AI	T
AuCoTrack	93.6	76.3	**96.4**	0.37	**42.6**
Zheng et al	**93.7**	**76.5**	95.6	0.21	60
Kitamura et al	93.5	70.9	92.5	**0.2**	160
Yang et al	90.6	74.2	95.9	0.25	120
Wolterink et al (Interactive)	93.7	81.5	97	0.21	10

Table 5: The comparison of our proposed AuCoTrack algorithm and the top automatic coronary artery centerline extraction techniques in terms of overlap (OV, in %), overlap until first error (OF, in %) and clinically relevant overlap (OT, in %), average accuracy inside (AI, in *mm*) and time taken (T, in s). The interactive CNN-based method by Wolterink et al. (2019) is separated by a dotted line.

7. Discussion

We aimed to provide a deep learning-based automatic approach for centerline extraction in CCTA images. The proposed algorithm was first tested on CCTA scans acquired using Philips scanners from multiple sites. The sweeps for hyper-parameter tuning were performed on Philips dataset using 33 CCTA scans for training and 10 CCTA scans for validation. The method was then evaluated using four-fold cross validation on these CCTA scans. A high average clinically relevant overlap of 89.1% and average sensitivity of 87.1% was obtained. The average accuracy inside for the Philips dataset was reported to 0.34 mm which is less than the average voxel dimensions.

The generalization of this approach was then evaluated by testing the proposed algorithm on the CAT08 training dataset. The model from four-fold cross validation on Philips dataset was used to extract centerlines for CAT08 training dataset. The images from CAT08 training dataset contained considerable variability in terms of calcium scores and image quality. An average overlap of 93.4% and clinically relevant overlap of 95.9% was obtained on evaluating CAT08 training dataset as a test set.

In order to benchmark the performance, we also tested the algorithm by submitting the extracted centerlines of 96 vessels on the evaluation framework of CAT08 challenge. In order to extract centerlines for CAT08 test set, the method was trained on all 43 CCTA scans from Philips dataset and 8 CCTA scans from CAT08 training set. The bifurcation detection failed in 7 out of these 132 vessels. A seed point was required in these cases in order to retrieve these coronary arteries.

The proposed algorithm achieves better overlap rank than the previously available fully automatic coronary artery centerline extraction algorithms. An overlap rank of 9.87 was achieved by AuCoTracker while the top three automatic algorithms on the CAT08 leaderboard by Zheng et al. (2013), Kitamura et al. (2012) and Yang et al. (2011) had an overlap rank of 10.43, 13.81 and 10.55 respectively. The accuracy inside for the CAT08 testing data set was 0.37 mm. The state-of-the-art automatic centerline extraction algorithm by Zheng et al. (2013) utilizes segmentation masks for their

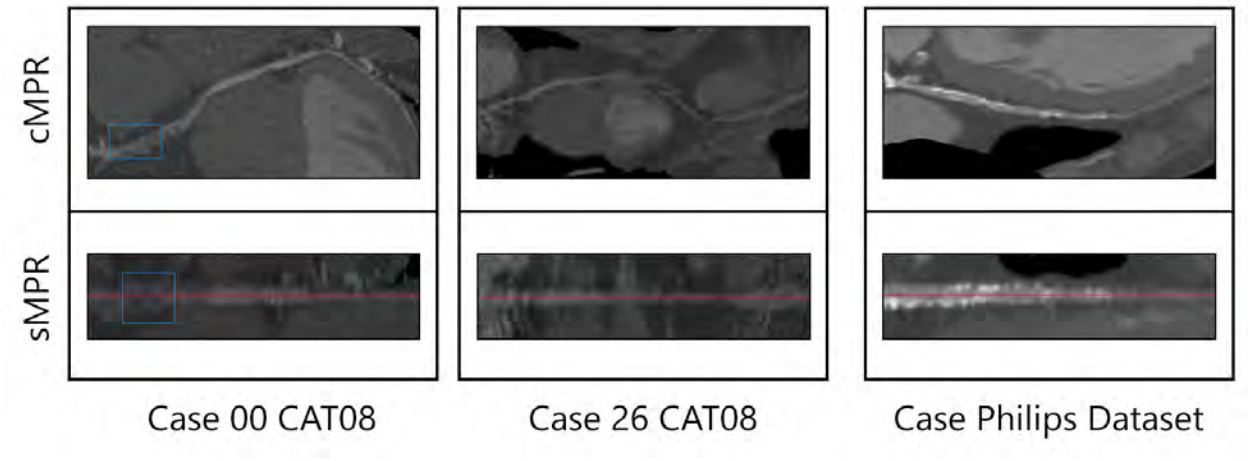


Figure 16: Stretched multiplanar reformation (sMPR) and curved multiplanar reformation (cMPR) images constructed from extracted centerlines using our proposed method **"AuCoTrack"**. The blue box in case 0 from training set of CAT08 challenge shows the presence of severe stenosis. Case 26 from testing set of CAT08 challenge has severe motion artifacts. MPR and cMPR images of a case from Philips dataset shows the presence of coronary calcification. The red lines shows the extracted centerline overlayed on the sMPR image.

model driven and data driven approach. Their algorithm uses 108 CCTA scans from their properiety dataset. Our proposed algorithm achieves state-of-the-art overlap metrics on training set of CAT08 challenge when trained with only 33 CCTA scans from a distinct in-house dataset. Hence, this method can be trained efficiently on low number of CCTA scans.

Table 4 shows the evaluation metrics for the testing set of CAT08 challenge with variable image quality and calcium score. An average overlap of 87.8%, 95.4% and 94.4% was achieved for CCTA scans with poor, moderate and good image quality respectively. Poor image quality is defined by the presence of image degrading artifacts and evaluation is only possible with low confidence (Schaap et al., 2009). Our algorithm's performance is effected by poor image quality but the performance is consistent over moderate and good image quality. CCTA images with calcium scores of low, moderate and severe had an average overlap of 92.9%, 94.0% and 94.2% respectively. This shows the performance of the algorithm is not strongly effected by the presence of coronary calcification.

The proposed algorithm aims at extracting the entire coronary tree from a single seed point. The comparative low metrics for the CCTA scans in Philips dataset as compared to CAT08 dataset can also be attributed to the fact that average number of annotated arteries in the Philips dataset is 9 as compared to 4 in the CAT08 dataset. High clinically relevant overlap in the cases with large number of annotated arteries show that our algorithm is capable of extracting the entire coronary tree. Some of the arteries that may be missed can retrieved by a single seed point.

Wolterink et al. (2019) achieved near state-of-the-art performance as an interactive method for the CAT08 dataset. This method was based on a CNN classifier which simultaneously predicts direction to the centerlines and radius. The main constraint of this method is that it requires one or more seed points per vessel. Our method removes the requirement of seed points per vessel. A seed point is required only when the bifurcation detection fails for the corresponding artery. Successful vessel detection was observed in 95 percent of the cases in CAT08 dataset. The ostia points required for tracker initialization were automatically obtained from an MBS model. Wolterink et al. (2019)'s tracker termination is guided by a moving average entropy criteria which fails in case of a severe stenosis. This is the reason that in some of the cases more than one point is required per vessel in order to warm-start the tracking process. Our method utilizes a model trained on patches beyond the end point in order to determine if the end of a coronary artery has been reached. We employ a voting mechanism of stop patch classification as well as moving average entropy in order to terminate the tracking.

The number of annotated centerlines in Philips dataset varies from 4 to 20. This attributes to severe label noise for bifurcation classification because the underrepresented bifurcation patches may be labelled as normal patches in cases where the number of annotated centerlines are low. This problem may be mitigated by labelling the missing bifurcation points on the annotated arteries. Alternatively, active learning or label noise suppression strategies can also be explored for the solution (Karimi et al., 2020; Wang and Smedby, 2008).

The extraction takes on average 42.1 s for the entire coronary tree in CAT08 dataset. The time complexity of the tracking algorithm in Listing 1 in worst case is softly bounded by $O(n^2)$ where n represents the total number of tracked points. Hence, the total time taken depends on the size of the extracted coronary tree. This time can be reduced by many folds by optimizing the tracker. The

different sub-trees can be processed in a parallel fashion making use of the bifurcation predication. Within the same sub-tree, different threads can access the main active queue and process the data in a parallel fashion keeping track of the visited points. The tracking result on the CCTA image can be displayed in real time as tracking is being performed.

Figure 16 shows stretched multiplanar reformation (sMPR) and curved multiplanar reformation (cMPR) images reconstructed using centerlines obtained from **"AuCoTrack"** algorithm. These reformatted images can be directly used for the diagnosis of coronary artery disease. The proposed algorithm makes use of the local intensity information in the patches in order detect the direction to the centerlines and the bifurcation. We have shown that the model trained on CCTA images from Philips dataset works well for the images from old Siemens scanners for CAT08 dataset.

8. Conclusions

We proposed a deep learning-based automatic coronary artery centerline extraction algorithm which consists of three major modules. The first module comprises of a novel multi-resolution CNN which simultaneously determines the direction to the coronary artery centerlines and detection of bifurcation in the patch. The second module consists of a similar architecture for the classification of end points of the coronary arteries based on patch extraction near the end points. The third module consists of a tracking algorithm that utilizes the information from the first two modules to obtain the entire coronary tree efficiently. This is the first automatic deep learning-based approach for centerline extraction based on a single seed point per coronary tree. Utilizing a model based segmentation module, we are able to automatically detect suitable ostium landmarks. Hence, the proposed overall pipeline requires zero user interaction. Previous CNN approach by (Wolterink et al., 2019) requires one or more seed points per vessel.

The proposed algorithm was first validated on dataset from Philips scanners which contained considerable variability in terms of annotation. The algorithm demonstrated high accuracy and speed. The algorithm was then benchmarked against previous automatic centerline extraction algorithms on CAT08 dataset. The proposed algorithm achieves better overlap rank as compared to previous state-of-the-art automatic centerline extraction techniques. Total overlap, clinically relevant overlap and overlap until first error metrics are approximately similar to the previous CNN approach requiring multiple seed points per vessel. The vessels that are missed by the proposed algorithm can also be retrieved by specifying a single seed point.

The high speed of coronary artery centerline extraction combined with high overlap performance make it suitable for deployment in real time applications. The generalization of the algorithm is demonstrated by the fact that it was trained on recent CCTA images from Philips scanners and tested on CAT08 dataset with considerable variability. Since the algorithm is based on local intensity of the patches, the same proposed pipeline/model can be used to obtain centerlines in other applications e.g. rib centerline extraction. A novel architecture was proposed employing multi-resolution patches with patch-type regularization. The proposed network can be trained to perform automatic tracking in many computer vision applications.

9. Future Work

We proposed a fully automatic deep learning-based centerline extraction algorithm. In future work, a re-centering of the extracted centerline points based on local intensity values could be performed in order to improve the accuracy inside. A rough segmentation algorithm may be used to obtain seed points for centerline extraction in the regions not already covered by automatic extraction performed by this method. This will aid in detecting the missed coronary artery centerlines due to failure in bifurcation detection. Further experiments can be performed in order to try the model trained on these CCTA scans to extract centerlines in 3D tubular structures such as lung, bronchia and other blood vessels. In any case, the proposed pipeline can be re-trained to detect centerlines in any tubular structure.

An anatomical prior such as fast segmentation of the ventricles can be used to define a volume of interest in order to apply constraints on the movement of the tracker. In future, the coronary centerlines extracted from the algorithm will also be used to obtain a segmentation of the coronary arteries by also predicting the radius simultaneously. This segmentation result will then be used to further evaluate the performance of AuCoTrack algorithm on Automated Segmentation of Coronary Arteries (ASOCA) challenge dataset.[3]

10. Acknowledgments

Zohaib Salahuddin holds EACEA Erasmus+ grant for the master in Medical Imaging and Applications (MAIA). This work has been done at Philips Research Hamburg as a master thesis under the supervision of Dr. Hannes Nickisch and Dr. Matthias Lenga. The authors would like to thank Philips Research Hamburg for providing the infrastructure and resources for the completion of this master thesis.

Zohaib Salahuddin would like to thank his supervisors Dr. Hannes Nickisch and Dr. Matthias Lenga for their guidance and support for the completion of the thesis. He would like to extend his sincerest gratitude to his

[3] https://asoca.grand-challenge.org/

friends Abdullah and Tewodros for always standing by him through the ups and downs during the MAIA master. He would like to thank all the MAIA family for the wonderful times during the two years. He would also like to acknowledge the support of his better half, Kiran and his entire family.

References

Benjamin, E., Muntner, P., Alonso, A., Bittencourt, M., Callaway, C., Carson, A., Chamberlain, A., Chang, A., Cheng, S., Das, S., Delling, F., Djoussé, L., Elkind, M., Ferguson, J., Fornage, M., Jordan, L., Khan, S., Kissela, B., Knutson, K., Virani, S., 2019. Heart disease and stroke statistics—2019 update: A report from the american heart association. Circulation 139.

Cademartiri, F., Grutta, L., Palumbo, A., Malagutti, P., Pugliese, F., Meijboom, W., Baks, T., Mollet, N., Bruining, N., Hamers, R., Feyter, P., 2007. Non-invasive visualization of coronary atherosclerosis: State-of-art. Journal of Cardiovascular Medicine (Hagerstown, Md.) 8, 129–37.

Cetin Karayumak, S., Demir, A., Yezzi, A., Degertekin, M., Unal, G., 2012. Vessel tractography using an intensity based tensor model with branch detection. IEEE transactions on medical imaging 32.

Cetin Karayumak, S., Unal, G., 2015. A higher-order tensor vessel tractography for segmentation of vascular structures. IEEE transactions on medical imaging 34.

Ecabert, O., Peters, J., Schramm, H., Lorenz, C., Berg, J., Walker, M., Vembar, M., Olszewski, M., Subramanyan, K., Lavi, G., Weese, J., 2008. Automatic model-based segmentation of the heart in ct images. IEEE transactions on medical imaging 27, 1189–201. doi:10.1109/TMI.2008.918330.

Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A., 1998. Multiscale vessel enhancement filtering, in: Wells, W.M., Colchester, A., Delp, S. (Eds.), Medical Image Computing and Computer-Assisted Intervention — MICCAI'98, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 130–137.

Frangi, R., Niessen, W., Vincken, K., Viergever, M., 2000. Multiscale vessel enhancement filtering. Med. Image Comput. Comput. Assist. Interv. 1496.

Friman, O., Kuehnel, C., Peitgen, H.O., 2020. Coronary centerline extraction using multiple hypothesis tracking and minimal paths .

Hampe, N., Wolterink, J., Velzen, S., Leiner, T., Išgum, I., 2019. Machine learning for assessment of coronary artery disease in cardiac ct: A survey. Frontiers in Cardiovascular Medicine 6.

Heron, M., 2020. Deaths: Leading causes for 2017.

Huang, W., Huang, L., Lin, Z., Huang, S., Chi, Y., Zhou, J., Zhang, J.M., Tan, R.S., Zhong, L., 2018. Coronary artery segmentation by deep learning neural networks on computed tomographic coronary angiographic images, pp. 608–611.

Kamnitsas, K., Ledig, C., Newcombe, V., Simpson, J., Kane, A., Menon, D., Rueckert, D., Glocker, B., 2016. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical Image Analysis 36.

Karimi, D., Dou, H., Warfield, S., Gholipour, A., 2020. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical Image Analysis 65, 101759.

Keinert, B., Innmann, M., Sänger, M., Stamminger, M., 2015. Spherical fibonacci mapping. ACM Transactions on Graphics 34, 1–7.

Kitamura, Y., Li, Y., Ito, W., 2012. Automatic coronary extraction by supervised detection and shape matching. Proceedings - International Symposium on Biomedical Imaging , 234–237.

Krissian, K., Bogunović, H., Pozo, J., Villa-Uriol, M.C., Frangi, A., 2008. Minimally interactive knowledge-based coronary tracking in cta using a minimal cost path .

Malakar, A., Choudhury, D., Halder, B., Paul, P., Uddin, D.A., Chakraborty, S., 2019. A review on coronary artery disease, its risk factors, and therapeutics. Journal of Cellular Physiology 234.

Paech, D., Weston, A., 2011. A systematic review of the clinical effectiveness of 64-slice or higher computed tomography angiography as an alternative to invasive coronary angiography in the investigation of suspected coronary artery disease. BMC Cardiovascular Disorders 11, 32.

Schaap, M., Metz, C., Walsum, T., van der Giessen, A., Weustink, A., Mollet, N., Bauer, C., Bogunović, H., Castro, C., Deng, X., Dikici, E., O'Donnell, T., Frenay, M., Friman, O., Hernandez Hoyos, M., Kitslaar, P., Krissian, K., Kühnel, C., Luengo-Oroz, M., Niessen, W., 2009. Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. Medical image analysis 13, 701–14.

Stimpel, B., Wetzl, J., Forman, C., Schmidt, M., Maier, A., Unberath, M., 2018. Automated curved and multiplanar reformation for screening of the proximal coronary arteries in mr angiography. Journal of Imaging 4, 124.

Tavakol, M., Ashraf, S., Brener, S., 2012. Risks and complications of coronary angiography: A comprehensive review. Global journal of health science 4, 65–93.

Wang, C., Smedby, O., 2008. An automatic seeding method for coronary artery segmentation and skeletonization in cta .

WHO, 2018. Global health estimates 2016: Deaths by cause, age, sex, by country and by region, 2000-2016.

Wolterink, J.M., van Hamersvelt, R.W., Viergever, M.A., Leiner, T., Išgum, I., 2019. Coronary artery centerline extraction in cardiac ct angiography using a cnn-based orientation classifier. Medical Image Analysis 51, 46 – 60.

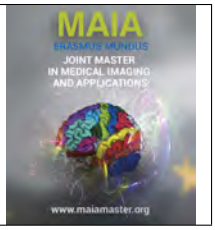
Yang, G., Kitslaar, P., Frenay, M., Broersen, A., Boogers, M., Bax, J., Reiber, J., Dijkstra, J., 2011. Automatic centerline extraction of coronary arteries in coronary computed tomographic angiography. The International Journal of Cardiovascular Imaging 28, 921–33.

Zheng, Y., Tek, H., Funka-Lea, G., 2013. Robust and accurate coronary artery centerline extraction in cta by combining model-driven and data-driven approaches, pp. 74–81.

Zreik, M., van Hamersvelt, R.W., Wolterink, J.M., Leiner, T., Viergever, M.A., Išgum, I., 2019. A recurrent cnn for automatic detection and classification of coronary artery plaque and stenosis in coronary ct angiography. IEEE Transactions on Medical Imaging 38, 1588–1598.

Medical Imaging and Applications

Master Thesis, August 2020



Computer-Aided Detection of Clinically Significant Prostate Cancer in mpMRI

Anindya Shaha, Matin Hosseinzadeh, Henkjan Huisman

Diagnostic Image Analysis Group, Radboud University Medical Center, The Netherlands

Abstract

Non-invasive multiparametric MR imaging (mpMRI) can facilitate the early detection of clinically significant prostate cancer (csPCa). However, interpretation of radiological findings is susceptible to overdiagnosis and low inter-reader agreement, as current assessment standards share a limited ability to distinguish csPCa from benign prostate cancer (PCa) and other non-malignant conditions. In this research, we propose a novel multi-stage computer-aided detection (CAD) model to perform automated voxel-level detection of csPCa in prostate mpMRI. The model is driven by convolutional neural networks (CNN), which use anisotropically-strided 3D convolutions to leverage the spatial context between adjacent MRI slices, without forgoing computational efficiency. It combines spatial and channel-wise attention mechanisms to adaptively target the most salient prostatic structures and discriminative feature dimensions in mpMRI volumes, at multiple resolutions. It uses an additional 3D residual classifier for independent false positive reduction. Finally, it exploits an anatomical prior, which captures the spatial prevalence of csPCa and its zonal distinction, to infuse clinical priori into the CNN architecture for guided inference and feature extraction. For 487 institutional testing scans, the 3D CAD system achieves 83.95% and 89.94% detection sensitivity at 0.5 and 1.0 false positive per patient, respectively, along with 0.884 AUROC in patient-based diagnosis. For 296 external testing scans, the 3D CAD system exhibits moderate agreement with a consensus of expert radiologists (77.70%; $kappa = 0.543$) and independent pathologists (78.04%; $kappa = 0.527$), thereby demonstrating a strong ability to generalize to histologically-confirmed csPCa detection using radiologist-supported training samples only.

Keywords: prostate cancer, magnetic resonance imaging, convolutional neural networks, computer-aided diagnosis and detection, anatomical prior, visual attention, zonal segmentation

1. Introduction

Prostate cancer (PCa) is among the most prevalent cancers in men worldwide. It is estimated that as of January, 2019, over 45% of all men living with a history of cancer in the United States had suffered from PCa (Miller et al., 2019). One of the main challenges surrounding the accurate diagnosis of PCa is its broad spectrum of clinical behavior. PCa lesions can range from low-grade, benign tumours that never progress into clinically significant disease to highly aggressive, invasive malignancies that can rapidly advance towards metastasis and death (Johnson et al., 2014). Successful clinical outcomes rely on the ability to distinguish between the former incidents of benign PCa from the latter strains of clinically significant PCa (csPCa) in the early stages of the disease. In the context of PCa staging and grading systems, the definitive standard in clinical practice is to use prostate biopsies to histologically assign a Gleason Score (GS) to each lesion, as a measure of cancer aggressiveness (Epstein et al., 2016). Although, non-targeted transrectal ultrasound (TRUS) is generally employed to guide biopsy extractions, it is severely prone to an underdetection of csPCa and overdiagnosis of benign PCa (Verma et al., 2017). Thus, multiparametric MR imaging (mpMRI) is used to compensate for these limitations of TRUS (Engels et al., 2020; Israël et al., 2020; Johnson et al., 2014). Negative mpMRI can rule out unnecessary biopsies by 33% (Elwenspoek et al., 2019), benefiting patients with a high risk of complications from repeated needle sampling. Multiple studies confirm that MRI-targeted biopsies can also replace or complement systematic TRUS-guided biopsies with indisputable improvements in diagnostic certainty (Ka-

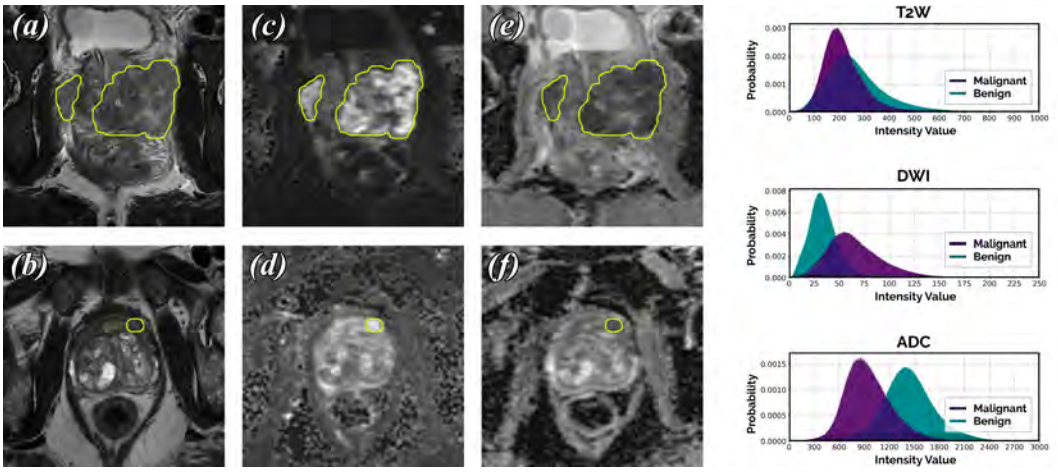


Figure 1: The challenge of discriminating csPCa due to its heterogeneous appearance. ***(a-b)*** T2-weighted imaging (T2W), ***(c-d)*** diffusion-weighted imaging (DWI) and ***(e-f)*** apparent diffusion coefficient (ADC) maps constituting the prostate mpMRI scans for two different patients are shown above, where yellow contours indicate csPCa lesions. While one of the patients has large, severe csPCa developing from both ends ***(top row)***, the other has a relatively focal csPCa lesion surrounded by perceptually similar nodules of non-malignant benign prostatic hyperplasia (BPH) ***(bottom row)***. Furthermore, normalized intensity histograms compiled from all 2733 scans used in this study ***(right)*** reveal a large overlap between the distributions of csPCa and non-malignant prostatic tissue for all three MRI channels.

sivisvanathan et al., 2018; Rouvière et al., 2019; van der Leest et al., 2019). As such, the Prostate Imaging Reporting and Data System (PI-RADS) (Weinreb et al., 2016) continues to be developed. PI-RADS is a comprehensive set of standardized guidelines that enables radiologists to estimate the likelihood of csPCa from the findings of prostate mpMRI. However, even PI-RADS v2 follows a qualitative or semi-quantitative assessment that mandates substantial expertise for proper usage. To make matter worse, csPCa can manifest as multifocal lesions of different shapes and sizes, while bearing a strong resemblance to numerous non-malignant conditions (as seen in Fig. 1) such as benign prostatic hyperplasia (BPH), scar tissue, hemorrhage, prostatitis, postradiation changes, etc. (Israël et al., 2020). In the absence of experienced radiologists, these factors can lead to low inter-reader agreement ($< 50\%$) and sub-optimal interpretation of prostate mpMRI scans (Garcia-Reyes et al., 2015; Rosenkrantz et al., 2016; Smith et al., 2019).

The development of proficient and reliable csPCa detection algorithms has therefore become an important research focus in medical image computing, offering the potential to aid radiologists' workflow and assist the early detection of csPCa with consistent quantitative analysis.

2. State of the Art

The advent of deep convolutional neural networks (CNN) has paved the way for powerful computer-aided detection (CAD) systems that rival human performance (Esteva et al., 2017; McKinney et al., 2020). Machine learning models are increasingly applied for PCa detection, leveraging the high soft-tissue contrast and rich blend of anatomical and functional information present in prostate mpMRI.

In recent years, a number of retrospective studies have investigated the growing potential of CAD systems relative to radiologists. Sanford et al. (2020) compares the PI-RADS classification performance of a four-class 2D ResNet with expert radiologists, reaching 56% agreement on 68 testing scans. Schelb et al. (2019) uses an ensemble of 2D U-Nets to achieve statistically similar csPCa detection performance as a cohort of trained radiologists on 62 testing scans.

Multiple studies have also explored architectural enhancements to extend functionality and improve pre-existing CAD systems. Cao et al. (2019a) proposes a hybrid 2D network titled *FocalNet* for joint PCa detection and GS prediction. Over 5-fold cross-validation using 417 patient scans, *FocalNet* achieves 87.9% sensitivity at 1.0 false positive per patient, trailing behind radiologists by only 1.5% at the same false positive rate. Meanwhile, Yu et al. (2020) proposes a dual-stage 2D U-Net for csPCa detection, with an integrated second-stage classifier solely for false positive reduction.

Cancerous lesions stemming from the prostatic peripheral zone (PZ) exhibit different morphology and pathology than those developing from the transitional zone (TZ) (Chen et al., 2000; Israël et al., 2020; Weinreb et al., 2016). Hosseinzadeh et al. (2019) highlights the merits of utilizing this priori using an early fusion of probabilistic zonal segmentations inside a 2D U-Net detector. The study demonstrates that the inclusion of PZ and TZ segmentations introduces an average increase of 5.3% in csPCa detection sensitivity at 0.5, 1.0 and 2.0 false positives per patient. In a separate study, Cao et al. (2019b) builds a probabilistic prevalence map, depicting the typical sizes, shapes and locations of PCa across

the prostate anatomy. The map is subsequently used as spatial prior in a weakly supervised 2D U-Net detector. Both methods emphasize the value of clinical priori and anatomical features, which also play an integral role in classical machine learning-based CAD systems for PCa as well (Lemaître et al., 2017; Litjens et al., 2014).

The vast majority of CAD systems for csPCa diagnosis operate on a 2D-basis, citing computational limitations and the non-isotropic imaging protocol of prostate mpMRI as their primary rationale. Yoo et al. (2019) tackles this challenge by employing dedicated 2D ResNets for each slice in a patient scan and aggregating all slice-level predictions with a Random Forest classifier. Aldoj et al. (2020) proposes a patch-based approach with highly localized regions of interest (ROI) passing through a standard 3D CNN. Alkadi et al. (2019) follows a 2.5D approach as a compromise solution, sacrificing the usage of multiple MRI channels for an additional pseudo-spatial dimension.

2.1. Contributions

In this research, we harmonize several state-of-the-art techniques from the past 5 years to present a novel end-to-end 3D CAD system for voxel-level detection of csPCa in prostate mpMRI scans. The following points summarize the main contributions of our study:

- We design a dual-attention detection network that adaptively targets the most spatially salient prostatic structures and discriminative features dimensions in mpMRI volumes, at multiple resolutions. Anisotropically-strided 3D convolutions are used to exploit the spatial context between adjacent MRI slices, without forgoing computational efficiency or the ability to harness multiple MRI channels.
- We study the effect of employing a 3D patch-wise residual classifier for independent false positive reduction and we investigate its utility in improving baseline specificity, without sacrificing high detection sensitivity.
- We hypothesize that an anatomical prior that captures the spatial prevalence of csPCa and its zonal distinction across the prostate volume, can be used to infuse domain-specific clinical priori into the CNN architecture. We analyze the impact of such a prior on the overall patient-based diagnosis and lesion-based detection performance, to understand the extent of its contribution to the CAD system.
- While similar studies are limited by a small number of test cases, we evaluate performance on 487 institutional and 296 external testing scans annotated using PI-RADS v2 and biopsy-derived GS, respectively. We benchmark our proposed CAD system against expert radiologists and we evaluate its ability to generalize csPCa detection to histologically-confirmed malignancies, despite using radiologist-supported training samples only.

3. Material and Methods

3.1. Dataset

The primary dataset consists of 2437 prostate mpMRI scans from Radboud University Medical Center (RUMC), paired with fully delineated annotations of csPCa from expert radiologists. From here, 1584 (65%) and 366 (15%) patient scans are partitioned into training and validation sets, respectively, via stratified sampling. The remaining 487 (20%) cases are used to form the testing set (TS1). An external testing set (TS2) is also curated using 296 prostate mpMRI scans from Ziekenhuisgroep Twente (ZGT), whose annotations are supported by independent, expert pathologists using clinically-graded biopsy samples. Special care is taken to ensure mutually exclusive patients between all four subsets of data.

3.1.1. Multi-Parametric MRI Scans

For this study, a total of 2437 prostate mpMRI exams are compiled from the Department of Radiology, Anatomy and Nuclear Medicine at RUMC over the period January, 2016 – December, 2017. Additionally, 296 prostate mpMRI exams are compiled from the Department of Radiology at ZGT over the period March, 2015 – January, 2017. Patients are biopsy-naive men (RUMC: {median age: 66 yrs, IQR: 61-70}, ZGT: {median age: 65 yrs, IQR: 59-68}) with elevated levels of PSA (RUMC: {median level: 8 ng/mL, IQR: 5-11}, ZGT: {median level: 6.6 ng/mL, IQR: 5-9}). Imaging is performed on 3T MR scanners (RUMC: {89.9% on Magnetom Trio/Skyra, 10.1% on Prisma}, ZGT: {100% on Skyra}; Siemens Healthineers, Erlangen, Germany). In both cases, acquisitions are obtained following standard mpMRI protocols in compliance with PI-RADS v2 (Engels et al., 2020). They include T2-weighted imaging (T2W) and diffusion-weighted imaging (DWI). Apparent diffusion coefficient (ADC) maps and high b-value DWI (b>1400 s/mm^2) are computed from the raw DWI scans. Prior to usage, all scans are spatially resampled to a common axial in-plane resolution of 0.5 mm^2 and slice thickness of 3.6 mm via B-spline interpolation. Due to the minimal temporal difference between T2W and DWI acquisitions in the imaging protocol, we observe negligible patient motion across the different sequences. Thus, no additional non-rigid registration is applied across the dataset in agreement with clinical recommendations (Epstein et al., 2016) and recent studies (Cao et al., 2019a).

3.1.2. Clinical Annotations

Patient mpMRI scans in both RUMC and ZGT datasets are reviewed by expert radiologists using PI-RADS v2, where any detected lesions marked PI-RADS

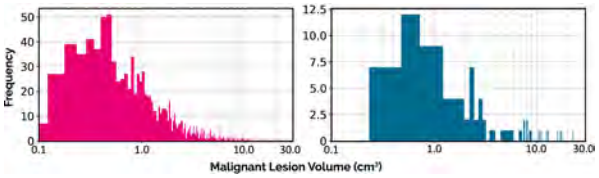


Figure 2: Distribution of csPCa lesion volumes (cm^3) observed in the prostate mpMRI datasets acquired from Radboud University Medical Center (RUMC) ***(left)*** and Ziekenhuisgroep Twente (ZGT) ***(right)***.

4 or 5 are flagged as csPCa$^{(PR)}$. When independently assigned PI-RADS scores are discordant, a consensus is reached through joint assessment. All instances of csPCa$^{(PR)}$ are then carefully delineated by three students on a voxel-level basis.

For ZGT dataset, all patients undergo TRUS-guided biopsies performed by a urologist, blinded to the imaging results. Additionally, in the presence of suspicious lesions (PI-RADS 3-5), patients undergo in-bore MRI-guided biopsies as detailed in van der Leest et al. (2019). Here, the lowest signal areas within a suspicious region on the ADC map are used to target extractions. Tissue samples are reviewed by experienced uropathologists, where cores containing cancer are assigned GS in compliance with the 2014 International Society of Urologic Pathology (ISUP) guidelines (Epstein et al., 2016). Any lesion graded GS > 3+3 is marked as csPCa$^{(GS)}$, and subsequently delineated by two students on a voxel-level basis.

Upon complete annotation, the RUMC and ZGT datasets contain 1532 and 210 *benign* cases, along with 902 and 86 *malignant* cases ($\geq$ 1 csPCa lesion), respectively. Moreover, on a lesion-level basis, the RUMC dataset contains 1095 csPCa$^{(PR)}$ lesions (mean frequency: 1.21 lesions per *malignant* scan; median size: 1.04 cm^3), while the ZGT dataset contains 90 csPCa$^{(GS)}$ lesions (mean frequency: 1.05 lesions per *malignant* scan; median size: 1.69 cm^3). Fig. 2 highlights the wide range of csPCa lesion sizes in both datasets.

3.1.3. Prostate Zonal Segmentations

Multi-class segmentations of prostate TZ and PZ are generated for each scan in the training dataset using a multi-planar, anisotropic 3D U-Net from a separate study (Riepe et al., 2020), where the network achieves an average Dice Similarity Coefficient of 0.90 ± 0.01 for whole-gland segmentation over 5 × 5 nested cross-validation. We use these zonal segmentations to construct the anatomical prior (as detailed in Section 3.2.3).

3.2. Model Architecture

Our proposed model comprises of two parallel 3D CNNs (M_1, M_2) followed by a decision fusion node N_{DF}, as shown in Fig. 3. The model is driven by the detection network M_1, which segments clinically significant instances of prostate malignancy, and it is supplemented by the classification network M_2, which independently scores and divides the scan into eight primarily *benign* or *malignant* regions. M_1 and M_2 rely on anisotropically-strided 3D convolutions to process the mpMRI data, which resemble multi-channel stacks of 2D images rather than full 3D volumes. In both cases, the input volume (x_1, x_2) undergoes intensity normalization. T2W and DWI channels are normalized to zero mean and unit standard deviation. ADC channel is linearly normalized from $[0, 3000]$ to $[0, 1]$, in order to retain its highly relevant numerical significance (Israël et al., 2020). Prevalence map P, constructed using all prostate zonal segmentations and csPCa annotations in the training dataset, is incorporated in M_1 and N_{DF} as an anatomical prior. At train-time, M_1 and M_2 are independently optimized using different loss functions and target labels. At test-time, N_{DF} is used to aggregate their predictions (y_1, y_2) into a single output detection map y_{DF}, with high sensitivity and reduced false positives.

3.2.1. Detection Network

The principal component of our proposed model is the detection network or M_1, as shown in Fig. 4. It is used to generate the preliminary voxel-level detection of csPCa in prostate mpMRI scans with high sensitivity.

Typically, a prostate gland occupies 45-50 cm^3, but it can be significantly enlarged in older males and patients afflicted by BPH (Basillote et al., 2003). The input ROI of M_1, measuring 144 × 144 × 18 voxels per channel or nearly 336 cm^3, includes and extends well beyond this window to utilize peripheral and global anatomical information. M_1 is trained on whole-image volumes equivalent to its total ROI, paired with fully delineated annotations of csPCa$^{(PR)}$ as target labels. Since the larger ROI and voxel-level labels contribute to a severe class imbalance (1:153) at train-time, we use a focal loss function to train M_1. Focal loss addresses extreme class imbalance in one-stage dense detectors by weighting the contribution of easy to hard examples, alongside conventional class-weighting (Lin et al., 2017). In a similar study for joint PCa detection and GS prediction in prostate mpMRI, the authors credit focal loss as one of the pivotal enhancements that enable their proposed solution *FocalNet* (Cao et al., 2019a).

For an input volume, $x_1 = ({x_1}^1, {x_1}^2, ..., {x_1}^n)$ derived from a given scan, let us name its target label $Y_1 = ({Y_1}^1, {Y_1}^2, ..., {Y_1}^n) \in \{0, 1\}$, where n represents the total number of voxels in x_1. We can formulate the focal loss function of M_1 for a single voxel in each scan, as follows, where $i \in [1, n]$:

$$FL({x_1}^i, {Y_1}^i) = -\alpha(1 - {y_1}^i)^\gamma {Y_1}^i \log {y_1}^i - (1 - \alpha)({y_1}^i)^\gamma (1 - {Y_1}^i) \log(1 - {y_1}^i)$$

Here, ${y_1}^i = p(O{=}1|{x_1}^i) \in [0, 1]$, represents the probability of ${x_1}^i$ being a *malignant* tissue voxel as predicted by M_1, while α and γ represent weighting hyperparameters

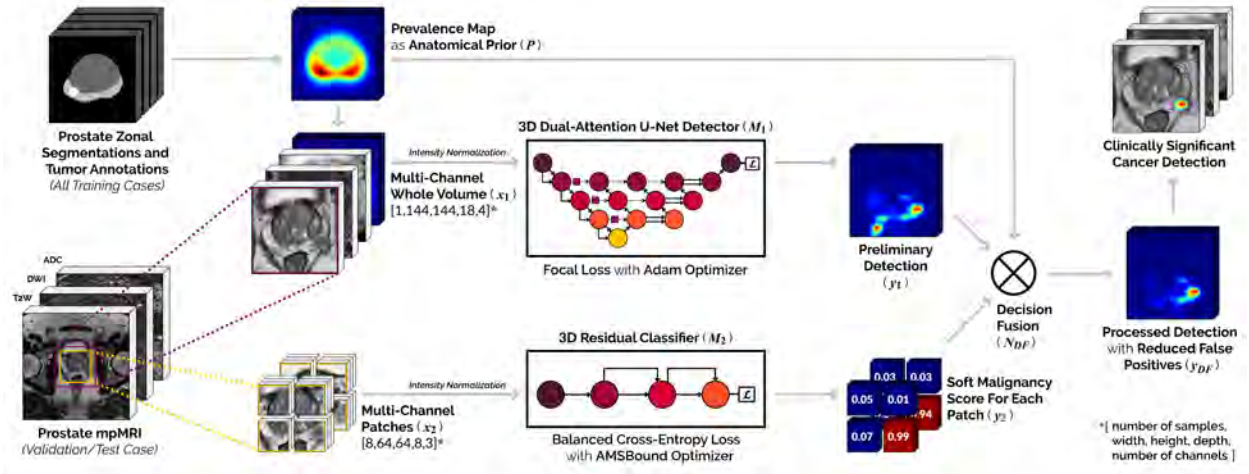


Figure 3: Proposed end-to-end framework for computing voxel-level detections of csPCa in validation/test samples of prostate mpMRI. The model center crops two ROIs from the multi-channel concatenation of a patient's T2W, DWI and ADC scans for the input of its detection and classification 3D CNN sub-models (M_1, M_2). M_1 leverages a prevalence map P in its input x_1 to synthesize spatial priori and generate a preliminary detection of csPCa y_1. M_2 infers on a set of overlapping patches x_2 and maps them to a set of probabilistic malignancy scores y_2. Decision fusion node N_{DF} aggregates y_1, y_2 with P to produce the model output y_{DF} in the form of a post-processed csPCa detection map with high sensitivity and reduced false positives.

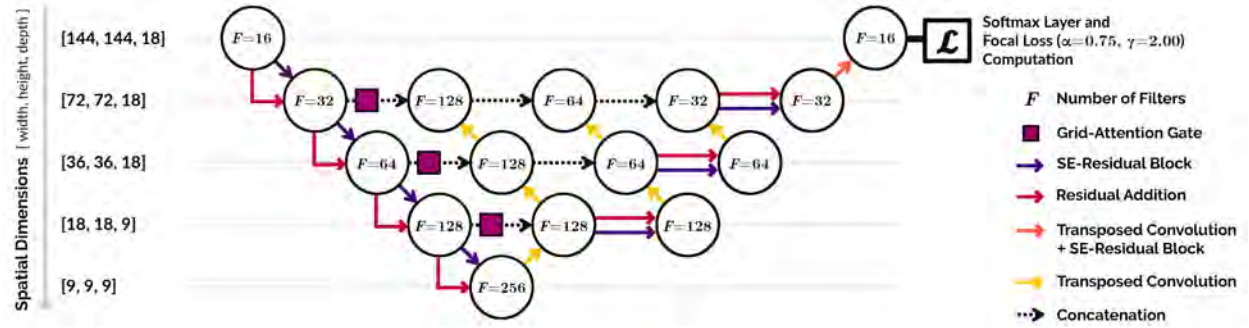


Figure 4: Architecture schematic for the 3D detection sub-model (M_1). M_1 is a modified adaptation of the UNet++ architecture (Zhou et al., 2020), utilizing a pre-activation residual backbone (He et al., 2016) with *Squeeze-and-Excitation* (SE) channel-wise attention mechanism (Hu et al., 2019) and grid-based attention gates (Schlemper et al., 2019). All convolutional layers in the encoder and decoder stages are activated by ReLU and LeakyReLU, respectively, and use kernels of size $3 \times 3 \times 3$ with L_2 regularization ($\beta = 0.001$). Both downsampling and upsampling operations throughout the network are performed via anisotropic strides. Dropout nodes (*rate*=0.50) are connected at the output layer and at each scale of the decoder, for improved generalization.

of the focal loss. At test-time, $y_1 = (y_1{}^1, y_1{}^2, ..., y_1{}^n) \in [0, 1]$, i.e. a voxel-level, probabilistic csPCa detection map for x_1, serves as the final output of M_1 for each scan.

We choose 3D U-Net (Çiçek et al., 2016; Ronneberger et al., 2015) as the base architecture of M_1, for its ability to summarize multi-resolution, global anatomical features (Dalca et al., 2018) and generate an output detection map with voxel-level precision. Pre-activation residual blocks are used at each scale of M_1 for feature extraction. Furthermore, the architecture of its decoder is redesigned into that of a modified UNet++ (Zhou et al., 2020) for dense feature aggregation. UNet++ is based on redesigned skip connections and a nested ensemble configuration. It demonstrates substantial improvements in semantic/instance segmentation, across a wide range of medical imaging modalities. In our adaptation, feature fusion from multiple semantic scales is used to achieve similar performance, while dense blocks and deep supervision from the original design are forgone to remain computationally lightweight.

Two types of differentiable, soft attention mechanisms are employed in M_1 to highlight salient information throughout the training process, without any additional supervision. Channel-wise SE attention (Hu et al., 2019) is used to amplify the most discriminative feature dimensions at each resolution. Attention gates (Schlemper et al., 2019) are used to automatically learn spatially important prostatic structures of varying shapes and sizes. While the former is integrated into every residual block to guide feature extraction, the latter is placed at the start of skip-connections to filter the semantic features being passed onto the de-

coder. During backpropagation, both attention mechanisms work collectively to suppress gradients originating from background voxels and inessential feature maps. Similar combinations of dual-attention mechanisms have reached state-of-the-art performance in semantic segmentation challenges, sharing an ability to integrate local features with their global dependencies (Fu et al., 2019).

3.2.2. Classifier for False Positive Reduction

The goal of the classification network, M_2, is to improve overall model specificity via independent, binary classification of each scan and its constituent segments. It is effectuated by N_{DF}, which factors in these predictions from M_2 to locate and penalize potential false positives in the output of M_1.

M_2 has an input ROI of $112 \times 112 \times 12$ voxels per channel or nearly 136 cm^3, tightly centered around the prostate. While training on the full ROI volume has the advantage of exploiting extensive spatial context, it results in limited supervision by the usage of a single coarse, binary label per scan. Thus, we propose patch-wise training using multiple, localized labels, to enforce fully supervised learning. We define an effective patch extraction policy as one that samples regularly across the ROI to densely cover all spatial positions. Sampled patches must also be large enough to include a sufficient amount of context for subsequent feature extraction. Random sampling within a small window, using the aforementioned criteria, poses the risk of generating highly overlapped, redundant training samples. However, a minimum level of overlap can be crucial, benefiting regions that are harder to predict by correlating semantic features from different surrounding context (Xiao et al., 2018). As such, we divide the ROI into a set of eight octant training samples, x_2, measuring $64 \times 64 \times 8$ voxels each with nearly 7.5% overlap shared between neighboring patches.

For input patches, $x_2 = ({x_2}^1, {x_2}^2, ..., {x_2}^8)$ derived from a given scan, let us name its set of target labels $Y_2 = ({Y_2}^1, {Y_2}^2, ..., {Y_2}^8) \in \{0, 1\}$. We can formulate the standard cross-entropy loss function of M_2 for a single patch in each scan, as follows, where $i \in [1, 8]$:

$$CE({x_2}^i, {Y_2}^i) = -{Y_2}^i \log {y_2}^i - (1 - {Y_2}^i)\log(1 - {y_2}^i)$$

Here, ${y_2}^i = p(O{=}1|{x_2}^i) \in [0, 1]$, represents the probability of ${x_2}^i$ being a malignant patch as predicted by M_2. Next, we introduce a pair of complementary class weights and switch to a balanced cross-entropy loss function to adjust for the patch-level class imbalance (1:4) as follows, where $i \in [1, 8]$:

$$BCE({x_2}^i, {Y_2}^i) = -\beta {Y_2}^i \log {y_2}^i - (1 - \beta)(1 - {Y_2}^i)\log(1 - {y_2}^i)$$

At test-time, $y_2 = ({y_2}^1, {y_2}^2, ..., {y_2}^8) \in [0, 1]$, i.e. a set of probabilistic malignancy scores for x_2, serves as the final output of M_2 for each scan.

Uncontrolled patch-wise training can create high label noise. For instance, a single octant patch contains $64 \times 64 \times 8$ or 32768 voxels per channel. In a naive patch extraction system, if the fully delineated ground-truth for this sample includes even a single voxel of *malignant* tissue, then the overall patch would be assigned the label *malignant* or '1', despite a voxel-level imbalance of 1:32768 supporting the opposite class. Such a training pair proves detrimental to the learning process, where the network associates semantic features to the wrong target class. To mitigate this challenge, we introduce a constraint, τ, that addresses patch-wise label assignment. τ represents the minimum percentage of *malignant* tissue voxels required in an image volume, for the overall volume itself to be considered *malignant*.

Due to their modularity and continued success in supporting state-of-the-art segmentation and detection performance in the medical domain (Jiang et al., 2020; McKinney et al., 2020; Yoo et al., 2019), we choose CNN architectures based on residual learning for feature extraction. In particular, we investigate the original pre-activation ResNet-v2 (He et al., 2016) and four of its derivative architectures to determine the most suitable configuration for M_2:

1. **Inception-ResNet-v2** (Szegedy et al., 2017): *Inception* blocks are built upon branches of convolutional layers with multiple kernel sizes, operating on the same level. This structure enables them to process spatially variable csPCa lesions across prostate mpMRI scans, with an appropriate kernel size for each scan. Additionally, in hybrid *Inception-ResNet* blocks, residual learning speeds up model convergence and allows the overall architecture to be simplified and/or extended, as required.
2. **Residual Attention Network** (Wang et al., 2017): In residual attention blocks, spatial masks are dynamically computed at each semantic level to target the most salient regions in prostate mpMRI scans. They are robust to noisy labels and can limit their detrimental gradient updates throughout the training stage.
3. **SEResNet** (Hu et al., 2019): *Squeeze-and-Excitation* (SE) mechanism adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between both prostate mpMRI channels (T2W, DWI, ADC) and its convolutional feature maps. This form of channel-wise attention can play an analogous role to a radiologist's task of correlating information from all MRI sequences to inform clinical diagnosis.
4. **SEResNeXt** (Hu et al., 2019): On top of SE attention, *SEResNeXt* blocks perform aggregated

transformations from multiple branches sharing the same topology (unlike *Inception* blocks) via pointwise grouped convolutional layers. The *-NeXt* architecture has been experimentally proven to be a more effective way of increasing network capacity, than going deeper or wider (Xie et al., 2017).

3.2.3. Anatomical Prior

Most CNN architectures are often conceived as *one-size-fits-all* solutions to computer vision challenges, where objects can belong to one of 1000 different classes and occupy any part of natural color images (Deng et al., 2009). In contrast, medical imaging modalities in radiology and nuclear medicine exhibit much lower inter-sample variability, where the spatial content of a scan is limited by the underlying imaging protocols and human anatomy. Traditional image analysis techniques, such as MALF (Wang et al., 2013), can exploit this property in the form of *atlases* or multi-expert labeled template images reflecting the target organ anatomy. Similarly, machine learning models can also adapt several techniques, such as reference coordinate systems (Kooi et al., 2017; Wachinger et al., 2018), to infuse domain-specific spatial priori into CNN architectures. Parallel to these recent studies (Cao et al., 2019b; Dalca et al., 2018; Hosseinzadeh et al., 2019), we hypothesize that M_1, as a variant architecture of U-Net, can benefit from an explicit anatomical prior for csPCa detection. To this end, we construct a probabilistic prevalence map, P, using 1584 prostate zonal segmentations and csPCa$^{(PR)}$ annotations from the training dataset. The goal of P is two-fold. It is used as an additional channel for every input scan fed to M_1, thereby guiding its learning process as spatial priori. It is also used as a spatial weight map in N_{DF}, while penalizing potential false positives in the output of M_1.

For the i-th mpMRI scan in the training dataset, let us consider its specific prevalence map as $p_i = ({p_i}^1, {p_i}^2, ..., {p_i}^n)$, where n represents the total number of voxels per channel. Let us consider the binary masks for the prostate TZ, PZ and csPCa$^{(PR)}$ (if present) in this sample as B_{TZ}, B_{PZ} and B_M, respectively. Therefore, the value of the j-th voxel in p_i can be computed as follows:

$$f({p_i}^j) = \begin{cases} 0.00 & {p_i}^j \in (B_{TZ} \cup B_{TZ} \cup B_M)' \\ \mu & {p_i}^j \in B_{TZ} \cap {B_M}' \\ 3\mu & {p_i}^j \in B_{PZ} \cap {B_M}' \\ 1.00 & {p_i}^j \in B_M \end{cases}$$

Here, $f({p_i}^j)$ aims to model the spatial likelihood of csPCa by drawing upon the empirical distribution of the training dataset. Nearly 75% and 25% of all PCa cases emerge from PZ and TZ, respectively (Chen et al., 2000; Israël et al., 2020). Thus, similar to PI-RADS v2, $f({p_i}^j)$ also incorporates the importance of zonal distinction during the assessment of csPCa. In terms of the likelihood of carrying csPCa, it assumes that voxels belonging to the background class are not likely ($f({p_i}^j) = 0.00$), those belonging to TZ are more likely ($f({p_i}^j) = \mu$), those belonging to PZ are three times as likely as TZ ($f({p_i}^j) = 3\mu$), and those containing csPCa are the most likely ($f({p_i}^j) = 1.00$), in any given scan. The value of $\mu \in [0, 0.20]$ is a hyperparameter that regulates the relative contribution of *benign* prostatic regions in the composition of p_i.

The mean of every specific prevalence map derived from the training dataset equates to a single probabilistic prevalence map, $P = (\sum p_i)/1584 \in [0, 1]$. P represents the anatomical prior used in the proposed model, as shown in Fig. 3, 5.

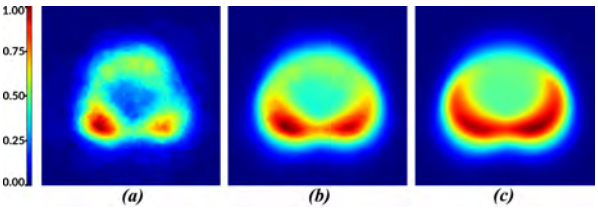


Figure 5: Anatomical prior P at different values of μ. **(*a*)** P at $\mu = 0.00$ is equivalent to the mean csPCa$^{(PR)}$ annotation in the training dataset. This map captures the common sizes, shapes and locations of *malignant* lesions. **(*b*)** P at $\mu = 0.01$ blends the information of csPCa$^{(PR)}$ annotations with that of prostate zonal segmentations in the training dataset. **(*c*)** P at $\mu = 0.20$ is equivalent to the weighted average of all prostate zonal segmentations in the training dataset.

3.2.4. Decision Fusion

Thus far, the model produces two distinct outputs for an input mpMRI scan. M_1 generates a voxel-level detection of csPCa termed y_1. M_2 generates a vector of patch-level malignancy scores termed y_2. The goal of the decision fusion node N_{DF} is to aggregate these predictions into a single output sharing near identical sensitivity as y_1, but with improved specificity.

False positives in y_1 are fundamentally clusters of positive values located in the *benign* regions of a scan. N_{DF} employs y_2 as a means of identifying these regions. To illustrate its working principle, let us assume y_2 = (0.05, 0.01, 0.07, 0.99, 0.03, 0.03, 0.01, 0.94) for the eight octants x_2 forming a given image volume, as shown in Fig. 6. We can set a threshold T_P on $(1 - {y_2}^i)$ to classify each patch ${x_2}^i$, where $i \in [1, 8]$. T_P represents the minimum probability of ${x_2}^i$ being a *benign* patch, required to classify it as such. A high value of T_P adapts M_2 as a highly sensitive classifier that yields very few false negatives, if any at all. In this case, at $T_P = 0.90$ where $(1 - y_2)$ = (0.95, 0.99, 0.93, 0.01, 0.97, 0.97, 0.99, 0.06), we can deduce that the image contains six *benign* patches. Once identified, any false positives in these patches are suppressed by multiplying their corresponding regions in y_1 with a penalty factor λ. Furthermore, we use the anatomical prior P as a spatial weight map for λ, instead of penalizing every voxel by the same value. The resultant detection y_{DF}, i.e. es-

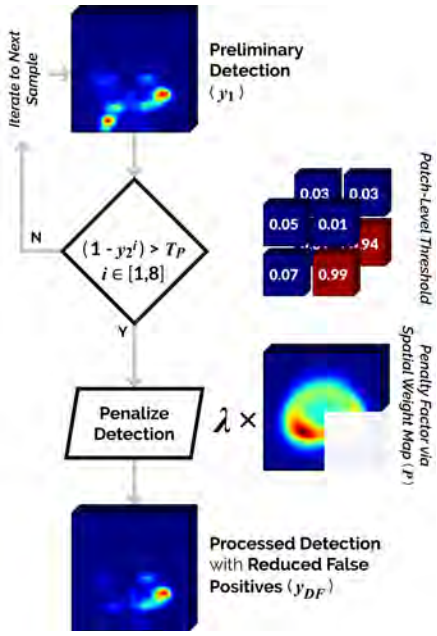


Figure 6: Working principle of the decision fusion node N_{DF}. Threshold T_P is applied on the complement of the classifier output y_2, to select patches of the scan that are predicted *benign* with a high degree of confidence. Positive values within these selection regions of the detector output y_1 are suppressed by a penalty factor λ, spatially weighted over the anatomical prior P. The resultant output y_{DF} shares similar sensitivity as y_1, but with reduced false positives.

sentially a post-processed y_1, serves as the final output of the proposed end-to-end model.

N_{DF} is limited to a simple framework consisting of only two hyperparameters to alleviate the risk of overfitting. An appropriate combination of T_P and λ can either reduce clear false positives without sacrificing any true positives, or facilitate an aggressive false positive reduction scheme at the expense of true positives in y_{DF}. In this research, we opt for the former policy for which the optimal values of T_P and λ are determined to be 0.98 and 0.90 respectively via a coarse-to-fine hyperparameter grid search.

3.3. Experimental Analysis

The following section shares a detailed account of the series of experiments we perform to analyze and optimize each component, statistically evaluate performance and thereby justify our primary design choices throughout the end-to-end model. We facilitate a fair comparison by maintaining an identical preprocessing, augmentation, tuning and train-validation pipeline for each candidate model in a given experiment. Patient-based diagnosis performance is evaluated using the Receiver Operating Characteristic (ROC), where the area under ROC (AUROC) is estimated from the normalized Wilcoxon/Mann-Whitney U statistic (Hanley and McNeil, 1982). To address PCa multifocality (Cao et al., 2019a), lesion-based performance is evaluated using the Free-Response Receiver Operating Characteristic (FROC), where detections sharing a minimum Dice Similarity Coefficient of 0.10 with the ground-truth annotation are considered true positives. Confidence intervals are estimated as twice the standard deviation from the mean of 5-fold cross-validation (applicable to training/validation sets) or 1000 replications of bootstrapping (applicable to testing sets). Statistically significant improvements are verified with a p-value on the difference in case-level AUROC and lesion-level sensitivity at clinically relevant false positive rates (0.5, 1.0) using 1000 replications of bootstrapping (Chihara et al., 2014). Bonferroni correction is used to adjust the significance level for multiple comparisons.

3.3.1. Structure of Anatomical Prior

Adjusting the value of μ can lead to remarkably different anatomical priors, as seen in Fig. 5. The structure of P can vary from a prevalence map reflecting the average csPCa$^{(PR)}$ annotation in the training dataset at $\mu = 0.00$, to a weighted probabilistic atlas of the prostate zonal anatomy at $\mu = 0.20$. We construct four different priors, switching the value of μ between 0.00, 0.01, 0.10 and 0.20, to investigate the range of its impact on csPCa detection. Each prior is used as an additional channel to train M_1 over 5-fold cross-validation, where performance is evaluated using lesion-level FROC and case-level ROC analyses, as shown in Fig. 7.

3.3.2. Classification Architecture

To determine the optimal architecture for M_2, five different 3D CNNs (ResNet-v2, Inception-ResNet-v2, Residual Attention Network, SEResNet, SEResNeXt) are implemented and tuned across their respective hyperparameters to maximize patch-level AUROC over 5-fold cross-validation. Furthermore, we train each candidate CNN using whole-images and patches, in separate turns, to draw out a comparative analysis surrounding the merits of spatial context versus localized labels. In the latter case, we study the effect of τ in regulating patch-wise label assignment (refer to Section 3.2.2). Although, higher values of τ produce lower label noise, they also remove smaller instances of csPCa from the train-validation cycle by counting them as *benign*. We investigate four different values of τ: 0.00%, 0.10%, 0.50% and 1.00%, which correspond to minimum csPCa volumes of size 1, 33, 164 and 328 voxel(s) per channel or 0.0009, 0.0297, 0.0594 and 0.1188 cm^3, respectively. Results are noted in Table 1. The top-performing CNN configuration, or M_2, is assessed qualitatively via 3D gradient-weighted class activation maps (GradCAM) (Selvaraju et al., 2017), as shown in Fig. 8, to ensure adequate interpretability for clinical usage.

3.3.3. Comparative Analysis of Detection Performance

To assess the efficacy of M_1, we compare it against the three baseline 3D CNNs (U-Net, UNet++, Attention U-Net) that inspire its design and construction. Initially,

each detection network is tuned and optimized over 5-fold cross-validation. Afterwards, we benchmark their performance on the testing datasets.

With regards to the overall end-to-end system, we analyze the individual contributions of the three fundamental components (M_1, M_2, P) constituting our proposed model, by comparing the following configurations on the testing datasets:

- **Proposed model**: Detection network supplemented by an independent classifier for false positive reduction and guided by an anatomical prior for csPCa detection.
- $\boldsymbol{M_1 \otimes M_2}$: Detection network supplemented by an independent classifier for false positive reduction only.
- $\boldsymbol{M_1}$: Detection network only.

To compare our proposed CAD system to expert radiologists, we analyze their relative performance on the external testing set TS2. Ground-truth annotations of TS2 are clinically graded by independent pathologists using TRUS/MRI-targeted biopsy samples, where csPCa$^{(GS)}$ corresponds to all lesions marked higher than GS 3+3 (see Section 3.1.2). Agreement between radiologists' patient-level diagnosis and that of our proposed CAD system is evaluated using Cohen's *kappa*.

Unlike csPCa$^{(GS)}$ lesions, which are histologically-confirmed by pathologists, csPCa$^{(PR)}$ lesions are only estimated from the findings of prostate mpMRI. Hence, the latter typically include false positives (Israël et al., 2020). Training a CAD system on the more readily available csPCa$^{(PR)}$ annotations can therefore introduce label noise into the learning cycle, which hampers performance. We evaluate the generalization capability of our proposed model by using it to detect csPCa$^{(GS)}$ lesions in TS2, while training it on csPCa$^{(PR)}$ lesions only.

In each case, the ability of a given configuration to accurately diagnose and localize csPCa is estimated via case-level ROC and lesion-level FROC analyses. Results for TS1 and TS2 are noted in Fig. 9, Table 2 and Fig. 10, Table 3, respectively. Furthermore, we draw a qualitative assessment of the detections generated by each candidate system (as shown in Fig. 11) to identify their key advantages and shortcomings, while tacking potentially challenging patient cases afflicted by multiple non-malignant conditions.

4. Results and Discussion

4.1. Structure of Anatomical Prior

Fig. 7 illustrates the effect of modifying μ to change the structure of the prior (as discussed in Section 3.3.1). We notice that an anatomical prior P that only captures the spatial prevalence of csPCa ($\mu = 0.00$) holds the least amount of useful priori to guide inference and feature extraction. Meanwhile, if P primarily incorporates zonal distinction ($\mu = \{0.10, 0.20\}$), it tends to be more beneficial. Finally, parallel to what we observe in Fig. 5, it is essentially a prior that balances both of these anatomical features ($\mu = 0.01$) that carries the most value for M_1. Thus, in our proposed model, we construct P with $\mu = 0.01$. An appropriate structure for the prior can improve case-level AUC by 2.80% and lesion-level detection sensitivity by 7.34% and 4.21% at 0.5 and 1.0 false positive per patient, respectively, over 5-fold cross-validation.

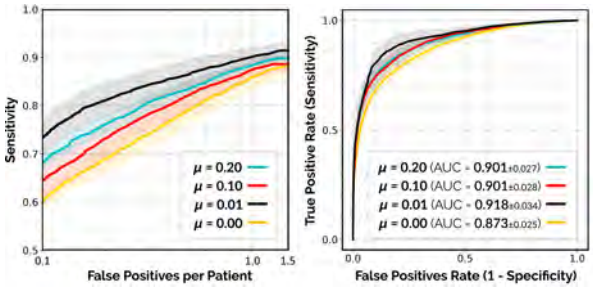


Figure 7: Effect of varying μ in anatomical prior P. Each variant of P is used, in turns, to train and evaluate the lesion-level FROC ***(left)*** and case-level ROC ***(right)*** performance of M_1 over 5-fold cross-validation. 95% confidence intervals are estimated as twice the standard deviation.

4.2. Classification Architecture

From Table 1, we deduce the top-performing architecture across every training scheme to be the 3D SEResNet. We also notice that a higher degree of supervision from patch-wise training proves more useful than the near 8× additional spatial context provided per sample via whole-image training. Increasing the value of τ consistently improves performance for all candidate classifiers (upto 10% in patch-level AUROC). While we attribute this improvement to lower label noise, it is important to note that the vast majority of csPCa lesions are small (refer to Fig. 2) and risk being discarded at higher values of τ. In fact, when τ is equal to 0.00%, 0.10%, 0.50% and 1.00%, the ground-truth for 0, 0, 1 and 9 *malignant* cases, respectively, are incorrectly set as *benign*. Hence, for our proposed model, we use the 3D SEResNet patch-wise classifier trained at $\tau = 0.10\%$ as M_2. At $\tau = 0.10\%$, we are able to suppress label noise and incrementally improve patch-level AUROC by nearly 2% relative to a naive patch extraction system ($\tau = 0.00\%$), without any cases of incorrect label assignment ($\tau = \{0.50, 1.00\}\%$). GradCAMs confirm that M_2 accurately targets csPCa lesions (if any) on a voxel-level basis, despite being trained on patch-level binary labels only (as highlighted in Fig. 8). In stark contrast to ensembling, which can scale up the number of trainable parameters multiple times over for small gains in performance, the addition of M_2 to M_1 bears negligible com-

putational overhead. More specifically, M_2 accounts for less than 0.6% of the total number of trainable parameters in $M_1 \otimes M_2$. Its independent training scheme also enables M_2 to be a modular component of our proposed CAD system. In other words, it can easily be tuned, upgraded or swapped out entirely upon future advancements without affecting the stand-alone performance of M_1 –making the model scalable for large datasets and clinical deployment. Additional details regarding the network and training configurations of M_1 and M_2 are noted in Appendix A.

4.3. Comparative Analysis of Detection Performance

In this section, the experimental results of Section 3.3.3 are analyzed sequentially on the basis of each testing set. First, we examine the proficiency of each candidate CNN system to accurately detect csPCa$^{(PR)}$ lesions in TS1. Next, we study their efficacy in retaining and generalizing detection performance to csPCa$^{(GS)}$ lesions in TS2.

4.3.1. Detection of Malignant PI-RADS Lesions

Lesion Localization: From the FROC analysis on the institutional testing set TS1 (refer to Fig. 9 and Table 2), we see that M_1 reaches 88.08 ± 0.99% detection sensitivity at 1.0 false positive per patient, outperforming the baseline U-Net (80.96 ± 1.64%), UNet++ (83.65 ± 1.51%) and Attention U-Net (84.93 ± 1.39%). With the addition of classifier M_2 to M_1, $M_1 \otimes M_2$ generates upto 12.89% less false positives per patient, while retaining the same maximum detection sensitivity (92.16%) as M_1. With the inclusion of anatomical prior P in $M_1 \otimes M_2$, our proposed model benefits from a further 3.2% increase in partial area under FROC (pAUC) between 0.1 to 2 false positives per patient, reaching 1.663 ± 0.026 pAUC. The proposed CAD system achieves 83.95 ± 1.55% detection sensitivity at

Table 1: Candidate network architectures and training schemes (whole-image versus patch-wise training with different values of τ to regulate label noise) for the classifier M_2. Performance scores indicate the mean of 5-fold cross-validation, followed by the 95% confidence interval estimated as twice the standard deviation.

Model	AUROC (Image)	AUROC (Patches)			
		τ = 0.00%	τ = 0.10%	τ = 0.50%	τ = 1.00%
ResNet-v2	0.8197±0.0175	0.8301±0.0101	0.8443±0.0110	0.8682±0.0130	0.8974±0.0079
Inception-ResNet-v2	0.8234±0.0173	0.8221±0.0137	0.8603±0.0147	0.8830±0.0094	0.9050±0.0079
Residual Attention Network	0.8261±0.0235	0.8373±0.0115	0.8500±0.0072	0.8758±0.0082	0.9005±0.0086
SEResNet	0.8357±0.0136	0.8420±0.0195	0.8608±0.0053	0.8856±0.0085	0.9121±0.0082
SEResNeXt	0.8201±0.0221	0.8333±0.0130	0.8430±0.0053	0.8748±0.0098	0.8960±0.0119

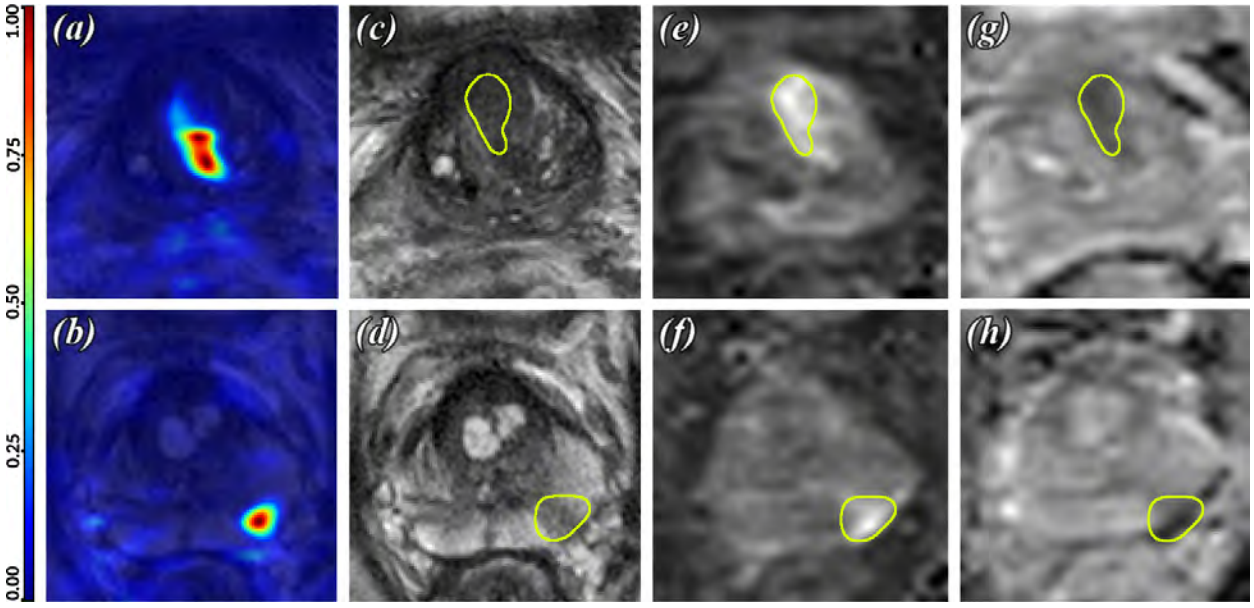


Figure 8: ***(a-b)*** Gradient-weighted class activation maps (GradCAM) of M_2 overlaid on their T2W scans, followed by the corresponding ***(c-d)*** T2W, ***(e-f)*** DWI and ***(g-h)*** ADC scans for two patient cases from the validation set. Both cases include a single instance of csPCa$^{(PR)}$ located in TZ (***top row***) or PZ (***bottom row***), as indicated by the yellow contours. Whole-image GradCAMs are generated by restitching and normalizing the eight patch-level GradCAMs per case. Maximum voxel-level activation is observed in the vicinity of csPCa$^{(PR)}$, despite training M_2 on patch-level binary labels only.

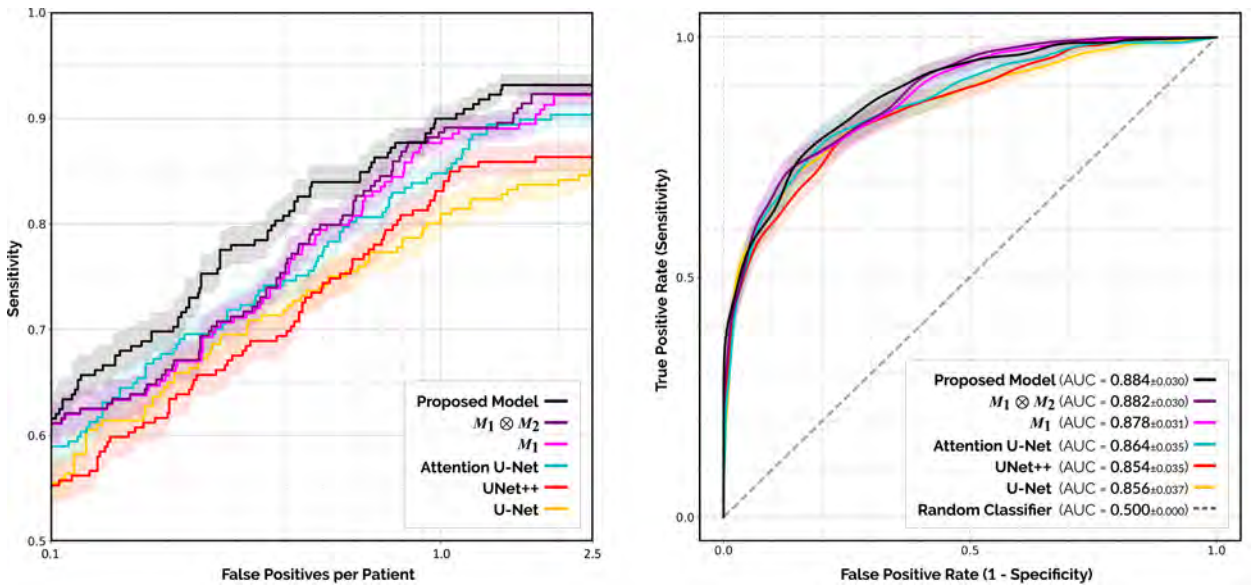


Figure 9: Lesion-level FROC (***left***) and case-level ROC (***right***) analyses of csPCa$^{(PR)}$ detection sensitivity against the number of false positives generated per patient scan using the baseline, ablated and proposed detection models on the institutional testing set TS1. Transparent areas indicate the 95% confidence intervals estimated as twice the standard deviation from 1000 replications of bootstrapping.

Table 2: False positives per patient scan at given csPCa$^{(PR)}$ lesion detection sensitivity (SENS) using the baseline and proposed detection models on the institutional testing set TS1. 95% confidence intervals are estimated as twice the standard deviation from 1000 replications of bootstrapping. TS1 includes 487 prostate mpMRI scans from the Departments of Radiology, Nuclear Medicine and Anatomy at Radboud University Medical Center, clinically reported by expert radiologists using PI-RADS v2, where csPCa$^{(PR)}$ correspond to all lesions marked PI-RADS 4, 5.

Model	Params	False Positives Per Scan				
		70% SENS	75% SENS	80% SENS	85% SENS	90% SENS
U-Net	1.62M	0.315±0.015	0.513±0.019	0.919±0.026	2.406±0.044	–
UNet++	14.93M	0.408±0.017	0.548±0.020	0.769±0.025	1.100±0.033	–
Attention U-Net	2.24M	0.263±0.013	0.423±0.017	0.576±0.021	1.030±0.028	1.889±0.043
M_1	15.25M	0.263±0.014	0.393±0.017	0.578±0.021	0.806±0.023	1.741±0.036
$M_1 \otimes M_2$	15.34M	0.257±0.013	0.385±0.016	0.549±0.020	0.753±0.023	1.526±0.036
Proposed Model	15.34M	0.204±0.012	0.243±0.013	0.376±0.016	0.652±0.023	1.087±0.031

0.5 false positive per patient, surpassing the best baseline (Attention U-Net) by 6.07% ($p \leq 0.001$), while detecting 4.52% ($p \leq 0.05$) and 4.04% ($p \leq 0.05$) more csPCa$^{(PR)}$ lesions than its component systems M_1 and $M_1 \otimes M_2$, respectively. It reaches a maximum detection sensitivity of 93.13 ± 0.98% at 1.43 false positives per patient, identifying a higher percentage of csPCa occurrences than all other candidate systems.

Patient-Based Diagnosis: From the ROC analysis on the institutional testing set TS1 (refer to Fig. 9), we observe that our proposed model reaches 0.884 ± 0.03 AUROC in case-level diagnosis, ahead of all other candidate systems by a margin of 0.2-2.8%. While it performs significantly better than the baseline U-Net ($p \leq 0.01$), UNet++ ($p \leq 0.001$) and Attention U-Net ($p \leq 0.01$), its ability to discriminate between *benign* and *malignant* patient cases is statistically similar to M_1 ($p = 0.170$) and $M_1 \otimes M_2$ ($p = 0.308$).

4.3.2. Generalization to Malignant GS Lesions

Both the FROC and ROC analyses on the external testing set TS2 (refer to Fig. 10 and Table 3) indicate similar patterns emerging as those observed in Section 4.3.1, but with an overall decrease in performance. We primarily attribute this decline to the disparity between the training annotations (csPCa$^{(PR)}$) and the testing annotations (csPCa$^{(GS)}$) in TS2. By comparing the relative drop in performance for each candidate model, we can effectively estimate their respective generalization and latent understanding of csPCa, beyond the training samples.

Lesion Localization: At 1.0 false positive per patient, our proposed CAD system achieves 86.84 ± 1.46% detection sensitivity on TS2 (refer to Fig. 10 and Table 3), performing significantly better ($p \leq 0.001$) than the baseline U-Net (66.68 ± 2.61%), UNet++ (76.72 ± 2.26%) and Attention U-Net (73.57 ± 2.32%). It also

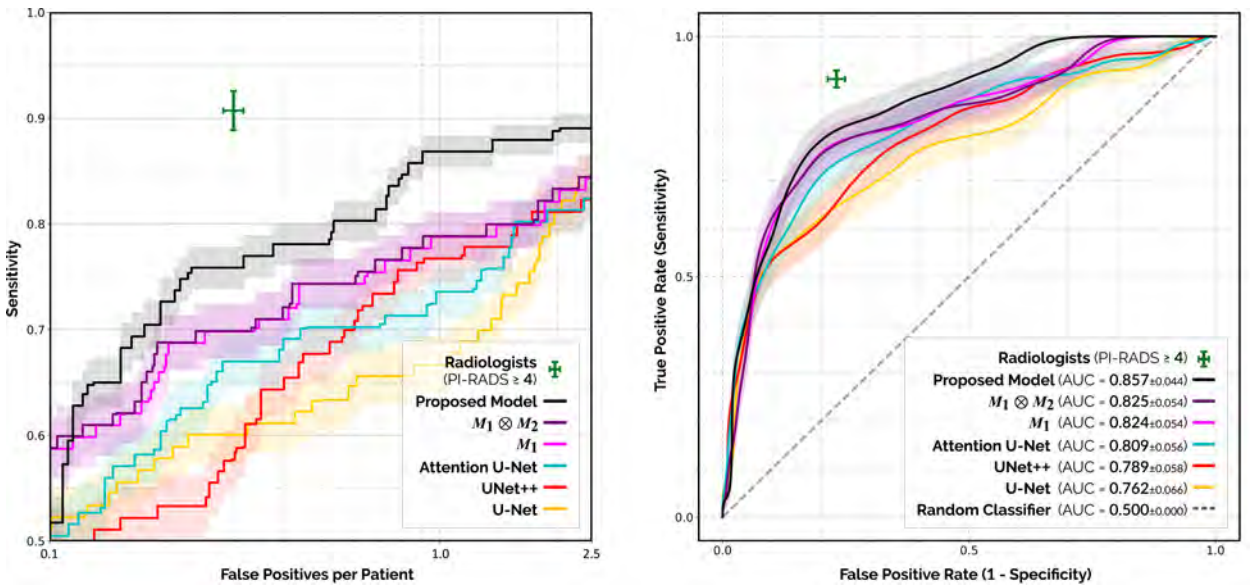


Figure 10: Lesion-level FROC (***left***) and case-level ROC (***right***) analyses of csPCa$^{(GS)}$ detection sensitivity against the number of false positives generated per patient scan using the baseline, ablated and proposed detection models on the external testing set TS2. Transparent areas indicate the 95% confidence intervals estimated as twice the standard deviation from 1000 replications of bootstrapping. Performance for the consensus of radiologists is shown by green markers, where lesions marked PI-RADS 4 or 5 are considered positive detections (as detailed in Section 3.1.2). Centerpoint and length of the markers indicate the mean and 95% confidence interval, respectively.

Table 3: False positives per patient scan at given csPCa$^{(GS)}$ lesion detection sensitivity (SENS) using the baseline and proposed detection models on the external testing set TS2. 95% confidence intervals are estimated as twice the standard deviation from 1000 replications of bootstrapping. TS2 includes 296 prostate mpMRI scans from the Department of Radiology at Ziekenhuisgroep Twente. All lesions are clinically graded by independent, expert pathologists using TRUS/MRI-targeted biopsy samples, where csPCa$^{(GS)}$ correspond to all lesions marked higher than GS 3+3.

Model	Params	False Positives Per Scan				
		65% SENS	70% SENS	75% SENS	80% SENS	85% SENS
U-Net	1.62M	0.594±0.024	1.338±0.036	1.688±0.041	1.931±0.043	–
UNet++	14.93M	0.398±0.017	0.595±0.019	0.770±0.021	1.625±0.034	–
Attention U-Net	2.24M	0.258±0.015	0.438±0.019	1.262±0.032	1.554±0.037	–
M_1	15.25M	0.185±0.013	0.334±0.016	0.641±0.020	1.793±0.033	–
$M_1 \otimes M_2$	15.34M	0.172±0.013	0.328±0.015	0.607±0.020	1.711±0.033	–
Proposed Model	15.34M	0.152±0.009	0.174±0.010	0.221±0.012	0.532±0.018	0.817±0.023

detects 8.06% ($p \leq 0.001$) and 8.01% ($p \leq 0.005$) more csPCa$^{(GS)}$ lesions than its ablated counterparts M_1 and $M_1 \otimes M_2$, respectively. The proposed model reaches a maximum detection sensitivity of 89.04 ± 1.33% at 2.03 false positives per patient, scoring higher than all other candidate systems. Relative to their performance on TS1, all baseline models undergo 7-14% drops in detection sensitivity at 1.0 false positive per patient. Similarly, even M_1 and $M_1 \otimes M_2$ undergo nearly 10% drops in detection sensitivity. With the inclusion of anatomical prior P in $M_1 \otimes M_2$, this decline comes down to only 3% for our proposed model. As such, we deduce that the anatomical prior plays a crucial role in enhancing the generalization capability of our proposed CAD system, far more than its architectural novelties. This is further verified by the overall 11.42% increase in pAUC between 0.1 to 2 false positives per patient, from the inclusion of P in $M_1 \otimes M_2$.

Patient-Based Diagnosis: With regards to case-level diagnosis performance on TS2 (refer to Fig. 10), we observe that our proposed model reaches 0.857 ± 0.04 AUROC, surpassing the baseline U-Net, UNet++ and Attention U-Net by 9.5% ($p \leq 0.001$), 6.8% ($p \leq 0.001$) and 4.8% ($p \leq 0.05$), respectively. Compared to TS1, our proposed CAD system demonstrates 2.7% decrease in AUROC on TS2, while all other candidate models undergo reductions between 5.5-9.4%. Once again, the anatomical prior proves vital, enabling our proposed model to outperform its immediate counterpart $M_1 \otimes M_2$ by 3.3% ($p \leq 0.05$).

Radiologists: FROC and ROC analyses for the consensus of radiologists on TS2 are indicated by green markers in Fig. 10. On a lesion-level basis, radiol-

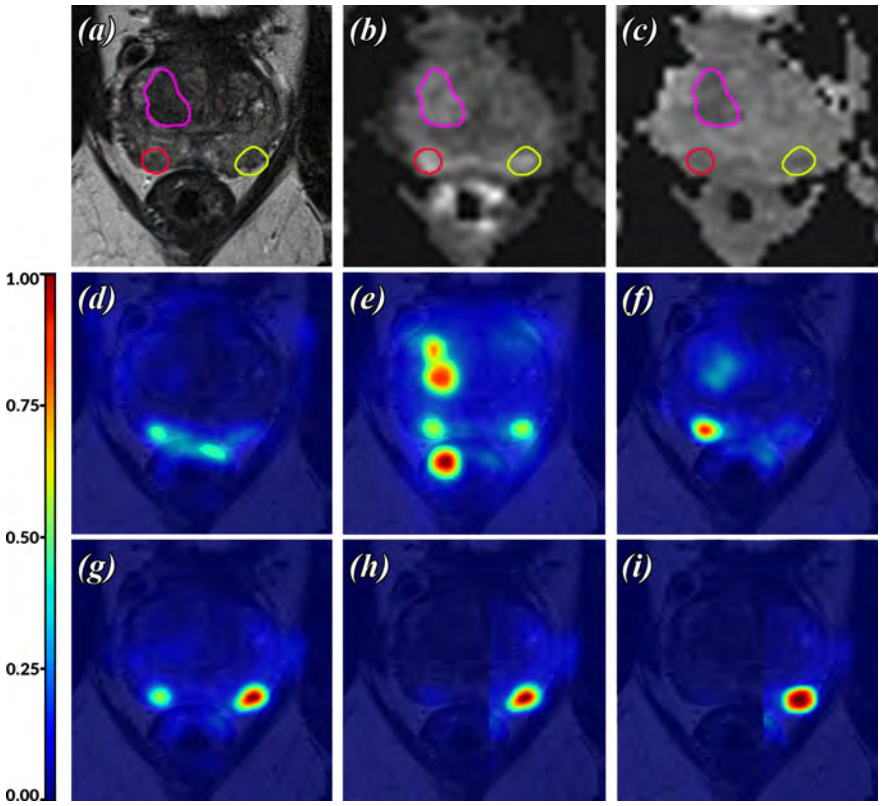


Figure 11: **(*a*)** T2W, **(*b*)** DWI and **(*c*)** ADC scans for a patient case in the external testing set TS2, followed by its corresponding csPCa detection maps from the **(*d*)** U-Net, **(*e*)** UNet++, **(*f*)** Attention U-Net, **(*g*)** M_1, **(*h*)** $M_1 \otimes M_2$ and **(*i*)** the proposed model. Yellow, red and magenta contours in **(*a*-*c*)** indicate csPCa, benign PCa and BPH occurrences, respectively, afflicting the patient's prostate gland.

ogists achieve 90.72 ± 1.74% detection sensitivity at 0.30 false positives per patient. On a case-level basis, they achieve 91.10 ± 1.67% sensitivity at 76.92 ± 1.53% specificity. In comparison, our proposed CAD system reaches 75.85 ± 1.97% detection sensitivity at the same false positive rate and 80.46 ± 4.09% sensitivity at the same specificity. With a probability threshold of 0.254 on its patient-level detections, our CAD system shares 77.70% (230/296 cases) agreement with radiologists, while operating at the same specificity. The corresponding *kappa* value is 0.543 ± 0.021. Furthermore, at the same operating point, radiologists and our CAD system share 81.42% (241/296 cases) and 78.04% (231/296 cases) agreement with *kappa* values of 0.609 ± 0.030 and 0.527 ± 0.033, respectively, with the independent pathologists. Its moderate agreement with both clinical experts (similar to inter-reader agreement stated in Section 1), demonstrates that our CAD system can potentially act as a viable second reader in a screening setting.

4.3.3. Qualitative Assessment of Model Detections

Fig. 11 illustrates the general differences between the positive detections of each candidate model. In this example, while nearly every detection network (U-Net, UNet++, Attention U-Net, M_1) incorrectly classifies an instance of benign PCa as csPCa with varying degrees of confidence, certain networks (UNet++, Attention U-Net) also mistake BPH for the same. Upon closer inspection, we notice how both of these non-malignant conditions closely resemble the actual malignancy across all three channels of prostate mpMRI, reiterating the difficult challenge of discriminating csPCa (as seen in Fig. 1). Only two stand-alone detection networks (UNet++, M_1) are able to successfully identify the csPCa lesion, albeit with additional false positives. In the case of our proposed CAD system, while the classifier in $M_1 \otimes M_2$ suppresses these false positives in M_1, the inclusion of an anatomical prior further strengthens the confidence and boundaries of the true positive.

4.4. Population Prior vs Case-Specific Priors

Currently, our anatomical prior is used as a population prior that is paired with every input scan to the CAD system, in an identical manner. Although this technique holds the merit of requiring less resources (eg. additional zonal segmentations for cases beyond the training dataset) and shares an indirect, weak dependency on the accuracy of the external segmentation model, it suffers from a lack of adaptability to offer specialized priori as per the varying prostate anatomy observed across different patients. In future work, we aim to investigate the effect of transforming the population prior into case-specific anatomical priors using the zonal segmentations of each scan.

4.5. Effect of 3D vs 2D Systems

Analyzing the impact of utilizing the spatial context along the axial plane in prostate mpMRI, is a focus area that remains underexplored. Although our proposed 3D CAD system outperforms the pre-existing 2D CAD system being used by the Diagnostic Image Analysis Group at Radboud University Medical Center (refer to Appendix C), it is difficult to differentiate the additional contribution of the third spatial dimension from that of their architectural differences, without further experiments. In future work, we aim to reconstruct our proposed CAD system in 2D to examine and understand the potential benefits of using a 3D system, if any.

5. Conclusion

In summary, the study proposed an end-to-end CAD system for the automatic voxel-level detection of clinically significant prostate cancer in mpMRI scans. We carefully designed a novel multi-stage 3D CNN architecture that combines an anisotropic dual-attention detection network with a supplementary classifier to exploit independent false positive reduction at high detection sensitivities. We demonstrated that an anatomical prior, which draws upon the spatial prevalence of prostate malignancy and its zonal distinction, can be a powerful tool to further inform and generalize CNN performance in the medical domain. Our proposed model was evaluated on 487 institutional and 296 external testing scans annotated via PI-RADS v2 and histologically-assigned Gleason Scores, respectively. For the former testing set, it reached 83.95% and 89.94% detection sensitivity at 0.5 and 1.0 false positive per patient, respectively. For the latter testing set, the CAD system shared moderate agreement with expert radiologists (77.70%; $kappa$ = 0.543) and independent pathologists (78.04%; $kappa$ = 0.527), demonstrating a strong ability to generalize from training samples annotated using PI-RADS v2 only. The model also scored 0.884 and 0.857 AUROC in patient-based diagnosis on the institutional and external testings sets, respectively, outperforming five other candidate 3D CNN systems and verifying the design choices made throughout its end-to-end framework.

To the best of our knowledge, this was the first demonstration of a deep learning-based fully 3D approach to clinically significant prostate cancer detection in mpMRI, trained using radiologist-supported annotations only and evaluated on large, multi-institutional testing sets. The promising results of this research motivates the ongoing development of comprehensive CAD systems that can be seamlessly integrated into the clinical workflow, to minimize inter/intra-reader variability among radiologists, reduce unnecessary biopsies and aid the early detection of cancer.

Acknowledgements

The authors would like to acknowledge the contributions of Maarten de Rooij and Ilse Slootweg from Radboud University Medical Center, in creating fully delineated annotations of prostate malignancy for every mpMRI scan used in this study. Anindya Shaha is supported by the Erasmus+ KA1: EMJMD scholarship of the European Commission: Education, Audiovisual and Culture Executive Agency.

References

Aldoj, N., Lukas, S., Dewey, M., Penzkofer, T., 2020. Semi-automatic classification of prostate cancer on multi-parametric mr imaging using a multi-channel 3d convolutional neural network. European Radiology 30, 1243–1253. doi:10.1007/s00330-019-06417-z.

Alkadi, R., El-Baz, A., Taher, F., Werghi, N., 2019. A 2.5d deep learning-based approach for prostate cancer detection on t2-weighted magnetic resonance imaging, in: Computer Vision – ECCV 2018 Workshops, Springer International Publishing. pp. 734–739.

Basillote, J.B., Armenakas, N.A., Hochberg, D.A., Fracchia, J.A., 2003. Influence of prostate volume in the detection of prostate cancer. Urology 61, 167–171. doi:10.1016/S0090-4295(02)02103-9.

Cao, R., Mohammadian Bajgiran, A., Afshari Mirak, S., Shakeri, S., Zhong, X., Enzmann, D., Raman, S., Sung, K., 2019a. Joint prostate cancer detection and gleason score prediction in mp-mri via focalnet. IEEE Transactions on Medical Imaging 38, 2496–2506.

Cao, R., Zhong, X., Scalzo, F., Raman, S., Sung, K., 2019b. Prostate cancer inference via weakly-supervised learning using a large collection of negative mri, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 434–439.

Chen, M.E., Johnston, D.A., Tang, K., Babaian, R.J., Troncoso, P., 2000. Detailed mapping of prostate carcinoma foci: biopsy strategy implications. Cancer 89, 1800–1809.

Chihara, L.M., Hesterberg, T.C., Dobrow, R.P., 2014. Mathematical Statistics with Resampling and R & Probability: With Applicati. John Wiley & Sons. OCLC: 941516595.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Springer International Publishing. pp. 424–432.

Dalca, A.V., Guttag, J., Sabuncu, M.R., 2018. Anatomical priors in convolutional networks for unsupervised biomedical segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9290–9299.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Elwenspoek, M.M.C., Sheppard, A.L., McInnes, M.D.F., Whiting, P., 2019. Comparison of Multiparametric Magnetic Resonance Imaging and Targeted Biopsy With Systematic Biopsy Alone for the Diagnosis of Prostate Cancer: A Systematic Review and Meta-analysis. JAMA Network Open 2, e198427–e198427. doi:10.1001/jamanetworkopen.2019.8427.

Engels, R.R., Israël, B., Padhani, A.R., Barentsz, J.O., 2020. Multiparametric magnetic resonance imaging for the detection of clinically significant prostate cancer: What urologists need to know. part 1: Acquisition. European Urology 77, 457 – 468. doi:10.1016/j.eururo.2019.09.021.

Epstein, J.I., Egevad, L., Amin, M.B., Delahunt, B., 2016. The 2014 International Society of Urological Pathology (ISUP) Consensus

Conference on Gleason Grading of Prostatic Carcinoma: Definition of Grading Patterns and Proposal for a New Grading System. Am. J. Surg. Pathol. 40, 244–252.

Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., 2017. Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115–118. doi:10.1038/nature21056.

Fu, J., Liu, J., Tian, H., Lu, H., 2019. Dual attention network for scene segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3141–3149.

Garcia-Reyes, K., Passoni, N.M., Palmeri, M.L., Kauffman, C.R., 2015. Detection of prostate cancer with multiparametric mri (mpmri): effect of dedicated reader education on accuracy and confidence of index and anterior cancer diagnosis. Abdominal Imaging 40, 134–142. doi:10.1007/s00261-014-0197-7.

Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143, 29–36. doi:10.1148/radiology.143.1.7063747. pMID: 7063747.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks, in: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016, Springer International Publishing. pp. 630–645.

Hosseinzadeh, M., Brand, P., Huisman, H., 2019. Effect of adding probabilistic zonal prior in deep learning-based prostate cancer detection, in: International Conference on Medical Imaging with Deep Learning – Extended Abstract Track, London, United Kingdom.

Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E., 2019. Squeeze-and-excitation networks. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–1.

Israël, B., van der Leest, M., Sedelaar, M., Padhani, A.R., Zámecnik, P., Barentsz, J.O., 2020. Multiparametric magnetic resonance imaging for the detection of clinically significant prostate cancer: What urologists need to know. part 2: Interpretation. European Urology 77, 469 – 480. doi:10.1016/j.eururo.2019.10.024.

Jaeger, P.F., Kohl, S.A., Bickelhaupt, S., Maier-Hein, K.H., 2020. Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection, in: Machine Learning for Health Workshop, pp. 171–183.

Jiang, Z., Ding, C., Liu, M., Tao, D., 2020. Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task, in: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, Springer International Publishing. pp. 231–241.

Johnson, L.M., Turkbey, B., Figg, W.D., Choyke, P.L., 2014. Multiparametric mri in prostate cancer management. Nature Reviews Clinical Oncology 11, 346–353. doi:10.1038/nrclinonc.2014.69.

Kasivisvanathan, V., Rannikko, A.S., Borghi, M., Panebianco, V., 2018. Mri-targeted or standard biopsy for prostate-cancer diagnosis. New England Journal of Medicine 378, 1767–1777. doi:10.1056/NEJMoa1801993.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), Ithaca, NY: arXiv.org. URL: http://arxiv.org/abs/1412.6980.

Kooi, T., Litjens, G., [van Ginneken], B., Gubern-Mérida, A., Sánchez, C.I., Mann, R., [den Heeten], A., Karssemeijer, N., 2017. Large scale deep learning for computer aided detection of mammographic lesions. Medical Image Analysis 35, 303 – 312. doi:10.1016/j.media.2016.07.007.

Lemaître, G., Martí, R., Rastgoo, M., Mériaudeau, F., 2017. Computer-aided detection for prostate cancer detection based on multi-parametric magnetic resonance imaging, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 3138–3141.

Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007.

Litjens, G., Debats, O., Barentsz, J., Karssemeijer, N., Huisman, H., 2014. Computer-aided detection of prostate cancer in mri. IEEE Transactions on Medical Imaging 33, 1083–1092.

Luo, L., Xiong, Y., Liu, Y., 2019. Adaptive gradient methods with dynamic bound of learning rate, in: International Conference on Learning Representations.

McKinney, S.M., Sieniek, M., Godbole, V., Godwin, J., 2020. International evaluation of an ai system for breast cancer screening. Nature 577, 89–94. doi:10.1038/s41586-019-1799-6.

Miller, K.D., Nogueira, L., Mariotto, A.B., Rowland, J.H., Yabroff, K.R., Alfano, C.M., Jemal, A., Kramer, J.L., Siegel, R.L., 2019. Cancer treatment and survivorship statistics, 2019. CA: A Cancer Journal for Clinicians 69, 363–385. doi:10.3322/caac.21565.

Riepe, T., Hosseinzadeh, M., Brand, P., Huisman, H., 2020. Anisotropic deep learning multi-planar automatic prostate segmentation, in: Proceedings of the 28th International Society for Magnetic Resonance in Medicine Annual Meeting. URL: http://indexsmart.mirasmart.com/ISMRM2020/PDFfiles/3518.html.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing. pp. 234–241.

Rosenkrantz, A.B., Ginocchio, L.A., Cornfeld, D., Froemming, A.T., 2016. Interobserver reproducibility of the pi-rads version 2 lexicon: A multicenter study of six experienced prostate radiologists. Radiology 280, 793–804. doi:10.1148/radiol.2016152542.

Rouvière, O., Puech, P., Renard-Penna, R., Claudon, M., 2019. Use of prostate systematic and targeted biopsy on the basis of multiparametric mri in biopsy-naive patients (mri-first): a prospective, multicentre, paired diagnostic study. The Lancet Oncology 20, 100 – 109. doi:10.1016/S1470-2045(18)30569-2.

Sanford, T., Harmon, S.A., Turkbey, E.B., Turkbey, B., 2020. Deep-learning-based artificial intelligence for pi-rads classification to assist multiparametric prostate mri interpretation: A development study. Journal of Magnetic Resonance Imaging n/a. doi:10.1002/jmri.27204.

Schelb, P., Kohl, S., Radtke, J.P., Bonekamp, D., 2019. Classification of cancer at prostate mri: Deep learning versus clinical pi-rads assessment. Radiology 293, 607–617. doi:10.1148/radiol.2019190938.

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., Rueckert, D., 2019. Attention gated networks: Learning to leverage salient regions in medical images. Medical Image Analysis 53, 197 – 207. doi:10.1016/j.media.2019.01.012.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626.

Smith, C.P., Harmon, S.A., Barrett, T., Bittencourt, L.K., 2019. Intra- and interreader reproducibility of pi-radsv2: A multireader study. Journal of Magnetic Resonance Imaging 49, 1694–1703. doi:10.1002/jmri.26555.

Smith, L.N., 2017. Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472.

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press. p. 4278–4284.

van der Leest, M., Cornel, E., Israël, B., Hendriks, R., 2019. Head-to-head comparison of transrectal ultrasound-guided prostate biopsy versus multiparametric prostate resonance imaging with subsequent magnetic resonance-guided biopsy in biopsy-naïve men with elevated prostate-specific antigen: A large prospective multicenter clinical study. European Urology 75, 570 – 578. doi:doi.org/10.1016/j.eururo.2018.11.023.

Verma, S., Choyke, P.L., Eberhardt, S.C., Oto, A., Tempany, C.M., Turkbey, B., Rosenkrantz, A.B., 2017. The current state of

mr imaging–targeted biopsy techniques for detection of prostate cancer. Radiology 285, 343–356. doi:10.1148/radiol.2017161684.

Wachinger, C., Reuter, M., Klein, T., 2018. Deepnat: Deep convolutional neural network for segmenting neuroanatomy. NeuroImage 170, 434 – 445. doi:10.1016/j.neuroimage.2017.02.035. segmenting the Brain.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X., 2017. Residual attention network for image classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6450–6458.

Wang, H., Suh, J.W., Das, S.R., Pluta, J.B., Craige, C., Yushkevich, P.A., 2013. Multi-atlas segmentation with joint label fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 611–623. doi:10.1109/TPAMI.2012.143.

Weinreb, J.C., Barentsz, J.O., Choyke, P.L., Cornud, F., 2016. Pi-rads prostate imaging – reporting and data system: 2015, version 2. European Urology 69, 16 – 40. doi:10.1016/j.eururo.2015.08.052.

Xiao, C., Deng, R., Li, B., Yu, F., Liu, M., Song, D., 2018. Characterizing adversarial examples based on spatial consistency information for semantic segmentation, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision – ECCV 2018, Springer International Publishing. pp. 220–237.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995.

Yoo, S., Gujrathi, I., Haider, M.A., Khalvati, F., 2019. Prostate cancer detection using deep convolutional neural networks. Scientific Reports 9, 19518. doi:10.1038/s41598-019-55972-4.

Yu, X., Lou, B., Shi, B., Szolar, D., 2020. False positive reduction using multiscale contextual features for prostate cancer detection in multi-parametric mri scans, in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1355–1359.

Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2020. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE Transactions on Medical Imaging 39, 1856–1867.

Zlocha, M., Dou, Q., Glocker, B., 2019. Improving retinanet for ct lesion detection with dense masks from weak recist labels, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 402–410.

Appendix A. Network Configurations

We implement the proposed CAD model, including its CNN sub-models (M_1, M_2), using the TensorFlow Keras and Estimator APIs. Special care is taken throughout the design stage (as detailed in Section 3.2) to ensure computational efficiency. As such, the end-to-end 3D system is fully trainable and deployable on a single NVIDIA RTX 2080 Ti GPU (11 GB) in less than 6 hours.

3D Dual-Attention U-Net (M_1): The network architecture (as detailed in Section 3.2.1)) comprises of 75 convolutional layers. Layers are activated by ReLU or Leaky ReLU ($\alpha = 0.10$), and account for 15.25M trainable parameters in total. M_1 is initialized using He uniform variance scaling (He et al., 2015) and trained using $144 \times 144 \times 18 \times 4$ multi-channel whole-images over 40 epochs. It trains with a minibatch size of 2 and an exponentially decaying cyclic learning rate ($\gamma = 0.99995$, step size = 5 epochs) (Smith, 2017) oscillating between 10^{-6} and 2.5×10^{-4}. Focal loss ($\alpha = 0.75, \gamma = 2.00$) is used with Adam optimizer ($\beta_1 = 0.90$, $\beta_2 = 0.99$, $\epsilon = 10^{-5}$) (Kingma and Ba, 2015) in backpropagation through the model. Train-time augmentations include horizontal flip, rotation ($-7.5°$ to $7.5°$), translation (0-5% horizontal/vertical shifts) and scaling (0-5%) centered along the axial plane.

3D SEResNet (M_2): The network follows a relatively shallow 3D adaptation of the SEResNet architecture proposed by Hu et al. (2019), comprising of two residual blocks with six convolutional layers each. Layers are activated by ReLU and account for 84.68K trainable parameters in total. M_2 is initialized using He uniform variance scaling (He et al., 2015) and trained using $64 \times 64 \times 8 \times 3$ multi-channel octant patches over 262 epochs. It trains with a minibatch size of 80 (equivalent to 10 full scans) and an exponentially decaying cyclic learning rate ($\gamma = 0.99995$, step size = 5 epochs) (Smith, 2017) oscillating between 10^{-6} and 2.5×10^{-4}. Balanced cross-entropy loss ($\beta = 0.80$) is used with AMSBound optimizer ($\gamma = 10^{-3}$, $\beta_1 = 0.90$, $\beta_2 = 0.99$) (Luo et al., 2019) in backpropagation through the model. Train-time augmentations include horizontal flip, rotation ($-10°$ to $10°$), translation (0-10% horizontal/vertical shifts) and scaling (0-5%) centered along the axial plane.

Appendix B. Pre-Existing 2D Detection System

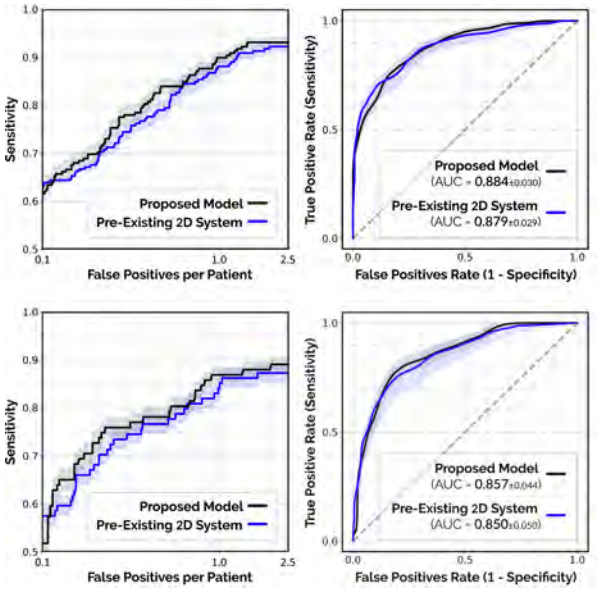


Figure 12: Lesion-level FROC ***(left)*** and case-level ROC ***(right)*** analyses using the proposed 3D CAD system and the pre-existing 2D CAD system at Radboud University Medical Center, on the testing sets TS1 ***(top)*** and TS2 ***(bottom)***. Transparent areas indicate the 95% confidence intervals estimated as twice the standard deviation from 1000 replications of bootstrapping. The 2D CAD system is based on a refined U-Net architecture that uses an early fusion of probabilistic zonal priors from a 3D segmentation model, to boost overall detection and diagnosis performance (Hosseinzadeh et al., 2019).

Appendix C. RetinaNet (*Discontinued*)

An earlier iteration of this research revolved around the application of state-of-the-art adaptations of RetinaNet (Lin et al., 2017) that are especially suited for medical object detection, such as the improved RetinaNet and Retina U-Net (Jaeger et al., 2020; Zlocha et al., 2019). Both of these models, along with the original RetinaNet, were implemented and employed for csPCa detection. Despite numerous trials, the models failed to converge. In the absence of pre-initialized weights, object detection models based on feature pyramid networks require a large number of samples for end-to-end training. This is why RetinaNets are generally deployed for tasks associated with large datasets of high-resolution medical images, such as mammograms or computed tomography (CT) exams (Jaeger et al., 2020; McKinney et al., 2020; Zlocha et al., 2019). Moreover, as per its default configuration, negative scans are discarded during its training cycle, further shrinking the available dataset. Thus, we presume that the limited number of *malignant* training samples in our dataset, paired with the low-resolution nature of prostate mpMRI, are the primary reasons why the standard RetinaNet and its derivative architectures could not be employed effectively for csPCa detection. To test this theory, we trained the same architecture for the simpler task of prostate detection using the same dataset. As opposed to less than 3000 slices of *malignant* cases, we could now use nearly 21000 slices of prostate glands as our training samples. In this case, we observed that all candidate RetinaNet architectures converge with high detection rates (as seen in Fig. 13), thereby supporting our claim. Furthermore, to the best of our knowledge, there are currently no available studies that employ the RetinaNet for csPCa detection in mpMRI. Hence, this approach was discontinued due to time constraints, and shelved for a future study.

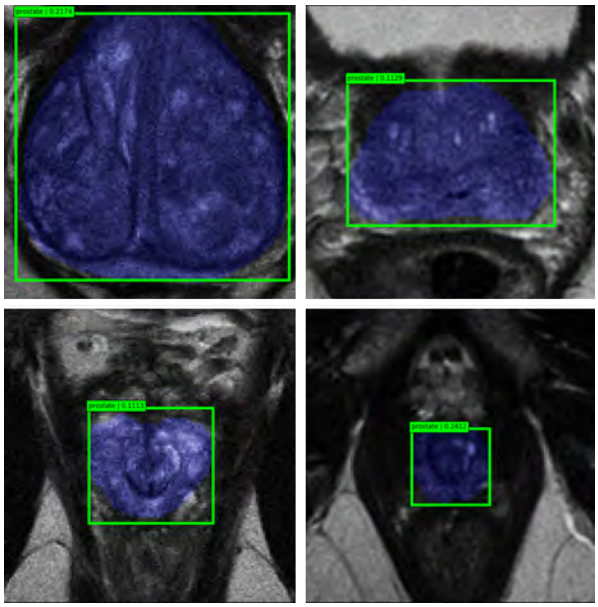
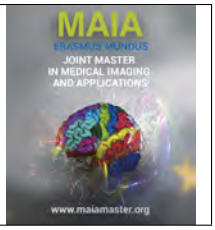


Figure 13: Bounding boxes for different scales and orientations of the prostate as predicted by the 2D RetinaNet ***(green)***, versus the segmentation ground-truth ***(blue)*** on T2W slices of four different validation patient cases.

Medical Imaging and Applications

Master Thesis, August 2020



Unsupervised 3D Brain Anomaly Detection

Jaime Simarro Viana, Ezequiel de la Rosa, Diana M. Sima

*ico**metrix**, Leuven (3000), Belgium.*

Abstract

Anomaly detection (AD) is the identification of data samples that do not fit a learned data distribution. As such, AD systems can help doctors to determine the presence, severity, and extension of a pathology. Deep generative models, such as Generative Adversarial Networks (GANs), can be exploited to capture the anatomical variability. Consequently, any outlier data (i.e., samples falling outside of the learned distribution) can be detected as an abnormality in an unsupervised fashion. By means of this methodology, we can not only detect expected or known lesions, but we can even unveil new disease-specific biomarkers. In this work we propose, to the best of our knowledge, the first AD approach able to efficiently handle volumetric data and detect 3D brain anomalies in one single model. Our proposal is a volumetric and high-detail extension of the 2D f-AnoGAN model obtained by combining a state-of-the-art 3D GAN with refinement training steps. In experiments using non-contrast computed tomography images from traumatic brain injury (TBI) patients, the model detects and localizes TBI abnormalities with an area under the ROC curve of ~75%. Moreover, we test the potential of the method for detecting other kind of anomalies such as low quality images, preprocessing inaccuracies, artifacts, and even the presence of post-operative signs (such as a craniectomy or a brain shunt). The method has potential for rapidly screening massive imaging data, for helping in abnormalities labeling, as well as for new biomarkers identification.

Keywords: Unsupervised learning, Anomaly detection, Deep generative networks, 3D GAN

1. Introduction

Nowadays, two main learning techniques are leading medical imaging research: *supervised learning* and *unsupervised learning*. A learning process is considered *supervised* if by observing examples from a labeled database, the model learns a mapping function from inputs to outputs through an explicit feedback (Stuart et al., 2003). In the past few years, supervised deep learning techniques have shown an outstanding performance in a wide diversity of medical imaging tasks. Supervised models have even outperformed radiologists in a clinical setting such as breast tumor identification (Stower, 2020) or lung cancer detection (Ardila et al., 2019). Although this learning process seems promising, it is limited by requiring large and manually annotated databases, which are expensive and time-consuming, hardly reproducible and prone to reader's bias. Furthermore, annotations are disease-specific and potentially incomplete; i.e., in most pathologies, the gamut of all possible lesions or abnormalities is not available. In consequence, supervised techniques are highly limited for abnormality screening in clinical practice (i.e., for recognition of abnormal anatomical configurations).

In contrast, *unsupervised* learning models are capable of discovering patterns from label-free databases. A current challenge in this field, involving extensive efforts from the research community, is unsupervised *anomaly detection (AD)*. AD is the task of identifying test data that is not fitting the data distributions seen during training (Schlegl et al., 2017). In clinical practice, AD represents a crucial step. Medical doctors learn the normal anatomical variability and recognize anomalies by their comparison against either normal cases or healthy surrounding areas. Most of the proposed AD models detect anatomical abnormalities by mimicking this human behaviour. In this way, any abnormality can be detected in an unconstrained fashion. Since only normal anatomy is needed for training, this strategy plays a crucial role in the detection of rare pathologies where

collecting training data is very challenging.

2. State of the art

Classical unsupervised learning models have captured the anatomical variability using statistical intensity features (such as histogram information) or texture-based features (such as grey level co-occurrence matrices) (Taboada-Crispi et al., 2009). Owing to the fact that medical image analysis involves a highly dimensional space, most of the recent approaches have been focused on reducing the dimensionality while taking into account the inter- and intra-subject variability occurring in the healthy population. For example, Sidibe et al. (2017) reduced the data dimension by applying PCA over optical coherence tomography images from healthy subjects.

Deep generative models can also be exploited for dimensionality reduction. These models have the ability to compress the data in a semantically rich space: the *latent space*. Note that literature also calls the latent space as *bottle neck* or *z-space*.

2.1. Deep generative models

Deep generative models, such as variational autoencoders (Kingma and Welling, 2013), generative adversarial networks (Goodfellow et al., 2014), and variants of these models have been widely used for image synthesis. These models are able to generate synthetic images by capturing the variability of the trained images. This is a very interesting feature for AD because, if a deep generative model is trained over data from healthy subjects, anomalies could be discovered by detecting samples that do not fit the normal healthy variability.

Autoencoder based approaches

Autoencoders (AE) are composed of two networks: an encoder that compresses an image into the latent space and a decoder that creates an image from this input (see Figure 1-a). Pawlowski et al. (2018) used Bayesian autoencoders to detect traumatic brain injury lesions in computed tomography (CT) mid-axial slices. The uncertainty of the AE was modeled using Monte Carlo Dropout. In Vaidhya et al. (2015), 3D patch-based Stacked Denoising Autoencoders were used as a pretraining step for supervised brain tumor segmentation on magnetic resonance images. Sato et al. (2018) trained an AE using 3D patches to target disorders of emergency head CT volumes. In addition, Seeböck et al. (2016) compressed the information with an AE and later on they used a one-class support vector machine to distinguish between normal and anomalous retina patches.

Variational autoencoders (VAE) share the same network architecture with AE. However, their latent space is constrained to a standard Gaussian distribution (i.e. $z \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$) by minimizing Kullback-Leibler divergence loss. In this way, the latent space is better controlled, since it cannot fall in any random space but just within the predefined ranges. Despite their ability for high resolution reconstructions, VAEs suffer from memorization and tend to produce blurry images (Baur et al., 2018). Addressing these issues, Larsen et al. (2016) proposed the VAEGAN architecture, which combines spatial VAE and an adversarial network (see Figure 1-b). This additional adversarial network allows generating sharper and more realistic images. Baur et al. (2018) used the VAEGAN architecture for delineating multiple sclerosis lesions in 2D brain MR images. A similar idea of combining both architectures is found in Adversarial autoencoders (AAE) (Makhzani et al., 2015) (see Figure 1-c). Chen and Konukoglu (2018) exploited the potential of the latent space of AAE to detect lesions in axial brain MRI slices. In a recent work, structural abnormalities in patients with brain metastasis were discovered using a VAE-based architecture. This network introspectively self-evaluates the differences between the input and reconstructed images to self-update (Kobayashi et al., 2020).

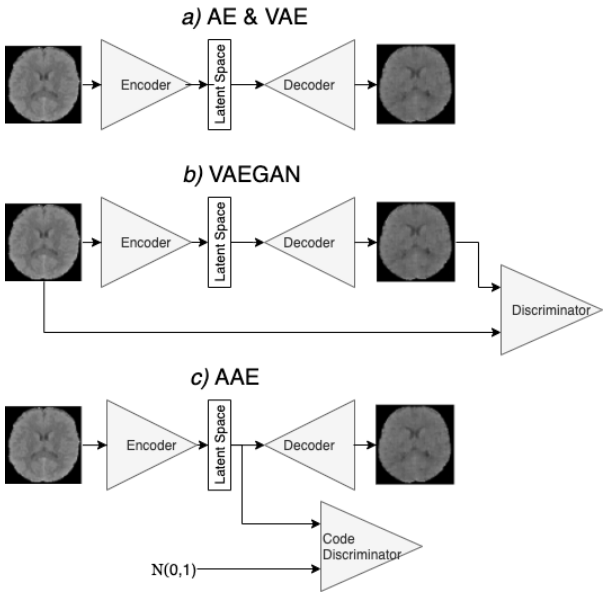


Figure 1: Autoencoder based architectures.

GAN based approaches

Generative adversarial networks (GANs) are composed of two networks: a generator and a discriminator. The disruptive idea that a GAN trained on images from healthy subjects should not be able to reconstruct abnormalities was proposed by Schlegl et al. (2017) to detect retina anomalies. When a GAN is trained, the latent space (which is obtained by randomly sampling from a standard Gaussian distribution) is input to the generator. In order to find the reconstruction of a

given image using the GAN network, we need to know the projection of this image in the GAN's latent space. Schlegl et al. (2017) used a slow iterative optimization algorithm to find the ideal projection. In f-AnoGAN, Schlegl et al. (2019) updated their research by replacing the iterative optimization algorithm by an encoder network and consequently, by reducing the inference time of the previous approach. In spite of sharing the same network architecture of VAEGAN, the training of f-AnoGAN is completely different. In a first step, the GAN networks (i.e., the generator and the discriminator) are trained. In a second step, the encoder trains while the GAN's weights remain frozen. Figure 2 shows f-AnoGAN training phases.

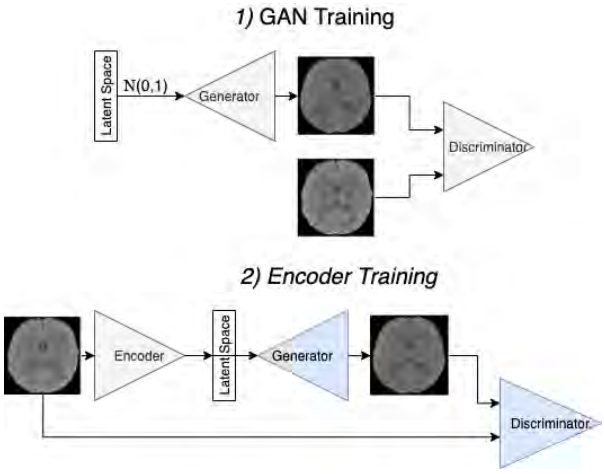


Figure 2: F-AnoGAN training phases. Networks in blue do not change their weight during this training phase

In a very recent deep generative comparative study, Baur et al. (2020) reported a good f-AnoGAN performance in diverse datasets. In this survey, only a VAE with a restoration of the latent space (You et al., 2019) was able to slightly outperform f-AnoGAN. However, this approach is based on an iterative approach that takes minutes for a single MR image, turning unfeasible for large volumetric screening.

Despite the potential of f-AnoGAN, Baur et al. (2020) suggest that this method does not preserve anatomical coherence between the input and the reconstruction. Probably, it is mainly caused by the training approach: use of independent 2D axial slices of each volume. The generator network is expected to reproduce the variability between axial slices plus the anatomical variability in healthy subjects per slice. Training the generator to cover this huge variability is very complex and hardly reachable.

2.2. Limitations of the current methods

In brain imaging, most recent works have been focused on anomaly detection using 2D axial brain images (Chen and Konukoglu, 2018), (Pawlowski et al., 2018), (Baur et al., 2020). All of these 2D-based approaches have several drawbacks: i) they do not consider volumetric information and consequently, they do not effectively handle the brain anatomy, which is deeply complex; ii) there is no prior information of the anomaly localization so AD systems must consider information from the whole brain image at once. As Han et al. (2019) underlines *"we can not discriminate diseases composed of the accumulation of subtle anatomical anomalies, without considering continuity between multiple adjacent slices"*; iii) training multiples models, i.e., one per batch of slices or patches is extremely inefficient, time-consuming and suboptimal when compared to the usage of a single model to generate an entire 3D volume.

It must be noticed that, besides deep generative models, also other strategies, such as Siamese networks (Alaverdyan et al., 2020) or Bayesians U-Net (Seeböck et al., 2019), use two-dimensional information in brain anomaly detection, thus suffer from not using information from the whole brain at once before performing AD.

2.3. Contributions of this work

To overcome these limitations, we propose, to the best of our knowledge, the first *3D brain anomaly detector*. This model effectively handles the complex brain structures and provides reliable 3D reconstructions based on normal brain anatomy. Moreover, we do not just prove the AD capability of the model in two independent traumatic brain injury datasets, but also we propose to use our AD model for quality control. Its AD potential is assessed in post-surgical anomalies (such as brain shunt or craniectomy) and poor quality images (such as the presence of artifacts, bad registrations, or bad orientations).

The present work is inspired by the recently proposed f-AnoGAN technique (Schlegl et al., 2019). However, the proposed methodology differs in several aspects from this AD methodology: i) the network learns from *volumetric information* creating densely new reconstructions image; ii) the networks architecture is updated by using a modified version of the state-of-the-art 3D GAN used in (Kwon et al., 2019); iii) a new training step is proposed to surpass the lack of details; iv) the model detects brain anomalies instead of retina lesions.

3. Material

3.1. Database

Traumatic brain injury (TBI) is the leading cause of mortality in young adults and a major cause of death and disability (Hyder et al., 2007). Non-contrast computed tomography (NCCT) is worldwide used to assess TBI, mainly because it allows fast acquisition of images. TBI includes a vast spectrum of pathoanatomic

findings, with some of them more common than others. Evaluating these abnormalities on NCCT is often complex and challenging, leading to notable inter-observer variation (Laalo et al., 2009). TBI englobes a wide variety of abnormalities for testing AD systems, while these AD systems have the potential to be a perfect screening tool to help the doctors understand the extent of TBI and decide on patient management. In order to ensure the reliability of the proposed methodology, the model is tested using two independents datasets:

CENTER-TBI. The collaborative European NeuroTrauma Effectiveness Research in Traumatic Brain Injury (CENTER-TBI) project included a database collection of NCCT images (Maas et al., 2015). The study protocol was approved by the national and local ethics committees for each participating center. Informed consent, including use of data for other research purposes, was obtained in each subject according to local regulations. Patient data was de-identified and coded by means of a Global Unique Patient Identifier. In this multicenter, multi-scanner, longitudinal study, all the NCCT images of TBI patients were visually reviewed and the abnormal findings were reported in a structured way by an expert panel. We retrieved a selection of images from a centralized imaging repository that stores the data collected and sent by the different sites. This dataset includes brain images without NCCT abnormal findings by expert review (n = 637 total scans) and manually annotated TBI scans (n = 102) with abnormal NCCT findings Jain et al. (2019). For the latter subset of images, the following lesions were manually annotated (Jain et al., 2019) and could serve as ground truth for AD: *epidural hematoma, subdural hematoma, and intraparenchymal (contusion and intracerebral) hematoma*.

PhysioNet. For further analysis and comparison, the model is also tested in a public database, available online at the PhysioNet repository[1] (Goldberger et al., 2000). This database is the result of a retrospective study where two radiologists annotated by consensus NCCT scans (Hssayeni et al., 2020). This dataset is composed by *epidural hematoma, subdural hematoma, intraventricular, intraparenchymal and subarachnoid* lesions. The performance of the model is validated using 37 normal subjects and 33 TBI patients.

The training of the model is performed over $\sim$ 80% (n = 532) of the CENTER-TBI data without abnormal NCCT findings. As test sets we use CENTER-TBI (remaining 20%, n = 105 and all TBI cases with abnormal findings) and PhysioNet (entire database).

[1] `https://physionet.org/content/ct-ich/1.3.1/`

4. Methods

The proposed methodology is based on three steps. Firstly, the misleading variability found in raw NCCT images is reduced through a preprocessing procedure. Secondly, a three-step training strategy is followed by only using images from patients without abnormal imaging findings. This training allows to not only capture the normal brain variability as seen on NCCT, but also to reconstruct a given image based on this learned variability. Finally, an anomaly score is calculated in order to classify a test image as inlier (normal) or outlier (anomalous).

4.1. Preprocessing

We propose a robust preprocessing methodology based on the following steps:

1. NCCT raw images are registered to the MNI space with an affine transformation using a uniform voxel resolution.
2. An automatic quality control process is performed using the FDA approved ico**brain** TBI software. Images highly corrupted by artifacts, missing the full head coverage, or even unfeasible to register to the MNI space are automatically discarded.
3. Using the same software tool, a skull-stripping operation is performed.
4. Keeping the images aligned, uninformative boundaries caused by the application of the brain mask are removed.
5. Images are resized to 64x64x64 due to constraints in memory of the graphics processing unit. Before the down-scaling using linear interpolation, aliasing artifacts are avoided by smoothing the image with a Gaussian filter.
6. We ensure that soft tissues have a good intensity representation by applying a windowing within the range -20 to 100 Hounsfield units, as similarly proposed in (Monteiro et al., 2019).
7. The images are min-max normalized between -1 and 1. In order to preserve the physical intensity relation between images provided from the Hounsfield units, the same min-max values are used in all the images.

A flowchart of these preprocessing steps is shown in Figure 3.

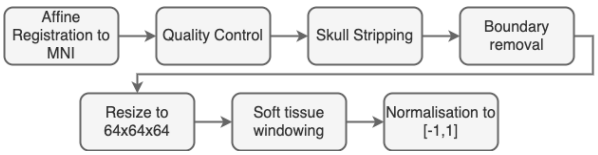


Figure 3: Preprocessing flow chart

4.2. Model architecture

The 3D GAN architecture proposed in Kwon et al. (2019) is used as foundation, while minor changes are performed over this architecture. Figure 4 exemplifies the network architecture.

The discriminator and the encoder share almost the same architecture based on five 3D convolutional layers. Following each convolutional operation, batch normalization (BatchNorm) and leaky rectified linear units (LeakyReLU) are used. However, in the first and the last layers, the batch normalization layers are removed. Both networks differ in the last activation layer: in the case of the encoder a hyperbolic tangent (tanh) is empirically set, while in the case of the discriminator a no-activation is required for matching the objective function (Equation 1).

In the generator network, instead of deconvolution layers, resize-convolutions are used in order to reduce the checkerboard artifacts (Odena et al., 2016). Moreover, BatchNorm and ReLU are used in each layer, except in the last one where tanh is preferred in order to match the original input range.

4.3. Training strategy

Following the idea proposed in f-AnoGAN, we have trained our networks through the following consecutive steps:

GAN training

The GAN training strategy is based on a competitive game between two networks: the *generator* network (G), which captures the data distribution by mapping input noise variable to the data space, and the *discriminator* network (D), which estimates the probability that a sample comes from the real data distribution rather than being generated.

During training, G maximizes the probability of D making a mistake, while D maximizes the probability of correctly predicting the real and generated samples. Formally, this two-players minimax game has the following objective function:

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log(1 - D(\tilde{x}))] \quad (1)$$

where $\mathbb{P}_r$ is the real data distribution and $\mathbb{P}_g$ is the model distribution defined by $\tilde{x} = G(z)$. The input z of G is sampled from a Gaussian distribution.

In practice, $\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log(1 - D(\tilde{x}))]$ can saturate the D in early learning, often leading to a vanishing gradient problem. Instead, we can obtain much stronger gradients by training the G to minimize $-\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log D(\tilde{x})]$ (Goodfellow et al., 2014). Thus, we obtain the following objective function:

$$\min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log D(\tilde{x})] \quad (2)$$

Wasserstein GANs (WGANs). Arjovsky et al. (2017) proposed to use Wasserstein-1 (also called Earth-Mover) distance to measure how close the model distribution and the real distribution are. Wasserstein distance not only improves the learning stability but also the WGAN value appears to correlate with the sample quality.

We can minimize $W(\mathbb{P}_r, \mathbb{P}_g)$ by using the following objective function:

$$\min_G \max_{D\epsilon\mathtt{D}} \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \quad (3)$$

where D is the set of 1-Lipschitz function, required to assume that $W(\mathbb{P}_r, \mathbb{P}_g)$ is continuous everywhere and differentiable almost everywhere. Arjovsky et al. (2017) enforces the Lipschitz constraint by clipping the weights of the discriminator within a compact space $[-c, c]$.

Gradient penalty in WGANs. Gulrajani et al. (2017) alternative enforces the Lipschitz constraint by constraining the gradient norm discriminator's output with respect to its input. Adding gradient penalty to our discriminator loss increases training stability. The resulting loss function in hence as follows:

$$\begin{aligned} L_{Discriminator} = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] \\ +\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2] \end{aligned} \quad (4)$$

$$L_{Generator} = -\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \quad (5)$$

where $\mathbb{P}_{\hat{x}}$ is sampled uniformly along straight lines between a pair of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$ and λ a weighting parameter.

Encoder projection

Once the adversarial training is completed, the G knows how to map from the latent space z to an image $\tilde{x}$, $G(z) = z \rightarrow \tilde{x}$. However, GANs can not perform the inverse mapping $E(x) = x \rightarrow z$. In other words, we do not know the representation and location of a given image in the latent space.

AE-based approaches, on the contrary, can map new images to the latent space. However, as the training of all the model's networks is performed in an end-to-end fashion, there is a need to predict the distribution of values in the latent space. VAE and AAE gain control over the latent space by constraining it to a Gaussian standard distribution. Differently, our approach does not require constraining the latent space. While the *encoder* network (E) is training, the weights of G and D remain frozen and only E weights are optimized. Consequently, E learns that in order to make realistic synthetic images, the approximation of the distribution $z \sim \mathcal{N}(\mu = 1, \sigma^2 = 1)$ is crucial. No remarkable differences are found when different activations are considered (as tahn, scaled tahn or no activation function).

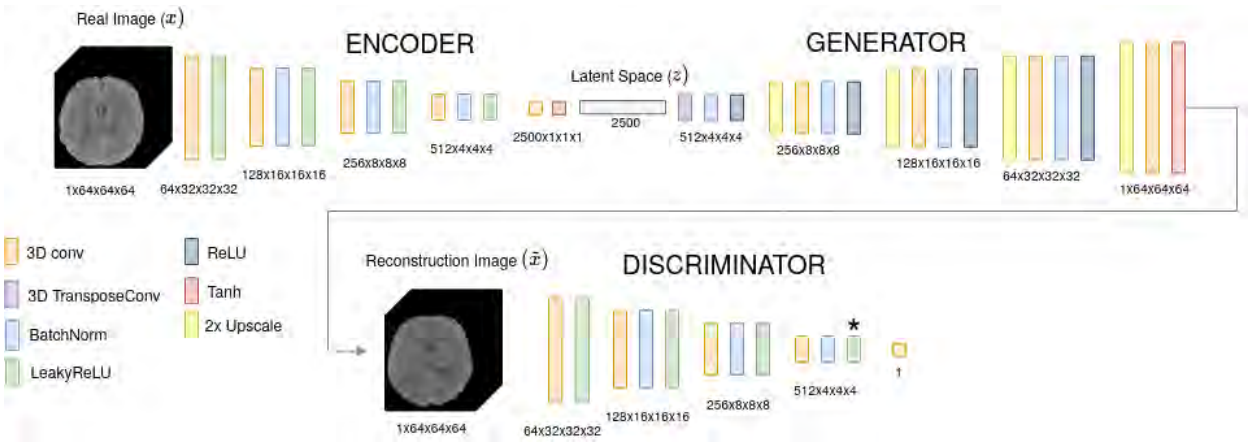


Figure 4: Architecture details. Dimensions of each feature maps are shown for each block. The asterisk (*) denotes the intermediate layer of the D used in $f(x)$

These results show that the E network exploits a proper latent space representation and, therefore, G outcomes proper reconstructions without requiring a forced constraint over z. Thanks to this methodology, the complexity of the training is reduced as a consequence of increasing the number of steps.

Encoder loss

A proper reconstruction would match the original image not only in intensity terms but also in semantically rich feature terms. Therefore, the E weights are optimized by minimizing a combination of these two losses: *the image space loss* and *the discriminator feature space loss*.

Image space loss. We reduce the visual differences by minimizing the mean squared error of input images x and reconstructed images $\tilde{x} = G(E(x))$. Equation 6 shows this pixel-wise mean difference commonly used in AE architectures:

$$L_{img} = \frac{1}{n} \|x - \tilde{x}\|^2 \tag{6}$$

where n is the number of voxels in the image.

Discriminator feature space loss. After the GAN training, the D network is able to differentiate the model distribution $\mathbb{P}_g$, which approximate the real data distribution $\mathbb{P}_r$, from not realistic images. Therefore, this network has created a low dimension but very informative feature space where its decision is supported. This idea was suggested by Schlegl et al. (2017) and it is inspired by the feature matching technique (Salimans et al., 2016). Letting $f(x)$ to denote the activation on an intermediate layer of the D, the discriminatory feature space loss is defined as follows:

$$L_{feat} = \frac{1}{m} \|f(x) - f(\tilde{x})\|^2 \tag{7}$$

where m is the dimensionality of the discriminator feature space.

By minimizing this loss in the E's training stage, we force this network to cooperate with the G to create similar features like the training data by using this rich intermediate activation layer. Notice that asterisk (*) in Figure 4 shows the $f(x)$ feature space used in training.

The final loss function which is used to train the E is the following:

$$L_{Encoder} = L_{img} + \kappa \cdot L_{feat} \tag{8}$$

where κ is a weighting parameter.

Techniques for improving the performance

After preliminary experiments with this training strategy, a lack of image details is observed. The complexity for training a 2D GAN considerably increases when training, instead, a 3D GAN for volumetric data generation. Therefore, we propose a new learning step that provides explicit learning feedback of the vast information that a 3D image contains. It is important to remark that the pretrained weights of the E are already able to project the images in proper regions of the latent space. In consequence, this extra training step provides a fine-tuning of the networks weights rather than a full model initial training.

We evaluate the effect of only optimizing the following networks while keeping the remaining ones frozen: i) generator; ii) encoder and generator networks (similar structure as a traditional AE training but with a different loss function); iii) combination of both AE and GAN in a fully end-to-end training of these networks: encoder, generator, and discriminator.

4.4. Anomaly score

The anomaly score quantifies the deviation of query images and corresponding reconstructions (Schlegl et al., 2019). Note that the reconstructions are generated

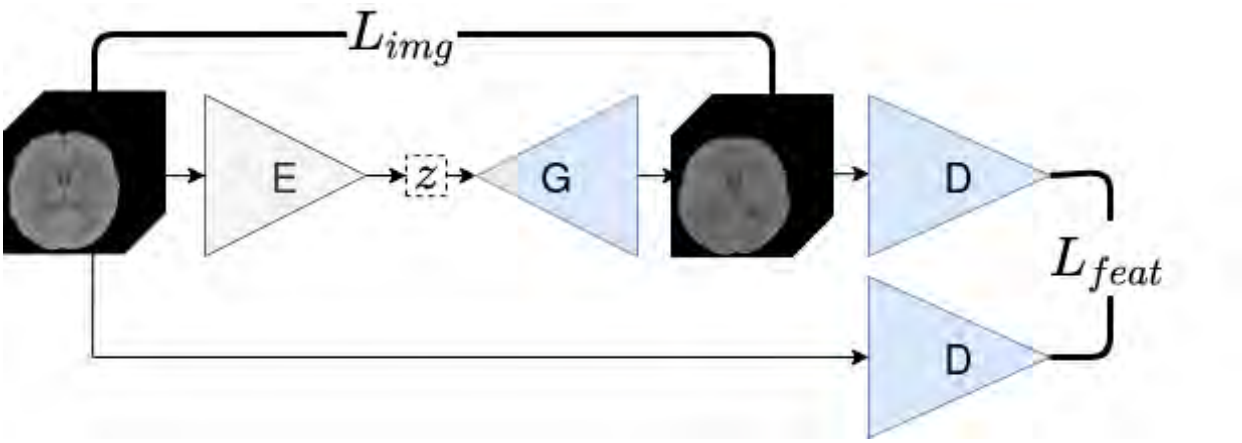


Figure 5: Image space loss and discriminator feature space loss. Networks in blue do not change their weights during encoder training phase.

by considering just the distribution of the normal data used for training, so we can understand this anomaly score as a distance metric between the input image and the learned normal variability. The anomaly score for a given image is obtained using the loss function shown in Equation 8. Thresholding the anomaly score provides a global classification of an image as abnormal or not. AD performance can thus be evaluated using an ROC analysis of the anomaly score.

4.5. Statistical Analysis

We use Receiver Operating Characteristic (ROC) curves and the corresponding area under the curve (AUC) to evaluate anomaly detection performance. Moreover, we also test the model's detection ability for different TBI lesions. Under heavy class imbalance (as happens when separately considering the different lesion types), ROC curves can be misleading as the higher frequency class has a much higher weight. To assess the anomaly performance under these circumstances we use Precision Recall curve and we calculate the average precision (AP). In addition, we plot iso-curves of the F1 score to facilitate performance comparison.

5. Results

In this section, we firstly show the AD improvement when 3D volumes are considered rather than 2D axial slices. Secondly, we give some empirical guidance needed to obtain the optimal model. Thirdly, we test the AD performance of our model in different datasets and TBI lesions types. The lesion localization capability of our methodology is also shown. Finally, we exemplify the potential use of the model for new biomarkers discovery.

Comparison with 2D f-AnoGAN

F-AnoGAN was primarily proposed for detecting retina lesions in 2D optical coherence tomography images. However, we use this methodology for 2D brain slices. As training one GAN per batch of slices is unfeasible, we train our 2D network in randomly selected middle axial slices. It must be noted that the preprocessing pipeline is also applied, ensuring that similar anatomical structures are shown in the selected slices. One of the main limitations of this model is to detect an abnormal brain when the abnormality is not present in a middle axial slice. Thus, in order to have a fair comparison, images that do not contain any abnormal pixel in the examined slice are removed. Despite this advantage, the 3D model outperforms the 2D network by 4% more of AUC. Possibly, the 2D model is not able to properly handle the complex volumetric brain structures.

Impact of the training strategy

We avoid collapse mode while training the GAN by updating more frequently the D parameters than the G ones. Moreover, stability in GAN's game is obtained by avoiding sparse gradients with the use of Wasserstein GAN with gradient penalty.

Experiments shows that both terms, L_{img} and L_{feat}, in the $L_{Encoder}$ (see Equation 8) are relevant. However, the L_{feat} provides a more robust information of the reconstruction results. In Table 1 is shown the influence of the parameter κ over the training performance. In addition, we also evaluate the similarity of the image and its reconstruction in the latent space as it is shown in Equation 9. No major differences where found when this loss is added, so L_{latent} is not used to optimize the encoder weights.

$$L_{latent} = \frac{1}{l} \|E(x) - E(\tilde{x})\|^2 \tag{9}$$

where l is the dimensionality of the latent space.

κ	0	0.1	1	**10**	100
AUC(%)	68.02	68.45	68.93	**71.73**	70.34

Table 1: Influence of the parameter κ in $L_{Encoder}$

After this training procedure, the image quality could increase by optimizing all the networks through the explicit feedback, i.e., optimizing all the network's weights by minimizing the $L_{Encoder}$. So, starting from the pretrained weights and while the remaining networks weights are frozen, we minimize the $L_{Encoder}$ by only training the following networks: i) the generator; ii) the encoder and the generator; iii) we also explore a combination of this training loss with the GAN training by optimizing the weights of the encoder, the generator and the discriminator. Table 2 shows the experiment results over the CENTER-TBI dataset.

Network trained	***AUC (%)***
None (pretrained weights)	71.73
Generator	74.05
Encoder & Generator	***74.92***
Encoder & Generator & Discriminator	70.11

Table 2: Training comparative of differences post-training techniques.

Anomaly detection performance

In order to test the generalization capabilities of the model, we compare the anomaly detection performance over two independents datasets: CENTER-TBI and PhysioNet. It should be underlined that the PhysioNet dataset has not been used in any previous step to train or to test. Figure 6 shows the ROC curves for each dataset and when both are combined. As we can appreciate the proposed methodology performs similarly in both datasets.

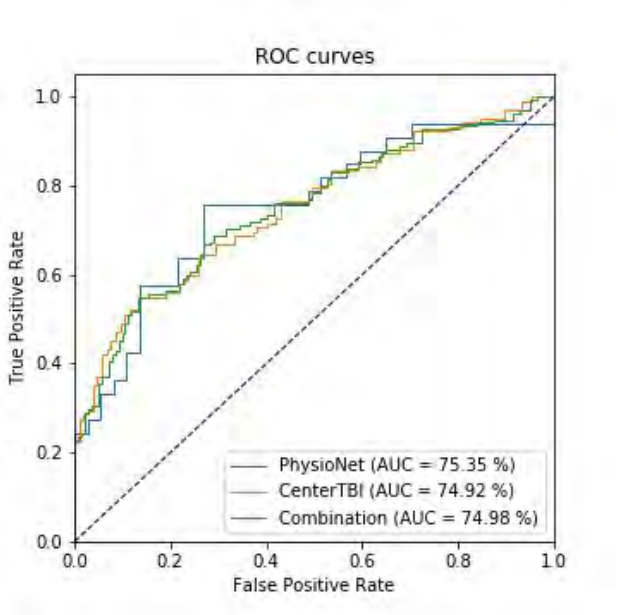


Figure 6: Comparison of ROC curves for the different datasets. AUC: Area under the ROC curve

Table 3 shows the performance statistics calculated at the Youden index of the ROC curve over the combined datasets. We can appreciate that at this operating point, the model has an outstanding specificity performance (also called true negative rate) in contrast with the recall performance (also called sensitivity or the true positive rate). Though a more balanced operative point can easily be found, we discard this kind of experiments given that are application-specific. Thus, the operating point could be better defined depending on the different clinical criteria. For this reason, we prefer using ROC curves to have a general overview of the proposed method's performance.

Accuracy	70.75%
Recall	54.07%
Specificity	86.66%
F1-score	64.32%

Table 3: Performance statistics calculated at the Youden index of the ROC curve

Detection performance over different lesion types

There is a big variability of lesion types in the TBI spectrum. The proposed 3D anomaly detector model should ideally be able to detect all of them since their appearance differs from normal brain variability in subjects without abnormal NCCT imaging findings. Subjects from both datasets with at least one of the following hematomas are used to evaluate the performance of the model: *epidural, subdural, and intraparenchymal*. Note, that if the model detects an anomalous case having one of these lesions, this is counted as a detection, no matter if other lesions are also present. Figure 7 shows the ability of the model for detecting these TBI lesions. In addition, the average precision (AP) combined with the iso-F1 curves are also included in the comparison. The model slightly changes the performance for different TBI lesions, having a better performance in detecting subdural hematomas.

Impact of the lesion volume in the AD score

The *anomaly score* provides a similarity measurement between the new given image and the learned training database. We investigate whether bigger lesions could have higher anomaly scores, or whether the method is independent of the lesion volumes. Figure 8 shows the anomaly score as a function of the lesion. Since a non-linear relationship is found, we measure the relationship using Spearman correlation, which achieves a ρ=0.65. Thus, there is a monotonically increasing relationship between the volume of the lesion and the anomaly score. Moreover, the non-parametric LOWESS (locally-weighted scatterplot smoothing) regression method shows the nonlinear relationship between the variables (see Figure 8).

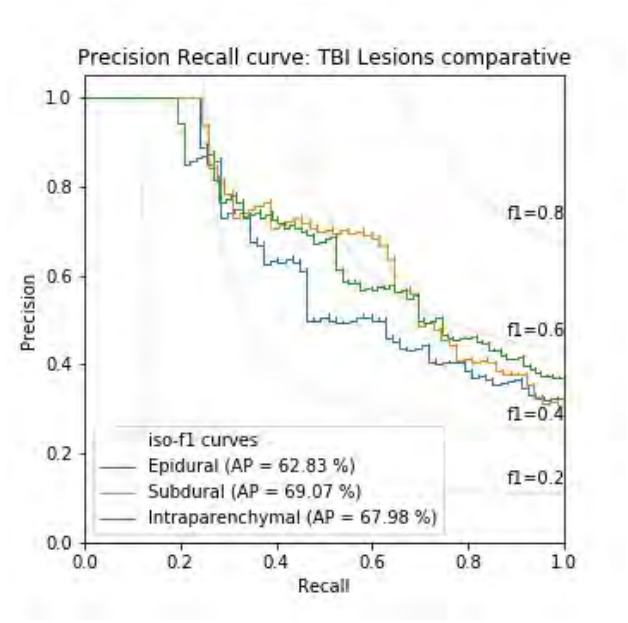


Figure 7: Comparison of Precision Recall curve for each TBI lesion. AP: Average precision

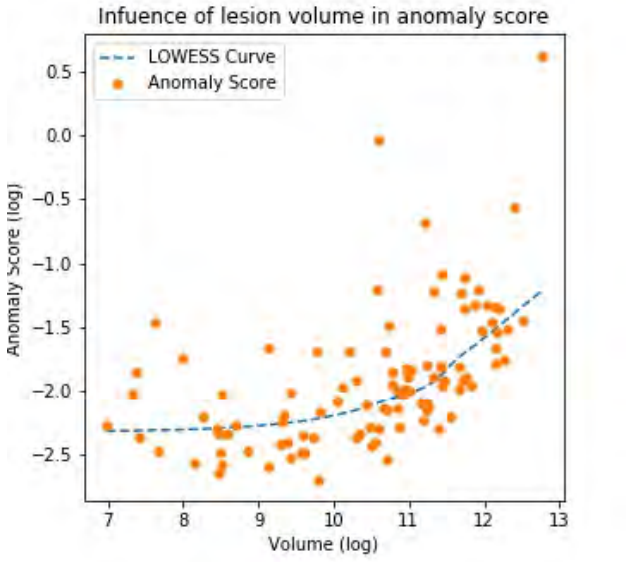


Figure 8: Lesion volume influence in AD score. LOWESS curve shows the monotonic relation of both variables.

Qualitative results and anomaly localization

The loss while training the encoder is a combination of the L_{img} and the L_{feat}. On one hand, the discriminator space gives informative features for the anomaly detection task. On the other hand, the L_{img} represents the mean squared error of a pixel-wise subtraction of the original x and its reconstruction $\tilde{x}$. This pixel-wise error image can be useful to *localize* the lesions. Figure 9 exemplifies the three most common cases in anomaly localization:

a) Case without findings. In the second row of this image, we can appreciate the reliable reconstructions of the input images. Indeed, no relevant region can be considered as a lesion in the pixel-wise error image (third row).

b) Undetected TBI lesions. Tiny lesions can be missed inside the normal anatomical variability, so the reconstruction image matches with the original one, making the TBI lesion hardly detectable for the system.

c) Detected TBI lesions. In contrast with the previous case, if lesions fall outside the learned distribution, the model will not be able to reconstruct this region and it will select the closest representation that has been learned, which could be thought of as a "healthy" representation of that brain. In consequence, lesions are well localized (see blue arrows). Refer to supplementary material Figure A.11 to visualize examples in different anatomical planes.

In case the reader is interested in visualize examples in different anatomical planes, refer to: axial view Figure A.11; sagital view Figure A.12; coronal view Figure A.13

Anomaly detection for biomarker discovery

The proposed unsupervised learning model is capable of detecting unknown or unannotated abnormalities. The Figure 9-c shows an epidural hematoma case, which has been labeled as ground truth. It can be noticed that the lesion compresses other structures causing *mass effect*, including *midline shift* and displacement of the lateral ventricles. This effect is not labeled in the dataset and consequently there is no supervised learning able to detect it. However, the proposed method overcomes this limitation highlighting and locating this abnormality (see orange arrows). This powerful property of detecting unannotated abnoarmalities has the potential to be used for new *biomarker discovery*.

Anomaly detection for quality control

In this experiment, we have shown that the model is capable of detecting a wide variety of anomalies. Given that an anomaly is defined as any type of data unrepresented by the normal data distribution, we can extend our AD model to detect any kind of "outlier" samples, not just a disease-specific lesion (i.e. TBI lesions). With this in mind, we extended our work for evaluating the potential use of our model for detecting low quality images and post-surgical images. It is worth mentioning that a low quality image can involve a wide variety of quality considerations, from artifacts, wrong registrations, and wrong orientations; while post-surgical cases include brain shunt and craniectomy. Figure 10 shows the anomaly score distributions for cases without abnormal findings in both datasets. In addition, the anomaly score for these anomalous images is also represented. As we can appreciate for most of the listed anomalous

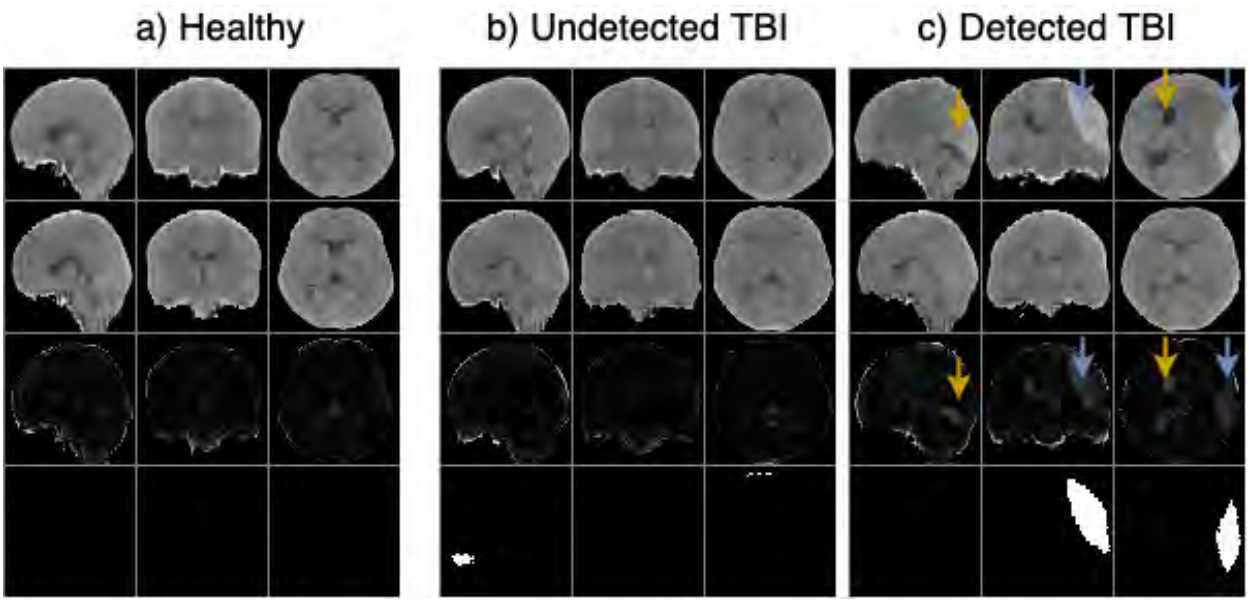


Figure 9: Anomalous region localization using pixel-level error. First row: Original image after preprocessing (x). Second row: Reconstruction image ($\tilde{x}$). Third row: Pixel-level error image. Fourth row: ground truth lesion segmentation. Arrows indicate the anomalies detected by the model; the blue ones show a labeled anomaly in the database while orange ones show an unlabeled anomaly.

cases this score is considerably higher than the the distribution without any radiological findings (note the logarithmic scale in anomaly score). Thus, we show that our model can also be used for detecting non-clinical anomalous data, but other types of outliers given by post-surgical anomalies or technical problems.

6. Discussion and conclusion

The present model is, to our knowledge, the first feasible attempt of developing a brain AD screening tool in a real-world scenario. We firstly identify, and secondly solve the main problems for deploying an AD tool in the clinical setting: i) we propose an unsupervised anomaly detector model able to efficiently handle volumetric information. The resulting model outperforms the equivalent 2D model on AD task, but also it is able to detect an anomaly without any prior information of the localization; ii) since only normal examples are used to train the model, it is capable of detecting a wide variety of abnormalities in an unconstrained manner, from TBI lesions to post-surgical pathologies, iii) in contrast with supervised learning techniques, the model does not have prior training bias of detecting any type of anomaly. Despite that the majority of TBI lesions are hematomas which are hyperdense, the model is able to detect midline shift lesions which are not only hypodense but also no labeled in the database, iv) the model offers good generalizations capabilities with a database-invariant anomaly score; v) the pixel-wise error image helps to increases the decision model interpretability. This error image is a fuzzy map that localizes the anomalies.

The proposed methodology reduces the training complexity due to the use of gradient penalty in the loss function of our Wasserstein GAN, a multi-step training methodology, and a state-of-the-art 3D GAN architecture. For instance, Kwon et al. (2019) suggest a 3D GAN model to generate brain MRI images. In this architecture, four models simultaneously fight between each other to reduce or increase the loss. Despite the notable results shown in this work, we have trained this architecture and due to the high number of network agents, obtaining proper a GAN training is hardly reachable.

6.1. Limitations

In spite of the AD potential of the model, it has limitations that could be tackled with further work. The model struggles to detect small abnormalities in these two specific cases: when the skull-stripping step removes small abnormalities close to the boundaries or when the anomaly appearance falls within the normal brain variability. In combination with these particular cases, the memory of the graphics processing unit limits the image resolution to 64x64x64, so small lesions suffer from a lack of details being prone to be undetected.

6.2. Future work

In order to improve the performance of the model, increasing the image resolution could add information for detecting tiny abnormalities. In addition, postprocessing steps such as a mean filter or connected component analysis could reduce the false positive in the pixel-error

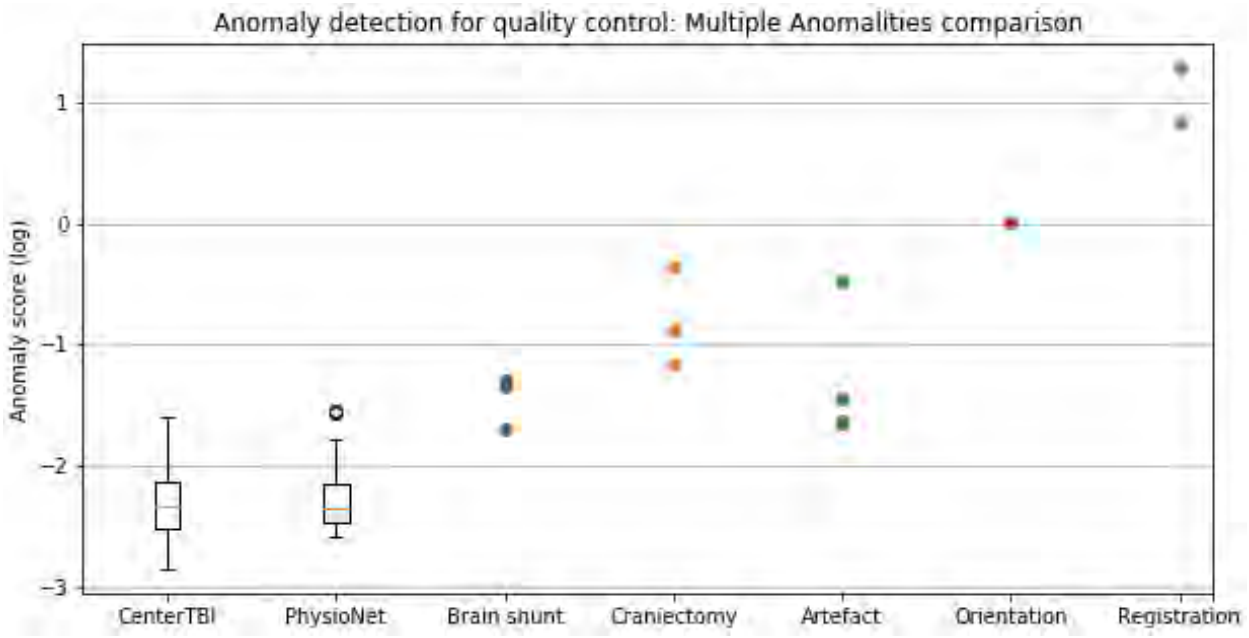


Figure 10: Different NCCT images: images without abnormal findings from CENTER-TBI and PhysioNet, followed by diverse post-surgical pathologies and low quality images. Logarithm of the anomaly scores are used for better visualization.

image. These improvements could lead to creating binary lesions maps, extending the model to an unsupervised anomaly segmentation. Moreover, the exploration of optimal cutoff values should be taken into account when considering translation to clinical settings.

A very interesting future work could be to include demographic variables in the model such as age; i.e., brain atrophy in elderly people should be included in the normal distribution while it is considered as an anomaly in young subjects.

It must be noticed that despite the huge variability on the tested datasets (different projects, centers, and scans), the model has a notable generalization capability thanks to the preprocessing and the image modality. NCCT density values represent a physical unit while, for MR images, this does not happen. Therefore, developing this AD methodology for MR images is more challenging. The GAN should capture both the normal brain variability and the intrinsic scan variability. However, it could be feasible with enough data.

7. Acknowledgments

This master thesis work is the end of my Erasmus Mundus Joint Master Degree in Medical Imaging and Applications. I want to express my gratitude to everyone who makes possible this amazing experience.

I would like to thank Thijs Vande Vyvere and David Robben for their helpful suggestions.

Data used in preparation of this manuscript were obtained in the context of CENTER-TBI, a large collaborative project with the support of the European Union 7th Framework program (EC grant 602150). Additional funding was obtained from the Hannelore Kohl Stiftung (Germany), from OneMind (USA) and from Integra LifeSciences Corporation (USA). We thank Charlotte Timmermans and Nathan Vanalken for performing the manual TBI lesion segmentations.

References

Alaverdyan, Z., Jung, J., Bouet, R., Lartizien, C., 2020. Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: application to epilepsy lesion screening. Medical Image Analysis 60, 101618.

Ardila, D., Kiraly, A.P., Bharadwaj, S., Choi, B., Reicher, J.J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al., 2019. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature medicine 25, 954–961.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks, in: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning, PMLR. pp. 214–223.

Baur, C., Denner, S., Wiestler, B., Albarqouni, S., Navab, N., 2020. Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study. arXiv preprint arXiv:2004.03271 .

Baur, C., Wiestler, B., Albarqouni, S., Navab, N., 2018. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images, in: International MICCAI Brainlesion Workshop, Springer. pp. 161–169.

Chen, X., Konukoglu, E., 2018. Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders. arXiv preprint arXiv:1806.04972 .

Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E., 2000. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. circulation 101, e215–e220.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adver-

sarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improved training of Wasserstein GANs, in: Advances in neural information processing systems, pp. 5767–5777.

Han, C., Rundo, L., Murao, K., Milacski, Z.Á., Umemoto, K., Nakayama, H., Satoh, S., 2019. GAN-based multiple adjacent brain MRI slice reconstruction for unsupervised Alzheimer's disease diagnosis. arXiv preprint arXiv:1906.06114 .

Hssayeni, M.D., Croock, M.S., Salman, A.D., Al-khafaji, H.F., Yahya, Z.A., Ghoraani, B., 2020. Intracranial hemorrhage segmentation using a deep convolutional model. Data 5, 14.

Hyder, A.A., Wunderlich, C.A., Puvanachandra, P., Gururaj, G., Kobusingye, O.C., 2007. The impact of traumatic brain injuries: a global perspective. NeuroRehabilitation 22, 341–353.

Jain, S., Vande Vyvere, T., Terzopoulos, V., Sima, D.M., Roura, E., Maas, A., Wilms, G., Verheyden, J., 2019. Automatic quantification of Computed Tomography features in acute Traumatic Brain Injury. J Neurotrauma:36(11).1794-1803 .

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Kobayashi, K., Hataya, R., Kurose, Y., Bolatkan, A., Miyake, M., Watanabe, H., Takahashi, M., Mihara, N., Itami, J., Harada, T., et al., 2020. Unsupervised brain abnormality detection using high fidelity image reconstruction networks. arXiv preprint arXiv:2005.12573 .

Kwon, G., Han, C., Kim, D.s., 2019. Generation of 3D brain MRI using auto-encoding generative adversarial networks, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 118–126.

Laalo, J.P., Kurki, T.J., Sonninen, P.H., Tenovuo, O.S., 2009. Reliability of diagnosis of traumatic brain injury by computed tomography in the acute phase. Journal of neurotrauma 26, 2169–2178.

Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O., 2016. Autoencoding beyond pixels using a learned similarity metric, in: Proceedings of The 33rd International Conference on Machine Learning, PMLR. pp. 1558–1566.

Maas, A.I., Menon, D.K., Steyerberg, E.W., Citerio, G., Lecky, F., Manley, G.T., Hill, S., Legrand, V., Sorgner, A., 2015. Collaborative european neurotrauma effectiveness research in traumatic brain injury (CENTER-TBI) a prospective longitudinal observational study. Neurosurgery 76, 67–80.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2015. Adversarial autoencoders. arXiv preprint arXiv:1511.05644 .

Monteiro, M., Kamnitsas, K., Ferrante, E., Mathieu, F., McDonagh, S., Cook, S., Stevenson, S., Das, T., Khetani, A., Newman, T., et al., 2019. TBI lesion segmentation in head CT: Impact of preprocessing and data augmentation, in: International MICCAI Brainlesion Workshop, Springer. pp. 13–22.

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill URL: `http://distill.pub/2016/deconv-checkerboard`, doi:`10.23915/distill.00003`.

Pawlowski, N., Lee, M.C., Rajchl, M., McDonagh, S., Ferrante, E., Kamnitsas, K., Cooke, S., Stevenson, S., Khetani, A., Newman, T., et al., 2018. Unsupervised lesion detection in brain CT using bayesian convolutional autoencoders .

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training GANs, in: Advances in neural information processing systems, pp. 2234–2242.

Sato, D., Hanaoka, S., Nomura, Y., Takenaga, T., Miki, S., Yoshikawa, T., Hayashi, N., Abe, O., 2018. A primitive study on unsupervised anomaly detection with an autoencoder in emergency head CT volumes, in: Medical Imaging 2018: Computer-Aided Diagnosis, International Society for Optics and Photonics. p. 105751P.

Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U., 2019. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical image analysis 54, 30–44.

Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G., 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International conference on information processing in medical imaging, Springer. pp. 146–157.

Seeböck, P., Orlando, J.I., Schlegl, T., Waldstein, S.M., Bogunović, H., Klimscha, S., Langs, G., Schmidt-Erfurth, U., 2019. Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal OCT. IEEE transactions on medical imaging 39, 87–98.

Seeböck, P., Waldstein, S., Klimscha, S., Gerendas, B.S., Donner, R., Schlegl, T., Schmidt-Erfurth, U., Langs, G., 2016. Identifying and categorizing anomalies in retinal imaging data. arXiv preprint arXiv:1612.00686 .

Sidibe, D., Sankar, S., Lemaitre, G., Rastgoo, M., Massich, J., Cheung, C.Y., Tan, G.S., Milea, D., Lamoureux, E., Wong, T.Y., et al., 2017. An anomaly detection approach for the identification of DME patients using spectral domain optical coherence tomography images. Computer methods and programs in biomedicine 139, 109–117.

Stower, H., 2020. AI for breast-cancer screening. Nature Medicine 26, 163–163.

Stuart, R., Peter, N., et al., 2003. Artificial intelligence: a modern approach. Prentice Hall.

Taboada-Crispi, A., Sahli, H., Hernandez-Pacheco, D., Falcon-Ruiz, A., 2009. Anomaly detection in medical image analysis, in: Handbook of research on advanced techniques in diagnostic imaging and biomedical applications. IGI Global, pp. 426–446.

Vaidhya, K., Thirunavukkarasu, S., Alex, V., Krishnamurthi, G., 2015. Multi-modal brain tumor segmentation using stacked denoising autoencoders, in: BrainLes 2015, Springer. pp. 181–194.

You, S., Tezcan, K.C., Chen, X., Konukoglu, E., 2019. Unsupervised lesion detection via image restoration with a normative prior, in: Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning, PMLR. pp. 540–556.

Appendix A. Additional Results

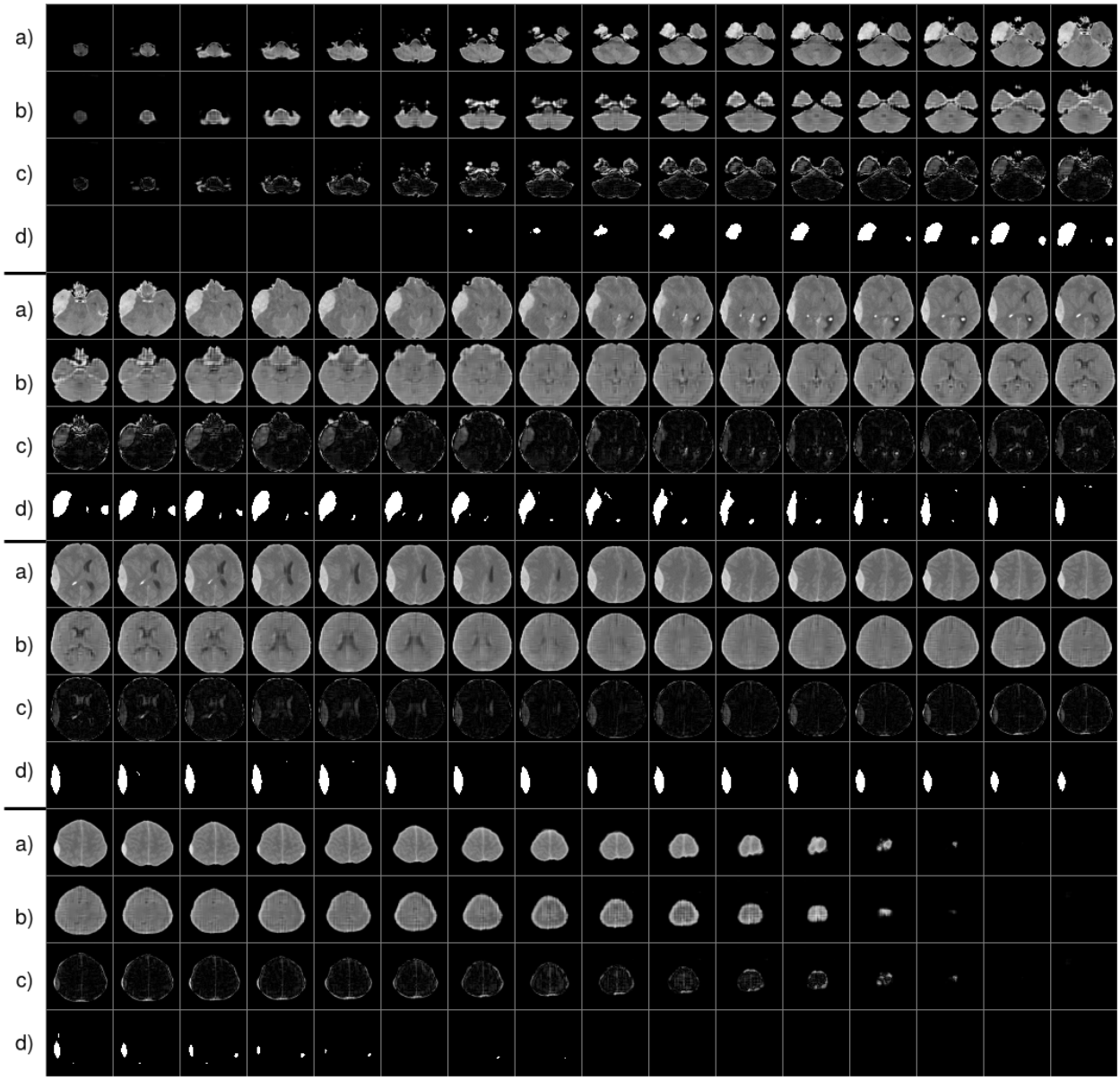


Figure A.11: Axial view: Anomalous region localization using pixel-level error. a) Input image b) Reconstruction c) Pixel-wise error d) Ground truth

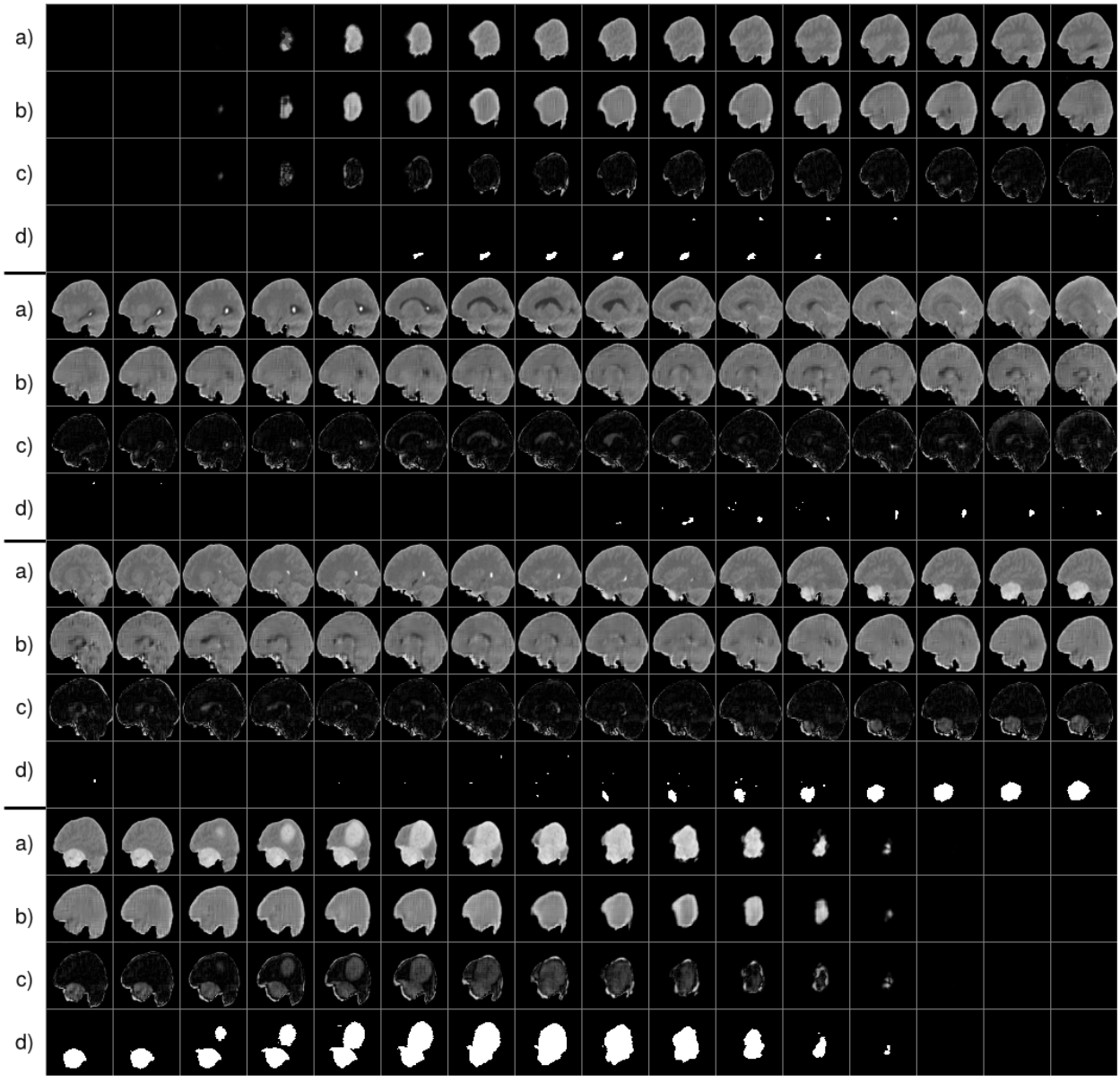


Figure A.12: Sagittal view: Anomalous region localization using pixel-level error. a) Input image b) Reconstruction c) Pixel-wise error d) Ground truth

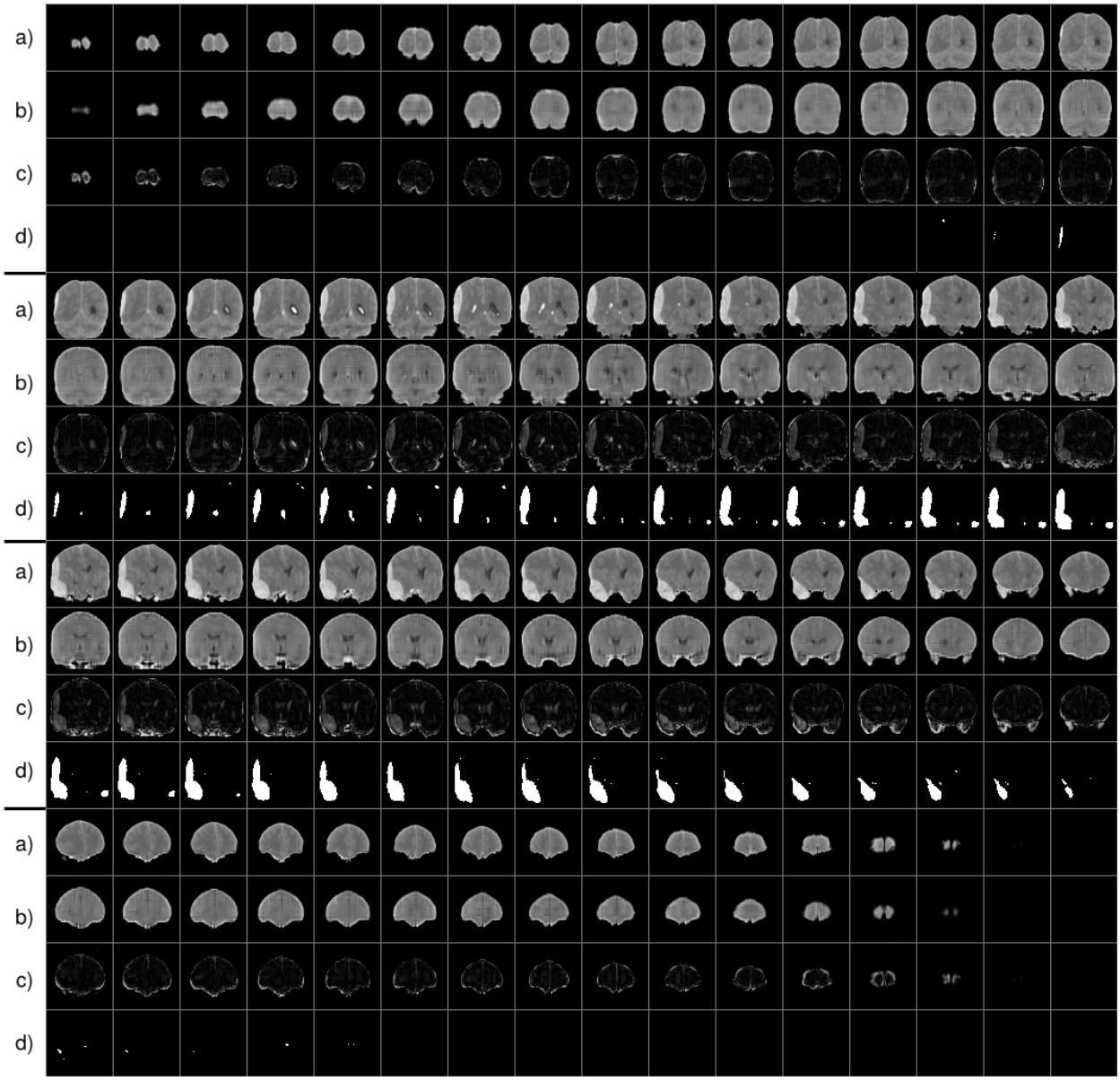
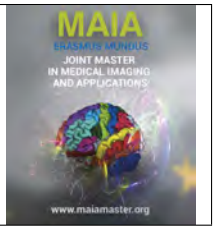


Figure A.13: Coronal view: Anomalous region localization using pixel-level error. a) Input image b) Reconstruction c) Pixel-wise error d) Ground truth

Medical Imaging and Applications

Master Thesis, August 2020



Multiple Sclerosis Lesion Segmentation Using Longitudinal Normalization and Convolutional Recurrent Neural Networks

Sergio Tascon-Morales, Martin Treiber, Stefan Hoffmann, Johannes Gregori

mediri GmbH, Heidelberg, Germany

Abstract

Magnetic resonance imaging (MRI) is the primary clinical tool to examine inflammatory brain lesions in Multiple Sclerosis (MS). Disease progression and inflammatory activities are examined by longitudinal image analysis to support diagnosis and treatment decision. Automated lesion segmentation methods based on deep convolutional neural networks (CNN) have been proposed, but are not yet applied in the clinical setting. Typical CNNs working on cross-sectional single time-point data have several limitations: changes to the image characteristics between single examinations due to scanner and protocol variations have an impact on the segmentation output, while at the same time the additional temporal correlation using pre-examinations is disregarded.

In this work, we investigate approaches to overcome these limitations. Within a CNN architectural design, we propose to use convolutional Long Short-Term Memory (C-LSTM) networks to incorporate the temporal dimension. To reduce scanner- and protocol dependent variations between single MRI exams, we propose a histogram normalization technique as pre-processing step. The ISBI 2015 challenge data were used for cross-validation.

We demonstrate that the combination of the longitudinal normalization and CNN architecture can increase the performance and the inter-time-point stability of the lesion segmentation. The proposed longitudinal architecture produced the highest Dice scores for all the analyzed cases. Furthermore, the combination of the proposed architecture and normalization led to the lowest variation for the Dice score, denoting a higher consistency of the results. The proposed methods can therefore be used to increase the performance and stability of fully automated lesion segmentation applications in the clinical routine or in clinical trials.

Keywords: Segmentation, multiple sclerosis, magnetic resonance imaging (MRI), deep learning, convolutional neural networks, recurrent neural networks, longitudinal normalization

1. Introduction

Multiple sclerosis (MS) is a chronic inflammatory disease of the central nervous system (CNS) that produces demyelination and axonal/neuronal damage (Cohen and Rae-Grant, 2012). The demyelinating process is associated with persistent inflammation throughout the CNS and, as a result, the demyelinated lesion, also known as plaque, is the main pathological feature of MS (Arnon and Miller, 2016; Compston et al., 2005). In terms of location of the lesions, there is a predilection for the periventricular white matter, optic nerves, brainstem, cerebellum and spinal cord (Lucchinetti and Parisi, 2006). Although the etiology of MS remains unknown, the disease appears to be determined by both genetic and environmental factors (Pryse-Phillips and Sloka, 2006). An autoimmune etiology has also been suggested, but remains unproven (Rinker II et al., 2006).

The course of MS can be described in terms of relapses, remissions and chronic progression either from onset or after a period of remissions (Compston et al., 2005). Relapses (attacks) are considered to represent the clinical correlate of recurrent episodes of inflammation and demyelination, often with axonal injury, in the CNS. Remissions are probably due to remyelination and resolution of inflammation and progression is believed to reflect a combination of ongoing demyelination, gliosis and axonal loss (Lucchinetti and Parisi, 2006). Four categories are commonly used to classify

the disease course: Relapsing-remitting, secondary progressive, primary progressive and progressive-relapsing (Compston et al., 2005). Relapsing-remitting MS is characterized by recurrent CNS inflammation with stable clinical manifestations between episodes, whereas in secondary progressive MS there is a gradual neurological deterioration, which occurs with or without superimposed relapses. Both primary progressive and progressive-relapsing MS exhibit gradual neurological deterioration from onset as main feature, but in the case of progressive-relapsing there are also superimposed relapses (Cohen and Rae-Grant, 2012).

According to the MS Atlas, which is a study carried out in 2008 and updated in 2013 by the World Health Organization (WHO) and the MS International Federation (MSIF), there were about 2.1 million people in the world with the disease in 2008 and 2.3 million in 2013. A more recent study reports about 2.2 million people with MS in the world and 18.932 deaths due to MS in 2016 (Wallin et al., 2019). The study also reports greater age-standardized prevalence in North America and some northern European countries (more than 120 cases per 100.000 population), moderate in some countries of Europe and Australasia (60-120 cases per 100.000 population) and lowest in North Africa and the Middle East, Latin America, Asia, Oceania, the Caribbean, and sub-Saharan Africa (<60 cases per 100.000 population).

Besides the uneven geographical distribution, MS has particular incidence and prevalence depending on sex and age. There is a female predominance of approximately 2.5 to 1 (Cohen and Rae-Grant, 2012). Regarding the onset age, although the disease can occur at virtually any age, the incidence of MS is low in childhood, with onset younger than age 10 occurring in about 0.3% of cases. After the age of 18 the incidence increases, reaching a peak between 25 and 35 and then declining. For this reason, MS is the most common non-traumatic neurological disease in young adults. Onset of the disease after the age of 50 is considered rare (Birenbaum and Greenspan, 2016; Lladó et al., 2012; Miller, 2006). The prevalence of MS is similar for boys and girls among preteen children. Divergence appears during adolescence, with higher prevalence among girls as compared to boys. This pattern continues until around the end of the sixth decade of life, when the sex ratio is about 2 to 1 in favor of women. For older people prevalence shows a continued increase for women, while for men there is a slow attenuation (Wallin et al., 2019).

Diagnosis of MS can involve several techniques or approaches that include physical examination, Magnetic Resonance Imaging (MRI), cerebrospinal fluid (CSF) analysis and evoked potentials (Cohen and Rae-Grant, 2012). MRI is widely used for diagnosis and monitoring of MS, due to the high sensitivity that it has for depicting white matter lesions, particularly in terms of dissemination in time and space, which is an important diagnostic criteria (Salem et al., 2019). Depending on the modality or sequence being examined, lesions may appear as hyperintensities, like in the case of T2-weighted (T2w), Proton Density weighted (PDw) and Fluid Attenuated Inversion Recovery (FLAIR), or as hypointensities, like in the case of T1-weighted (T1w) images (Brosch et al., 2016b). Imaging biomarkers such as lesion load and lesion count, which are based on delineation of lesions, are important for determining the progression and treatment effects of MS (Brosch et al., 2016a). Although feasible in practice and considered as the gold standard, manual lesion segmentation from 3D scans is tedious, time-consuming and prone to errors caused by inter- and intra-rater variability (Andermatt et al., 2018; Roy et al., 2018; Valverde et al., 2017). For this reason, automated strategies have been proposed based on traditional machine learning and atlas based techniques (Lladó et al., 2012). More recently, deep neural networks have attracted interest, specially convolutional neural networks (CNN) by proving their effectiveness in tissue segmentation and also brain tumor segmentation (Salem et al., 2019).

From the perspective of how the data is used to train a model, MS lesion segmentation algorithms can be classified as either longitudinal of cross-sectional. Longitudinal approaches make use of the temporal information provided by subsequent scans (known as time-points or visits) of the same patient. In cross-sectional approaches, all scans, even if belonging to the same patient, are treated as independent scans and no time information is considered. Most of the automated methods for MS lesion segmentation found in the literature treat the data as cross-sectional even in the cases in which the images have been acquired in a longitudinal manner.

This master thesis focused on the impact of longitudinal approaches from two perspectives: (1) architecture and (2) data normalization. Both perspectives aim to exploit the temporal information of longitudinal data to produce more consistent segmentation of the MS lesions.

This document is organized as follows. Section 2 presents different normalization methods that have been proposed or adopted for longitudinal MRI in presence of MS lesions. It also presents the state of the art as a comparison between cross-sectional and longitudinal approaches, focusing on deep learning approaches. In Section 3 the methodology and materials are described in detail. The results and their corresponding analysis are presented in Sections 4 and 5, respectively. Finally, conclusions are provided in Section 6.

2. State of the art

2.1. Longitudinal Normalization

One important issue when using longitudinal data is the normalization across time-points, or longitudinal

normalization. The goal is to increase the similarity, in terms of image intensity regarding tissue classes, of the different time-points, without modifying the structures whose changes are due to pathological conditions. MS lesions are an example of those structures, as they can persist, change or disappear in time (Roy et al., 2013). A statistical normalization method is proposed by Shinohara et al. (2014), in which all histograms are centered using statistical measures obtained from the white matter voxels. Sweeney et al. (2013) followed a very similar approach by expressing intensities as a departure from the white matter mean. Other methods based on matching of histograms use landmarks from a reference template to increase similarity through a piecewise linear transformation (Nyúl et al., 2000). This type of approaches can cause, however, undesired mappings, altering key anatomical structures (Roy et al., 2013).

Another longitudinal normalization method was introduced by Roy et al. (2013). In this case voxel changes in time are modeled mathematically depending on the behaviour and the lesion priors are used for keeping the lesion voxels unchanged.

2.2. MS Lesion Segmentation

Automated MS lesion segmentation is not a trivial task due to the fact that lesions vary in size, shape, intensity and location within the brain (Brosch et al., 2016a). A wide variety of algorithms have been proposed in the past to address this problem. We can distinguish between traditional machine learning approaches and deep-learning-based approaches. In the traditional machine learning group, algorithms based on both supervised and unsupervised learning can be found. The supervised learning subgroup includes algorithms based on probabilistic atlases and algorithms that are trained with manual segmentation masks. In the unsupervised learning subgroup, methods can either focus on segmenting brain tissue and detecting lesions as outliers, or they can directly focus on segmenting the lesions (Lladó et al., 2012).

With some recent exceptions such as the one proposed by Wang et al. (2020), in which a Bayesian model is built using Markov and Gibbs random field theorems, the vast majority of modern approaches are based on deep learning, to the point that deep learning approaches outnumber the approaches based on traditional machine learning. More specifically, deep convolutional neural networks (CNN) eliminate the need for handcrafted features or prior guidance and have shown, as mentioned before, outstanding performance in different brain imaging tasks. Furthermore, CNN-based approaches are now in the top of the rankings of international MS lesion segmentation challenges (Salem et al., 2019).

2.2.1. Cross-sectional MS lesion segmentation

Most of the deep learning approaches for MS lesion segmentation are cross-sectional, as shown in Fig. 1. Leading the entry of deep learning into the MS lesion segmentation field, Yoo et al. (2014) used a patch-based deep neural network to extract features that could then be used by a random forests classifier. Shortly thereafter, Vaidya et al. (2015) and Ghafoorian and Bram (2015) used 2D and 3D patch-based CNNs, respectively, not only for extracting features, but also for performing voxel classification using fully connected layers. Brosch et al. (2015) proposed an encoder-decoder called Convolutional Encoder Network (CEN) architecture without skip connections that used whole slices instead of patches. In an attempt to combine the advantages of the CEN with the classic U-Net (Ronneberger et al., 2015) architecture, the same authors added skip connections to the CEN model and used deconvolution instead of upsampling (Brosch et al., 2016a). Following the encoder-decoder architectures, McKinley et al. (2016) proposed to use several networks, one for each orientation (axial, sagittal and coronal) with only one skip connection at the top level. This multi-view style was also exploited by Aslani et al. (2018) using skip connections for all levels, and then by Aslani et al. (2019), who used three parallel independent encoders based on residual blocks, to generate features from three different modalities. These features were then combined and upsampled with one single decoder. The tendency towards the encoder-decoder architectures can be observed in several other approaches (Brugnara et al., 2020; Duong et al., 2019; Gabr et al., 2019; Narayana et al., 2020). As an alternative to this encoder-decoder architectures, a method based on convolutional recurrent neural networks was proposed by Andermatt et al. (2018), but the sequential power of the recurrent networks was not used for considering the time dimension, but rather for treating the spatial dimensions as sequential data. Their method is based on multi-dimensional gated recurrent units (GRU) and considers a filtered version of the images as an additional channel, under the assumption that the filtered images announce changes before they actually occur.

One of the problems faced when segmenting MS lesions is the number of false positive lesions that can be generated by an automated algorithm, due to the high class imbalance (Salehi et al., 2017). To address this problem, Valverde et al. (2017) proposed a special type of CNN architecture. It is a cascaded 3D CNN in which a first network is trained to have high sensitivity so that candidate lesions can be detected, and a second network is trained to reduce the amount of false positives (FP). One advantage of this architecture is that it allows domain adaptation, meaning that after being trained on a certain dataset, it does not have to be completely re-trained for evaluation on another dataset. Fur-

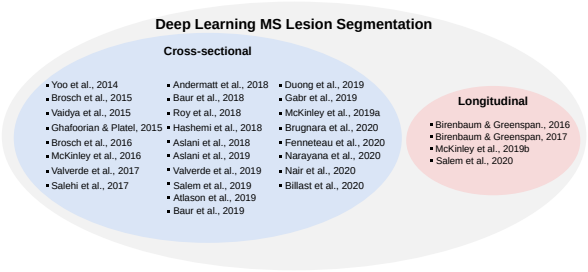


Figure 1: Overview of deep learning methods for MS lesion segmentation according to categories cross-sectional and longitudinal. For years 2014 to 2019 publications with at least 3 citations are considered, whereas for 2020 all found publications are included.

thermore, only some of the fully connected layers have to be re-trained with few new examples (Valverde et al., 2019). Instead of focusing on the architecture itself, Salehi et al. (2017) and Hashemi et al. (2019) used an asymmetric loss function based on the Tversky index. This loss function, which is a generalization of the Dice coefficient and the F_β scores, allows to give more importance to sensitivity or to precision, as determined by two parameters α and β.

Although supervised learning is the predominant type of learning for MS lesion segmentation, other types of learning such as unsupervised (Atlason et al., 2019; Baur et al., 2019b), semi-supervised (Baur et al., 2019a) and self-supervised (Fenneteau et al., 2020) have been explored too.

2.2.2. Longitudinal MS Lesion Segmentation

Only few deep learning approaches can be found in the literature that tackle the problem of MS lesion segmentation in a longitudinal manner. The first CNN-based longitudinal method found in the literature was pro-posed by Birenbaum and Greenspan (2016, 2017). Although their CNN is used only for classifying candidates extracted using intensity and atlas information, the method employs different time-points to perform the task.

McKinley et al. (2020) proposed a method to detect the lesion load change using CNNs. To achieve this, an architecture known as DeepSCAN was used as basis, which is a hybrid between the U-Net and the Dense-Net (Huang et al., 2017). A special type of loss function allows to output, for each voxel and tissue class, the probability that a voxel contains the tissue class, as well as the probability that the predicted label does not match the label of the ground truth. These probabilities and the mask provided by the model are then used to obtain information about lesion change. Following this idea of detecting changes, Salem et al. (2020) proposed an architecture that consists of a first block based on Voxelmorph (Balakrishnan et al., 2019) to learn deformation fields and register baseline image to the follow-up images, and a second block to perform the segmentation of new lesions using the results of the first block.

3. Materials and methods

3.1. Dataset and Pre-processing

One of the most popular datasets for MS lesion segmentation is the one provided by the longitudinal MS lesion segmentation challenge, which was part of the International Symposium on Biomedical Imaging (ISBI) in 2015 and continues to be publicly available. The dataset, acquired with a 3T scanner, is subdivided into training (5 subjects) and testing (14 subjects) sets. Only the training set contains lesion segmentation masks generated by two different expert raters. These masks will be referred to as *mask1* and *mask2* in this document. Each subject contains between 4 and 6 time-points, each of which consists of a T1-weighted (T1-w) magnetization prepared rapid gradient echo (MPRAGE) with TR = 10.3 ms, TE = 6 ms, flip angle = 8°, $0.82 \times 0.82 \times 1.17$ mm^3 voxel size; a double spin echo (DSE) which produces the PD-w and T2-w images with TR = 4177 ms, TE1 = 12.31 ms, TE2 = 80 ms, $0.82 \times 0.82 \times 2.2$ mm^3 voxel size; and Fluid Attenuated Inversion Recovery (FLAIR) with TI = 835 ms, TE = 68 ms, $0.82 \times 0.82 \times 2.2$ mm^3 voxel size. The average time between subsequent time-points is 1 year (Carass et al., 2017; IACL, 2018).

Both the original and the pre-processed images are available for use. The pre-processing steps for each subject are the following: First, the baseline (first time-point) MPRAGE image is corrected using the N4 algorithm, then it is skull-stripped, then dura stripped. After this a second N4 correction takes place and, finally, it is registered to a 1 mm isotropic MNI template. This pre-processed baseline MPRAGE image is then used as target for remaining images of the current patient, which are N4 corrected and then rigidly registered to the baseline MPRAGE image. The masks obtained from skull and dura stripping the baseline image are used on the remaining images (IACL, 2018). For this work, the pre-processed images were used.

3.2. Longitudinal Normalization

A simple yet effective MRI longitudinal normalization based on the Chi-Square metric χ^2 is proposed. The Chi-Square test is commonly used for analyzing the difference between observed and expected distributions (Weaver et al., 2017), but in this case only the metric is used to measure and maximize the similarity between the histograms of volumes of the same modality for different patients and different time-points. Eq. 1 shows the Chi-Square metric as a means of comparison of two histograms H_a and H_b, for voxel intensities I.

Let s, t and m represent subject, time-point and modality, respectively. For each modality m a reference volume $V^{(m)}_{\hat{s}\hat{t}}$ is selected to normalize the other volumes

$V_{st}^{(m)}$ of that modality, with $s \neq \hat{s}, t \neq \hat{t}$. For each $V_{st}^{(m)}$ an optimal scalar $\theta_{st}^{(m)}$ is found using Eq. 2, where $H_{\hat{s}\hat{t}}^{(m)}$ and $H_{st}^{(m)}$ are the histograms of the normal appearing white matter (NAWM) of $V_{\hat{s}\hat{t}}^{(m)}$ and $V_{st}^{(m)}$, respectively. The normalized images are the result of the product $\theta_{st}^{(m)} V_{st}^{(m)}$. To obtain the NAWM masks, the Computational Anatomy Toolbox (CAT12) applied within the Statistical Parametric Mapping (SPM12) toolkit was used (Gaser, C., Dahnke, 2016; Penny et al., 2011).

$$dist_{\chi^2}(H_a, H_b) = \sum_I \frac{[H_a(I) - H_b(I)]^2}{H_a(I)} \quad (1)$$

$$\theta_{st}^{(m)} = \underset{\theta}{\operatorname{argmin}} \sum_I \frac{[H_{\hat{s}\hat{t}}^{(m)}(I) - H_{st}^{(m)}(\theta \cdot I)]^2}{H_{\hat{s}\hat{t}}^{(m)}(I)} \quad (2)$$

3.3. Patch sampling

Training a patch-based model that considers multiple time-points and multiple MRI modalities requires a proper temporal sampling strategy. In this work, patches with dimensions (T, M, H, W, D) are used, where T and M represent the number of selected time-points to process for each sample and the number of modalities, respectively. H, W and D represent the spatial dimensions, i.e. the height, width and depth of the patches in each volume.

We can subdivide the sampling process into spatial sampling, modality sampling and time sampling. Spatial sampling determines how the 3D patches are selected within each 3D volume. Sub-patches with size (H, W, D) are extracted for each subject in an uniform way from the brain only, using a brain mask generated as the non-zero voxels of the FLAIR image of the first time-point. Because of the multi-modal and longitudinal character, the selected sub-patches are also extracted across modalities and across time-points, as determined by the modality and time sampling.

Modality sampling refers to which modalities are used for generating the patches. Regarding modality sampling, considering all four modalities has been shown to bring the best performances in MS lesion segmentation as compared to using only some of them (Narayana et al., 2020). For this reason, all four available modalities are used, therefore $M = 4$ in all cases.

Finally, sampling in time refers to how the patches are selected across time-points, as determined by the desired number of time-points to be analyzed in each sample (parameter T). This sampling is made by slicing a window of size T through all available time-points. Choosing an odd value for T becomes convenient, so that the segmentation can be provided for the time-point in the middle, which is possible thanks to the bi-directional implementation of the C-LSTMs, as explained later on. This, however, raises the question about how to segment the $\lfloor T/2 \rfloor$ first and last time-points. This was solved by applying time padding, i.e. by repeating the first and last $\lfloor T/2 \rfloor$ time-points. Fig. 2 shows this strategy for the case when $T = 3$. The first and last time-points are copied for all modalities, which is indicated with blue arrows in the figure. It is also shown which time-point is selected in the ground truth, which, as mentioned before, is chosen to be the one in the middle of the window. Thus, this padding allows to generate samples in the positions where the sliding window would not have information available.

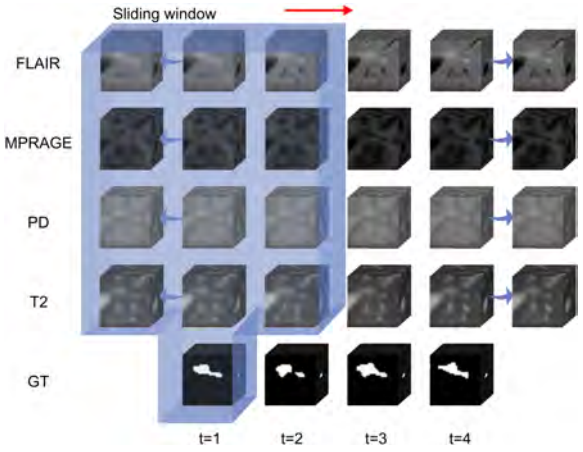


Figure 2: Time padding strategy when $T = 3$. First and last time-points are repeated for all modalities (blue arrows). The red arrow indicates how the temporal sliding window moves.

3.4. Architecture

In order to exploit temporal information, a 3D extension of the architecture presented by Novikov et al. (2019) is proposed for the segmentation of MS lesions from longitudinal multi-modal brain images. This architecture is a hybrid between the well known U-Net and a variant of the Convolutional Long Short-Term Memory (C-LSTM) network (Xingjian et al., 2015). Figure 4 shows the architecture. Just like in the U-Net, there is an encoder for extracting hierarchical features, but these are extracted for each time-point separately. These features are then combined, at the deepest level and for all input time-points, by a bidirectional C-LSTM. After the features are processed by the first C-LSTM, a decoder upsamples them so that the input dimensions can be reached again. At the output of the decoder, a second bidirectional C-LSTM combines the features of the different time-points again. Finally, the feature maps corresponding to a specific time-point (e.g. the one in the middle, if T is odd) are selected and a last convolution takes place to reduce the number of maps to 2, one for each class, lesion or non-lesion. The selection of a time-point after the second C-LSTM implies that when training the network, the ground truth mask of the same time-point must be used, as indicated previously in Fig. 2.

The inner structure of the units or cells that compose each bidirectional C-LSTM block is shown in Fig. 3, where C and h correspond to the cell and hidden states, respectively. Contrary to traditional LSTM networks used in other fields and although not visible in the figure, the C-LSTM uses convolutions (Xingjian et al., 2015), as determined by Eq. 3 to 8, where σ corresponds to the sigmoid function, tanh is the hyperbolic tangent function, $*$ denotes convolution and $\circ$ represents the Hadamard product.

Figure 5 shows how the C-LSTM blocks are built for the case when $T = 3$. The bidirectional nature is achieved by processing the sequences in both possible directions of the time dimension and then adding the outputs. This allows to better capture the temporal behaviour of the lesions and also makes possible to take advantage of using the time-point in the middle during training.

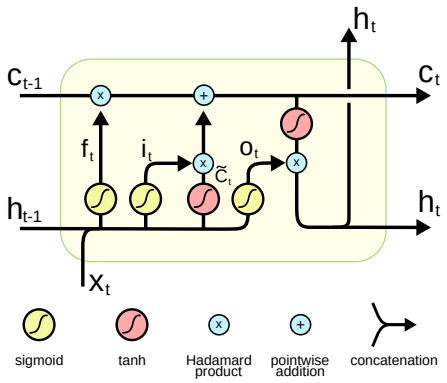


Figure 3: C-LSTM basic unit. Convolutions described by Eq. 3 to 8 are not shown in the figure. Diagram based on Phi (2018).

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{3}$$

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{4}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{5}$$

$$\tilde{C}_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{6}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{7}$$

$$h_t = o_t \circ \tanh(C_t) \tag{8}$$

3.5. Post-Processing

After a segmentation is produced, a post-processing step is performed to exclude potential false-positive (FP) detected lesions. This is achieved by imposing a minimal lesion size of 3 mm^3, as it has been found to improve the performance of MS lesion segmentation methods (Fartaria et al., 2018).

3.6. Experimental Setup

3.6.1. Normalization Configuration

One important step in the proposed longitudinal normalization is the white matter segmentation, which was performed on each volume using CAT12 with the default parameters. For finding the values of $\theta_{st}^{(m)}$, the first time-point of subject 01 was selected as reference for each modality. The Nelder–Mead Simplex method (Dennis Jr and Woods, 1985) was employed for the minimization of the distance function.

For comparison purposes, the min-max normalization (Eq. 9) and the standardization (Eq. 10) are also considered, since they are widely used in MS lesion segmentation. In min-max normalization the intensity values are mapped to the interval $[0, 1]$, whereas in standardization the goal is to have zero mean and standard deviation one. In Eq. 9, I_{orig}, I_{min} and I_{max} represent the original, minimum and maximum intensities of a volume, respectively, and I_{norm} is the assigned intensity. In Eq. 10 the term μ corresponds to the mean of the intensities and σ is the standard deviation. I_{orig} and I_{norm} have the same meaning explained for Eq. 9.

$$I_{norm} = \frac{I_{orig} - I_{min}}{I_{max} - I_{min}} \tag{9}$$

$$I_{norm} = \frac{I_{orig} - \mu}{\sigma} \tag{10}$$

3.7. Training and Cross-validation

After having normalized the pre-processed images, a leave-one-out (subject-wise) cross-validation was performed on the training set. For each fold, from the 4 subjects not used for testing, one was used for validation and 3 for training. The model was trained using 32×32×32 spatial patches with step size 16×16×16. All four modalities were used ($M = 4$) and three time-points were considered for each training sample ($T = 3$). This means the size of each sample patch is (3,4,32,32,32). Training was performed using the Adam optimizer (Kingma and Ba, 2015) for a maximum of 200 epochs with an early stopping condition of 20 epochs, and a batch size of 16. To reduce the effect of the class imbalance (more normal tissue as compared to lesion tissue), the dice loss function (Milletari et al., 2016) was used, as defined in Eq. 11, where p_i and g_i denote the predicted binary segmentation and ground truth binary volume, respectively, and N is the total number of voxels. All models were separately trained for both available masks, and the subjects were assigned for each subject according to Table 1, where the validation subjects were randomly chosen once and then set to be the same for all experiments. No data augmentation was performed in order to increase the comparability between different experiments.

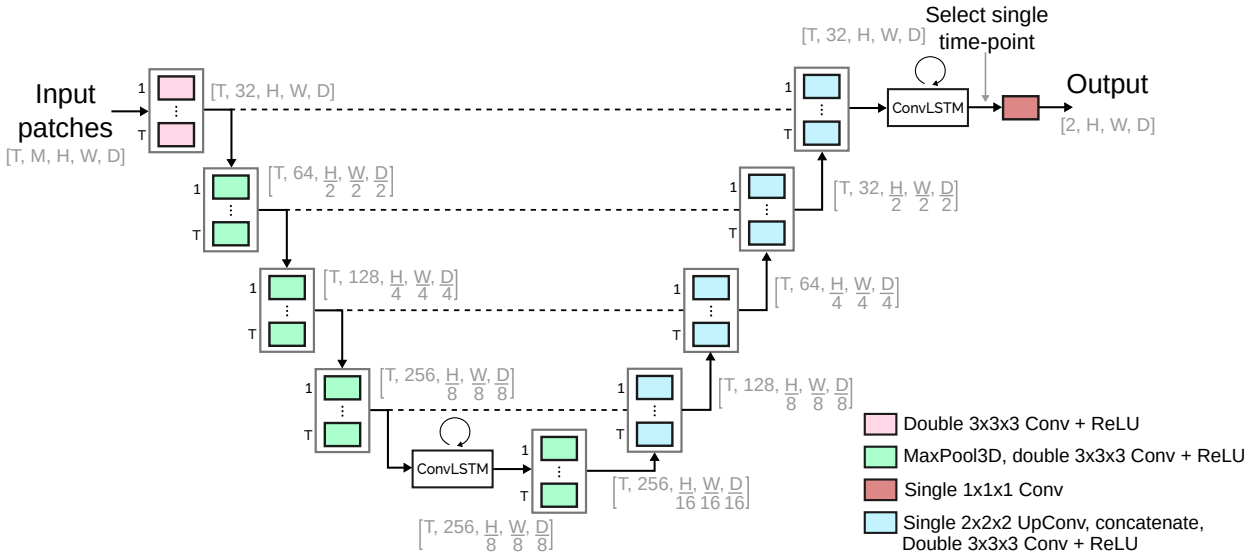


Figure 4: U-Net ConvLSTM architecture. Patch dimensions are included in gray text, where T and M denote the number of selected time-points and number of modalities, respectively. H, W and D denote the spatial dimensions of the patches in the volumes. Horizontal dashed lines denote skip connections by copying and concatenation.

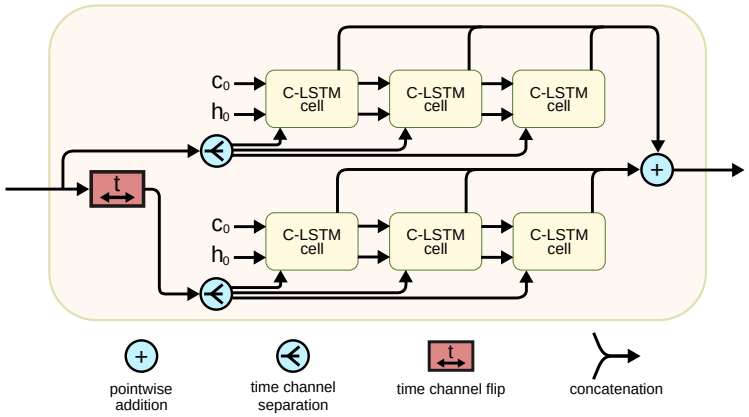


Figure 5: Bidirectional C-LSTM block for the case when $T = 3$. Both the cell and hidden states C_0 and h_0 are initialized to zero for the first unit.

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_{i}^{N} p_i g_i}{\sum_{i}^{N} p_i^2 + \sum_{i}^{N} g_i^2} \tag{11}$$

Table 1: Cross-validation subject selection

Fold	Train	Validation	Test
1	02, 04, 05	03	01
2	01, 04, 05	03	02
3	01, 02, 05	04	03
4	01, 02, 03	05	04
5	01, 03, 04	02	05

A cross-sectional version of the model, was also trained under the same parameters described before. This cross-sectional model, which is shown in Fig. 6, is the version resulting from the proposed model when the C-LSTM blocks are removed and the time-dimension is not included in the samples.

Both models (U-Net and U-Net ConvLSTM) were trained using cross-validation for the three described normalization methods (min-max, standardization and the proposed one), and for both available segmentation masks (mask1 and mask2). This means a total of 12 experiments were carried out to determine the advantages of the proposed normalization method as well as the advantages of incorporating time information to the U-Net with the bi-directional C-LSTM blocks.

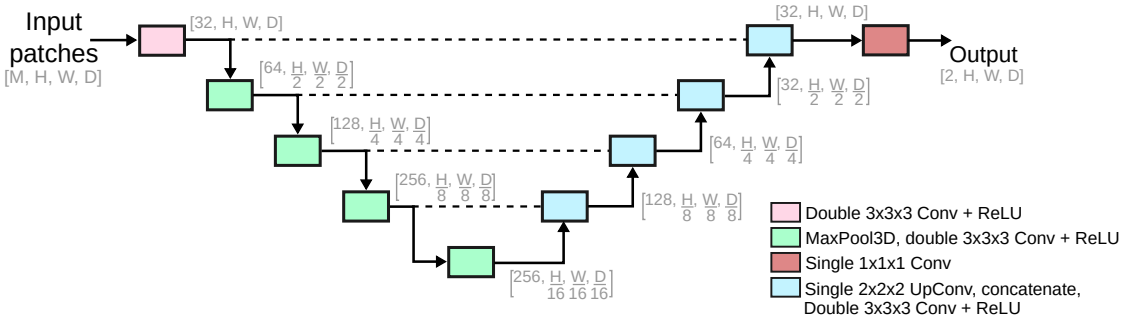


Figure 6: U-Net architecture. Patch dimensions are included in gray text, where M denotes number of modalities, respectively. H, W and D denote the spatial dimensions of the patches in the volumes. Dashed lines denote skip connections implemented by copying and concatenation.

3.8. Evaluation Metrics

To evaluate the performance of the longitudinal method, the Dice score (DSC), lesion-wise false positive rate (LFPR) and lesion-wise true positive rate (LTPR) were used. The DSC is computed according to Eq. 12, where TP, FP and FN denote number of true positive, false positive, and false negative voxels, respectively. The LFPR (Eq. 13) is the number of lesions in the produced segmentation that do not overlap with a lesion in the ground truth, divided by the total number of lesions in the produced segmentation. The LTPR (Eq. 14) is computed as the number of lesions in the ground truth that overlap with a lesion in the produced segmentation, divided by the total number of lesions in the ground truth (Aslani et al., 2018).

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{12}$$

$$LFPR = \frac{LFP}{\#PL} \tag{13}$$

$$LTPR = \frac{LTP}{\#RL} \tag{14}$$

3.9. Implementation

The models were implemented in PyTorch, using a GPU NVIDIA Tesla T4.

4. Results

The histograms after normalization are presented in Fig. 7 for all pre-processed training subjects of the dataset. Background voxels are ignored for the computation of the histograms.

Tables 2 to 5 show the results of the cross-validation process for all four possible combinations: training and evaluation with mask1 (Table 2), training with mask1 and evaluation with mask2 (Table 3), training with mask 2 and evaluation with mask1 (Table 4), and training and evaluation with mask2 (Table 5). The metrics are computed as global averages for all time-points of all subjects and standard deviations are shown in parentheses. When the same ground truth is used for both training and evaluation (Tables 2 and 5), the proposed pipeline produces the best DSC (0.711) in the case of mask1 and the second best (0.676) in the case of mask2, compared to the other evaluated methods. In both situations, however, the proposed pipeline leads to the lowest standard deviation of the DSC. When different masks are used for training and evaluation (Tables 3 and 4), the proposed pipeline leads to the highest DSC and also the lowest standard deviations. The best results could be achieved when training and evaluating the network on mask1, as well as when training on mask2 and evaluating on mask1 (Tables 2 and 4), with a DSC of > 0.71 and standard deviation of $\leqslant 0.085$.

To demonstrate how the standard deviation changes for the different evaluated methods, Fig. 8 and 9 show scatter plots of the DSC metric for the cases in which both training and evaluation are performed using the same ground truth. Particularly, with respect to the simple cross-sectional model with min-max normalization (leftmost), the proposed pipeline (rightmost) reduces the standard deviation of the DSC by 56.2% and 33.8% for mask1 and mask2, respectively. Especially, the amount of results with low DSC is reduced.

Resulting lesion segmentation examples are shown as overlay on the FLAIR images in Fig. 10 and 11 for one slice of specific subjects.

5. Discussion

5.1. Longitudinal Normalization

A longitudinal pipeline for the segmentation of MS lesions has been presented in this document. The first step in the pipeline is the longitudinal normalization, which is based on the optimization of the Chi-Square metric, and which is performed not only across time-points for every single subject, but also across all subjects. This allows to increase the homogeneity of the whole dataset while preserving the contrast characteristics of the lesions and other structures. As shown in Fig 7, the alignment of the histograms is higher for the

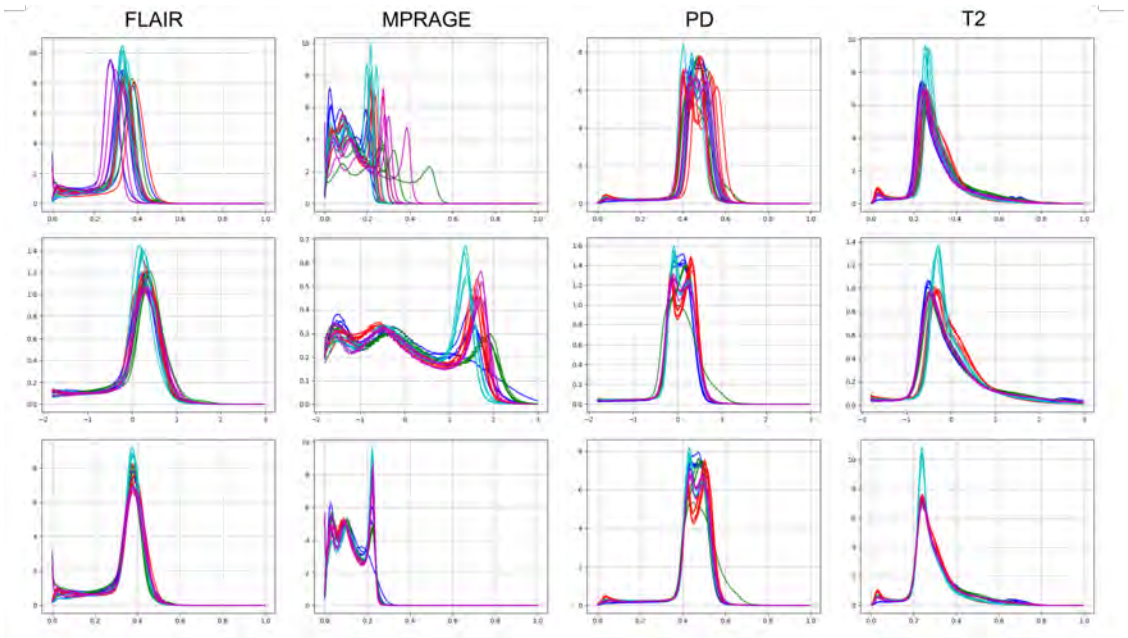


Figure 7: Histograms of the training set images for all four modalities after min-max normalization (top row), after standardization (middle row) and after the Chi-Square based normalization (bottom row).

Table 2: Segmentation results for different models and normalization methods. For the cross-validation mask1 was used for both training and evaluation. Metrics are computed as the averages for all time-points of all subjects.

Method	Normalization	Mean DSC	Mean LFPR	Mean LTPR
U-Net	min-max	0.636 (0.194)	0.396 (0.143)	0.616 (0.189)
	standardization	0.685 (0.159)	0.348 (0.163)	0.662 (0.199)
	proposed	0.651 (0.148)	0.453 (0.240)	0.664 (0.198)
Proposed	min-max	0.646 (0.179)	0.407 (0.170)	0.645 (0.160)
	standardization	0.684 (0.143)	0.371 (0.178)	0.656 (0.174)
	proposed	0.711 (0.085)	0.398 (0.134)	0.667 (0.171)

Table 3: Segmentation results for different models and normalization methods. For the cross-validation mask1 was used for training and mask2 for evaluation. Metrics are computed as the averages for all time-points of all subjects.

Architecture	Normalization	Mean DSC	Mean LFPR	Mean LTPR
U-Net	min-max	0.608 (0.185)	0.360 (0.165)	0.445 (0.182)
	standardization	0.635 (0.165)	0.338 (0.190)	0.455 (0.149)
	proposed	0.605 (0.148)	0.411 (0.265)	0.485 (0.158)
Proposed	min-max	0.605 (0.169)	0.392 (0.176)	0.458 (0.143)
	standardization	0.625 (0.144)	0.355 (0.193)	0.465 (0.144)
	proposed	0.658 (0.085)	0.377 (0.189)	0.479 (0.139)

proposed method in comparison to the classic standardization, in which the overall alignment is not always achieved.

In comparison to other normalization methods that require a reference such as histogram matching, the proposed method allows to preserve the basic shape of the histograms, which prevents from loosing key intensity information about the lesions. The optimization of the similarity metric reduces problems that peak/landmark based methods can exhibit when the histograms differ too much before normalization, especially in MPRAGE and PD images, where several peaks can be observed in the histograms. We chose an approach using a pre-segmented WM mask, assuming that normalizing the

Table 4: Segmentation results for different models and normalization methods. For the cross-validation mask2 was used for training and mask1 for evaluation. Metrics are computed as the averages for all time-points of all subjects.

Architecture	Normalization	Mean DSC	Mean LFPR	Mean LTPR
U-Net	min-max	0.670 (0.129)	0.420 (0.187)	0.678 (0.162)
	standardization	0.695 (0.167)	0.441 (0.161)	0.740 (0.135)
	proposed	0.659 (0.129)	0.544 (0.118)	0.750 (0.138)
Proposed	min-max	0.680 (0.124)	0.406 (0.179)	0.694 (0.130)
	standardization	0.712 (0.127)	0.446 (0.111)	0.750 (0.136)
	proposed	0.713 (0.080)	0.455 (0.134)	0.720 (0.118)

Table 5: Segmentation results for different models and normalization methods. For the cross-validation mask2 was used for both training and evaluation. Metrics are computed as the averages for all time-points of all subjects.

Architecture	Normalization	Mean DSC	Mean LFPR	Mean LTPR
U-Net	min-max	0.664 (0.142)	0.359 (0.180)	0.505 (0.145)
	standardization	0.663 (0171)	0.375 (0.139)	0.561 (0.089)
	proposed	0.638 (0.135)	0.481 (0.163)	0.580 (0.122)
Proposed	min-max	0.673 (0.137)	0.329 (0.156)	0.542 (0.135)
	standardization	0.685 (0.140)	0.385 (0.116)	0.550 (0.108)
	proposed	0.676 (0.094)	0.392 (0.191)	0.534 (0.099)

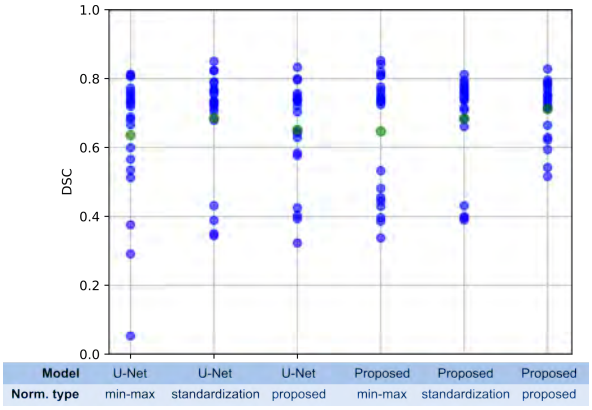


Figure 8: Scatter plot of DSC metric for models trained and evaluated with mask1 using cross-validation, for different normalization methods. Blue circles represent the value of the metric for all subjects and time-points, and green circles represent the average value.

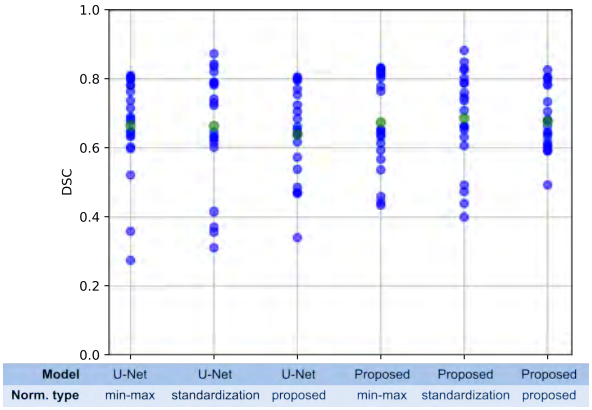


Figure 9: Scatter plot of DSC metric for models trained and evaluated with mask2 using cross-validation, for different normalization methods. Blue circles represent the value of the metric for all subjects and time-points, and green circles represent the average value.

surrounding tissue value of white matter lesions optimally supports the detection of the pathological lesions. This approach relies on a rough segmentation of the white matter before applying the CNN. The WM segmentation can be affected by the presence of lesions, but the influence in the final displacement of the histograms was found to be very small. However, when a WM mask is not available, the normalization can also be applied on the original histograms, at the cost of a higher influence of the lesion volume in the quality of the normalization, but still allowing the alignment of the histograms.

One disadvantage of the proposed normalization is the fact that it requires a reference time-point, which is used for all remaining images. Even though the effect of the selection of this reference or the generation of a synthetic reference was not studied in this work, it is expected to have an impact in the performance of the subsequent steps of the pipeline.

5.2. Lesion Segmentation

The second step in the pipeline is the lesion segmentation, for which an improvement in the DSC is observed when the mask1 is used for both training and evaluation, whereas for training and evaluation performed with mask2 the standardization produced, to-

gether with the proposed architecture, a better DSC as compared to the other cases. When training and evaluation is performed with different masks, the proposed pipeline produced the highest DSC. Furthermore, in all 4 possible combination of masks for training an evaluation the proposed architecture produced the best results in terms of DSC, either with standardization or with the proposed normalization procedure. This contributes to the validness of the initial hypothesis that considering time information can help produce better segmentation results, as suggested previously by Birenbaum and Greenspan (2016, 2017).

Table 6 shows a summary and comparison of the results, with respect to previous reported deep learning approaches in the literature for the same general cross-validation procedure followed in this work. The table also includes the metrics computed between both raters. Taking into account that no data augmentation took place for the proposed pipeline, the obtained results are close and even higher to some of the previously proposed methods in some of the metrics, particularly the DSC and the LFPR. However, the table does not reflect the inter- and intra time-point consistency of the results, which is also one important advantage of the proposed pipeline.

In order to characterize MS lesion activity and the temporal change of lesion size correctly, it is crucial that all time-points lead to consistent results compared to a human expert rater. To study this, tables 2 to 5 include the standard deviation of the obtained DSC for the respective method. Outliers lead to higher standard deviations, while results consistent with the human rater should yield low standard deviations of DSC. Thus, the standard deviation of DSC is an important quality metric in this work, to characterize the quality for a segmentation when longitudinal data is used.

Another important fact to be considered in the longitudinal setting is that, for a given subject, the images of subsequent time-points are not expected to have significant or aggressive changes. Instead, they are expected to be relatively similar considering also that the average interval between time-points for the used dataset is one year. This consistency in the volumes implies consistency in the segmentations of the lesions. While the proposed histogram normalization method is expected to reduce outliers induced by time-points with different image contrast, the proposed longitudinal architecture including C-LSTM is expected to improve the temporal consistency of the segmentations.

The cross-sectional approach with min-max normalization can produce segmentations that are highly different for a certain time-point of a subject, as shown in the first column of Fig. 10, where no lesion is detected in the second time-point for the shown slice. This is corrected by implementing a better normalization strategy. However, in order to increase the intra and inter time-point consistency, the use of a longitudinal model together with the proposed normalization led to the most compact ranges for the DSC metric in terms of low standard deviation, as shown in the scatter plots and standard deviations. This does not mean that the model can only detect lesions that appear in all time-points. Fig. 11 shows an example of a lesion that changes in time, and for which the proposed architecture, when combined with the proposed normalization or with standardization, is able to capture the change in the lesion.

Regarding the LFPR and LTPR, it does not seem to be an improvement nor deterioration of the obtained values, or at least a general trend. In some cases the longitudinal model led to higher values, whereas in some other cases the cross-sectional approach caused higher values for both LFPR and LTPR.

In terms of training and inference times, although this aspect was not analyzed thoroughly, the addition of the bidirectional C-LSTM blocks causes additional computation time which is highly dependent on the implementation of these blocks. The training time of the proposed architecture was found to be about 1.5x the time of a normal cross-sectional U-Net. This factor is of course dependent on the implementation of the C-LSTM blocks, which were not optimized for time efficiency in this work.

Diagnosis and treatment decision based on lesion inspection on MRI data is a central aspect in MS. The clinical workflow also contains the comparison to pre-examinations to assess inflammatory activity. This process is tedious when looking at up to above 100 slices in high resolution imaging, at least four modalities and several pre-examinations. Still, common solutions for automated lesion segmentation do not rely on neural networks and are not typically applied in the clinical setting. Thus, the work presented in this thesis is highly relevant as it investigates ways to improve the state-of-the-art regarding the important aspect of longitudinal analysis, in order to make longitudinal lesion segmentation applicable in clinical MS neuroimaging.

6. Conclusions and Future Work

In this study we have proposed a supervised longitudinal pipeline for MS lesion segmentation from multi-modal MR images. The approach combines a whole-volume longitudinal normalization scheme with a patch-based 3D CNN architecture that exploits time information. The method was evaluated on data from the ISBI 2015 challenge, obtaining result that are consistent in time as well as across subjects, allowing also improvements in the segmentation metrics, especially in the DSC. Lesion segmentation consistency in time for each subject should be an important goal of the segmentation algorithms, as it is a natural consequence of the non-sudden variations in the different scans that a subject can have in longitudinal studies.

Table 6: Comparison of different deep learning segmentation methods for leave-one-out cross-validation on the ISBI challenge dataset. The word proposed represents in this case the whole pipeline i.e. the proposed normalization followed by the proposed model.

Method	**mask1**			**mask2**		
	DSC	**LFPR**	**LTPR**	**DSC**	**LFPR**	**LTPR**
Rater 1	-	-	-	0.732	0.174	0.645
Rater 2	0.732	0.355	0.8260	-	-	-
Brosch et al., 2016 (mask1)	0.684	0.546	0.746	0.644	0.529	0.633
Brosch et al., 2016 (mask2)	0.683	0.646	0.783	0.659	0.620	0.693
Aslani et al., 2018 (mask1)	0.698	0.482	0.746	0.651	0.451	0.641
Aslani et al., 2018 (mask2)	0.694	0.497	0.784	0.664	0.442	0.695
Aslani et al., 2019 (mask1)	0.765	0.120	0.670	0.699	0.123	0.536
Aslani et al., 2019 (mask2)	0.765	0.202	0.700	0.713	0.190	0.572
Proposed (mask1)	0.711	0.398	0.667	0.658	0.377	0.479
Proposed (mask2)	0.713	0.455	0.720	0.676	0.392	0.534

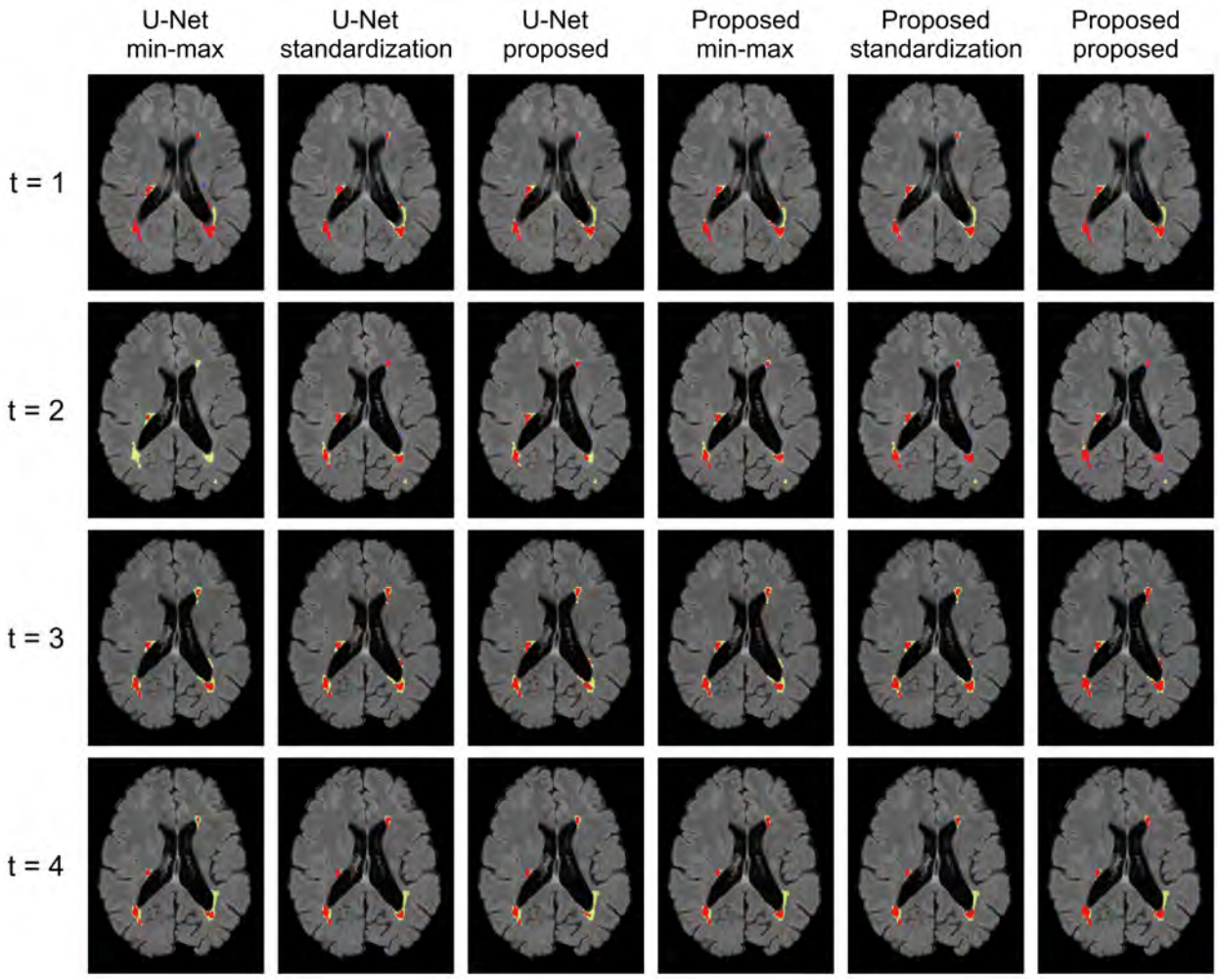


Figure 10: Example of resulting segmentation masks from cross-validation experiment for patient 01, slice 89 from the ISBI training dataset for cross-sectional and longitudinal models, and for three types of normalization. *t* denotes time-point index. Pixel colors correspond to true positives (red), false negatives (yellow) and false positives (blue), using *mask1* as reference. The proposed pipeline (rightmost column) produces the highest DSC score.

The longitudinal normalization pre-processing method increased the robustness of a trained network

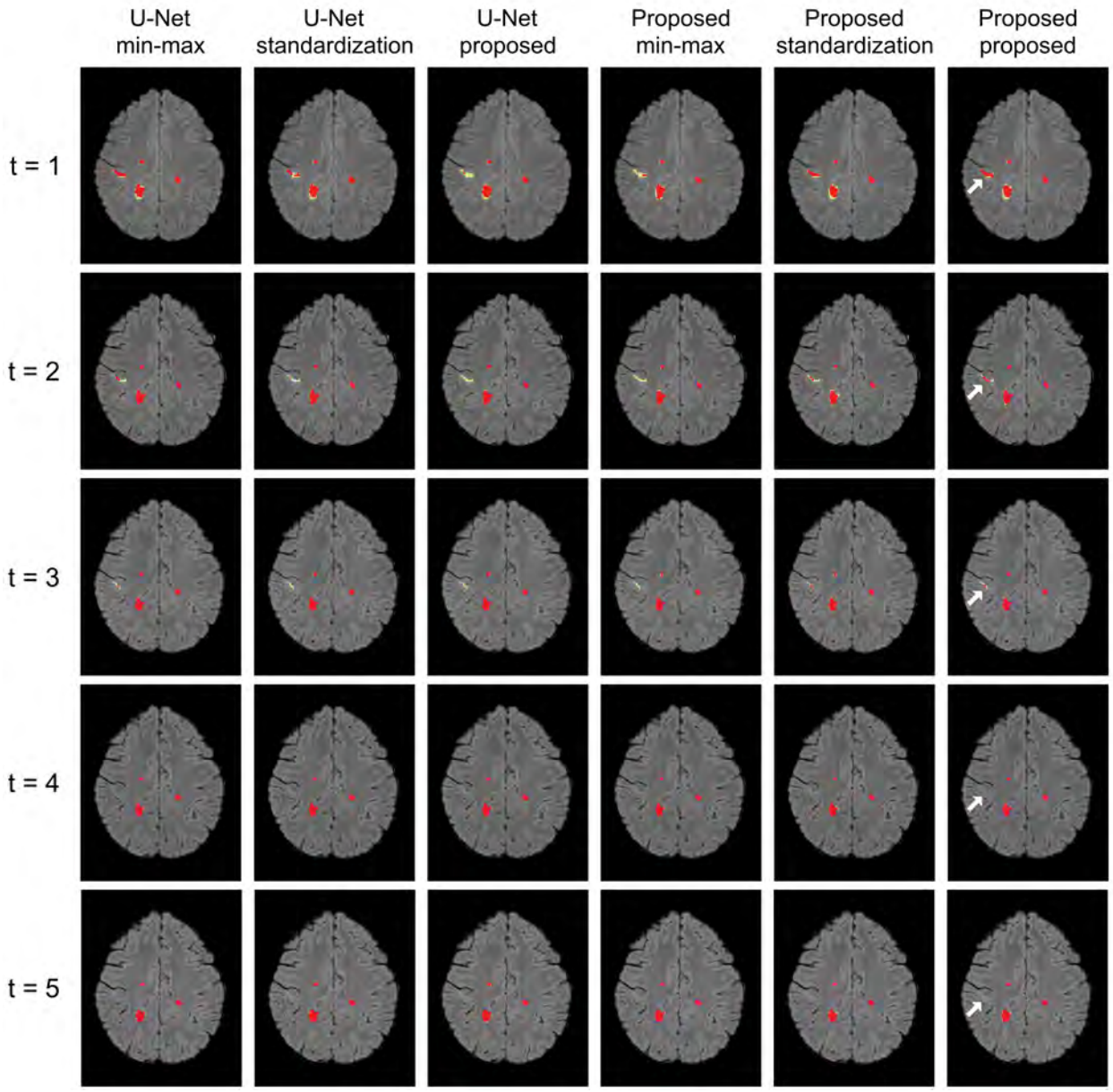


Figure 11: Example of resulting segmentation masks from cross-validation experiment for patient 03, slice 109 from the ISBI training dataset for cross-sectional and longitudinal models, and for three types of normalization. *t* denotes time-point index. Pixel colors correspond to true positives (red), false negatives (yellow) and false positives (blue), using *mask1* as reference. The white arrow points (for simplicity only in the last column) to a lesion that disappears in time, and whose change can be properly detected by the longitudinal pipeline.

in respect to the histogram variations of the input data, which were present in the ISBI 2015 training data. Thus, it is a promising technique to be applied also on MRI data from various sources, e.g. in the context of multi-center trials. Future work of our group will therefore include the validation of the algorithm on heterogeneous data from clinical studies and the evaluation of diagnostic relevance together with clinical partners. Future improvements also include the automatic selection of the reference images for the normalization process or eventually the generation of a synthetic template.

7. Acknowledgments

We would like to thank Prof. Dr. Matthias Günther and Annika Hänsch for their support whenever questions related with their fields of expertise arose. Additionally, special thanks to Daniel Mensing for his feedback to some parts of this document.

References

Andermatt, S., Pezold, S., Cattin, P.C., 2018. Automated segmentation of multiple sclerosis lesions using multi-dimensional gated recurrent units. BrainLes 2017 10670, 31–42. doi:10.1007/978-3-319-75238-9_3.

Arnon, R., Miller, A., 2016. Translational neuroImmunology in multiple sclerosis. 1st ed., Academic Press, Inc., London.

Aslani, S., Dayan, M., Murino, V., Sona, D., 2018. Deep 2D encoder-decoder convolutional neural network for multiple sclerosis lesion segmentation in brain MRI, in: International MICCAI Brainlesion Workshop, pp. 132–141. doi:10.1007/978-3-030-11723-8_13.

Aslani, S., Dayan, M., Storelli, L., Filippi, M., Murino, V., Rocca, M.A., Sona, D., 2019. Multi-branch convolutional neural network for multiple sclerosis lesion segmentation. NeuroImage 196, 1–15. doi:10.1016/j.neuroimage.2019.03.068, arXiv:1811.02942.

Atlason, H.E., Love, A., Sigurdsson, S., Gudnason, V., 2019. Unsupervised brain lesion segmentation from MRI using a convolutional autoencoder, in: Medical Imaging 2019: Image Processing, International Society for Optics and Photonics. p. 109491H. arXiv:arXiv:1811.09655v1.

Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V., 2019. VoxelMorph: A Learning Framework for Deformable Medical Image Registration. IEEE Transactions on Medical Imaging 38, 1788–1800. doi:10.1109/TMI.2019.2897538, arXiv:1809.05231.

Baur, C., De Benedikt Wiestler, C.B., Albarqouni, S., Navab, N., 2019a. Fusing Unsupervised and Supervised Deep Learning for White Matter Lesion Segmentation. Proceedings of Machine Learning Research 102, 63–72. URL: https://openreview.net/pdf?id=ryxNhZGlxV.

Baur, C., Wiestler, B., Albarqouni, S., Navab, N., 2019b. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11383 LNCS, 161–169. doi:10.1007/978-3-030-11723-8_16, arXiv:1804.04488.

Billast, M., Meyer, M.I., Sima, D.M., Robben, D., 2020. Improved inter-scanner ms lesion segmentation by adversarial training on longitudinal data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11992 LNCS, 98–107. doi:10.1007/978-3-030-46640-4_10, arXiv:2002.00952.

Birenbaum, A., Greenspan, H., 2016. Longitudinal multiple sclerosis lesion segmentation using multi-view convolutional neural networks 10008, 58–67. URL: http://link.springer.com/10.1007/978-3-319-46976-8, doi:10.1007/978-3-319-46976-8_7.

Birenbaum, A., Greenspan, H., 2017. Multi-view longitudinal CNN for multiple sclerosis lesion segmentation. Engineering Applications of Artificial Intelligence 65, 111–118. URL: http://dx.doi.org/10.1016/j.engappai.2017.06.006, doi:10.1016/j.engappai.2017.06.006.

Brosch, T., Tang, L.Y., Yoo, Y., Li, D.K., Traboulsee, A., Tam, R., 2016a. Deep 3D Convolutional Encoder Networks With Shortcuts for Multiscale Feature Integration Applied to Multiple Sclerosis Lesion Segmentation. IEEE Transactions on Medical Imaging 35, 1229–1239. doi:10.1109/TMI.2016.2528821.

Brosch, T., Yoo, Y., Tang, L., Tam, R., 2016b. Deep learning of brain images and its application to multiple sclerosis, in: Wu, G., Shen, D., Sabuncu, M.R. (Eds.), Machine learning and medical imaging. 1st ed.. Academic Press, Inc., London. chapter Deep learn, pp. 69–97.

Brosch, T., Yoo, Y., Tang, L.Y.W., Li, D.K.B., Traboulsee, A., Tam, R., 2015. Deep Convolutional Encoder Networks for Multiple Sclerosis Lesion Segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 3–11. URL: http://link.springer.com/10.1007/978-3-319-24574-4_1, doi:10.1007/978-3-319-24574-4_1.

Brugnara, G., Isensee, F., Neuberger, U., Bonekamp, D., Petersen, J., Diem, R., Wildemann, B., Heiland, S., Wick, W., Bendszus, M., Maier-Hein, K., Kickingereder, P., 2020. Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis. European Radiology 30, 2356–2364. doi:10.1007/s00330-019-06593-y.

Carass, A., Roy, S., Jog, A., Cuzzocreo, J.L., Magrath, E., Gherman, A., Button, J., Nguyen, J., Bazin, P.L., Calabresi, P.A., Crainiceanu, C.M., Ellingsen, L.M., Reich, D.S., Prince, J.L., Pham, D.L., 2017. Longitudinal multiple sclerosis lesion segmentation data resource. NeuroImage 12, 346–350. URL: http://dx.doi.org/10.1016/j.dib.2017.04.004, doi:10.1016/j.dib.2017.04.004.

Cohen, J.A., Rae-Grant, A., 2012. Handbook of multiple sclerosis. 1st ed., Springer Healthcare, London. doi:10.1007/978-1-907673-50-4.

Compston, A., Confavreux, C., Lassmann, H., McDonald, I., Miller, D., Noseworthy, J., Smith, K., Wekerle, H., 2005. McAlpine's multiple sclerosis. 4th ed., Churchill Livingstone, Elsevier, Inc.

Dennis Jr, J.E., Woods, D.J., 1985. Optimization on microcomputers. The nelder-mead simplex algorithm.

Duong, M.T., Rudie, J.D., Wang, J., Xie, L., Mohan, S., Gee, J.C., Rauschecker, A.M., 2019. Convolutional neural network for automated flair lesion segmentation on clinical brain MR imaging. American Journal of Neuroradiology 40, 1282–1290. doi:10.3174/ajnr.A6138.

Fartaria, M.J., Todea, A., Kober, T., O'brien, K., Krueger, G., Meuli, R., Granziera, C., Roche, A., Bach Cuadra, M., 2018. Partial volume-aware assessment of multiple sclerosis lesions. NeuroImage: Clinical 18, 245–253. URL: https://doi.org/10.1016/j.nicl.2018.01.011, doi:10.1016/j.nicl.2018.01.011.

Fenneteau, A., Bourdon, P., Helbert, D., Habas, C., Guillevin, R., Fenneteau, A., Bourdon, P., Helbert, D., Fernandez-maloigne, C., 2020. Learning a CNN on multiple sclerosis lesion segmentation with self-supervision, in: 3D Measurement and Data Processing, IST Electronic Imaging 2020 Symposium, San Francisco.

Gabr, R.E., Coronado, I., Robinson, M., Sujit, S.J., Datta, S., Sun, X., Allen, W.J., Lublin, F.D., Wolinsky, J.S., Narayana, P.A., 2019. Brain and lesion segmentation in multiple sclerosis using fully convolutional neural networks: A large-scale study. Multiple Sclerosis Journal , 1–10doi:10.1177/1352458519856843.

Gaser, C., Dahnke, R., 2016. CAT-A Computational Anatomy Toolbox for the Analysis of Structural MRI Data. Human Brain Mapping 32, 336–348.

Ghafoorian, M., Bram, P., 2015. Convolutional Neural Networks for MS Lesion Segmentation, Method Description of DIAG team, in: Proceedings of the 2015 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge, pp. 1–2.

Hashemi, S.R., Salehi, S.S.M., Erdogmus, D., Prabhu, S.P., Warfield, S.K., Gholipour, A., 2019. Asymmetric Loss Functions and Deep Densely-Connected Networks for Highly-Imbalanced Medical Image Segmentation: Application to Multiple Sclerosis Lesion Detection. IEEE Access 7, 1721–1735. doi:10.1109/ACCESS.2018.2886371, arXiv:1803.11078.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-Janua, 2261–2269. doi:10.1109/CVPR.2017.243, arXiv:1608.06993.

IACL, 2018. The 2015 Longitudinal MS Lesion Segmentation Challenge. URL: http://iacl.ece.jhu.edu/MSChallenge.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings , 1–15arXiv:1412.6980.

Lladó, X., Oliver, A., Cabezas, M., Freixenet, J., Vilanova, J.C., Quiles, A., Valls, L., Ramió-Torrent, L., Rovira, L., 2012. Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches. Information Sciences 186, 164–185. doi:10.1016/j.ins.2011.10.011.

Lucchinetti, C.F., Parisi, J.E., 2006. Pathology: What may it tell us?, in: Cook, S.D. (Ed.), Handbook of multiple sclerosis. 4th ed.. Taylor & Francis Group, New York. chapter Pathology:, pp. 114–115.

McKinley, R., Gundersen, T., Wagner, F., Chan, A., Wiest, R., Reyes, M., 2016. Nabla-net: a deep dag-like convolutional architecture for biomedical image segmentation: application to white-matter lesion segmentation in multiple sclerosis, in: MSSEG Challenge Proceedings: Multiple Sclerosis Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure, pp. 37–43. URL: http://www.hal.inserm.fr/inserm-01397806.

McKinley, R., Wepfer, R., Aschwanden, F., Grunder, L., Muri, R., Rummel, C., Verma, R., Weisstanner, C., Reyes, M., Salmen, A., Chan, A., Wagner, F., Wiest, R., 2019. Simultaneous lesion and neuroanatomy segmentation in multiple sclerosis using deep neural networks. arXiv preprint arXiv:arXiv:1901.07419.

McKinley, R., Wepfer, R., Grunder, L., Aschwanden, F., Fischer, T., Friedli, C., Muri, R., Rummel, C., Verma, R., Weisstanner, C., Wiestler, B., Berger, C., Eichinger, P., Muhlau, M., Reyes, M., Salmen, A., Chan, A., Wiest, R., Wagner, F., 2020. Automatic detection of lesion load change in Multiple Sclerosis using convolutional neural networks with segmentation confidence. NeuroImage: Clinical 25, 102104. URL: https://doi.org/10.1016/j.nicl.2019.102104, doi:10.1016/j.nicl.2019.102104, arXiv:1904.03041.

Miller, A.E., 2006. Clinical features, in: Cook, S.D. (Ed.), Handbook of Multiple Sclerosis. 4th ed.. Taylor & Francis Group, New York. chapter 6, pp. 153–178.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016 , 565–571doi:10.1109/3DV.2016.79, arXiv:1606.04797.

Nair, T., Precup, D., Arnold, D.L., Arbel, T., 2020. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. Medical image analysis 59. URL: http://dx.doi.org/10.1007/978-3-030-00928-1_30, doi:10.1007/978-3-030-00928-1, arXiv:1805.10884.

Narayana, P.A., Coronado, I., Sujit, S.J., Sun, X., Wolinsky, J.S., Gabr, R.E., 2020. Are multi-contrast magnetic resonance images necessary for segmenting multiple sclerosis brains? A large cohort study based on deep learning. Magnetic Resonance Imaging 65, 8–14. URL: https://doi.org/10.1016/j.mri.2019.10.003, doi:10.1016/j.mri.2019.10.003.

Novikov, A.A., Major, D., Wimmer, M., Lenis, D., Buhler, K., 2019. Deep sequential segmentation of organs in volumetric medical scans. IEEE Transactions on Medical Imaging 38, 1207–1215. doi:10.1109/TMI.2018.2881678, arXiv:1807.02437.

Nyúl, L.G., Udupa, J.K., Zhang, X., 2000. New variants of a method of MRI scale standardization. IEEE Transactions on Medical Imaging 19, 143–150. doi:10.1109/42.836373.

Penny, W.D., Friston, K.J., Ashburner, J.T., Kiebel, S.J., Nichols, T.E., 2011. Statistical parametric mapping: the analysis of functional brain images. Elsevier.

Phi, M., 2018. Illustrated Guide to LSTM's and GRU's: A step by step explanation. URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Pryse-Phillips, W., Sloka, S., 2006. Etiopathogenesis and epidemiology: clues to etiology, in: Cook, S.D. (Ed.), Handbook of multiple sclerosis. 4th ed.. Taylor & Francis Group, New York. chapter 1, pp. 1–39.

Rinker II, J.R., Naismith, R.T., Cross, A.H., 2006. Multiple sclerosis: an autoimmune disease of the central nervous system?, in: Cook, S.D. (Ed.), Handbook of multiple sclerosis. 4th ed.. Taylor & Francis Group, New York. chapter 4, pp. 95–112.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. doi:10.1007/978-3-319-24574-4_28, arXiv:1505.04597.

Roy, S., Butman, J.A., Reich, D.S., Calabresi, P.A., Pham, D.L., 2018. Multiple Sclerosis Lesion Segmentation from Brain MRI via Fully Convolutional Neural Networks URL: http://arxiv.org/abs/1803.09172, arXiv:1803.09172.

Roy, S., Carass, A., Shiee, N., Pham, D.L., Calabresi, P., Reich, D., Prince, J.L., 2013. Longitudinal intensity normalization in the presence of multiple sclerosis lesions. 2013 IEEE 10th International Symposium on Biomedical Imaging , 1384–1387doi:10.1109/ISBI.2013.6556791.

Salehi, S.S.M., Erdogmus, D., Gholipour, A., 2017. Tversky loss function for image segmentation using 3D fully convolutional deep networks, in: International Workshop on Machine Learning in Medical Imaging, pp. 379–387. doi:10.1007/978-3-319-67389-9_44, arXiv:1706.05721.

Salem, M., Valverde, S., Cabezas, M., Pareto, D., Oliver, A., Salvi, J., Rovira, A., Llado, X., 2019. Multiple Sclerosis Lesion Synthesis in MRI Using an Encoder-Decoder U-NET. IEEE Access 7, 25171–25184. doi:10.1109/ACCESS.2019.2900198, arXiv:1901.05733.

Salem, M., Valverde, S., Cabezas, M., Pareto, D., Oliver, A., Salvi, J., Rovira, À., Lladó, X., 2020. A fully convolutional neural network for new T2-w lesion detection in multiple sclerosis. NeuroImage: Clinical 25, 102149. URL: https://doi.org/10.1016/j.nicl.2019.102149, doi:10.1016/j.nicl.2019.102149.

Shinohara, R.T., Sweeney, E.M., Goldsmith, J., Shiee, N., Mateen, F.J., Calabresi, P.A., Jarso, S., Pham, D.L., Reich, D.S., Crainiceanu, C.M., 2014. Statistical normalization techniques for magnetic resonance imaging. NeuroImage: Clinical 6, 9–19. URL: http://dx.doi.org/10.1016/j.nicl.2014.08.008, doi:10.1016/j.nicl.2014.08.008.

Sweeney, E.M., Shinohara, R.T., Reich, D.S., Crainiceanu, C.M., Mri, M.L., 2013. Automatic Lesion Incidence Estimation and Detection in Multiple Sclerosis Using Multisequence Longitudinal MRI. American Journal of Neuroradiology 34, 68–73.

Vaidya, S., Chunduru, A., Muthuganapathy, R., Krishnamurthi, G., 2015. Longitudinal Multiple Sclerosis Lesion Segmentation using 3D Convolutional Neural Networks, in: Proceedings of the 2015 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge, pp. 1–2.

Valverde, S., Cabezas, M., Roura, E., González-Villà, S., Pareto, D., Vilanova, J.C., Ramió-Torrentà, L., Rovira, À., Oliver, A., Lladó, X., 2017. Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. NeuroImage 155, 159–168. doi:10.1016/j.neuroimage.2017.04.034, arXiv:1702.04869.

Valverde, S., Salem, M., Cabezas, M., Pareto, D., Vilanova, J.C., Ramió-Torrentà, L., Rovira, À., Salvi, J., Oliver, A., Lladó, X., 2019. One-shot domain adaptation in multiple sclerosis lesion segmentation using convolutional neural networks. NeuroImage: Clinical 21, 101638. URL: https://doi.org/10.1016/j.nicl.2018.101638, doi:10.1016/j.nicl.2018.101638, arXiv:1805.12415.

Wallin, M.T., Culpepper, W.J., Others, 2019. Global burden of diseases, injuries, and risk factors study (GBD). The Lancet Neurology 18, 269–285. doi:https://doi.org/10.1016/S1474-4422(18)30443-5.

Wang, J., Liu, M., Zhang, C., Xu, H., Zhang, L., Zhao, Y., 2020. An adaptive sparse Bayesian model combined with probabilistic label fusion for multiple sclerosis lesion segmentation in brain MRI. Future Generation Computer Systems 105, 695–704. URL: https://doi.org/10.1016/j.future.2019.12.035, doi:10.1016/j.future.2019.12.035.

Weaver, K.F., Morales, V., Dunn, S.L., Godde, K., Weaver, P.F., 2017. An Introduction to Statistical Analysis in Research. John Wiley Sons, Inc., Hoboken, NJ, USA. URL: http://doi.wiley.com/10.1002/9781119454205, doi:10.1002/9781119454205.

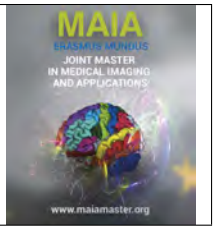
Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, pp. 802–810.

Yoo, Y., Brosch, T., Traboulsee, A., Li, D.K., Tam, R., 2014. Deep learning of image features from unlabeled data for multiple sclero-

sis lesion segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8679, 117–124. doi:10.1007/978-3-319-10581-9_15.

Medical Imaging and Applications

Master Thesis, August 2020



PET Image Harmonization Using Conditional GANs

Abdullah Thabit, Pierrick Bourgeat

Australian e-Health Research Center - CSIRO, Queensland, Australia

Abstract

Alzheimer's disease (AD) is the most common cause of dementia. It is characterized by irreversible memory loss and degradation of cognitive skills. Amyloid PET imaging has been used in the diagnosis of AD to measure the amyloid burden in the brain. It is quantified by the Standard Uptake Value Ratio (SUVR), which is defined as the amyloid signal of the cortical region in relation to a reference region that is known not to store amyloid. However, there is great variability in SUVR measurements when different tracers or scanner models are used. Therefore, quantitative assessments of amyloid burden in a multi-center study require standardization and harmonization of PET images. Conventionally, PET image harmonization has been mainly tackled either by standardization protocols at the time of image reconstruction, or by applying a smoothing function to PET images in order to bring them to a common resolution. In this work, we propose to learn to match the data distribution of PET images across different scanners in a fully automatic approach using conditional GANs (cGANs). Five different cGANs were adopted for PET image harmonization: a 3D pix2pix with a new proposed SUVR-based objective function for supervised PET harmonization, a 3D CycleGAN for unsupervised PET harmonization, a 3D StarGAN and Multi-cycleGAN for multi-domain harmonization across multiple scanner models, and a Smoothing-cycleGAN that specifically estimates the optimum smoothing function to bring PET images into a common spatial resolution. We validate the proposed approaches and perform a qualitative and quantitative analysis using two sets of datasets for different reconstruction methods and scanner models. For the reconstruction-methods dataset, all five approaches showed better agreement in the SUVR measurements after image translation compared to before, with the Mean Absolute Error (MAE) in the SUVR difference for Ultra to Plain image translation reduced from 0.059 to 0.011 for pix2pix with the SUVR loss, and to 0.015, 0.017, 0.023 and 0.048 for CycleGAN, StarGAN, Multi-cycleGAN and Smoothing-cycleGAN respectively. However, for the unsupervised approaches, there was some variability in the SUVR measurements and results reproducibility, therefore further investigation into their training stability and reproducibility is required for PET image harmonization.

Keywords: PET image harmonization, Amyloid imaging, image-to-image translation, conditional GANs

1. Introduction

Alzheimer's disease (AD) is a major neurodegenerative disease, it is characterized by a group of symptoms such as irreversible memory damage, deterioration of cognitive abilities, and difficulties in speaking and other behavioral skills that can affect a person's ability to perform day-to-day activities (Wilson et al., 2012).AD is the most common cause of dementia. It accounts for 60-70% of all dementia (Wimo et al., 2003). In 2010, it was estimated that more than 36 million people lived with dementia (Prince et al., 2013), and as the numbers of elderly people increase, the numbers of AD patients are expected to double every 20 years reaching 115 million by 2050. The prevalence of AD in population is a great challenge for healthcare systems. Despite the technological advances and the rapid increase in the understanding of the disease, there is no cure for AD and it is ultimately fatal. However, most of the AD treatment and developed drugs can slow the disease progression, and therefore early detection is vital for early treatment.

Medical imaging has always played a key role in the diagnosis and management of AD. Magnetic Resonance Imaging (MRI) was considered to be the main imaging modality for neurologists to help with AD diagnosis. It was used to identify typical anatomical changes

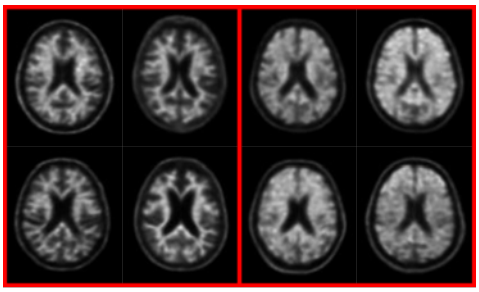


Figure 1: A sample of PET images for patients with (a) low and (b) high amyloid deposition.

that characterizes AD (Suppiah et al., 2018). However, a structural imaging modality such as MRI is affected by the aging process and other co-existing conditions that can cause changes to the brain volume (Dhikav et al., 2014). Therefore, functional imaging modalities such as Positron Emission Tomography (PET) offers a more reliable replacement with increased diagnostic accuracy.

In AD diagnosis, PET imaging has been utilized to detect two main categories of radiotracers. At first, 2-Deoxy-2[^{18}F]fluorodeoxyglucose ([^{18}F]FDG) was used for imaging AD, however it was reported to have a lower accuracy in detecting AD in older patients (Ng et al., 2007). Recently, a new set of biomarkers have been developed known as the amyloid precursors. These tracers have the ability to cross the blood brain barrier and selectively bind to beta amyloid (Aβ) plaques (Anand and Sabbagh, 2017) (Degenhardt et al., 2016) (Daerr et al., 2017) (Lowe et al., 2017), where the presence of amyloid deposits along with neurofibrillary tangles have been recognized as the hallmark pathological features of AD (Nordberg, 1992) (Perry, 1986). In a typical AD patient, Amyloid is present as small insoluble Aβ peptides (Masters et al., 1985), which can be an early event in the pathogenesis of AD that appears first in the basal neocortex, and then spread to all areas of the brain cortex (Braak and Braak, 1991). The most 5 commonly used amyloid PET tracers for AD imaging are the Pittsburgh compound ^{11}C-PiB (PiB), ^{18}F-NAV4694 (NAV), ^{18}F-Florbetaben (FBB), ^{18}F-Flutemetamol (FLUTE) and ^{18}Florbetapir (FBP) (Rowe and Villemagne, 2013).

In clinics, visual assessment and interpretation of the Amyloid PET scans is used for assessing the significance of Aβ burden in the brain, where the signal intensity of the target regions known to be rich of amyloid plaques such as the frontal cortex is compared to that of regions known to be poor of amyloid plaques such as the subcortical white matter. Figure 1 shows PET images of patients with low and high Aβ burden. However, this approach of assessing Amyloid PET scans for A burden suffers from a couple of limitations. First, it is required to be conducted by a medical imaging expert with adequate training for Amyloid imaging. Second, a standardized interpretation protocol needs to be followed to ensure inter-rater agreement. Third, PET scans are of low resolution and have low signal to noise ratio (SNR) and therefore can be hard to interpret (Johnson et al., 2013).

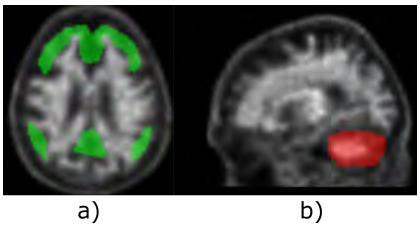


Figure 2: SUVR measurement: (a) An axial slice of the cortical region, (b) a sagittal slice of the reference cerebellum region.

For quantitative assessment, the Standard Uptake Value Ratio (SUVR) is calculated by assessing the amyloid signal of the cortical region that reflect amyloid plaque deposition in AD patients, in relation to a reference region that is known not to store amyloid (Schmidt et al., 2015):

$$SUVR = \frac{CTX}{RX}, \tag{1}$$

where CTX is the amyloid uptake in the cortical region, and RX is the amyloid uptake in the reference region. For the selection of the reference region, the cerebellum cortex is usually selected due to its low fibrillary amyloid plaques. Figure 2 shows the cortical and cerebellum regions for calculating the SUVR.

However, performing quantitative assessment of amyloid deposition in a multi-center or patient-follow up study would require a set of standardized guidelines for scans acquisition, dosage calibration, and selection of the reference region (Schmidt et al., 2015). A few organizations such as The European Association of Nuclear Medicine (EANM) and the Japanese Society of Nuclear Medicine (JSNM) tried to address this by proposing some acquisition and harmonization protocols for Amyloid imaging in order to achieve comparable quantitative measurements (Minoshima et al., 2016) (Senda, 2017). Despite these standardization efforts, different technical factors such as the scanner model, the reconstruction method, and the tracer used still lead to differences between the PET scans and therefore variability in the SUVR measurements.

The differences in the reconstructed PET images due to differences in the scanner models can be regarded as either high frequency differences or low frequency differences. The high frequency differences are depicted as differences in image resolution, which can be attributed to the scanner crystal sizes, the detector material, and the number of rings (Joshi et al., 2009). While the low frequency differences are mainly characterized by the image uniformity and the contrast difference between gray matter and white matter. It is caused by the differences in handling attenuation and scatter correction across scanner models. Figure 3 shows visually the dif-

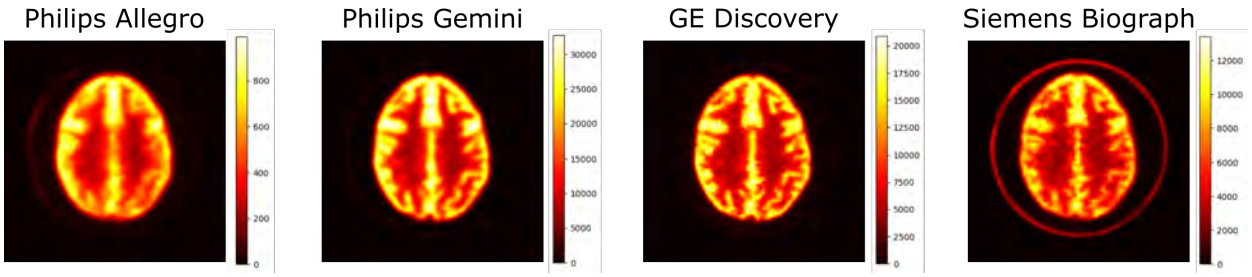


Figure 3: A PET Hoffman brain phantom imaged with 4 different scanner models.

ferences caused by changing the PET scanner model for a brain phantom.

In the efforts of standardizing and harmonizing PET amyloid measurements, Klunk et al. (2015) proposed a reference scale for A measurements that is based on a reference PiB dataset that they made publicly available. The Centiloid scale have values that range between 0 and 100, where 0 represents the typical values of negative young controls, and 100 represents the typical values of mild AD patients. Moreover, they provide a framework that allows for the linear mapping between other Aβ amyloid tracers, provided that their corresponding PiB scans are available. One limitation of their framework is that it requires the PET images to be spatially normalized by their corresponding MRI images. Bourgeat et al. (2015) tried to alleviate the need for MRI images by proposing a PET-only-based pipeline for mapping other amyloid tracers into the Centiloid scale. However, in a recent work, Bourgeat et al. (2018) demonstrated the need for harmonizing PET scans across scanner models before mapping them into the centiloid scale, otherwise it results in biases in the SUVR measurements.

Conventionally, PET harmonization is addressed at the time of image reconstruction from sinograms, by following some standardized protocols to adjust the reconstruction parameters to obtain similar quality images across different PET scanner models (Senda, 2017) (Laforest et al., 2018). Some other works, proposed to perform PET harmonization in the image space by applying image filtering to bring all the PET images into a common spatial resolution (Tsutsui et al., 2018) (Joshi et al., 2009). In the work of Joshi et al. (2009) two PET image harmonization steps were proposed to reduce the differences between different scanner models for the ADNI multi-center study (Mueller et al., 2005). They first applied a range of smoothing kernels with different PSF parameters to learn the optimum full width half maximum (FWHM) values for a harmonized spatial resolution based on phantom data. Then as a second step, they applied an affine correction to reduce the attenuation and scatter correction errors. However, these harmonization techniques are based on phantom data, and hinder their applicability when phantom data are not available. In addition, these methods are cumbersome and prone to errors as the smoothing parameters are estimated from a single scan.

In recent years, deep learning has been extensively investigated in multiple research areas such as computer vision, speech recognition, and natural language processing. The great advantage of deep learning networks lies in the fact that they can learn an increasingly higher abstract data representation as they transform inputs to outputs. The most successful type of deep learning networks for image analysis are the Convolutional Neural Networks (CNN) (Litjens et al., 2017)which contain many layers that transform their input with small size convolution filters. All these advances have led to great achievements in the area of medical image analysis, covering a range of applications such as lesion detection and classification (Shin et al., 2016) (Dou et al., 2016), image segmentation (Havaei et al., 2017) (Kamnitsas et al., 2017), image registration (Miao et al., 2016), and image enhancement (Chen et al., 2017) (Bahrami et al., 2016) (Oktay et al., 2016)

In 2014, Goodfellow et al. (2014) have introduced Generative Adversarial Networks (GANs), which are generative models that aims at learning the underlying distribution of training data to generate new data samples that are realistic and indistinguishable from the input samples. They consist of two networks, one generates synthetic data samples from random noise, and the other works as a binary classifier that tries to distinguish between the synthetic fake data samples and the real data samples. These networks are known as the generator and the discriminator respectively. The two networks are trained simultaneously with an opposing loss functions in a net-sum game. The generator is trained to maximize the probability of passing the synthetic data samples as real and fool the discriminator, while the discriminator is trained to maximize the probability of real-fake classification rate by minimizing the cross-entropy loss between the two samples. Convergence is met when the two networks reach Nash equilibrium (Zhao et al., 2016), and that is when the discriminator is incapable of distinguishing real from fake data as the generator became very good at generating realistic synthetic samples.

Some recent work built on GANs to have the generator and discriminator conditioned on some input infor-

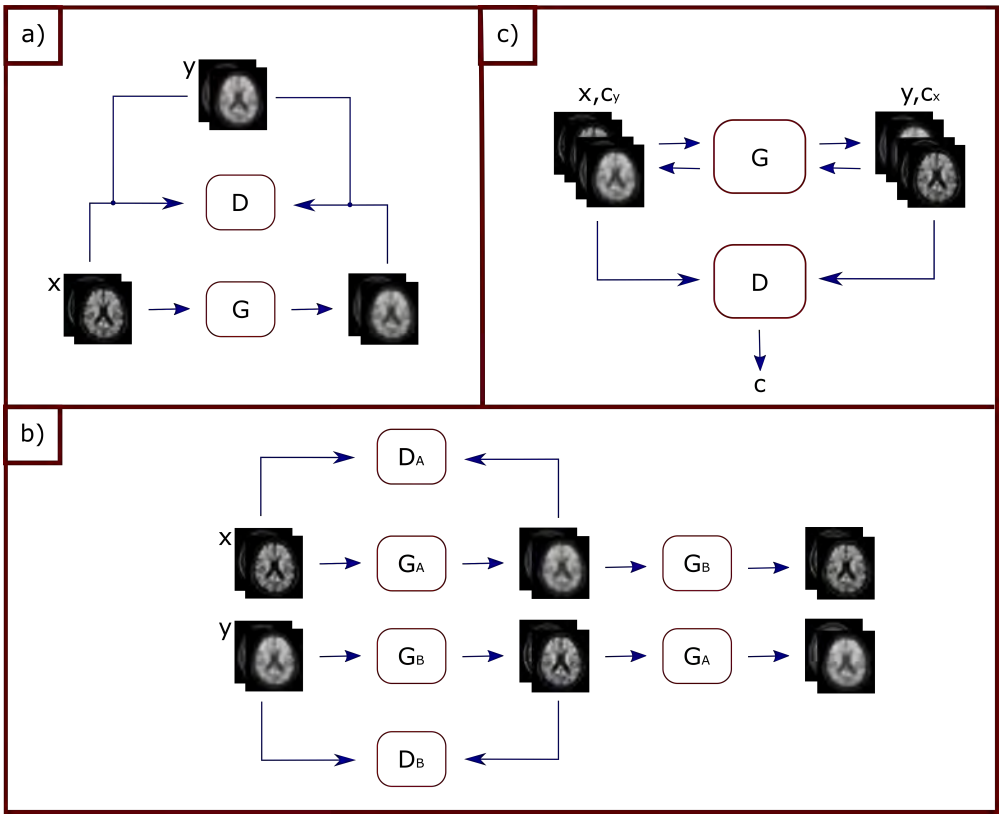


Figure 4: Training diagrams of (a) pix2pix, (b) CycleGAN and (c) StarGAN, where G is a generator, D is a discriminator, x and y are real images of two different domains, c in StarGAN is the class domain.

mation. This information could be either the class of the input or some other correlated features. These networks are known as conditional GANs (cGANs) (Mirza and Osindero, 2014) (Odena, 2016) (Odena et al., 2017). Image-based cGANs have been applied successfully in multiple applications, including domain transfer (Kim et al., 2017) (Taigman et al., 2016), superresolution imaging(Ledig et al., 2017), and photo editing (Brock et al., 2016) (Shu et al., 2017).

In this work we aim at tackling the PET harmonization problem by utilizing cGANs to translate images from one domain to the other. Specifically, we aim at harmonizing PET images across different PET reconstruction methods and different scanner models. We adopt three main networks that have demonstrated their success in other image-to-image translation tasks, namely pix2pix (Isola et al., 2017), CycleGAN (Zhu et al., 2017), and StarGAN (Choi et al., 2018). Moreover, we investigate the usage of a 3D pix2pix for supervised learning with an addition of a new proposed SUVR loss to force the generator to learn to match the SUVR distribution between the two domains. We investigate the utilization of the 3D CycleGAN and a 3D StarGAN for the unsupervised learning, and we adapt the two networks to handle the harmonization between multiple domains for PET harmonization. We further propose a new cycleGAN-based network that aims at specifically learning a Point Spread Function (PSF) for bringing the PET scans into a harmonized spatial resolution. We validate our work using two sets of datasets in a supervised and unsupervised manner, and we perform qualitative and quantitative analysis of the harmonized PET results.

2. State of the art

GANs have achieved impressive results in multiple areas in computer vision, such as image generation (Denton et al., 2015) (Huang et al., 2017) (Radford et al., 2015) (Zhao et al., 2016), super resolution imaging (Ledig et al., 2017) and image translation (Isola et al., 2017) (Kim et al., 2017) (Zhu et al., 2017). Image-to-image translation, in particular, has received great attention lately, with many proposed image-based cGANs. The goal of image-to-image translation is to learn a mapping function to transform images from one domain (source-domain) to another domain (target-domain). For example, in 2016 Isola et al. (2017) introduced the pix2pix GAN framework that learns image-to-image translation in a supervised manner using cGANs (Mirza and Osindero, 2014). It combines an adversarial loss with an L1 loss, and therefore requires paired data samples. Similar works have been used to generate images from sketches (Sangkloy et al., 2017)

or images from attributes and semantic layouts (Karacan et al., 2016). However, this approach is limited to supervised settings and requires pairs of corresponding images in the two domains. This, therefore, hinders its applicability when having paired data is not a possibility.

In order to overcome the need for obtaining data pairs, unpaired image-to-image translation frameworks (Kim et al., 2017) (Liu et al., 2017) (Zhu et al., 2017) have been proposed. Liu et al. (2017) proposed UNIT, which combines variational autoencoders (VAEs) (Kingma and Welling, 2013) with CoGAN (Liu and Tuzel, 2016), a GAN network that have shared weights between two generators in order to learn the joint distribution of images in cross domains. CycleGAN (Zhu et al., 2017) and DiscoGAN (Kingma and Welling, 2013) use a cycle consistency loss to attain key attributes between the input and the translated image. However, these frameworks can only be applied to two different domains at one time. Therefore, their applicability cannot be scaled up to for translating images across multiple domains, and hence would need to have the network trained for every two domains separately.

Recently, StarGAN was proposed by Choi et al. (2018) as a unified framework capable of translating images in a multi-domain setting using a single universal generator. The generator was conditioned on a label encoding the target domain, and the discriminator is trained to classify fake images to their correct domains, therefore making sure the network learns the differences between the multiple domains while translating images. Siddiquee et al. (2019) proposed Fixed-Point GAN, which is built on StarGAN but modified to learn the identity mapping in an explicit way by forcing the network to translate images to the source domain as well and therefore learns a minimal mapping function across domains.

In the medical field, image-to-image translation using cGANs has gained more attention recently. For example, Nie et al. (2018) used pix2pix with an additional gradient-based loss to translate MR images to CT. Wolterink et al. (2017a) used a CycleGAN network to perform the translation between MR and CT in an unsupervised manner. For denoising CT images, Wolterink et al. (2017b) utilized a pix2pix GAN network to translate low dose CT images into their high dose counterpart. In another work, Yang et al. (2018) addressed the task of CT denoising by utilizing a GAN network with Wasserstein distance loss and perceptual similarity. Odena et al. (2016) replaced the first generator in a CycleGAN with a smoothing kernel to do image deconvolution for microscopy super-resolution. In PET imaging, Zhou et al. (2020) used a CycleGAN with an additional supervised loss to boost the quality of low dose PET images using paired data. In a different work, Dong et al. (2020) proposed to use a patch-based 3D CycleGAN to perform attenuation correction for PET images without the need for CT images. In a non-GANs based work, Dewey et al. (2019) proposed a U-Net architecture to harmonize between MRI scanner models. However, up to our knowledge, no work has been done to tackle PET harmonization using a deep learning approach.

3. Material and methods

3.1. Approaches

In this work, three different cGAN networks have been adopted for PET harmonization, namely, pix2pix (Isola et al., 2017), CycleGAN (Zhou et al., 2020) and StarGAN (Choi et al., 2018). Two other variants of CycleGAN have also been implemented; one leverages multi-domain translation (Multi-cycleGAN), and one aims at specifically harmonizing the differences in spatial resolution between scanner models (Smoothing-cycleGAN).

3.1.1. pix2pix

For a supervised PET harmonization (i.e translating between paired PET images across two different domains), a 3D implementation of pix2pix was adopted. In Pix2pix, the generator is conditioned on PET images from the source domain and tries to match their distribution with that of the target domain. The discriminator challenges the generator to ensure that it produces images that cannot be distinguished from the real ones. Figure 4a shows the training diagram of pix2pix for PET harmonization.

As in the originally proposed pix2pix, the generator and the discriminator were optimized by two main objective functions: L1 loss to learn the content mapping between the source and target domains, and an adversarial loss that pushes the generator to learn the high frequency (i.e details) of the target domain. The two losses are defined as follow:

$$L_{L1} = E_{x,y}[\|G(x) - y\|_1], \tag{2}$$

$$L_{adv} = E_{x,y}[logD(x, y)] + E_x[log(1 - D(x, G(x))], \tag{3}$$

where x and y are paired sample images in two different domains, G and D are the generator and discriminator respectively.

In addition, we propose to optimize the generator with a loss specific for PET harmonization, we call it SUVR loss. It focuses on forcing the generator to learn to match the SUVR distribution of the target domain.

$$L_{SUVR}(G) = E_{x,y}[|SUVR_{G(x,y)} - SUVR_y|] \tag{4}$$

The generator and discriminator were trained simultaneously in rivalry, where the generator is trained to

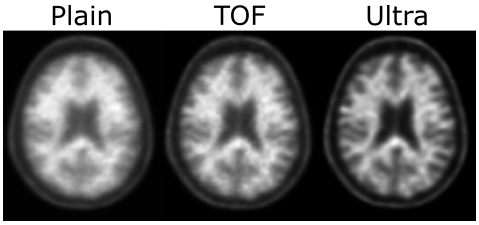


Figure 5: PET sample image of Recon-AIBL for the three Reconstruction methods: Plain, TOF and Ultra

minimize the adversarial loss while the discriminator tries to maximize it. The overall loss function can be written as:

$$L = \arg\min_G \max_D L_{adv}(G, D) + \lambda_{L1} L_{L1}(G) + \lambda_{SUVR} L_{SUVR}(G) \quad (5)$$

The network architecture is similar to the originally proposed in pix2pix but in 3D, where the generator is composed of an encoder-decoder U-Net architecture. The skip connections help in preserving similar content as that of the source domain . On the other hand, the discriminator, is a PatchGAN CNN, which divides the input image into patches and classify them into real or fake instead of classifying the whole image. This approach helps in focusing on small details of the target domain.

For all experiments, pix2pix was trained with Adam as an optimizer and a starting learning rate of 0.0002 and linearly decaying to 0 over 200 training epochs. λ_{L1} and λ_{SUVR} were set to 100 and 1 respectively.

3.1.2. CycleGAN

Having the same subjects imaged on different PET scanners or using different tracers is rarely feasible, as PET is an invasive imaging method that exposes each subject to a significant amount of radiation. To overcome this limitation, a 3D CycleGAN implementation was adopted. CycleGAN leverages two cGAN networks to generate pseudo pairs and learns a cyclic mapping between the two domains. As shown in figure 4b, Generator G_A learns the mapping function from domain x to domain y, while Generator G_B learns the mapping from domain y to domain x.

As in the original CycleGAN, the two cGAN networks were being optimized simultaneously and work to improve each other during training. The generator and discriminator of each cGAN were optimized by an adversarial loss similar to that in pix2pix. The adversarial loss for the forward mapping can be written as follow:

$$L_{adv} = E_y[logD_A(y)] + E_x[log(1 - D_A(G_A(x))] \quad (6)$$

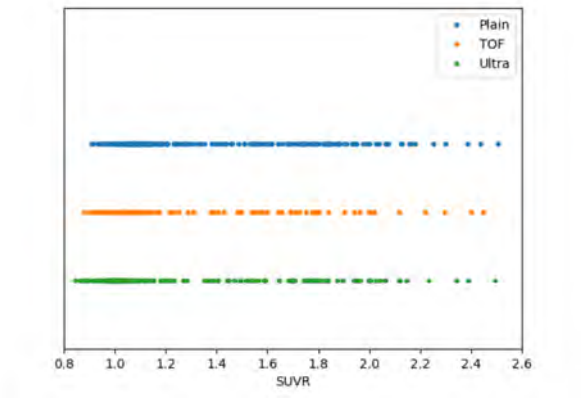


Figure 6: SUVR distribution in Recon-AIBL for Plain, TOF and Ultra

The same loss is applied for the inverse mapping, but with G_A and D_A replaced by G_B and D_B respectively. In addition, a cycle consistency loss is used to enforce a one-to-one mapping in the absence of paired images. It is defined as:

$$L_{cycle} = E_x[\|G_B(G_A(x)) - x\|_1] + E_y[\|G_A(G_B(x)) - y\|_1] \quad (7)$$

The full objective function can then be written as:

$$L_G = \arg\min_G \max_D L_{adv}(G_A, D_A) + L_{adv}(G_B, D_B) + \lambda_{cyc} L_{cycle}(G_A, G_B) \quad (8)$$

We adopt the same architecture as in the original CycleGAN, where the generator is composed of a two-level downsampling encoder, a bottleneck of 6 residual blocks and a two-level upsampling decoder. For the discriminator, a PatchGAN architecture was used.

CycleGAN was trained for 300 epochs for all experiments, and similar to pix2pix, Adam was used as an optimizer but with a fixed learning rate of 0.0001. λ_{cyc} was set to 100.

3.1.3. Star GAN

The presence of many scanner models and many amyloid tracers requires multi-domain PET harmonization. CycleGAN can only learn a two-ways mapping at one time. Star GAN was adopted in this work in order to handle multi-domain PET translation and allow for a generalized harmonization framework. StarGAN uses a single generator capable of learning the mapping across many domains (e.g. scanner models) by conditioning the generator on the class of the target domain when translating input images. The discriminator, on the other hand, has an auxiliary part to classify the inputs into their respective domains. Figure 4c shows the training diagram of the adopted StarGAN.

Table 1: Multi-AIBL demographics

Tracer	Allegro	Discovery	Gemini	Biograph
PiB	1006	-	2	495
AV45	317	273	40	-
NAV	106	108	580	112
FlUET	44	262	1	86
FBB	2	3	11	-

We trained StarGAN in a manner similar to that in Fixed-Point GAN (Siddiquee et al., 2019), where inputs are translated to both the same domain (identity mapping) and cross domains. This is achieved by conditioning the generator on the source domain label c_x in addition to the cross domains label c_y. We train our network with 4 objective functions similar to that in Fixed-Point GAN: an adversarial loss, a domain classification loss, a reconstruction loss and a conditional identity loss.

For the adversarial loss, Wasserstein GAN with gradient penalty was used, which was found to perform better than the Vanilla GAN originally proposed by Goodfellow et al. (2014). Therefore, the adversarial loss becomes:

$$L_{adv} = E_x[logD_{rf}(x)] - E_{x,c}[log(D_{rf}(G(x,c))] - \lambda_{gp}E_{\hat{x}}[(\|\nabla_{\hat{x}}D_{GAN}(\hat{x})\|_2 - 1)^1], \quad (9)$$

where $c \in \{c_x, c_y\}$ is the class of the target domain, D_{rf} is the discriminator's part for real/fake classification, and $\hat{x}$ is uniformly sampled along a straight line between a pair of real and fake images.

The domain classification loss is of two sides, one is optimizing the discriminator for real images L^r_{dom}, and the other is optimizing the generator for fake images L^f_{dom}.

$$L^r_{dom} = E_{x,c}[-logD_{dom}(c_x|x)] \quad (10)$$

$$L^f_{dom} = \sum_{c\in\{c_x,c_y\}} E_{x,c}[-logD_{dom}(c|G(x,c))] \quad (11)$$

The reconstruction loss is similar to the cycle consistency loss in CycleGAN. It uses an L1 loss between the input image and the fake image conditioned first on c for the forward mapping and then c_x for the inverse mapping:

$$L_{recon} = \sum_{c\in\{c_x,c_y\}} E_{x,c_x,c}[\|G(G(x,c),c_x) - x\|_1] \quad (12)$$

The conditional identity loss forces the generator to retain the domain identity when performing same domain translation. Hence, same domain translation is optimized while cross domain translation is regularized:

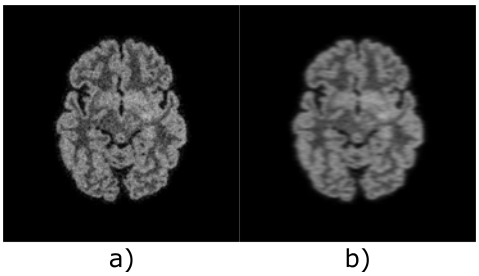


Figure 7: sample synthetic image of Brats 2013 used for evaluating Smoothing-cycleGAN: (a) is the source image, and (b) is the target image with the source image being smoothed to 5mm and 1.5mm in the x and y axis respectively

$$L_{id} = E_{x,c}[\|G(x,c) - x\|_1] \quad if c = c_x, 0\ otherwise \quad (13)$$

Therefore, the overall objective functions of the discriminator and generator are defined as follow:

$$L_D = -L_{adv} + \lambda_{dom}L^r_{dom} \quad (14)$$

$$L_G = L_{adv} + \lambda_{dom}L^f_{dom} + \lambda_{recon}L_{recon} + \lambda_{id}L_{id} \quad (15)$$

The network architecture of the generator and the discriminator is exactly the same as that of CycleGAN. For all experiments, StarGAN was trained for 750 epochs with Adam as an optimizer and a starting learning rate of 0.0001 and linearly decaying. λ_{gp}, λ_{recon}, λ_{cls} were set to 10, 10 and 1 respectively.

3.1.4. Multi-cycleGAN

In PET harmonization, we are mainly interested in bringing all scanner models and tracer types into one common scanner model and tracer, and not in translating between each other. Therefore, we implemented a variant of CycleGAN that does a mapping from many-to-one and one-to-many instead of many-to-many as in StarGAN. Multi-cycleGAN has two generators with the first generator conditioned on the common target domain label and the second generator conditioned on the source domains label with the same training settings as in CycleGAN.

We also optimized the network with the same objectives as in CycleGAN, but with the addition of the domain classification loss as in StarGAN. The network architecture of the generator and discriminator is similar to that of CycleGAN and StarGAN. For all experiments, we trained Multi-cycleGAN for 300 epochs and with the same hyper-parameters used in CycleGAN.

3.1.5. Smoothing-cycleGAN

cGANs, similar to other deep neural networks, are considered to be a black-box. In a PET image-translation task, it is hard to understand what scanner

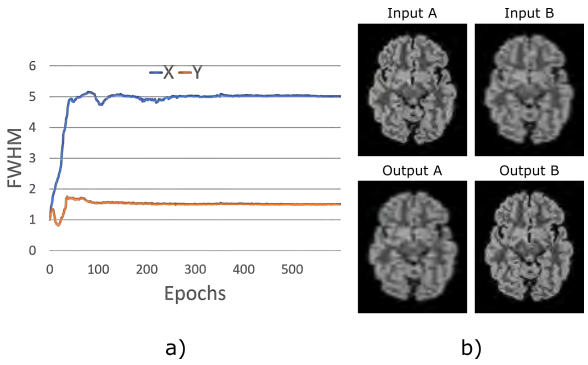


Figure 8: Training Smoothing-cycleGAN on synthetic Brats. (a) Shows the optimization of the 2D smoothing parameters in an unsupervised manner. (b) presents visual results of Smoothing-cycleGAN showing an input and an output image of the smoothing kernel (generator A) and the deconvolution network (generator B)

differences are being picked up by the network and why. Hence, we propose Smoothing-cycleGAN to address this limitation by modeling one of the differences across scanner models that is spatial resolution. We aim at bringing all scanner models into a common spatial resolution. Our framework is similar to that of cycleGAN shown in Figure 4b but with the first generator replaced by a smoothing kernel.

The smoothing kernel is only one *conv3d* layer that convolve the input with a 3D Gaussian filter:

$$g(xy, z) = \frac{1}{2\pi\sigma^2}.\exp -(\frac{xy^2}{2\sigma_{xy}^2} + \frac{z^2}{2\sigma_z^2}), \qquad (16)$$

where σ_{xy} and σ_z are the standard deviation in the x-y axes and z axis respectively. Therefore, we have only two learnable parameters (σ_{xy} and σ_z), since the PSF of PET scanners is mainly isotropic for x and y axis. We can then represent these parameters in terms of FWHM as:

$$FWHM = 2\sqrt{2\ln 2}\sigma \qquad (17)$$

We adapt the discriminator of the smoothing kernel to have only 3 *3dconv* layers with batch normalization and *leakyRelu* activation in between. For the deconvolution path (i.e. inverse mapping), we used the same architecture as in pix2pix.

Two training setups for the network were implemented: One that has only the smoothing kernel and its corresponding discriminator (Smoothing-singleGAN). It was used to determine the optimum FWHM values using paired images, and the second is the full cyclic network proposed for unsupervised harmonization (Smoothing-cycleGAN). We trained the latter for 600 epochs, with the smoothing kernel being optimized every 10 epochs in order to allow the deconvolution generator to converge first since it has many more parameters. The same loss functions as in CycleGAN was used to optimize the network, but with the gradient flow in

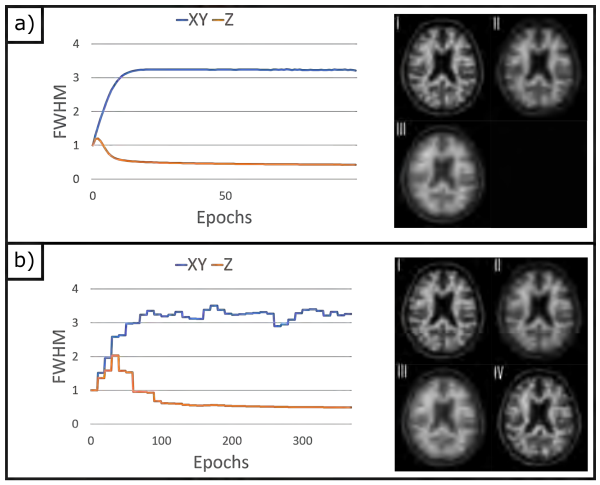


Figure 9: Training Smoothing-cycleGAN on Recon-AIBL. (a) and (b) shows the training of Smoothing-singleGAN and Smoothing-cycleGAN respectively. The optimization of the 3D smoothing kernel parameters is shown on the left side, and the visual results on the right side. Images I and II are the input and output of the smoothing kernel, while III and IV are the input and output of the deconvolution generator

the cycle consistency loss detached for each generator to accommodate for the architecture imbalance. For training we used Adam as an optimizer with a learning rate of 0.001 for the smoothing kernel generator and 0.0001 for the discriminators and the 2^{nd} generator.

3.2. Data

Data used in this work comes from the Australian Imaging, Biomarkers and Lifestyle Flagship Study of Aging (AIBL) (http://www.aibl.csiro.au/) (Ellis et al., 2009). AIBL is a longitudinal study aimed at providing a large-scale dataset to assist in AD research. It has a large number of paired MRI and Aβ PET images for both healthy controls and AD patients. In this work, MRI scans were spatially normalized to the MNI-152 template using the SPM8 unified segmentation method (Klunk et al., 2015). The transformation parameters from the MRIs were then used to co-register their corresponding PET images to the normalized space. After normalization, PET images had an isotropic voxel spacing of 2mm and a dimension of 91x109x91 in x, y and z respectively. PET images were then zero-padded to 128x128x128 and normalized to between -1 and 1 for all experiments. Brain symmetry around the y-axis was leveraged for augmentation by flipping PET images around the y-axis.

For this work two main sub-dataset of AIBL were used:

Recon-AIBL: It contains PET images scanned by a Siemens Biograph scanner but were reconstructed with three different reconstruction methods, namely, Ultra, Plain and Time of Flight (TOF). Plain was reconstructed using OSEM3D algorithm, while TOF was re-

Table 2: Comparison of the SUVR agreement for all approaches in Recon-AIBL for Ultra-Plain harmonization. Methods are compared in terms of MAE(SD) of the difference in SUVR values before and after image translation

Approach		Ultra to Plain	Plain to Ultra
w/o harmonization		0.059 (0.026)	0.059 (0.026)
Supervised	pix2pix w/o SUVR loss	0.015 (0.014)	0.021 (0.015)
	pix2pix w/ SUVR loss	**0.011 (0.009)**	**0.012 (0.009)**
Unsupervised	CycleGAN	**0.015 (0.018)**	**0.015 (0.020)**
	StarGAN	0.017 (0.009)	0.030 (0.014)
	Multi-cycleGAN	0.023 (0.010)	0.020 (0.010)
	Smoothing-cycleGAN	0.048 (0.016)	0.026 (0.019)

constructed using OSEM3D + Time of Flight, and Ultra was reconstructed using PSF + Time of Flight.

The dataset has about 549 PET scans from 165 subjects. Scans are divided across the three reconstruction methods as follow: Plain has 267 scans, ultra 161 and TOF 121 scans. The number of paired scans across the three Reconstruction methods is 86 scans, while ultra and plain shares 160 paired scans. Recon-AIBL has been used in supervised and unsupervised settings and the presence of paired scans was utilized for the validation of both settings. Figure 5 shows a sample of paired scans across the three reconstruction methods.

In Recon-AIBL, the number of negative AD subjects is much more than the number of positive AD patients. Figure 6 shows the distribution of the SUVR values for each reconstruction method, which reflects the ratio of healthy controls to AD patients.

Multi-AIBL: A multi-scanner multi-tracer dataset, where the subjects have been imaged by 5 different tracers: PiB, AV45, NAV, FLUTE, and FBB, and 4 different scanners: Philips Allegro, GE Discovery, Philips Gemini, and Siemens Biograph. Table 1 shows the demographics of the dataset. However, in contrast with Recon-AIBL, Multi-AIBL has no paired PET scans for the different scanners and tracers, therefore, only unsupervised methods have been evaluated for this dataset.

Smoothing-cycleGAN was also evaluated on a 3^{rd} synthetic dataset, Brats 2013 (Menze et al., 2014), which contains synthetic MRI brain images. It was utilized to demonstrate the capability of Smoothing-cycleGAN in estimating the PSF required to harmonize between two domains of different spatial resolution. For this dataset, the mid-slice of 150 synthetic MRI images were extracted and a Gaussian filter of 5mm and 1.5mm FWHM in the x and y axis was applied in order to generate the target domain images of a lower spatial resolution. Figure 7 shows a sample image for the source and target smoothed domains.

3.3. Evaluation metrics

In this work, the main metric used for evaluating the different harmonization approaches is the SUVR, which was calculated as the ratio of the mean signal intensity reflecting the amyloid uptake in the cortical region to that in the whole cerebellum. For Recon-AIBL,

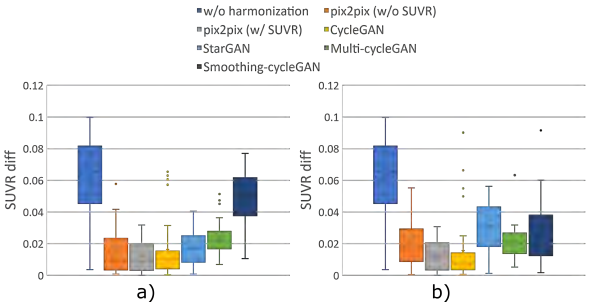


Figure 10: Box plots comparing all approaches for the Ultra-Plain harmonization. (a) shows image translation from Ultra to Plain, and (b) from Plain to Ultra

the SUVR values were measured before and after harmonization and the results' agreement was assessed by the Mean Absolute Error (MAE) and Standard Deviation (SD) of the difference in SUVR values. The slope and intercept of the SUVR regression line and Bland-Altman plots were further used to assess the harmonization of the implemented approaches. For Multi-AIBL, the performance of the proposed approaches have been assessed qualitatively by visual assessment of the PET scans.

4. Results

4.1. Smoothing-cycleGAN evaluation on Brats Synthetic dataset

As a proof of concept, a 2D Smoothing-cycleGAN was first evaluated on the Brats 2013 synthetic dataset. It was trained to estimate the PSF smoothing parameters (i.e. FWHM values) required to bring the original brain MRI images to the spatial resolution of their smoothed counterparts. As shown in Figure 8a, Smoothing-cycleGAN optimized the parameters of the smoothing kernel and reached equilibrium when the FWHM values are equal to 5mm and 1.5mm for the x and y axis respectively, which were the FWHM values of the target domain images. Visual results of the inputs and outputs of the network is shown in Figure 8b.

4.2. Networks evaluation on Recon-AIBL

4.2.1. Smoothing-cycleGAN PSF optimization

For Smoothing-cycleGAN, the optimum PSF parameters for bringing Ultra to the spatial resolution of Plain were first determined using Smoothing-singleGAN with paired data. Figure 9a shows the training optimization of the FWHM values for paired Ultra-Plain harmonization. The smoothing kernel reaches equilibrium when the FWHM values of the smoothing kernel are equal to 3.2mm and 0.5mm for x-y and z axes respectively. As shown in Figure 9b, Smoothing-cycleGAN with unpaired data was capable of optimizing the PSF parameters to the same values in unsupervised learning. A sam-

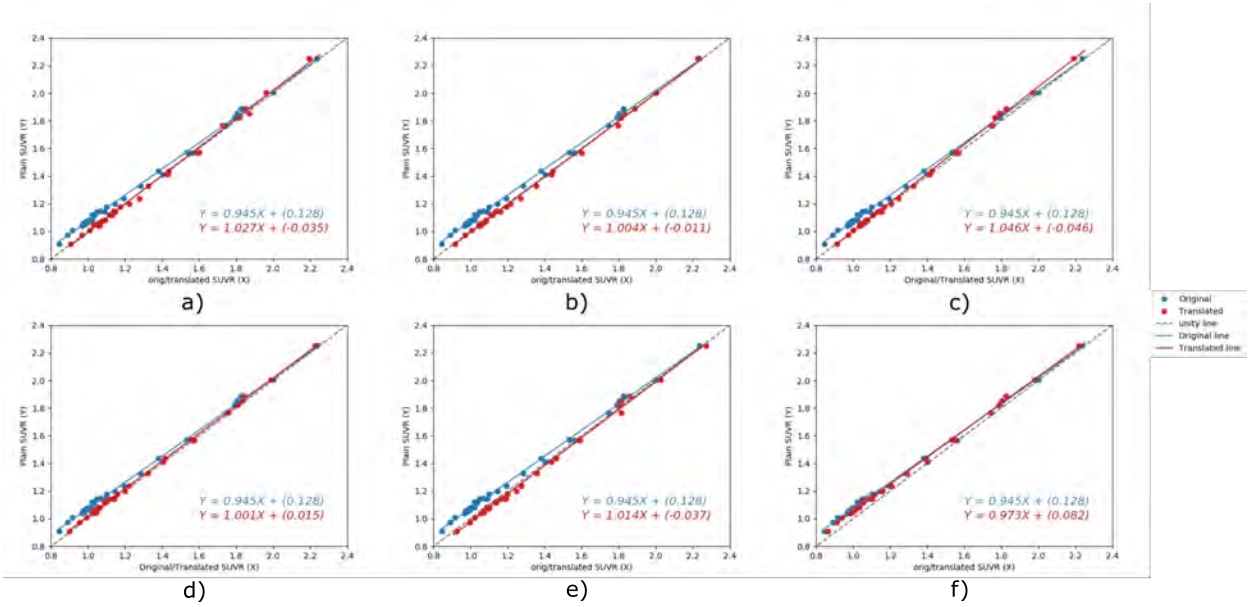


Figure 11: Scatter plots comparing all approaches for image translation from Ultra to Plain:(a) pix2pix w/o SUVR loss, (b) pix2pix w/ SUVR loss, (c) CycleGAN, (d) StarGAN), (e) Multi-cycleGAN and (f) Smoothing-cycleGAN

ple visual results of Smoothing-cycleGAN for Ultra-Plain harmonization is also shown in Figure 9.

4.2.2. one-to-one PET image translation

For the evaluation of all proposed approaches on Recon-AIBL, a subset of 32 images fore each reconstruction method were set aside and the rest were used for training. A comparison of all supervised and unsupervised approaches for Ultra-Plain harmonization is presented in Table 2. We chose Ultra-Plain harmonization to present because they have the highest disagreement among the three reconstruction methods. Table 2 compares the MAE and SD of the difference in SUVR values before and after harmonization. For the Supervised approaches we compared the performance of pix2pix with and without the proposed SUVR loss.Training pix2pix with the SUVR loss showed lower MAE compared to that without the SUVR loss, for both translating PET images from Ultra to Plain and vice versa. Figure 11a&b shows the scatter plot of the SUVR values for Ultra-Plain harmonization for pix2pix, with and without the proposed SUVR loss. The slope and intercept of the regression line for pix2pix with SUVR loss resulted in an overestimation of only 0.4% when translating from ultra to plain, compared to 5.5% without harmonization and 2.7% without the SUVR loss.

Table 3: Comparison of Multi-domain approaches for PET harmonization in Recon-AIBL in terms of MAE(SD) for the SUVR difference before and after image translation

Approach	Ultra to Plain	Plain to Ultra	TOF to Plain	Plain to TOF
w/o harmonization	0.059 (0.026)	0.059 (0.026)	0.036 (0.013)	0.036 (0.013)
StarGAN	**0.017 (0.009)**	0.030 (0.014)	0.015 (0.013)	0.011 (0.009)
Multi-cycleGAN	0.023 (0.010)	**0.020 (0.010)**	**0.012 (0.012)**	**0.008 (0.007)**

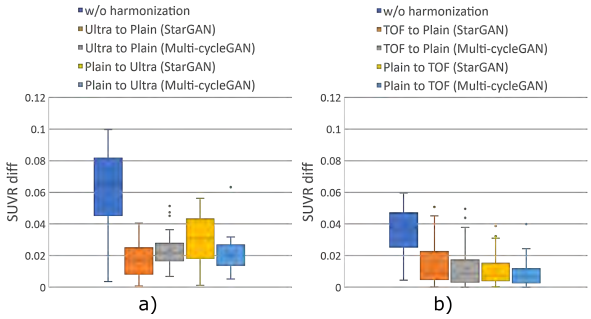


Figure 12: pox plots of StarGAN and Mulit-cycleGAN for multi-domain PET harmonization

For unsupervised learning, CycleGAN achieved the highest agreement in SUVR values after image translation compared to other approaches. While, smoothing-cycleGAN showed the lowest improvement in SUVR values agreement for the Ultra to Plain translation. The box plots of the difference in SUVR values are shown in Figure 10. Furthermore, Figure 11c,d,e&f compares the scatter plots for translating PET images from Ultra to Plain for unsupervised approaches. The slope of the regression line of StarGAN showed the lowest divergence from unity slope, with an overestimation of only 0.1%. CycleGAN, on the other hand, showed the highest divergence from the unity slope with an overestimation of 4.6%.

4.2.3. Multi-domain PET image translation

For multi-domain harmonization, StarGAN and Multi-CycleGAN were used for Ultra-Plain and TOF-Plain image harmonization. The Generator in StarGAN was conditioned to translate between the three recon-

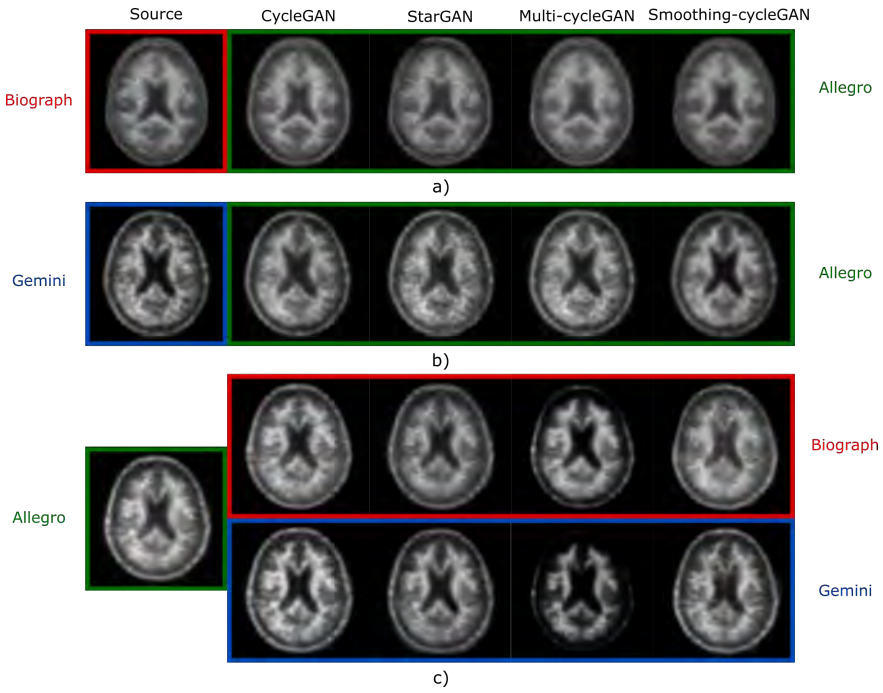


Figure 13: visual samples (unpaired) of translating images across scanner models in Multi-AIBL. (a) Biograph to Allegro, (b) Gemini to Allegro, (c) Allegro to Biograph and Gemini

struction methods randomly during training. In Multi-cycleGAN, the first generator was strictly conditioned on translating PET images to Plain only, and the second generator was conditioned to translate images back to both Ultra and TOF. The MAE and SD of the difference in SUVR before and after translation is presented in Table 1. The two approaches achieved comparable results for Ultra-plain harmonization, however, Multi-cycleGAN achieved better agreement in the SUVR values for TOF-Plain harmonization. The results in SUVR agreement for Ultra-Plain and TOF-Plain of both methods is further shown in the box plots in Figure 12.

4.3. Evaluation of networks on Multi-AIBL

For Multi-AIBL, since paired PET scans across the different scanner models were not available, only unsupervised approaches have been evaluated for this dataset. Furthermore, with the lack of paired PET images, only qualitative evaluation of the proposed approaches could be used for assessing their performance. For Multi-AIBL experiments, Allegro was set as the target domain and Biograph, Gemini, and Discovery as the source domains. To eliminate the variability in PET tracers and focus on only harmonizing between scanner models, tracer NAV was chosen since it has PET images of all the scanner models. 96 PET images of each scanner model were used for training CycleGAN, StarGAN, Multi-cycleGAN and Smoothing-CycleGAN. Figure 13a shows a sample image translated from Biograph to Allegro for all the unsupervised approaches, while Figure 13b shows the visual results of a sample image translated from Gemini to Allegro. Figure 13c shows a sample image of the inverse mapping that is translating PET images from Allegro to Biograph and Gemini.

5. Discussion

In contrast to the traditional way of doing PET harmonization, in this work we showed that deep learning, specifically cGANs, can learn to match the distribution of PET images across different domains, resulting in more consistent amyloid measurements. We alleviate the need to do PET harmonization during image reconstruction or the need to use phantoms as traditionally proposed (Senda, 2017) (Joshi et al., 2009) by building deep convolution networks from existing datasets. We demonstrated that the proposed networks for harmonizing PET images improve the SUVR agreement in Recon-AIBL between the translated images and their pairs in the target domain, as shown in Table 2 and Figure 10.

For pix2pix, the results in Table 2 shows the advantage of using the proposed SUVR loss, which forces the network to learn to match the SUVR distribution of the target domain. As shown in Figure 3, the SUVR loss helped in bringing the slope of the regression line close to that of the unity line. The SUVR loss mainly helped in reducing the class imbalance in the SUVR measurements by giving more attention to images with high SUVR values. This is more evident in the Bland-Altman plots in Figure 14, where the images with

high SUVR values have lower SUVR difference when pix2pix was trained with the SUVR loss. Pix2pix however, can only be applied in a supervised manner and when paired data are available.

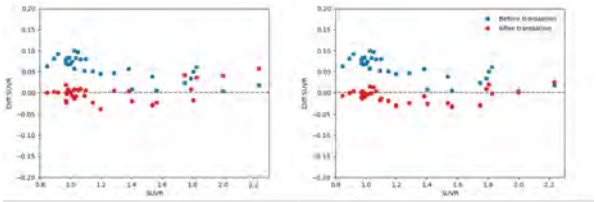


Figure 14: Bland-Altman plots for translating images from Ultra to Plain using pix2pix (a) with out SUVR loss, and (b) with SUVR loss

For unsupervised mapping, CycleGAN showed good performance in bringing the SUVR values into good agreement for Recon-AIBL, as shown in Table 2 and Figure 10. It showed the lowest MAE among the unsupervised approaches. However, because of the class imbalance in Recon-AIBL, CycleGAN seemed to perform well in translating images with low SUVR values, while failing to bring images with high SUVR values into good agreement with those of the target domain. In fact, CycleGAN resulted in a greater misalignment of the SUVR values for subjects with high SUVR (AD patients). This bias in the SUVR harmonization is demonstrated by the CycleGAN's 4.6% overestimation in the slope of the regression line in Figure 11. This is more evident in the Bland-Altman plots in Figure 15.

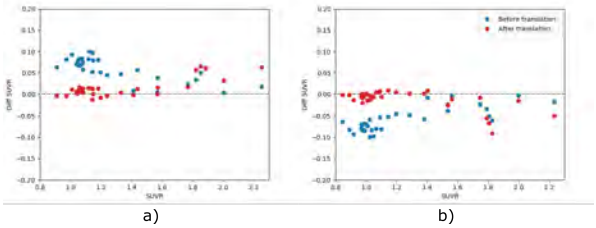


Figure 15: Bland-Altman plots for Ultra-Plain harmonization using CycleGAN:(a) Ultra to Plain, (b) Plain to Ultra

In contrast to CycleGAN, StarGAN and Multi-cycleGAN demonstrated good performance in translating images with high SUVR, as shown by the slopes of their regression lines in Figure 11. They mainly take advantage of the shared latent space across multiple domains and the larger set of training inputs. While this improved the generalizability of the network and helped reducing the effects of class imbalance, it also affected their specificity, See the Bland-Altman plots of StarGAN for Ultra-Plain harmonization in Figure 16.

For Smoothing-cycleGAN, although it showed an inferior performance in harmonizing Recon-AIBL PET images, it offered an alternative explainable image translation in comparison to the black-box image translation of the other approaches. It specifically learns to model a smoothing function required to bring the spatial resolution of the source domain to that of the target domain. Conventionally, the parameters of the smoothing function are determined manually using a set of phantom images as in (Joshi et al., 2009). This is prone to errors since the number of phantoms is very small and they may not be a true representative of the real PET images. In Smoothing-cycleGAN, the network learns to optimize the PSF parameters from many real PET images, capturing the overall distribution of the data. Moreover, Smoothing-cycleGAN can be applied retrospectively on existing datasets where phantom scans are not available. As shown in Figure 9 Smoothing-cycleGAN successfully estimated the optimum PSF parameters required for harmonizing PET data that were of different spatial resolutions.

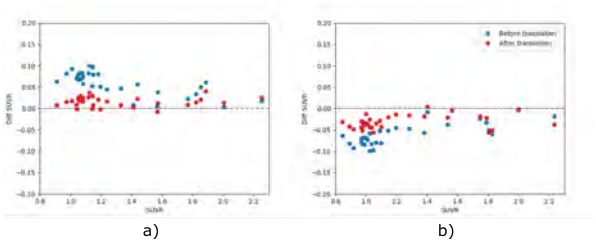


Figure 16: Bland-Altman plots for Ultra-Plain harmonization using StarGAN:(a) Ultra to Plain, (b) Plain to Ultra

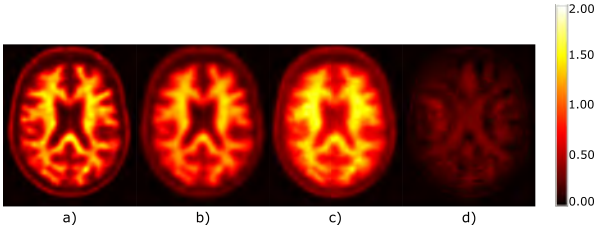


Figure 17: Image difference between the output of the smoothing kernel in Smoothing-cycleGAN and the target image. (a) smoothing kernel input, (b) smoothing kernel output, (c) target image, and (d) image difference of (b-c)

A similar structure to our Smoothing-cycleGAN, was proposed by Lim et al. (2020) for microscopy super-resolution. However, the focus of their work was on the deconvolution mapping, and not in the estimation of the optimum PSF parameters. Moreover, PET image harmonization differs in a way that it constitutes the harmonization of not only spatial resolution but also other factors, such as noise and contrast. Therefore, our problem is more complex, and the inverse mapping is not simply a deconvolution. This led to instability in the training of Smoothing-cycleGAN, where the smoothing kernel lacks the necessary complexity to bring the translated images fully into the distribution of the target domain. As a result, the second generator failed to keep a stable output because it was being misled by the output

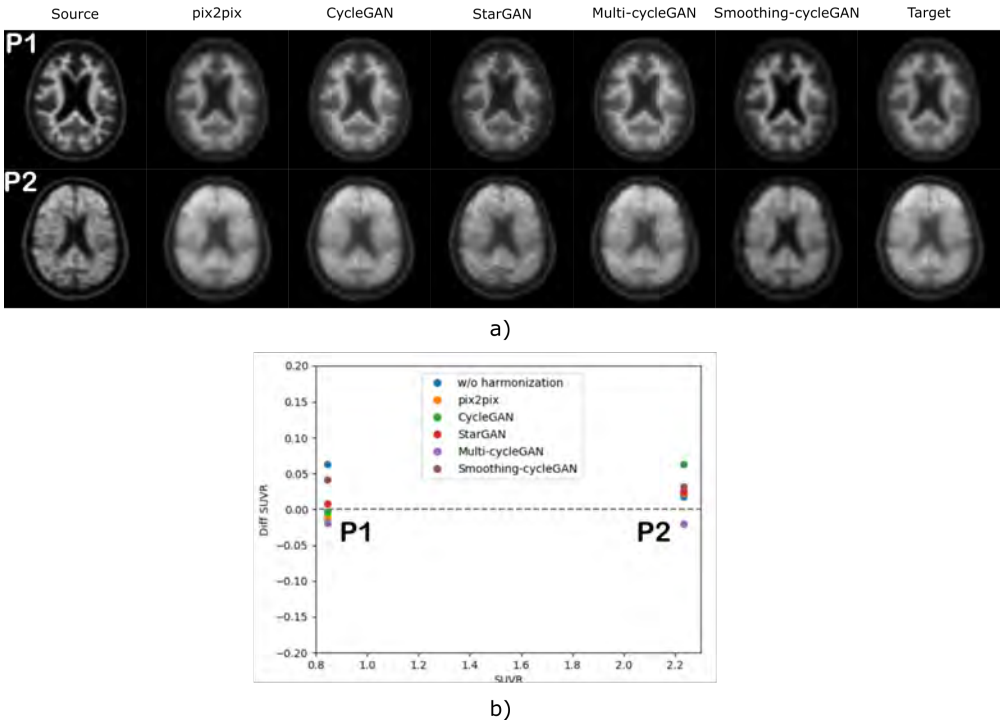


Figure 18: Visual assessment difficulty of cGANs-based PET harmonization approaches. (a) shows a visual output of the proposed approaches for two PET images, where the difference between them is hard to be noticed, while in (b) their SUVR measurements varies greatly

of the smoothing kernel in the cycle consistency loss. Two different approaches were evaluated to address this issue. First, the smoothing kernel was updated every 10 epochs instead of every epoch in order to increase the training stability. This has helped to account for the large number of tunable parameters in the deconvolution generator and led to a more stable training. Second, we tried to model the noise differences in the two domains, assuming it follows a Gaussian distribution, by adding a Gaussian noise layer with one parameter to learn. However, due to the random nature of the noise, the network ignored it and optimized its parameter to almost zero. Moreover, as shown in Figure 17 the differences between the output of the smoothing kernel and the target image is mainly structural and not because of noise. A possible solution is to add other layers to model the attenuation and scatter correction in addition to the smoothing kernel, similar to (Joshi et al., 2009).

For multi-AIBL, not having paired data prevented from evaluating the harmonization performance of the proposed approaches quantitatively. Qualitatively however, we can see from Figure 13 that visually CycleGAN and Smoothing-CycleGAN seemed to be able of learning the mapping from Gemini and Biograph to Allegro quite okay, and vice versa. However, For multi-domain mapping using StarGAN and Multi-cycleGAN, we notice that StarGAN showed a minimum translation to be noticed visually, while, Multi-cycleGAN appears to work well for the down-resolution mapping from Gemini and Biograph to Allegro but fails in translating Allegro back to the high resolution domains (Biograph and Gemini). This is mainly due to the transpose deconvolution layer in the upsampling part of the generator (Odena et al., 2016). A possible improvement to tackle this issue would be to replace the transpose deconvolution layers by an upsampling layer followed by convolution instead of doing upsampling and convolution in one layer.

Despite the great potential demonstrated for using deep learning cGANs for PET harmonization. This approach suffers from two main limitations:

- In computer vision, cGANs proposed for the task of image-to-image translation are being mainly evaluated by the realism of their generated outputs, which can be visually determined. However, this is not the case in PET harmonization. PET images are of low resolution and noisy. Therefore, the changes introduced by the different types of scanners, for example, are very subtle and cannot be easily noticed visually. However, quantitatively the effects can be significant enough to overestimate or underestimate the measures of amyloid retention. As reported by (Bourgeat et al., 2018), the distortion in the amyloid measurements tends to multiply when multiple factors come into play, such as changing the scanner and the tracer at the same time. To better explain this, Figure 18 shows a sample of two images translated from Ultra to plain using all proposed approaches. In Figure 18a

we can see that the output images of the different approaches look very similar, however, their SUVR uptake in Figure 18b shows quite a bit of variation.

- In the absence of paired images, unsupervised cGAN image translation is very unstable and output images can vary drastically from one epoch to the other. In computer vision applications, this behavior of non-deterministic mapping and variability in the generated outputs is generally encouraged. However, in PET harmonization this variation poses an issue of determining the optimum epoch for evaluation, given the sensitivity of PET harmonization to subtle changes. In an effort to address this issue, we tried to implement a cyclic SUVR metric, in which we compare the SUVR of the input image to that of the reconstructed one, similar to the cycle consistency loss. However, this approach is ill posed and led to choosing epochs with minimum translation of both the forward and inverse mapping.

6. Conclusions

This is the first work studying the possibility of utilizing deep learning networks, specifically, cGANs for unsupervised PET harmonization. We showed that good agreement in SUVR measurements can be obtained for the translated images using the proposed approaches. Furthermore, we showed that the PSF of PET images that have different spatial resolutions can be estimated automatically using Smoothing-cycleGAN, which offers an alternative for the phantom-based PSF estimation that is cumbersome and cannot be applied retrospectively on existing datasets with no reference phantoms. However, further work is required to improve the networks training stability and evaluation criteria for reproducibility and optimum PET harmonization.

7. Acknowledgments

Abdullah Thabit receives a grant from EACEA as part of the Eurpean Union Erasmus+ program. This work has been developed at the Australian e-Health Research Center (AEHRC) as the final project for a Joint Master Degree in Medical Imaging and Applications.

The author would like to express his deep appreciation for Dr. Pierrick Bourgeat for all his guidance and support throughout this project. He also would like to thank his academic professors for all the mentoring and teaching during this master. Thanks also goes to all MAIA friends for making the journey pleasant and memorable. Special thanks goes to Tewodros Arega and Zohaib Salahuddin for their immense and endless support over the past two years.

References

Anand, K., Sabbagh, M., 2017. Amyloid imaging: poised for integration into medical practice. Neurotherapeutics 14, 54–61.

Bahrami, K., Shi, F., Rekik, I., Shen, D., 2016. Convolutional neural network for reconstruction of 7t-like images from 3t mri using appearance and anatomical features, in: Deep Learning and Data Labeling for Medical Applications. Springer, pp. 39–47.

Bourgeat, P., Doré, V., Fripp, J., Ames, D., Masters, C.L., Salvado, O., Villemagne, V.L., Rowe, C.C., research group, A., et al., 2018. Implementing the centiloid transformation for 11c-pib and β-amyloid 18f-pet tracers using capaibl. Neuroimage 183, 387–393.

Bourgeat, P., Villemagne, V.L., Dore, V., Brown, B., Macaulay, S.L., Martins, R., Masters, C.L., Ames, D., Ellis, K., Rowe, C.C., et al., 2015. Comparison of mr-less pib suvr quantification methods. Neurobiology of aging 36, S159–S166.

Braak, H., Braak, E., 1991. Neuropathological stageing of alzheimer-related changes. Acta neuropathologica 82, 239–259.

Brock, A., Lim, T., Ritchie, J.M., Weston, N., 2016. Neural photo editing with introspective adversarial networks. arXiv preprint arXiv:1609.07093 .

Chen, H., Zhang, Y., Kalra, M.K., Lin, F., Chen, Y., Liao, P., Zhou, J., Wang, G., 2017. Low-dose ct with a residual encoder-decoder convolutional neural network. IEEE transactions on medical imaging 36, 2524–2535.

Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J., 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8789–8797.

Daerr, S., Brendel, M., Zach, C., Mille, E., Schilling, D., Zacherl, M.J., Bürger, K., Danek, A., Pogarell, O., Schildan, A., et al., 2017. Evaluation of early-phase [18f]-florbetaben pet acquisition in clinical routine cases. NeuroImage: Clinical 14, 77–86.

Degenhardt, E.K., Witte, M.M., Case, M.G., Yu, P., Henley, D.B., Hochstetler, H.M., D'Souza, D.N., Trzepacz, P.T., 2016. Florbetapir f18 pet amyloid neuroimaging and characteristics in patients with mild and moderate alzheimer dementia. Psychosomatics 57, 208–216.

Denton, E.L., Chintala, S., Fergus, R., et al., 2015. Deep generative image models using a laplacian pyramid of adversarial networks, in: Advances in neural information processing systems, pp. 1486–1494.

Dewey, B.E., Zhao, C., Reinhold, J.C., Carass, A., Fitzgerald, K.C., Sotirchos, E.S., Saidha, S., Oh, J., Pham, D.L., Calabresi, P.A., et al., 2019. Deepharmony: a deep learning approach to contrast harmonization across scanner changes. Magnetic resonance imaging 64, 160–170.

Dhikav, V., Sethi, M., Anand, K., 2014. Medial temporal lobe atrophy in alzheimer's disease/mild cognitive impairment with depression. The British journal of radiology 87, 20140150.

Dong, X., Lei, Y., Wang, T., Higgins, K., Liu, T., Curran, W.J., Mao, H., Nye, J.A., Yang, X., 2020. Deep learning-based attenuation correction in the absence of structural information for whole-body positron emission tomography imaging. Physics in Medicine & Biology 65, 055011.

Dou, Q., Chen, H., Yu, L., Qin, J., Heng, P.A., 2016. Multilevel contextual 3-d cnns for false positive reduction in pulmonary nodule detection. IEEE Transactions on Biomedical Engineering 64, 1558–1567.

Ellis, K.A., Bush, A.I., Darby, D., De Fazio, D., Foster, J., Hudson, P., Lautenschlager, N.T., Lenzo, N., Martins, R.N., Maruff, P., et al., 2009. The australian imaging, biomarkers and lifestyle (aibl) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of alzheimer's disease. International psychogeriatrics 21, 672–687.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H., 2017. Brain tumor

segmentation with deep neural networks. Medical image analysis 35, 18–31.

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S., 2017. Stacked generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5077–5086.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134.

Johnson, K.A., Minoshima, S., Bohnen, N.I., Donohoe, K.J., Foster, N.L., Herscovitch, P., Karlawish, J.H., Rowe, C.C., Carrillo, M.C., Hartley, D.M., et al., 2013. Appropriate use criteria for amyloid pet: a report of the amyloid imaging task force, the society of nuclear medicine and molecular imaging, and the alzheimer's association. Journal of Nuclear Medicine 54, 476–490.

Joshi, A., Koeppe, R.A., Fessler, J.A., 2009. Reducing between scanner differences in multi-center pet studies. Neuroimage 46, 154–159.

Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B., 2017. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical image analysis 36, 61–78.

Karacan, L., Akata, Z., Erdem, A., Erdem, E., 2016. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215 .

Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J., 2017. Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192 .

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 .

Klunk, W.E., Koeppe, R.A., Price, J.C., Benzinger, T.L., Devous Sr, M.D., Jagust, W.J., Johnson, K.A., Mathis, C.A., Minhas, D., Pontecorvo, M.J., et al., 2015. The centiloid project: standardizing quantitative amyloid plaque estimation by pet. Alzheimer's & dementia 11, 1–15.

Laforest, R., Khalighi, M.M., Byrd, D., Hongyu, A., Larson, P., Sunderland, J., Kinahan, P., Hope, T., 2018. Harmonization of pet image reconstruction parameters on multiple pet/mri system. Journal of Nuclear Medicine 59, 291–291.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al., 2017. Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4681–4690.

Lim, S., Park, H., Lee, S.E., Chang, S., Sim, B., Ye, J.C., 2020. Cyclegan with a blur kernel for deconvolution microscopy: Optimal transport geometry. IEEE Transactions on Computational Imaging 6, 1127–1138.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical image analysis 42, 60–88.

Liu, M.Y., Breuel, T., Kautz, J., 2017. Unsupervised image-to-image translation networks, in: Advances in neural information processing systems, pp. 700–708.

Liu, M.Y., Tuzel, O., 2016. Coupled generative adversarial networks, in: Advances in neural information processing systems, pp. 469–477.

Lowe, V.J., Lundt, E., Knopman, D., Senjem, M.L., Gunter, J.L., Schwarz, C.G., Kemp, B.J., Jack Jr, C.R., Petersen, R.C., 2017. Comparison of [18f] flutemetamol and [11c] pittsburgh compound-b in cognitively normal young, cognitively normal elderly, and alzheimer's disease dementia individuals. NeuroImage: Clinical 16, 295–302.

Masters, C.L., Simms, G., Weinman, N.A., Multhaup, G., McDonald, B.L., Beyreuther, K., 1985. Amyloid plaque core protein in alzheimer disease and down syndrome. Proceedings of the National Academy of Sciences 82, 4245–4249.

Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al., 2014. The multimodal brain tumor image segmentation benchmark (brats). IEEE transactions on medical imaging 34, 1993–2024.

Miao, S., Wang, Z.J., Liao, R., 2016. A cnn regression approach for real-time 2d/3d registration. IEEE transactions on medical imaging 35, 1352–1363.

Minoshima, S., Drzezga, A.E., Barthel, H., Bohnen, N., Djekidel, M., Lewis, D.H., Mathis, C.A., McConathy, J., Nordberg, A., Sabri, O., et al., 2016. Snmmi procedure standard/eanm practice guideline for amyloid pet imaging of the brain 1.0. Journal of Nuclear Medicine 57, 1316–1322.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 .

Mueller, S.G., Weiner, M.W., Thal, L.J., Petersen, R.C., Jack, C.R., Jagust, W., Trojanowski, J.Q., Toga, A.W., Beckett, L., 2005. Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging initiative (adni). Alzheimer's & Dementia 1, 55–66.

Ng, S., Villemagne, V.L., Berlangieri, S., Lee, S.T., Cherk, M., Gong, S.J., Ackermann, U., Saunder, T., Tochon-Danguy, H., Jones, G., et al., 2007. Visual assessment versus quantitative assessment of 11c-pib pet and 18f-fdg pet for detection of alzheimer's disease. Journal of nuclear medicine 48, 547–552.

Nie, D., Trullo, R., Lian, J., Wang, L., Petitjean, C., Ruan, S., Wang, Q., Shen, D., 2018. Medical image synthesis with deep convolutional adversarial networks. IEEE Transactions on Biomedical Engineering 65, 2720–2730.

Nordberg, A., 1992. Neuroreceptor changes in alzheimer disease. Cerebrovascular and brain metabolism reviews 4, 303–328.

Odena, A., 2016. Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583 .

Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill 1, e3.

Odena, A., Olah, C., Shlens, J., 2017. Conditional image synthesis with auxiliary classifier gans, in: International conference on machine learning, pp. 2642–2651.

Oktay, O., Bai, W., Lee, M., Guerrero, R., Kamnitsas, K., Caballero, J., de Marvao, A., Cook, S., O'Regan, D., Rueckert, D., 2016. Multi-input cardiac image super-resolution using convolutional neural networks, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 246–254.

Perry, E.K., 1986. The cholinergic hypothesis—ten years on. British Medical Bulletin 42, 63–69.

Prince, M., Bryce, R., Albanese, E., Wimo, A., Ribeiro, W., Ferri, C.P., 2013. The global prevalence of dementia: a systematic review and metaanalysis. Alzheimer's & dementia 9, 63–75.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 .

Rowe, C.C., Villemagne, V.L., 2013. Brain amyloid imaging. Journal of nuclear medicine technology 41, 11–18.

Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J., 2017. Scribbler: Controlling deep image synthesis with sketch and color, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409.

Schmidt, M.E., Chiao, P., Klein, G., Matthews, D., Thurfjell, L., Cole, P.E., Margolin, R., Landau, S., Foster, N.L., Mason, N.S., et al., 2015. The influence of biological and technical factors on quantitative analysis of amyloid pet: points to consider and recommendations for controlling variability in longitudinal data. Alzheimer's & Dementia 11, 1050–1068.

Senda, M., 2017. Standardization and quality control of brain pet data in a multicenter study, in: Neuroimaging Diagnosis for Alzheimer's Disease and Other Dementias. Springer, pp. 269–279.

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging 35, 1285–1298.

Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., Samaras, D., 2017. Neural face editing with intrinsic image disentan-

gling, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5541–5550.

Siddiquee, M.M.R., Zhou, Z., Tajbakhsh, N., Feng, R., Gotway, M.B., Bengio, Y., Liang, J., 2019. Learning fixed points in generative adversarial networks: From image-to-image translation to disease detection and localization, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 191–200.

Suppiah, S., Ching, S.M., MFamMed, A.J.N., MRad, S.V., 2018. The role of pet/ct amyloid imaging compared with tc99m-hmpao spect imaging for diagnosing alzheimer's. Med J Malaysia 73, 147.

Taigman, Y., Polyak, A., Wolf, L., 2016. Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200 .

Tsutsui, Y., Daisaki, H., Akamatsu, G., Umeda, T., Ogawa, M., Kajiwara, H., Kawase, S., Sakurai, M., Nishida, H., Magota, K., et al., 2018. Multicentre analysis of pet suv using vendor-neutral software: the japanese harmonization technology (j-hart) study. EJNMMI research 8, 1–10.

Wilson, R.S., Segawa, E., Boyle, P.A., Anagnos, S.E., Hizel, L.P., Bennett, D.A., 2012. The natural history of cognitive decline in alzheimer's disease. Psychology and aging 27, 1008.

Wimo, A., Winblad, B., Aguero-Torres, H., von Strauss, E., 2003. The magnitude of dementia occurrence in the world. Alzheimer Disease & Associated Disorders 17, 63–67.

Wolterink, J.M., Dinkla, A.M., Savenije, M.H., Seevinck, P.R., van den Berg, C.A., Išgum, I., 2017a. Deep mr to ct synthesis using unpaired data, in: International workshop on simulation and synthesis in medical imaging, Springer. pp. 14–23.

Wolterink, J.M., Leiner, T., Viergever, M.A., Išgum, I., 2017b. Generative adversarial networks for noise reduction in low-dose ct. IEEE transactions on medical imaging 36, 2536–2545.

Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M.K., Zhang, Y., Sun, L., Wang, G., 2018. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. IEEE transactions on medical imaging 37, 1348–1357.

Zhao, J., Mathieu, M., LeCun, Y., 2016. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 .

Zhou, L., Schaefferkoetter, J.D., Tham, I.W., Huang, G., Yan, J., 2020. Supervised learning with cyclegan for low-dose fdg pet image denoising. Medical Image Analysis , 101770.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, pp. 2223–2232.

Medical Imaging and Applications

Master Thesis, August 2020

Pediatric Bone Age Assessment of X-Ray images based on Detection of Ossification Regions

Esteban Alejandro Vaca Cerda, Adriyana Danudibroto

AGFA Radiology Solutions, Septestraat 27, 2640 Mortsel, Belgium,

Abstract

Bone age or skeletal maturity assessment is a common practice in paediatrics. It aims to evaluate children development and diagnose pediatric syndromes or growth disorders. The general practices for determining the skeletal maturity are the Greulich and Pyle (G&P) atlas and the Tanner Whitehouse (TW) method, both use hand X-Ray images and rely on visual evaluation of the radiograph. However, these standard methodologies have high inter and intra-rater variability. In order to overcome the variability, multiple automatic techniques based on image processing have been developed to aid physicians in accurately addressing skeletal maturity. This project presents an automatic deep-learning approach to perform bone age assessment of hand X-Ray images using the dataset provided by the Radiological Society of North America (RSNA), which includes 14,236 hand radiographs along with the gender and the skeletal age of each subject in months. The approach consists of a convolutional neural network which performs ossification region detection on the hand X-ray image. The model uses gender metadata to predict the bone age of the patient combining the predicted age of the detected regions. The metric used to evaluate the performance of this work is the mean absolute error (MAE) between the predicted and the actual bone age. We achieved an MAE of 4.139 months with a standard deviation of 3.84 and a coefficient of determination of $R^2 = 0.98$, which is comparable with the current state of the art for bone age assessment.

Keywords: bone age, ossification regions, hand X-rays, hand segmentation, object detection, active learning, CNNs

1. Introduction

1.1. Motivation

Bone age assessment (BAA) is a measure of bone development. It takes into account the size, shape and degree of ossification (mineralization) of the bones, which estimates the proximity of the bones to full maturity (Gilsanz and Ratib, 2005). During child development, multiple factors may cause a difference between the chronological age and the bone age, including, among others: paediatric syndromes, endocrine disorders, growth problems, poor nutrition or genetic diseases (Poznanski et al., 1978).

In paediatrics, the use of BAA has a broader application that is not limited to the detection of growth disorders but includes the detection of elite athletes up to civil registration programmes. In which, the application of BAA enables the verification of the age of children that do not have birth registration documents, allowing children to claim their rights (Mansourvar et al., 2013). In the last years, BAA has also been useful in the case of children seeking asylum, where it allows the correct allocation of resources (Creo and Schwenk, 2017).

The predominant clinical procedure of BAA is the atlas (G&P) by Greulich and Pyle (1959), which uses an X-ray of the left hand. The method consists of visual observation of the skeletal maturation of the whole hand. It compares a subject radiograph with an atlas which is a collection of hand radiographs of children gathered in the United States from 1931 to 1942; the physician assigns the bone age (BA) according to the most similar image in the atlas. Despite the popularity of this method, it does not set a standard procedure to assess the BA, and it depends on the expertise of the physicians. Thus it may be biased by human interpretation.

In order to overcome the drawbacks of the G&P atlas,

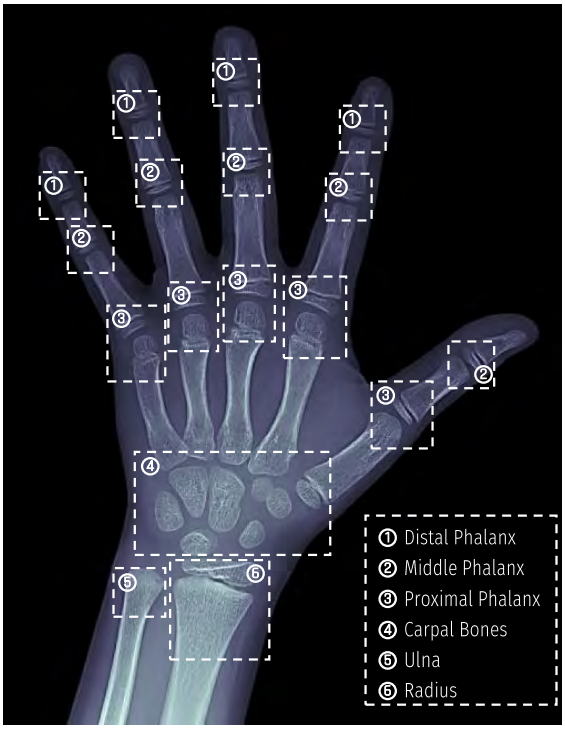


Figure 1: Regions of interest of TW2 method

TW methods were developed (TW2 and TW3) by Tanner et al. (2001). Initially designed in 1975, TW methods analyse particular bones of the hand including the radius, ulna, carpal bones of the wrist, metacarpals, and phalanges as depicted in Figure 1. The methods take into account metadata like the gender and the race of the subject. The study presented by Zhang et al. (2009) showed that there exist differences in growth patterns according to the ethnic and race of the subjects; such differences directly affect the BAA. Nevertheless, the G&P atlas did not acknowledge this fact.

Furthermore, TWs methods can predict the adult height with some additional information like the current subject height, the chronological age and some height coefficients. The TW methods subdivide the radiograph into several regions of interest (ROIs) and assign a maturity stage per ROI with a letter between (A, B, ..., I). They take into account race and gender information and some age coefficients and convert the marked stages into a numerical score. The resulting bone age is the addition of the numeric scores of the regions of interest (Spampinato et al., 2017).

Regardless of the acceptance of the G&P and TWs methods, both require thorough physician analysis and have a high dependency on the expertise of the radiologists, which in a practical setting, is very time-consuming. These facts, coupled with the growing demand for this analysis in the immigration and paediatrics, make these methods susceptible to high inter and intra-rater variability (Berst et al., 2001).

In this context, technological advances in computer vision have been used to reduce rating variability in BAA. Most of these advances are based on image processing aiming to replicate the clinical approach of G&P and TWs methods. One approach to ease BAA was the digital atlas of skeletal maturity presented by Gilsanz and Ratib (2005), which introduced a portable computer containing the G&P atlas to help the radiologists perform BAA manually. Towards automating BAA, several strategies were developed in recent years, most of them inspired by the TW methods (Mansourvar et al., 2013). Indeed, the sequential structure of the TW method based on the score of various ROIs enables the development of automated algorithms primarily focused on the localisation of the ROIs and performing feature extraction to predict the skeletal maturity of the subject.

The evolution of machine learning and deep learning has demonstrated performance improvements in medical applications such as computer-assisted diagnosis, image segmentation, and object localisation (Ching et al., 2018). In this context, numerous proposals have been developed to perform automatic BAA with deep-learning, as presented in the review by Dallora et al. (2019). The approaches range from the direct application of convolutional neural networks (CNNs) with a regression head to ensembles of predictions of the ROIs of the TW method.

1.2. Contribution

In this work, we developed two automatic methods for BAA based on CNNs. We investigated if the application of new cutting edge architectures can improve the performance on this problem. We developed an architecture capable of performing dense BAA on the whole hand and considering the TW ossification regions, which allows the prediction of the skeletal maturity of the subjects considering crucial local information. Thus, leveraging the possibility to perform multiple score fusion rules of the different ROIs, which has not mainly been studied by other methodologies.

1.3. Organisation

This work is structured as follows: section 2, presents a review of the state of the art for automatic BBA. Section 3 describes the materials and the development of the proposed BAA algorithms. Section 4, details the experiments performed with the proposed method. Section 5 contains a discussion of the obtained results, Finally section 6 presents conclusions and future work.

2. State of the art

The objective of the present work is to investigate possible improvements in BAA in hand X-rays using CNNs. According to Dallora et al. (2019), the most frequently used techniques for BAA rely on regression-based methods and deep-learning. Thus, in this section,

we present a review of the existing methods and their scopes. Furthermore, Table 1 presents a summary of the performance of the methods presented in this review.

2.1. Regression based methods for BAA

Based on the theory of G&P and TW methods, BoneXpert software developed by Thodberg et al. (2008) is one of the few available in the market and approved for distribution within Europe. It predicts automatic BAA through the use of an active appearance model. First, it performs the segmentation of 13 bones, followed by the extraction of features like shape, intensity and texture scores of the different ROIs, which enables the prediction of the bone age for each ROI in the X-ray. A consensus among the predicted age is calculated considering the gender of the patient. Nevertheless, a disadvantage of this framework is that it rejects low-resolution radiographs and does not takes into account the carpal bones of the youngest ages.

Another algorithm developed following the G&P and TW methods was presented by Seok et al. (2016). It performs the localisation of 17 ROIs and a relevant grouping of the regions. Then the bone age is calculated as a weighted sum of the different parts of the image using a least-square regression. This method mainly relies on decision rules and does not cover the whole paediatrics range [0-19 years]. Similarly, the method presented by Kashif et al. (2016), presents an extensive comparison of different well-known feature descriptors in computer vision that can be used to perform BAA, descriptors such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), BRIEF (Calonder et al., 2010) and BRISK (Leutenegger et al., 2011). The features were extracted within the ROIs defined in the TW method and subsequently, a support vector machine performed a regression to estimate the BA.

Mansourvar et al. (2015) presented an automated system of BAA based on extreme learning machines, which introduced the possibility to predict the BA without performing the segmentation of the hand or the extraction of ROIs. It used the features extracted from the full image and metadata of the gender and the race.

2.2. Deep Learning based BAA

The first approach that used CNNs for BAA was presented by Spampinato et al. (2017); they conducted a comparison of several architectures such as OverFeat by Sermanet et al. (2013), and GoogLeNet by Szegedy et al. (2015). Additionally, they developed the architecture called BoNet, which was trained using the digital hand atlas dataset released by Cao et al. (2003). BoNet uses the first convolutional layers of a pre-trained version of OverFeat. It uses a spatial-transformer layer (Jaderberg et al., 2015) to affinely transform the image, enabling the removal of any orientation misalignment of the hand in the X-Ray image. Afterwards, it uses an additional convolutional layer which then feeds a single fully-connected regression layer of 2048 neurons that predicts the BA. The performance achieved by this method is presented in Table 1.

Similarly, Larson et al. (2018) performed an automatic BAA based on the well-known architecture Resnet-50 designed by He et al. (2016). It introduced the large-scale dataset by RSNA containing a total of 14,236 clinical radiographs of the left hand obtained for BAA. Additionally, They show an evaluation with the digital hand atlas dataset where their method obtained improved the performance of BAA in such dataset. The metrics of this method are presented in Table 1.

The disadvantage of the methodologies of Larson et al. (2018) and Spampinato et al. (2017) is that they rely entirely on a transfer learning approach, and they did not explore further improvements in terms of image preprocessing or data augmentation. They also did not take advantage of the ROIs defined by the G&P and TW methods or any other combination. Instead, they used the whole image to transfer the knowledge from pre-trained networks that were trained on natural images from ImageNet (Deng et al., 2009).

Lee et al. (2017) introduced an approach to perform preprocessing for BAA for patients between 5 to 18 years. They used first a detection CNN (patch-based CNN) for tissue and bone by removing the background, collimation, and annotation marker. Once the image was segmented, they standardised it by applying contrast enhancement, denoising and edge sharpening. Finally, with the enhanced image, they performed transfer learning using GoogleNet. However, this research did not include samples of 0–4-year-old patients.

In 2017 the RSNA launched the pediatric bone age challenge, releasing a dataset of 14,236 images. Since the release of such dataset, multiple deep learning-based approaches have been published, especially in the context of the challenge where the winner (Bilbily and Cicerot, 2018) achieved an MAE of 4.26 months. The winner method consisted of first resizing all the images to 500×500 pixels. Then they used an InceptionV3 network (Szegedy et al., 2015) for feature extraction and, a subnetwork of 32-neuron dense layers for the gender information. The resulting features of the CNN and the gender subnetwork were concatenated. Finally, they used two 1000-neuron dense layers with ReLu activation to estimate the BA (Halabi et al., 2019).

Another approach published after the release of the RSNA dataset is the work of Iglovikov et al. (2018), which performed a multistage method aimed to improve the generalisation of the learning models and to increase the quality of the images. Their algorithm first performs segmentation of the hand in the X-Ray with the help of a U-Net (Ronneberger et al., 2015). To this end, they manually labelled some images, and they employed a technique called positive mining that combines man-

ual labelling with automatic processing. Once the images were segmented, they removed irrelevant information of the image such as background, collimation and orientation labels. The second stage of their proposal consisted of a keypoint detection algorithm based on the VGG architecture (Simonyan and Zisserman, 2014), which recognised 3 points on the hand (the centre of the carpal bones, the top of the thumb and the middle finger) which enabled them to perform an orientation alignment. Then, once the images were aligned and without background, they trained multiple regression models with the whole hand and with ROIs such as the carpal bones and the proximal phalanges. Finally, they explore possible ensembles of the different ROIs, which increased the performance of their models.

The complexity for segmenting the images presented in Iglovikov et al. (2018), inspired the works of Ren et al. (2018) and Wu et al. (2019), which replaced the segmentation preprocessing by attention modules as an initial input layer of a DensenetV3 model (Iandola et al., 2014), which performed the regression of the BA. These works did not align the images in a specific orientation, and they depended on data augmentation to handle the misalignments and the dose variation, which affected the contrast of the images. The results obtained by these works are presented in Table 1.

Towards improving the performance of BAA, Pan et al. (2019) investigated improvements that can be obtained through model ensembling. For this work, they took into account 48 submissions from the 2017 RSNA challenge. They conducted a bootstrap analysis using the 200 test images. They found that the average performance of a single model was 4.55 MAE in months. Meanwhile, the best-performing ensemble consisted of four models obtaining an MAE of 3.79 months. It is worth mentioning that in this work, the researchers did not train any model. Instead, they perform cross-validation of the top-5 solutions of the RSNA challenge.

Reddy et al. (2020) investigated the possibility to execute BAA using a reduced part of the hand X-Ray like the index finger region. They subsequently compared the BAA using the full hand, the cropped index finger region and the estimation of three radiologists. They found that the results using a cropped section of the X-ray are similar to using the whole hand. Furthermore, they demonstrated that the automatic approaches obtained higher confidence than radiologist predictions, since the algorithms lack of rater-variability. Their research opened a gate to perform BAA with the use of X-rays of only specific parts of the hand.

A relevant work for preprocessing the X-ray images towards BAA was published by Koitka et al. (2018). They opened a gateway to the research of two-stage systems that recognises ossification ROIs based on TW methods and BAA with the use of high-resolution patches of the image. They released a dataset with 240 manually annotated radiographs containing labels for regions of the distal, intermediate and proximal phalanges, metacarpals, the carpal bones, the ulna and the radius. Although BAA was not performed in their work, the automatic detection of ROIs is an essential step towards an explainable BAA method.

Table 1: Comparison of previous works in automatic BAA

Author	Method	Dataset	Dataset size	Age range	MAE (months)	Other metrics	Comment
Bon-eXpert, Thodberg et al. (2008)	Active appearance model	Multiple Sources	1,559	2–17	-	0.42 years MSE	Metric according G&P
						0.80 years MSE	Metric according TW2
		RSNA Dataset	12,480	0–19	4.50	-	RSNA Test Data
Mansourvar et al. (2015)	Extreme learning	Private	1,100	0-18	-	0.22 years MSE	-
Seok et al. (2016)	Feature based	Private	135	-	-	0.19 years MSE	-
Kashif et al. (2016)	Feature based	Private	1,100	0–18	7.26	-	-
Spampinato et al. (2017)	Deep-learning	Digital Hand Atlas Database System	1,391	0–18	9.48	-	-
Lee et al. (2017)	Deep-learning	Massachusetts General Hospital MGH	8,325	7-17	-	0.82 years RMSE	-
Zhao et al. (2018)	Deep-learning	RSNA Dataset	12,611	0–19	7.66	-	RSNA Validation
Larson et al. (2018)	Deep-learning	RSNA Dataset	12,611	0–19	7.32	-	RSNA Validation
Bilbily and Cicerot (2018)	Deep-learning	RSNA Dataset	12,480	0–19	4,27	-	RSNA Test Data
Ren et al. (2018)	Deep-learning	RSNA Dataset	12,480	0–19	5.2	-	-
Iglovikov et al. (2018)	Deep-learning	RSNA Dataset	11,600	0–19	7.52	-	RSNA Validation
				0–19	4,97	-	RSNA Test Data
Wu et al. (2019)	Deep-learning	RSNA Dataset	12,480	0–19	7.38	-	RSNA Validation
Pan et al. (2019)	Ensembling	RSNA Dataset	200	0–19	3.79	-	RSNA Test Data
Escobar et al. (2019)	Deep-learning	RSNA Dataset	12,480	0–19	4.41	-	RSNA Test Data
		Radiological Hand Pose EstimationDataset	6,288		6.86		-
Reddy et al. (2020)	Deep-learning	RSNA Dataset	12,480	0–19	4.7	-	One finger Prediction

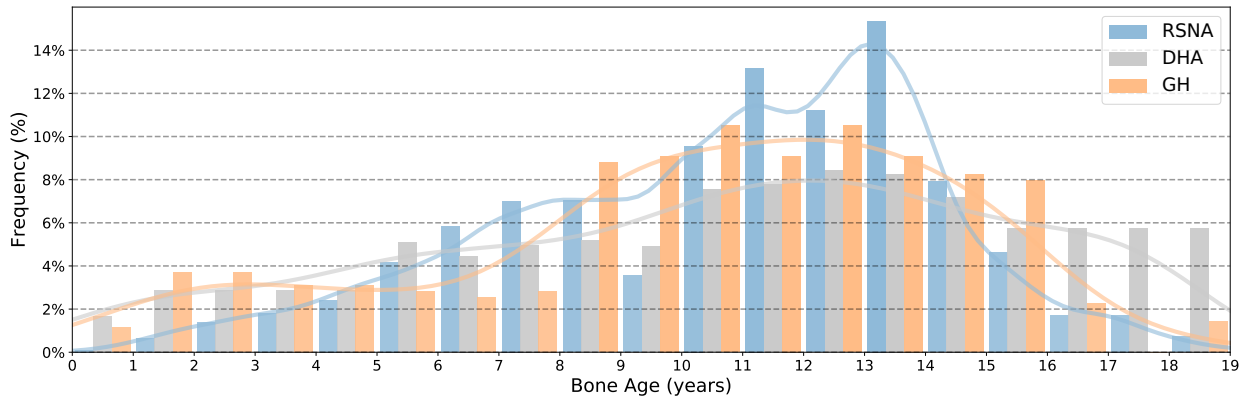


Figure 2: Bone age distribution for radiographs of the three datasets used in this work

The latest improvement towards BAA prediction was introduced by Escobar et al. (2019); they performed hand pose estimation for pediatric hand X-rays. In their work, they manually annotated keypoints of the RSNA dataset to detect the pose of the hand before assessing the skeletal maturity. Their approach consisted of three stages. First, they localised and cropped the region of the hand using a fast region-based convolutional network (Fast R-CNN) by Ren et al. (2015). The second stage estimates the pose of the hand using the baseline of human pose estimation and tracking proposed by Xiao et al. (2018). Finally, they performed BAA by modifying the method of the winner of the RSNA challenge (Bilbily and Cicerot, 2018). They additionally introduced the dataset Radiological Hand Pose Estimation (RHPE), a dataset that contains 6,288 images from a population with different characteristics of the RSNA dataset.

3. Material and methods

3.1. Datasets

There are three datasets used in this work. First, the dataset released by the RSNA for the Pediatric Bone Age Machine Learning Challenge in 2017, in which all the images have skeletal maturity and gender information. BA was assigned by multiple expert observations considering the rater variability and the bias produced by each expert. The RSNA dataset is subdivided into 12,611 X-ray images for training, 1425 for validation and 200 for testing.

The second dataset is the publicly available Digital Hand Atlas (DHA) by the University of Southern California Image Processing and Informatics Laboratory introduced in Gertych et al. (2007). It contains 1390 images; each of them contains the bone age in years, the gender and the race of the patients. Larson et al. (2018) shows that there is no patient overlap among the RSNA and the DHA datasets.

Table 2: Gender Distribution of the three datasets used in this work

Dataset	Subject Number			Age range (months)	Gender ratio (F:M)
	Female	Male	Total		
RSNA Train	6,488	6,123	12,611	[1-228]	48:51
RSNA Val	652	773	1,425	[3-228]	46:54
RSNA Test	100	100	200	[11-219]	50:50
DHA	690	700	1,390	[0-216]	50:50
GH	159	193	352	[0-216]	45:55
TOTAL	7,724	8,254	15,978	[0-228]	47:53

The third dataset is a private collection of the Gasthuisberg Hospital (GH) from the University Hospital Leuven - Belgium, compiled from 1998 to 2001. The dataset consists of 352 images of patients with growth disorders. The dataset bone age was estimated by two experts using the TW2 method assigning a score letter for the different ROIs of the hand; the dataset contains information about skeletal age, current age and the qualitative maturity score for all the ROIs in hand. The Digital Hand Atlas and the Gasthuisberg dataset are used only for testing the accuracy of the developed algorithm.

Figure 2 depicts the age distribution of the three datasets used in this project. It is possible to observe that there exists an imbalance in the age distribution. The number of samples of infants and toddlers [0-4 years] and late adolescents [17-19 years] is underrepresented compared with the samples available for mid-adolescence [12-15 years].

In addition to the bone age labels, the gender of the subjects is used in our methodology. The gender information has a high correlation with the BAA. Table 2 shows the distribution of the images according to the gender of the patients in the three datasets. It can be seen that the distribution concerning the gender of the subjects is almost balanced.

3.2. Methodology

This subsection describes the proposed strategies for BAA; we investigated whether the application of cutting edge architectures used for classification and object de-

tection of conventional images could improve the accuracy of BAA.

We evaluated different approaches for BAA not only using the whole image but also using the information of different anatomical structures from the hand. Therefore, we present three major tasks to perform BAA. First, we perform hand segmentation as preprocessing; second, we develop our first strategy based on feature fission and whole hand segmentation. Finally, we describe the design of an algorithm based on the ossification ROIs of the TW methods. In contrast with previous studies based on multiple-stage algorithms (Iglovikov et al., 2018), (Escobar et al., 2019), our introduced approach performs BAA at once and exploring the different ROIs in the X-ray images simultaneously.

Since the labels presented in the datasets include only the bone age of the different patients, we performed an unsupervised hand segmentation algorithm together with some deep learning-based segmentation methods. Furthermore, an active learning framework was implemented to label the ossification ROIs for the object detection task interactively.

3.3. Hand Segmentation

To perform hand segmentation, we modify the segmentation pipelines presented by Hsieh et al. (2012), which show a detailed methodology for phalanges segmentation. An overview of the unsupervised segmentation model implemented is presented in Appendix A.

Since our goal is to embed the segmentation of the hand within a deep-learning model (U-Net), we used positive mining similar to (Iglovikov et al., 2018). However, instead of manually segmenting the initial samples, we used the result of our unsupervised segmentation algorithm. Since the output of the unsupervised segmentation method contains some errors, we manually refined some of the generated masks to improve the performance of the segmentation models. Besides, we labelled some of the images that were not successfully segmented by the unsupervised method.

For performing segmentation, we adopted a modified version of U-net which contains an EfficientNet encoder (Tan and Le, 2019). We call this architecture as Efficient-UNet. EfficientNets have outperformed other well-known architectures in multiple tasks such as image classification and object detection. Furthermore, we selected EfficientNet as an encoder for the U-Net since once it is optimised to perform hand segmentation, the encoder weights can be transferred for other tasks such as BAA or object detection. An overview of a U-Net with an EfficientNet encoder applied to medical imaging was introduced by Nguyen et al. (2020).

The losses used to optimise the parameters of the segmentation models were cross-entropy loss,

$$\mathcal{L}_{BC}(y_i, \hat{y}_i) = \frac{-1}{n} \sum_{i=0}^{n} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\right), \quad (1)$$

and the dice loss:

$$\mathcal{L}_{Dice}(y_i, \hat{y}_i) = \frac{2 \sum_i^N \hat{y}_i y_i}{\sum_i^N \hat{y}_i^2 + \sum_i^N y_i^2}, \quad (2)$$

proposed by Milletari et al. (2016). This combination accounts for class imbalance without the need of assigning weights to the samples of specific classes. In equations (1) and (2), $\hat{y}_i$ corresponds to the predicted segmentation and y_i to the ground truth.

Thus, the total loss for the segmentation task is the sum of the cross-entropy and the dice loss:

$$\mathcal{L}_{seg} = \mathcal{L}_{BC} + \mathcal{L}_{Dice} \,. \quad (3)$$

During training, weights optimisation is performed using decoupled weight decay regularisation (AdamW) by Loshchilov and Hutter (2017) with an initial learning rate of 0.001. The architecture is initialised with the weights of ImageNet. To account for different X-ray variations such as collimation and radiation dose, we included data augmentation. Our augmentation pipeline includes image rotation within $-90°$ to $90°$, random horizontal and vertical image translation up to 20%, image scaling within the ratio [0.7 - 1], horizontal and vertical image flip, and image random brightness was applied with a probability of 50%. The augmentation is performed using the Albumentation library by Buslaev et al. (2020)

3.4. BAA based on Dense Predictions

The general pipeline for BAA with dense predictions consists of four main components. First, the EfficientNet backbone based on the trained hand segmentation model introduced in Section 3.3, second a BiFPN, then a metadata layer and finally, a dense regression convolutional network.

3.4.1. Bidirectional Feature Pyramid (BiFPN)

The features extracted for the segmentation task are reused to feed a BiFPN. Tan et al. (2019) compared different approaches for multi-scale feature fusion, such as conventional top-down feature pyramid by Lin et al. (2017a); PANet that adds an additional bottom-up pathway on top of FPN (Liu et al., 2018); and the FPN based in neural architecture search (NAS-FPN) by Ghiasi et al. (2019). In comparison, Tan et al. (2019) demonstrated that a better approach to fusing the multi-scale features effectively could be achieved by BiFPN, which has shown improvement in performing one-stage object detection. Figure 3 shows the architecture of BiFPN.

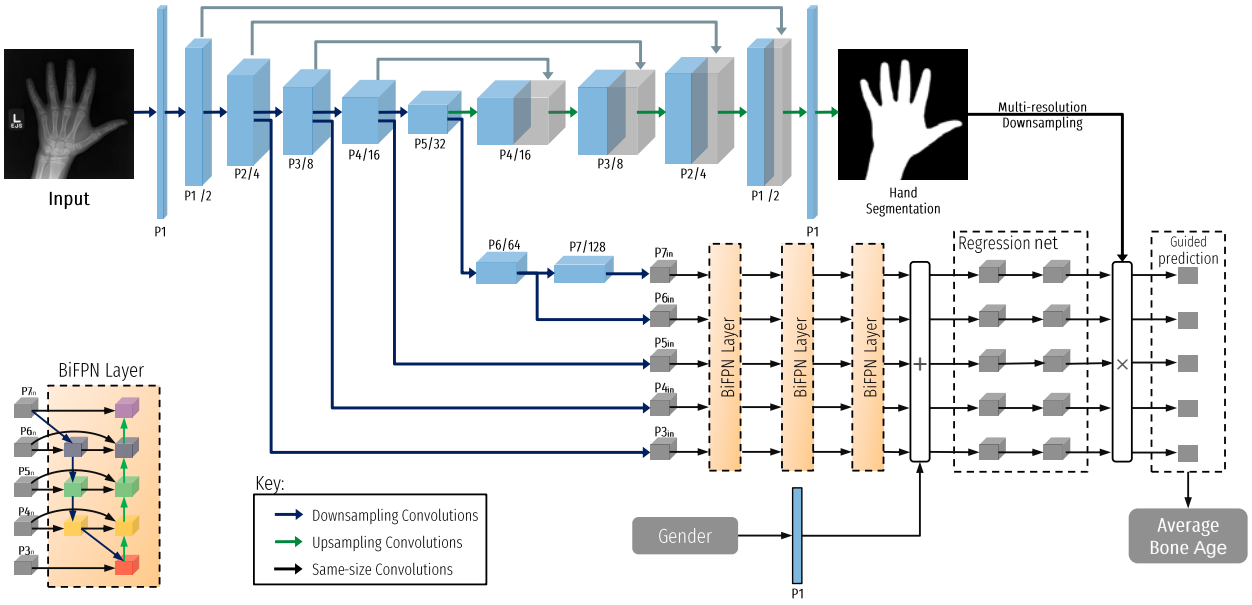


Figure 3: BAA proposed architecture – It employs EfficientNet as the backbone network, BiFPN as the feature network, a U-net based segmentation autoencoder, and regression prediction network. The final prediction is guided by the generated segmentation mask.

In contrast to conventional feature pyramids that are unidirectional, BiFPN performs top-down and bottom-up feature fusion, leveraging the multi-scale fusion compared to the limited one-way FPN. Additionally, BiFPN includes a weight to the different resolutions that enable the network to learn the best weights to combine the features during training (Tan et al., 2019).

3.4.2. Metadata

According to the TW methods of, gender is crucial metadata to be considered for BAA (Pietka et al., 2001). Therefore, in the winning solution of the RSNA challenge by Bilbily and Cicerot (2018), a layer of 32 neurons was used to merge gender information with 2048 features extracted from the image. In our approach, we designed the prediction network in a BiFPN respecting the compound scaling model of EfficientNets.

We used a fully connected layer with size according to the compound scaling concept of Tan et al. (2019). The features are pointwise added to each output of the BiFPN similar to the approach of Levine et al. (2018). The number of neurons applied for each EfficientDet model is presented in Table 3.

Table 3: EfficientDet parameters, the values presented in the table follow the design of Tan et al. (2019). The number of neurons adopted for the metadata respects the number of channels of the BiFPN.

Model	image size	#neurons metadata	#BiFPN		#box, regression, BA layers
			#channels	#layers	
D0	512	64	64	3	3
D1	640	88	88	4	3
D2	768	112	112	5	3
D3	896	160	160	6	4
D4	1024	224	224	7	4

3.4.3. Regression Network

Instead of the conventional fully connected final layer of regression and classification CNNs, we proposed a set of same size convolutional layers that performed multi-scale dense predictions of BA. This approach leveraged the possibility of not only considering the ROIs defined by conventional methods such as TW and G&P but also to search for other regions in the hand that could be important for BAA. Keeping attention only in the hand region and removing the markers and frames that affect the prediction.

The segmentation results of the Efficient-UNet guide the dense predictions. To this aim, the segmentation results are downsampled to match the dimension of the different feature maps obtained by the prediction network. This approach provides the possibility of only taking into account the pixels that lay inside the segmentation of the hand, hence, predicting only in hand regions of the images.

3.4.4. Training Overview

To optimise the parameters of the dense regression network, we used a modified version of the loss function employed in YOLO (You Only Look Once) network by Redmon et al. (2016). Equation (4) shows the proposed loss function:

$$\mathcal{L}_{reg}(y_i, \hat{y}_i) = \sum_{i=0}^{S^2} \mathbb{1}_i^{seg} |y_i - \hat{y}_i| , \tag{4}$$

where S^2 represents the multi-scale feature space that produces the network, $\mathbb{1}_i^{seg}$ is the segmentation response of the Efficient-UNet downsampled to the output scale,

y_i is the bone age ground truth, and $\hat{y}_i$ is the pixel-wise predictions of the network.

The loss presented in Equation (4) calculates the mean absolute error pixel-wise. The optimisation back-propagates the loss through the regression network and the BiFPN. We left the parameters of the EfficientNet backbone unchanged since they are shared with the hand segmentation task.

To alleviate the imbalance in the dataset depicted in Figure 2, we used random minority oversampling (Buda et al., 2018), which replicated randomly selected samples from minority classes. To effectively balance the distribution of the dataset, we resampled the number of images considering the bone age in intervals of six months. Additionally, to avoid the possibility of overfitting of random oversampling, we performed data augmentation (Buda et al., 2018), (Chawla et al., 2002).

The preprocessing pipeline and augmentation used for this task are as follows:

- Re-scale the image so that the maximum side is equal to the desired compound scale, keeping the aspect ratio of the initial X-ray.
- Random resize the input in the range of 70% to 100% of the original size.
- One random selected intensity transformation among random brightness contrast, sharpening or emboss.
- Random rotation of the image between $-180°$ and $180°$.

The different intensity augmentations allowed us to count for the dose variations on the dataset, and the broad rotation took care of images that are present in different orientations. Thus, we mitigate the rotation and translation invariance issues by using these transformations instead of registering all the images to a predefined orientation.

We optimised the loss function of Equation (4) with AdamW with an initial learning rate of 0.001 with a weight decay coefficient of 0.01, we reduce the learning rate on plateau with a weight decay of 0.1 when the validation loss has stopped improving for more than ten epochs. The training process is stopped when the validation loss stopped improving for 30 epochs.

3.5. BAA based on Ossification ROIs

In order to design a deep-learning version of the TW methods, we localise the epiphyseal growth-regions, also called ossification regions. The regions include four regions between the distal and intermediate phalanges (DIP), five spaces between the intermediate and proximal phalanges (PIP), the five regions between the proximal phalanges and the metacarpals (MCP) the carpal bones (Wrist), the radius and ulna.

The design of our object detection algorithm was based on the work presented by Koitka et al. (2018), who released the bone annotation of 329 X-ray images of the RSNA dataset. In their study, they localised the ossification regions of the hand with a region proposal network (RPN).

In recent years, new approaches have been proposed to perform object detection. EfficientDet by Tan et al. (2019) achieved the state of the art performance in the Common Objects in Context (COCO) dataset released by Lin et al. (2014), their strategy uses a one-stage detector with feature fusion, and classification/box subnetworks that is flexible to modifications. Therefore, we replaced the RPN proposed by Koitka et al. (2018) with an EfficientDet to localise the ossification regions.

Another limitation of the work of Koitka et al. (2018) lies in the number of annotated images which is small compared with the number of samples in the RSNA dataset. To overcome this issue, we performed active learning for object detection to increase the number of labelled data, as presented in section 3.5.1.

Once we labelled the ROIs of the whole dataset, we added a regression head in the EfficientDet architecture to predict the BA of the subjects. The new regression head follows the methodology presented in subsection

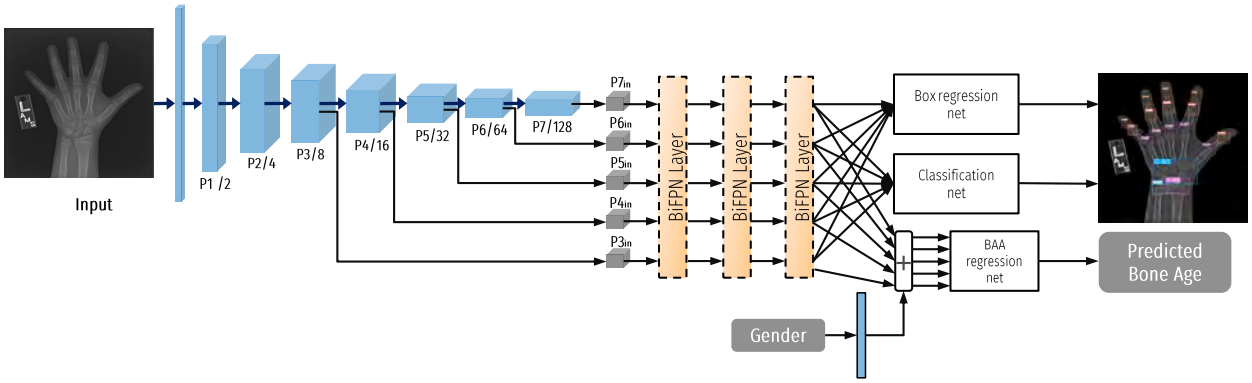


Figure 4: Modified version of the EfficientDet by Tan et al. (2019), we add an additional regression box to perform BAA, the box regression and classification heads allow the detection of the ossification regions.

3.4.2 for handling the metadata and the design of the regression head of subsection 3.4.3. The details of the active learning process and the training of the object detection network are presented in the following subsection.

Figure 4 presents a scheme of the proposed model. It contains three prediction heads including box regression, classification of the regions and BAA head. The presented model can be trained in an end to end way using the labels of the three task at the same time.

3.5.1. Active Learning for Detection of TW ROIs

To increase the number of available labels of the dataset released by Koitka et al. (2018), we adapted a procedure called localisation-aware active learning for object detection introduced by Kao et al. (2018), which is an active learning technique that enables automatic labelling of images with the predictions of a deep network by taking into account the localisation stability when noise is added to the unlabelled data pool.

A significant component of active learning is the image selection criteria, which allows to either reject or accept a predicted outcome. Thus, it is crucial to define a metric to assess the quality of the predictions. From this perspective, we introduced a metric that assesses the anatomical structure of the predictions besides analysing the prediction uncertainty and the localisation stability proposed by Kao et al. (2018).

The uncertainty of a predicted box $U_{B_j}(B_j)$ is computed considering its predicted confidence $P(B_j)$, thus:

$$U_B(B_j) = 1 - P(B_j), \tag{5}$$

a value of $P(B_j)$ close to one means that the algorithm is sure about the box prediction, on the contrary, if the confidence is low, the algorithm is not sure about the class of the object in the box. In order to compute the confidence uncertainty, $U_C(I_i)$ within n predicted boxes, the average box uncertainty in the image is computed as:

$$U_C(I_i) = \frac{1}{N}\sum_{j=1}^{N} U_{B_j}(B_j). \tag{6}$$

The second metric used for assessing the quality of the images is the localisation stability, which measures the robustness of the model by quantifying the change of the predicted boxes when the input images are corrupted by noise. Figure 5 presents the intuition of the procedure to calculate this metric.

To calculate the localisation stability, first, we predict the boxes in the original image, these boxes are considered as reference and are denoted as B_0^j, where j is the index of the box in the image. Then, object detection is carried out in the images affected by different noise levels n. After detecting the boxes for each noise level, we match the reference boxes with the noisy predictions by considering the highest intersection over union (IoU) among all the overlapping boxes considering the labelling consistency. A matching box is denoted as $C_n(B_0^j)$.

Thus, the localisation stability for each box $S_B(B_0^j)$ is computed as the average IoU between the reference box and the corresponding box in the noisy images:

$$S_B(B_0^j) = \frac{\sum_{n=1}^{N} IoU(B_0^j, C_n(B_0^j))}{N}, \tag{7}$$

where N is the number of noise levels. Finally, to estimate the localisation stability per image $S_I(I_i)$ a weighted sum of the stability of all the M boxes is performed, considering as weight the confidence of each box in the reference image $P(B_0^j)$:

$$S_I(I_i) = \frac{\sum_{j=1}^{M} P(B_0^j) S_B(B_0^j)}{\sum_{j=1}^{M} P(B_0^j)}, \tag{8}$$

we use six noise levels $N = 6$ with a standard deviation of $\{4, 8, 12, 16, 20, 24\}$.

The last metric to quantify the quality of the predictions evaluates the anatomical structure of the predicted boxes. Therefore, we consider the number of predicted boxes per type in each image. Since the boxes delineate the hand joints, the number of ROIs in each image is always fixed, except in X-rays with two hands or some abnormalities.

The ossification regions of the hand ($H_S(I_i)$) considered in this work follows the labelling by Koitka et al. (2018), that includes four DIP, five PIP, five MCP, wrist, radius and ulna. Thus, $H_S(I_i)$ is a vector containing the number of regions of each class. Since the number of

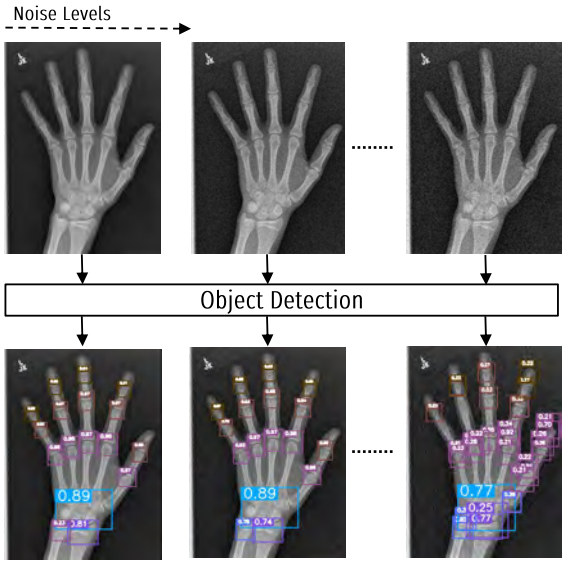


Figure 5: Localisation stability intuition, on the left the original image and its reference predictions, on the middle and the right the images corrupted by noise and their predictions. The noise applied can be better observed zooming the images

hands in the images is variable (1 - 2), we estimate the number of hands taking the median of the number of boxes that localised the wrist, radius and ulna in the prediction. The number of hands $N_h(I_i)$ is used to scale the base structure. We define a structural error ($e_S(I_i)$) based on the number of localised boxes as follows:

$$e_S(I_i) = \frac{|freq(B^M) - N_h(I_i)H_S(I_i)|}{N_h(I_i)H_S(I_i)}, \quad (9)$$

where B^M are all the boxes in an image, and $freq()$ calculates the absolute frequency of the boxes. Besides checking the consistency of the number of ROIs, we compute the distance between predicted ROIs of the same class (DIP, PIP and MCP) to account for miss-detections that fulfil the number of predictions.

In order to accept a sample of the unlabelled data pool, we first check if the anatomical structural error is zero $e_S(I_i) = 0$. Then if the predicted sample follows the anatomical structure, we analyse the confidence uncertainty $U_C(I_i)$ together with the localisation stability $S_I(I_i)$. Thus we compute the total uncertainty of $U_T(I_i)$ as:

$$U_T(I_i) = \alpha U_C(I_i) + (1 - \alpha)(1 - S_I(I_i)) + \beta e_S(I_i), \quad (10)$$

where α is a constant value between [0,1], that weights the confidence uncertainty and localisation stability, as both metrics are equivalent, we set the value of α to 0.5. Meanwhile, the constant weight of β is set to 10 since we want to penalise missing any ROI actively.

We sort the images according to their total uncertainty plus the total structural error, and we select the images with the highest uncertainty to be labelled by the user, in each iteration, we fed the algorithm with the ten most uncertain samples. Additionally, since the uncertainty prediction is expensive, we remove from the unlabelled pool, the "easy samples", which are those images without structural error and with a total uncertainty less than 0.4, this value was empirically defined with visual observation of the uncertainty among the samples. An overview of the active learning strategy adopted in this work is presented in the Algorithm 1.

Algorithm 1: Active learning object detection

Input: $L = \{\{I_1, y_1\}, \ldots, \{I_n, y_n\}\}$: Labelled data
$U = \{I_1, I_2, \ldots, I_m\}$: Unlabelled data
$S = \{\}$: Easy samples
$f(I_i)$: Object detection model
while *any*(U) **do**
  Train $f(x)$ in L;
  Predict $f(x)$ in U;
  Calculate total uncertainty $U_T(I_i)$ of U (10);
  if $U_T(I_i) < 0.4$ **then**
    Move "easy" samples from U to S;
  end
  Query user annotations of samples with maximum $U_T(I_i)$;
  Update L and U;
end

The selection of the model to perform active learning was done using the performance of the different trained models with the small dataset. The results obtained in subsection 4.2 demonstrated that the EfficientDet-D1 presented a trade-off between network size and performance to perform the active learning process.

3.5.2. *Training Overview for Object Detection*

To optimise the classification parameters of the EfficientDet, we employed focal loss as presented by Lin et al. (2017b):

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t), \quad (11)$$

where $\alpha = 0.25$ is a weighting factor that address the imbalance of the data, $\gamma = 2$ is a focusing parameter of the loss and p_t is defined as:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases} \quad (12)$$

with p representing the model confidence output and y the ground truth class.

For the parameters of the regression bounding box the Huber loss function is used, it combines the properties of squared error when the error is close to zero and the properties of absolute error when the error is large Hastie et al. (2009). It is defined as:

$$L_\delta(y_b, y_{bt}) = \begin{cases} \frac{1}{2}(y_{bt} - y_b)^2 & \text{if } |y_{bt} - y_b| \leq \delta, \\ \delta\,|y_{bt} - y_b| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases} \quad (13)$$

where y_{bt} represents the ground truth of the bounding boxes, and y_b are the predicted boxes. The parameter $\delta = 1$ defines the point in which the loss switches the optimisation from squared-error to absolute error.

The parameters of the BAA regression head were optimised using a similar loss that the one employed for the dense prediction network L_{reg} in section 3.4.4, but instead of considering the segmentation of the hand the ground truth bounding boxes were used to highlight only the ossification regions for the prediction.

The augmentation pipeline and the optimiser used for training the model were the same presented in section 3.4.4. The network was initialised with the parameters of EfficientDet for the COCO challenge since such parameters help to converge faster than using random initialisation. To successfully train the network, we first train the model using only the box regression and classification heads to solve the object detection task. Once these two heads are trained with the small dataset, we

start training the model with the whole dataset, including the BAA task. All the prediction heads are trained at the same time since they depend on each other.

3.6. BAA Evaluation

The statistical model selected to assess the most suitable algorithm to perform BAA is the MAE that was used for evaluating the RSNA challenge (Halabi et al., 2019). The MAE is formulated as follows:

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|\,, \tag{14}$$

where y_i and $\hat{y}_i$ are defined in Equation (4) and e_i represents the prediction error.

Aside from the error, other useful statistics to assess the quality of the model is the R-squared (R^2). R^2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model. The higher the R-squared, the better the model (Kassambara, 2018). Mathematically, R^2 is defined as:

$$R^2 \equiv 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}}\,, \tag{15}$$

where SS_{res} is the sum of squares of residuals calculated as follows:

$$SS_{\mathrm{res}} = \sum_i (y_i - \hat{y}_i)^2. \tag{16}$$

and SS_{tot} is the total sum of squares which is proportional to the variance of the data:

$$SS_{\mathrm{tot}} = \sum_i (y_i - \bar{y})^2, \tag{17}$$

where $\bar{y}$ is the mean of the predictions.

4. Results

The experiments performed to test the proposed algorithms are structured as follows. The subsections 4.1 and 4.2 presents the results of the preprocessing steps towards BAA: subsection 4.1 displays hand segmentation results; subsection 4.2 depicts the experiments conducted for the object detection task and the results of the localisation-aware active learning framework. Finally, subsection 4.3 displays a benchmark of the implemented BAA algorithms.

4.1. Hand Segmentation

To assess the quality of the segmentation models, we manually segmented the 200 images of the RSNA test set. We train multiple segmentation U-Net networks with EfficientNet encoder changing the compound scaling factor, to test if the input size of the image improves the segmentation results.

Table 4: Segmentation results, the metrics are obtained in 200 images of the test set of the RSNA dataset

Model	Image size	DICE	HD (pixels)	AVD (pixels)
Unsupervised	1024	0.9308	165.311	25.22
Efficient-Unet D0	512	0.9636	41.44	6.50
Efficient-Unet D1	640	0.9655	42.81	6.22
Efficient-Unet D2	768	0.9665	36.61	5.97
Efficient-Unet D3	896	0.9667	41.77	6.15
Efficient-Unet D4	1024	0.9680	38.46	5.85

The metrics used to benchmark the different segmentation algorithms were the DICE score, the Hausdorff distance (HD) and the average distance (AVD) that are well-defined by Taha and Hanbury (2015). The results obtained by the segmentation techniques are displayed in Table 4.

From Table 4, it is possible to see that the different segmentation models based on Deep Learning achieved a satisfying dice score of 96%, which enables to correctly label most of the images in the dataset and predict new hand X-rays images. The unsupervised algorithm fails in segmenting images with low contrast and with the presence of big frames, thus obtaining 93% DICE score.

Besides the DICE score, the HD and AVD results enable to see if there exist presence of outliers in the segmentation, It is possible to see that only in the case of the unsupervised algorithm exist a considerable distance, this occurs because in some cases the borders of the X-ray are segmented as part of the hand thus producing such high distance. Meanwhile, the deep-learning-based segmentation models show consistent results among each other and low values of HD and the AVD, which means that the outline of the segmentation is closer to the ground truth.

4.2. Ossification ROIs Detection

The performance of the ossification region detection was measured by the average precision (AP) of the Intersection over Union (IoU). A detected region is considered as a good match if the predicted area overlaps with at least 50% with the ground truth as defined in the COCO challenge metrics (Lin et al., 2014). The global performance of a model is computed as the mean average precision (mAP). Figure 6 depicts the results of the proposed algorithm for ossification regions localisation.

Table 5 presents the results of the proposed methodology for object detection and its comparison with the baseline model of Koitka et al. (2018), which demonstrates that our selected architectures outperform the base RPN. EfficientDet-D3 obtains the best result. However, since the size of the input image of this architecture is significant, it demands longer training time. Therefore, due to limited resources and time constraints,

Table 5: Evaluation of object detection models for ossification region detection. Results are stated as mean and standard deviation of five runs on the test set of 89 images defined by Koitka et al. (2018)

Model	Image size	AP@0.5IoU						mAP@0.5IoU
		DIP	PIP	MCP	Radius	Ulna	Wrist	
Koitka et al. (2018)	1024	89.79 ± 5.10	**88.29 ± 4.98**	94.82 ± 1.45	97.96 ± 1.10	87.78 ± 3.24	98.87 ± 0.01	92.92 ± 1.93
EfficientDet-D0	512	76.14 ± 0.78	77.21 ± 0.21	**95.32 ± 0.22**	**97.75 ± 0.07**	92.39 ± 0.79	99.16 ± 1.15	89.66 ± 0.22
EfficientDet-D1	640	91.24 ± 0.35	85.69 ± 1.10	95.05 ± 0.58	96.95 ± 0.01	96.21 ± 0.02	97.84 ± 1.21	93.83 ± 0.4
EfficientDet-D2	768	88.76 ± 1.45	82.42 ± 0.57	94.48 ± 0.15	96.92 ± 0.00	96.68 ± 0.10	97.58 ± 1.35	92.81 ± 0.44
EfficientDet-D3	896	**93.97 ± 0.89**	88.08 ± 0.59	93.64 ± 0.67	97.43 ± 0.10	**97.52 ± 0.67**	**100.0 ± 0.00**	**95.11 ± 0.26**
EfficientDet-D4	1024	91.20 ± 0.85	87.73 ± 0.07	94.85 ± 0.67	97.20 ± 0.44	96.33 ± 0.05	98.66 ± 1.22	94.33 ± 0.24

we selected the architecture D1 to proceed with the active learning framework.

The results of applying the localisation aware, active learning framework are depicted in Figure 7. Since the initial performance for detecting the ossification ROIs is high, it was possible to label all the remaining samples of the dataset in only eight active learning iterations, obtaining a significant mAP increase from 93.83%± 0.4 to 96.18%± 0.59. At the end of this procedure, we increased the number of manually labelled samples for object detection from 329 to 409, and we obtained automatically generated labels for the whole dataset.

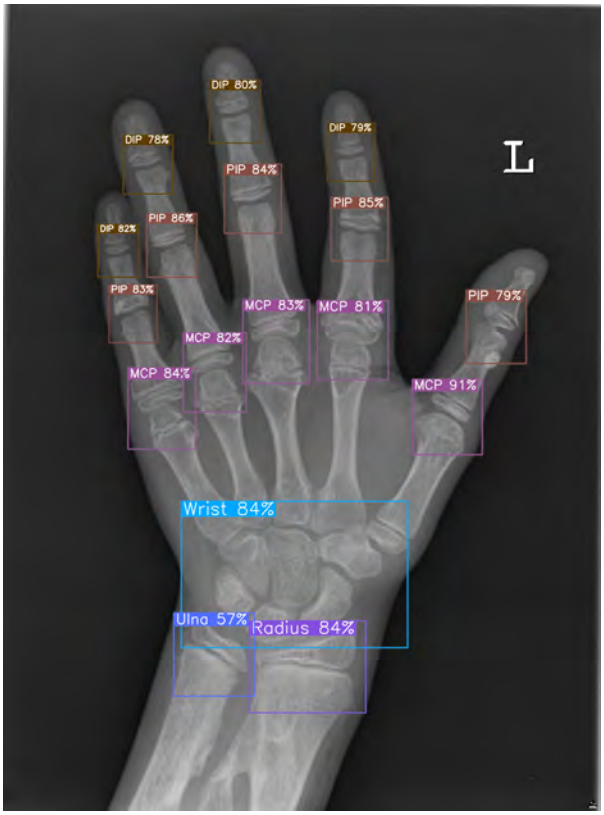


Figure 6: Example of the algorithm proposed for ossification regions detection using EfficientDet-D2. The boxes present the class of the ROI and the confidence of the algorithm.

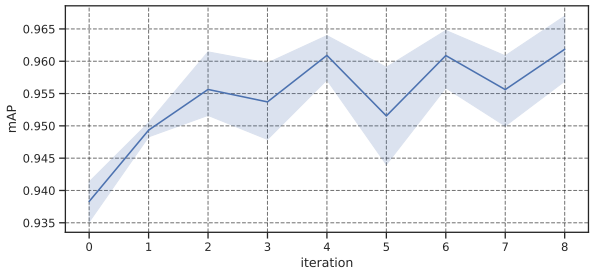


Figure 7: Mean average precision of 5 runs in each iteration of the object aware, active learning framework towards improving the performance of the ossification region localisation.

4.3. BAA Results

A summary of the performance of the implemented methodologies for BAA is presented in Table 6. It presents a comparison of the different methodologies developed for BAA, starting from a simple transfer learning with EfficientNet-B5, and the CNN families developed for BAA with dense predictions through a U-Net and based on the detection of the ossification ROIs. It is possible to observe that including the ossification ROIs in the model increases the performance. Table 6 additionally presents different metrics of the obtained errors such as standard deviation, the R^2 coefficient, and the 5^{th}, 95^{th}, and 99^{th} percentiles of the errors.

In order to have a clear overview of the performance of the outcome of each model for BAA we obtained a scatter plot of the ground truth age vs the predicted BA, such plot allowed us to visually inspect the quality of the algorithms and also to determine the existence of outliers or other problems within the predictions in multiple subjects. Figure 8 presents the result for the localisation based model Efficient-Det D3 with preprocessing; the plot depicts that most of the predictions correctly lay on the ground-true band. It is also possible to observe that there exist few samples that can be considered outliers, which shows the successful operation of our method.

In addition to the age scatter plots, we obtained the Bland–Altman plots of all the presented strategies. Figure 9 presents the Bland–Altman plot of the best performing model, it shows that more than 95% of all the predictions were within one year of the ground truth.

Table 6: Comparison of absolute errors of proposed CNNs. The statistics are determined with the bone age in months for the 200 test samples in the RSNA dataset.

Type	Model Name	Image size	MAE	STD	R2	5th%^{tile}	95th%^{tile}	99th%^{tile}
Transfer Learning	EfficientNet-B5	512	4.773	3.867	0.980	0.277	13.14	15.575
EfficientNet + dense predictions	Efficient-UNet D0	512	5.021	4.03	0.978	0.36	12.85	15.82
	Efficient-UNet D1	640	5.174	4.27	0.976	0.49	13.67	17.82
	Efficient-UNet D2	768	4.719	3.89	0.980	0.23	11.65	17.09
Localisation BAA	Efficient-Det D0	512	5.169	4.77	0.973	0.30	14.43	21.08
	Efficient-Det D1	640	4.928	4.17	0.977	0.43	13.93	17.45
	Efficient-Det D2	768	4.733	3.93	0.980	0.31	12.03	15.75
	Efficient-Det D3	896	4.559	3.94	0.980	0.21	11.75	17.26
Localisation BAA + Preprocessing	Efficient-Det D0	512	4.962	3.90	0.978	0.51	12.29	15.20
	Efficient-Det D1	640	4.766	3.83	0.980	0.54	11.86	17.58
	Efficient-Det D2	768	4.276	**3.41**	**0.984**	0.33	**10.94**	**14.29**
	Efficient-Det D3	896	**4.139**	3.84	0.983	**0.22**	12.32	17.09

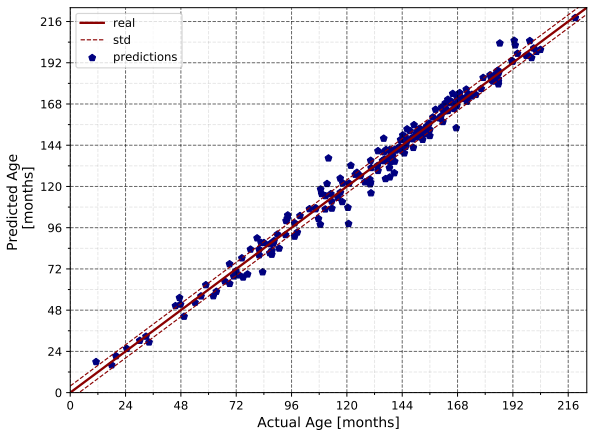


Figure 8: Scatter plot of the predicted BAA of the 200 samples of the RSNA dataset vs the ground-true age. The solid red line represents the perfect prediction, and the dashed red lines are the bounding values that separate the perfect prediction with one standard deviation.

It is possible to observe that only five predictions have a mean absolute error above one year. Thus the Bland–Altman plots show a strong correlation between the BA predictions and the ground truth.

Figure 10 presents some detection output with their predicted BA per ROI. The depicted predictions correspond to the object detection algorithm without preprocessing, which demonstrates that the detection algorithm works even in the presence of dose variations, markers, borders and collimation.

4.3.1. BAA evaluation with external datasets

As a final experiment to further validate the results of the results obtained by the proposed methods, we evaluate the trained methodology in the digital hand atlas (DHA) and the Gasthuisberg (GH) datasets. It is essential to mention that the training process of our developed models did not use samples from the external datasets, and there exist significant variations in their distribution since they contain samples of different demographic, health status and racial group out of the scope from those of the RSNA dataset. Table 7 presents the results obtained in the two external datasets by our best performing model. We include the mean average error, the standard deviation and the R^2 coefficient, which shows a good correlation of the prediction on the external data.

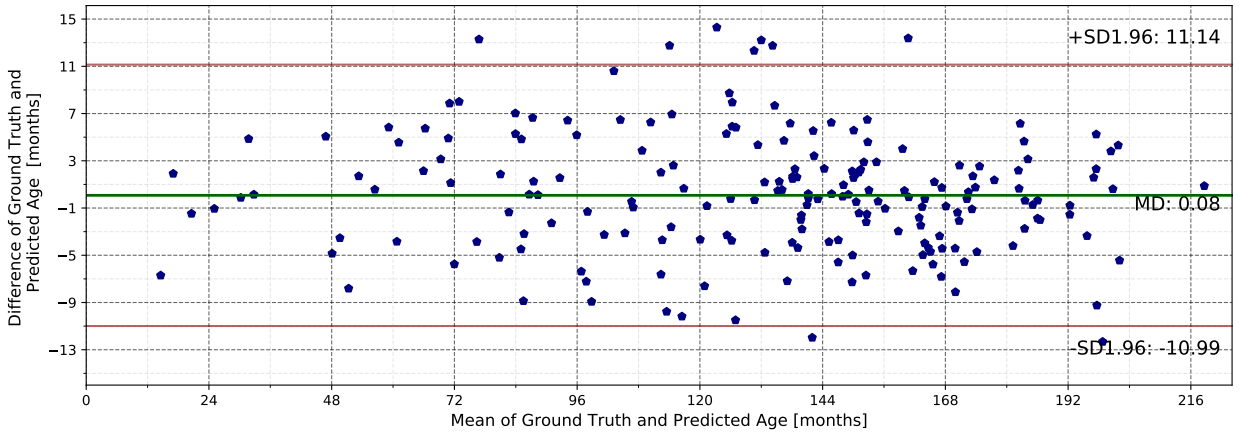


Figure 9: Bland–Altman plot of the 200 samples of the RSNA dataset. Red lines indicate 1.96 standard deviation of the difference

Table 7: Results obtained in months by the proposed model in external datasets of Digital Hand Atlas (DHA) and the Gasthuisberg dataset (GH)

Dataset	Model Name	MAE	STD	R2
DHA	Efficient-Det D0	**8.818**	6.005	0.968
	Efficient-Det D1	8.923	**5.880**	**0.968**
	Efficient-Det D2	8.873	6.095	0.968
	Efficient-Det D3	9.109	6.053	0.967
GH	Efficient-Det D0	7.739	6.053	0.959
	Efficient-Det D1	7.654	**6.322**	0.959
	Efficient-Det D2	**7.480**	6.356	0.960
	Efficient-Det D3	7.812	6.243	0.958

Table 8: BAA metrics by ossification ROIs for the family of Efficient-Det's with preprocessing step

Model Name	MAE						
	DIP	PIP	MCP	Wirst	Radius	Ulna	Total
Efficient-Det D0	5.02	4.98	4.95	5.03	5.12	4.97	4.962
Efficient-Det D1	4.67	0.85	4.79	4.95	4.83	5.05	4.766
Efficient-Det D2	4.33	**4.29**	**4.31**	4.50	4.51	**4.35**	4.276
Efficient-Det D3	**4.29**	4.33	4.40	**4.38**	**4.47**	4.43	**4.139**

5. Discussion

5.1. Ablation Study

In this work, we first study the effect observed in the performance of each component in the proposed methodology. The experimental results presented in Table 6 starts with the base methodology of this study with the EfficientNet-B5, such architecture was selected considering the winner of the RSNA challenge that used an input image of 500 × 500 pixels. The base architecture produces the MAE score of 4.774 months.

In order to improve the performance of the base model, preprocessing steps such as background stripping and histogram equalisation of the hand region were performed. However, the result obtained by training the model with the standardised images did not improve the performance of the base methodology. Thus, considering that the original BAA methodology relay on different ROIs, we modified the architecture using the segmentation of the whole hand and then to use the regions of the TW methods.

Since it was possible to obtain segmentation models, we introduced a variation of EfficientDet with a U-net head to guide the predictions on the hand. The application of this methodology, improving the MAE score only in 0.053 months. Therefore, we introduced an active learning framework to label the ossification ROIs based on preexisting annotations of 341 images.

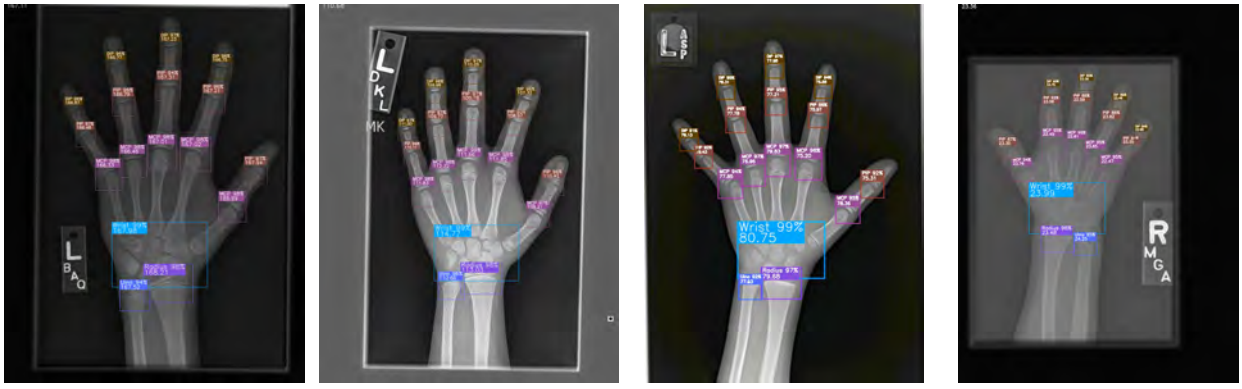
The ossification ROIs extraction produced an improvement of 0.214 months in MAE compared to the base model. Furthermore, we introduced a preprocessing step in the prediction stage to improve further the performance of the model for BAA achieving the final MAE of 4.139 in months; such result is comparable with the best-published model on the RSNA data by Escobar et al. (2019), which reached an MAE of 4.14 months using only the data of one RSNA dataset. Besides, they demonstrated that such performance could be further improved using additional data, in their research, they reached an MAE of 3.85 months, adding 6,288 images.

5.2. BAA by ossification ROIs

This experiment evaluates the performance of the best performing model considering the different types of ROIs. Table 8 shows the MAE by ROI type in the 200 images of the RSNA test set.

From Table 8, we can observe that in general, averaging the output of the ROIs by patient enhance the prediction performance, since it collects all the anatomical information of the subject. It is possible to observe that the prediction obtained by the phalanges (DIP, MCP and PIP) keeps a higher correlation with the total output compared with the radius, ulna and wrist regions.

The obtained results by ROI are consistent with the



(a) gt:169.67 pred: 167.11 (b) gt:117.17 pred: 110.68 (c) gt:73.26 pred: 77.90 (d) gt: 24.52 pred: 23.36

Figure 10: Some prediction examples of the proposed algorithm for BAA, each bounding box depicts the category, confidence and the assessed BA per ROI.The final BA is computed as the average of all the detected ROIs

study performed by Iglovikov et al. (2018). In which they trained different models for the carpal bones(wrist), the metacarpals (MCP) and the whole hand, finding that merging the outcome of the different regions improves the performance of BAA. Our strategy considers all the anatomical regions stated in the TW and G&P methods, which is a step closer in performing BAA more intuitively and in a more explainable way using CNNs.

Our approach could potentially help in highlighting abnormalities in the final BAA if any of the detected ROIs is significantly different from the rest of the predictions.

5.3. BAA in external datasets

The results obtained for BAA in external datasets in the subsection 4.3.1 shows that the proposed model can predict the BA in new samples, although there is still a discrepancy with the result on the test set of the RSNA.

In the case of the DHA dataset, the BA labels of the whole dataset present quantisation in intervals of one year. Thus, introducing a gap between the real BA and the available labels. Besides this fact, the DHA is divided into racial groups. It is possible that the DHA contains a variety of samples that are out of the scope of the RSNA dataset. Comparing our method with other work that was evaluated with this dataset, Spampinato et al. (2017) reported an MAE score of 9.48 months with training on the same dataset, which is coherent with our best-reported model of 8.81 months shown in Table 7.

Meanwhile, in the case of the GH dataset, we demonstrated that the proposed framework could work with images of patients with growth disorders. It is crucial to notice that the quality of the images of the GH datasets differs notably from the RSNA dataset since most of the images include collimation, and the resolution of the images differs from the images of the RSNA.

The obtained results are consistent with the state of the art by Escobar et al. (2019), which show their performance comparing results of the RSNA and the RHPE datasets. Their best-obtained MAE within the RSNA is 4.14 months and in an external dataset is 7.60 months. This discrepancy could be caused by the different way of labelling of the external data, which has not taken into account the inter-rater variability and bias introduced by the individual radiologist, described in section 1.

Overall, the obtained results with external datasets show that there is a domain adaptation problem between the new data and the trained data. It will be necessary for future research to train the model with images of the specific manufacturer and demographic samples in order to have a more consistent prediction.

6. Conclusions

In the present work, we developed multiple methodologies to perform BAA based on CNNs.We demonstrated that hand segmentation and ossification ROIs detection are valid ways to exploit local information of the X-rays and to improve the performance of the deep-learning methodologies.

Towards improving the generalisation of BAA, we performed active learning-based labelling of the ossification ROIs of the hands, which enabled the development of new object detection based architecture that emulates the TW and the G&P methods for BAA. The obtained results demonstrated that the use of anatomical information improves the quality of BAA.

Further investigations are required to create a multi-stage pipeline that extracts high-resolution ROIs that could improve the overall generalisation of the model, and to include additional metadata of the patients such as the chronological age and the race that could leverage the performance of BAA. It is crucial to consider the domain adaptation necessity for applying the model with a specific demographic group and manufacturer machines.

7. Acknowledgments

I want to thank my supervisor for her support throughout my internship and her direction towards accomplishing this work. I am also thankful with all the team of the MAIA master for all the knowledge you provide me during these two years; it has been not only a remarkable academic experience but also a life experience. Finally, I would like to thank the AGFA and its Discovery program, for their support and the access to all their computational resources needed for the implementation of this work.

References

Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, in: European conference on computer vision, Springer. pp. 404–417.

Berst, M.J., Dolan, L., Bogdanowicz, M.M., Stevens, M.A., Chow, S., Brandser, E.A., 2001. Effect of knowledge of chronologic age on the variability of pediatric bone age determined using the greulich and pyle standards. American Journal of Roentgenology 176, 507–510.

Bilbily, A., Cicerot, M., 2018. Rsna bone age challenge 16bit solution. https://github.com/us/bone-age-prediction. (Accessed on 06/08/2020).

Breu, H., Gil, J., Kirkpatrick, D., Werman, M., 1995. Linear time euclidean distance transform algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 529–533.

Buda, M., Maki, A., Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249–259.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: Fast and flexible image augmentations. Information 11. URL: https://www.mdpi.com/2078-2489/11/2/125, doi:10.3390/info11020125.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. Brief: Binary robust independent elementary features, in: European conference on computer vision, Springer. pp. 778–792.

Cao, F., Huang, H., Pietka, E., Gilsanz, V., Dey, P.S., Gertych, A., Pospiech-Kurkowska, S., 2003. Image database for digital hand atlas, in: Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation, International Society for Optics and Photonics. pp. 461–470.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.M., Zietz, M., Hoffman, M.M., et al., 2018. Opportunities and obstacles for deep learning in biology and medicine. Journal of The Royal Society Interface 15, 20170387.

Creo, A.L., Schwenk, W.F., 2017. Bone age: A handy tool for pediatric providers. Pediatrics 140.

Dallora, A.L., Anderberg, P., Kvist, O., Mendes, E., Ruiz, S.D., Berglund, J.S., 2019. Bone age assessment with various machine learning techniques: A systematic literature review and meta-analysis. PloS one 14.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee. pp. 248–255.

Elad, M., 2002. On the origin of the bilateral filter and ways to improve it. IEEE Transactions on image processing 11, 1141–1151.

Escobar, M., González, C., Torres, F., Daza, L., Triana, G., Arbeláez, P., 2019. Hand pose estimation for pediatric bone age assessment, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 531–539.

Gertych, A., Zhang, A., Sayre, J., Pospiech-Kurkowska, S., Huang, H., 2007. Bone age assessment of children using a digital hand atlas. Computerized medical imaging and graphics 31, 322–331.

Ghiasi, G., Lin, T.Y., Le, Q.V., 2019. Nas-fpn: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7036–7045.

Gilsanz, V., Ratib, O., 2005. Hand bone age: a digital atlas of skeletal maturity. Springer Science & Business Media.

Greulich, W.W., Pyle, S.I., 1959. Radiographic atlas of skeletal development of the hand and wrist. Stanford University Press.

Halabi, S.S., Prevedello, L.M., Kalpathy-Cramer, J., Mamonov, A.B., Bilbily, A., Cicero, M., Pan, I., Pereira, L.A., Sousa, R.T., Abdala, N., et al., 2019. The rsna pediatric bone age machine learning challenge. Radiology 290, 498–503.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hsieh, C., Chen, C., Jong, T., Liu, T., Chiu, C., 2012. Automatic segmentation of phalanx and epiphyseal/metaphyseal region by gamma parameter enhancement algorithm. Measurement Science Review 12, 21–27.

Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 .

Iglovikov, V.I., Rakhlin, A., Kalinin, A.A., Shvets, A.A., 2018. Paediatric bone age assessment using deep convolutional neural networks, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 300–308.

Jaderberg, M., Simonyan, K., Zisserman, A., et al., 2015. Spatial transformer networks, in: Advances in neural information processing systems, pp. 2017–2025.

Kao, C.C., Lee, T.Y., Sen, P., Liu, M.Y., 2018. Localization-aware active learning for object detection, in: Asian Conference on Computer Vision, Springer. pp. 506–522.

Kashif, M., Deserno, T.M., Haak, D., Jonas, S., 2016. Feature description with sift, surf, brief, brisk, or freak? a general question answered for bone age assessment. Computers in biology and medicine 68, 67–75.

Kassambara, A., 2018. Machine Learning Essentials: Practical Guide in R. sthda.

Koitka, S., Demircioglu, A., Kim, M.S., Friedrich, C.M., Nensa, F., 2018. Ossification area localization in pediatric hand radiographs using deep neural networks for object detection. PloS one 13.

Larson, D.B., Chen, M.C., Lungren, M.P., Halabi, S.S., Stence, N.V., Langlotz, C.P., 2018. Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs. Radiology 287, 313–322.

Lee, H., Tajmir, S., Lee, J., Zissen, M., Yeshiwas, B.A., Alkasab, T.K., Choy, G., Do, S., 2017. Fully automated deep learning system for bone age assessment. Journal of digital imaging 30, 427–441.

Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. Brisk: Binary robust invariant scalable keypoints, in: 2011 International conference on computer vision, Ieee. pp. 2548–2555.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D., 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. The International Journal of Robotics Research 37, 421–436.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, pp. 2980–2988.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European conference on computer vision, Springer. pp. 740–755.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768.

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 .

Lowe, G., 2004. Sift-the scale invariant feature transform. Int. J 2, 91–110.

Mansourvar, M., Ismail, M.A., Herawan, T., Gopal Raj, R., Abdul Kareem, S., Nasaruddin, F.H., 2013. Automated bone age assessment: motivation, taxonomies, and challenges. Computational and mathematical methods in medicine 2013.

Mansourvar, M., Shamshirband, S., Raj, R.G., Gunalan, R., Mazinani, I., 2015. An automated system for skeletal maturity assessment by extreme learning machines. PloS one 10.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 Fourth International Conference on 3D Vision (3DV), IEEE. pp. 565–571.

Nguyen, T.N., Tran, Q.D., Nguyen, T.N., Nguyen, Q.H., 2020. Detection and segmentation of endoscopic artefacts and diseases using deep architectures. medRxiv .

Pan, I., Thodberg, H.H., Halabi, S.S., Kalpathy-Cramer, J., Larson, D.B., 2019. Improving automated pediatric bone age estimation using ensembles of models from the 2017 rsna machine learning challenge. Radiology: Artificial Intelligence 1, e190053.

Pietka, E., Pospiech, S., Gertych, A., Cao, F., Gilsanz, V., 2001. Computer automated approach to the extraction of epiphyseal regions in hand radiographs. Journal of Digital Imaging 14, 165.

Poznanski, A.K., Hernandez, R.J., Guire, K.E., Bereza, U.L., Garn, S.M., 1978. Carpal length in children—a useful measurement in the diagnosis of rheumatoid arthritis and some congenital malformation syndromes. Radiology 129, 661–668.

Reddy, N.E., Rayan, J.C., Annapragada, A.V., Mahmood, N.F., Scheslinger, A.E., Zhang, W., Kan, J.H., 2020. Bone age determination using only the index finger: a novel approach using a convolutional neural network compared with human radiologists. Pediatric Radiology 50, 516–523.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition,

pp. 779–788.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, pp. 91–99.
Ren, X., Li, T., Yang, X., Wang, S., Ahmad, S., Xiang, L., Stone, S.R., Li, L., Zhan, Y., Shen, D., et al., 2018. Regression convolutional neural network for automated pediatric bone age assessment from hand radiograph. IEEE journal of biomedical and health informatics 23, 2030–2038.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.
Seok, J., Kasa-Vubu, J., DiPietro, M., Girard, A., 2016. Expert system for automated bone age determination. Expert Systems with Applications 50, 75–88.
Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint:1312.6229 .
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .
Spampinato, C., Palazzo, S., Giordano, D., Aldinucci, M., Leonardi, R., 2017. Deep learning for automated skeletal bone age assessment in x-ray images. Medical image analysis 36, 41–51.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
Taha, A.A., Hanbury, A., 2015. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. BMC medical imaging 15, 29.
Tan, M., Le, Q.V., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 .
Tan, M., Pang, R., Le, Q.V., 2019. Efficientdet: Scalable and efficient object detection. arXiv preprint arXiv:1911.09070 .
Tanner, J.M., Whitehouse, R., Cameron, N., Marshall, W., Healy, M., Goldstein, H., et al., 2001. Assessment of skeletal maturity and prediction of adult height (TW2 method). Saunders London.
Thodberg, H.H., Kreiborg, S., Juul, A., Pedersen, K.D., 2008. The bonexpert method for automated determination of skeletal maturity. IEEE transactions on medical imaging 28, 52–66.
Walter, T., Klein, J.C., 2001. Segmentation of color fundus images of the human retina: Detection of the optic disc and the vascular tree using morphological techniques, in: International Symposium on Medical Data Analysis, Springer. pp. 282–287.
Wu, E., Kong, B., Wang, X., Bai, J., Lu, Y., Gao, F., Zhang, S., Cao, K., Song, Q., Lyu, S., et al., 2019. Residual attention based network for hand bone age assessment, in: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), IEEE. pp. 1158–1161.
Xiao, B., Wu, H., Wei, Y., 2018. Simple baselines for human pose estimation and tracking, in: Proceedings of the European conference on computer vision (ECCV), pp. 466–481.
Zhang, A., Sayre, J.W., Vachon, L., Liu, B.J., Huang, H., 2009. Racial differences in growth patterns of children assessed on the basis of bone age. Radiology 250, 228–235.
Zhao, C., Han, J., Jia, Y., Fan, L., Gou, F., 2018. Versatile framework for medical image processing and analysis with application to automatic bone age assessment. Journal of Electrical and Computer Engineering 2018.

Appendix A. Unsupervised hand segmentation

The proposed unsupervised hand segmentation pipeline adopts some steps of algorithm by Hsieh et al. (2012), which uses K-means to differentiate the bone and hand tissue from the background. It considers the non-uniform background produced by the X-ray irradiation process, which varies the appearance of the X-ray images. Furthermore, most of the samples in the RSNA dataset contains collimation borders and dose variation. Thus, our algorithm is developed to alleviate the effect of such variations, and overcoming the different contrast presented in the dataset. The segmentation is performed in 6 steps as follows:

(a) Resize the image so that maximum side is equal to 1024 pixels, keeping the aspect ratio of the initial image. We then apply bilateral filter , to remove the noise while preserving the borders (Elad, 2002).

(b) Extract the edges of the image by means of morphological edge detection operation and obtain the region of interest of the hand by selecting a bounding box of the part of the image with the biggest contour in the edges image.

(c) Apply histogram equalisation to the cropped image and K-means with K=3 to separate the different structures in the image.

(d) Select the segmentation region corresponding to the bones and remove all the external frame lines through top-hat operations with linear kernels with different orientations (Walter and Klein, 2001).

(e) Generate a marker containing the sure foreground and the sure background by means of distance transform (Breu et al., 1995).

(f) Apply watershed to segment the hand and restore the mask to the original image size.

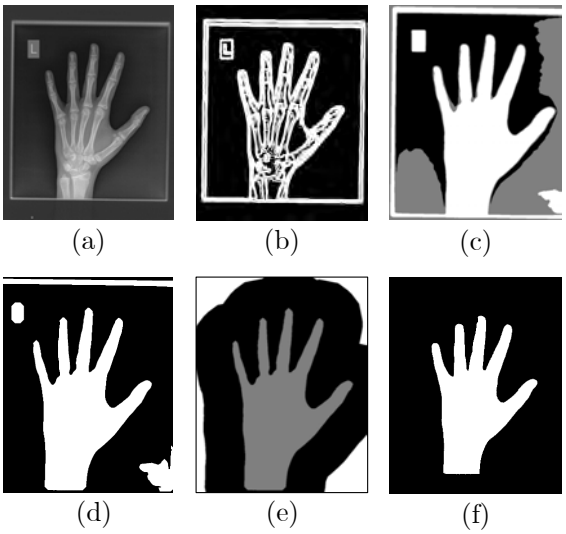


Figure A.11: Unsupervised segmentation pipeline for the hand in X-ray images.

Medical Imaging and Applications

Master Thesis, August 2020

A Qualitative and Quantitative Analysis of state of the art Techniques for MRI Brain Image Synthesis

Pierpaolo Vendittelli, **Supervisors**: Arnau Oliver, Xavi Lladó

Universitat de Girona, 17003, Girona, Spain

Abstract

Medical Image synthesis is nowadays a crucial topic for reducing the costs and the acquisition timing of the images. By reducing these two important factors, much more patients can be tested in less time and get diagnosis faster. Among all the imaging techniques, MRI is one of the most used, therefore, the purpose of this research is to implement, test and analyze the different state of the art techniques usually used for brain image synthesis on two public datasets (WMH and BraTS), trying so to discuss each of them according to two different problems: synthesizing FLAIR sequences starting from T1-Weighted sequences, and the opposite. In this work three main families of architectures are tested, **Deep Convolutional Neural Network**s (DCNN) such as Unet and ResUnet, **Generative Adversarial Networks** (GANs) and **Cycle Generative Adversarial Networks** (CycleGANs). Experimental results performed on both the datasets, showned that the task of synthesizing FLAIR sequences from T1-Weighted sequences was easier than the opposite, and furthermore, it was found that a complex architecture such as CycleGAN was performing worse that more simple architectures both when synthesizing FLAIR and when synthesizing T1-Weighted.

Keywords: Image Synthesis, Brain MRI, Deep Learning, GANs, Domain adaptation

1. Introduction

Medical Imaging is the process of representing the exterior or the interior of a body through visual representation (images) which will be used for clinical analysis and medical intervention. There exist different techniques of acquisition using different physical basis, such as Ultrasounds, Positron Emission Tomography, Radiography, Computed Tomography, Magnetic Resonance Imaging etc. each one with a different objective and used for different purposes (i.e. screening or treatment). Among these, Magnetic Resonance Imaging (MRI) is a technique generating detailed images of the organs and tissues in the body with the usage of a magnetic field. The configuration of the machine allows the acquisition of multiple sequences each one highlighting different properties of the analyzed tissues. The most common sequences are T1-Weighted, PD, T2-Weighted, FLAIR and DWI. The acquisition of these different sequences of the same organ is highly expensive in terms of resource and time. Thus, during the last years research is being driven towards the automatic generation of multiple sequences from another one.

Image synthesis is indeed the process of creating new images from some form of image description which can be either noise or another image. In the medical domain this leads to creation of new data useful to fulfil some important tasks such as domain adaptation (Perone et al., 2019) or improving lesion segmentation (Salem et al., 2019) , as well as data augmentation (Shin et al., 2018) and modalities generation (Lee et al., 2019).

As it will be better described in Section II, since image synthesis is a really broad topic, a lot of different strategies are being developed during the years. These strategies can be divided in two main areas which are: traditional methods (Freeman et al., 2000) and Deep Learning techniques (Vemulapalli et al., 2015), with the exploit of Adversarial Learning techniques, these last (GANs) being a really hot topic nowadays because of the impressive performance they shown in multiple applications (Wang et al., 2020).

Through the usage of advanced Deep Learning techniques, this master thesis project is focused on analyzing and developing strategies for synthesizing Brain MRI images. More specifically, we will focus on doing a qualitative and quantitative evaluation of different and recent state of the art techniques to perform image synthesis, studying the behavior and proposing improvements on the analyzed approaches including standard convolutional neural networks approaches such as Unet and ResUnet in 3D and more advanced Adversarial approaches both in 2D and 3D .
The main tasks we will cover are two: the synthesis of FLAIR images having as input T1-Weighted images, and the opposite problem, the synthesis of T1-Weighted images having as input FLAIR. One of the main difficulties of these tasks is that when a lesion is present in the brain, while it appears clear in the FLAIR modality, in T1-Weighted it can be confused with the gray matter because of its intensities value, although usually lesions in brain appears close (around) the ventricles and the gray matter is the outside part of the brain. Figure 1 shows a clear example of this problem which can confuse neural networks and produce inconsistent results. The green circle on the left represents how the lesion appears in FLAIR images, while on the right how it appears (and can be confused with gray matter) in T1-Weighted images. The rest of this paper is organized as follow: *Section II* gives an overview of the most advanced techniques for Image Synthesis which were used as an input to the work, *Section III* describes the material used and *Section IV* introduces and discuss the background and the implementation of each approach. *Section V* presents a commented analysis of the experiments conducted for each of the developed architectures while *Section VI* discuss the experimental results obtained. In the end, *Section VII* offers a conclusion to the project mentioning some interesting approaches for future work.

2. State of the art

In this section we discuss all the most advanced strategies for image synthesis in general and proceed then analyzing more in depth state of the art techniques related to brain MRI image synthesis and cross-domain adaptation.

2.1. Image synthesis

As mentioned, image synthesis is the process of creating new images starting from some sort of descriptor. It is a really broad topic and a lot of different strategies have been developed over the last years to solve different problems, from generating new lesions for improving segmentation (Salem et al. (2019)) to pure image synthesis (Liu et al. (2018)). We can group these strategies in two main categories which respectively are: Traditional Methods, and Deep Learning techniques, which usually use Adversarial Learning techniques such as GANs.(Yi et al., 2019)

2.1.1. Traditional Methods

Traditional methods are usually atlas/multi atlas-based methods which are able to estimate the intensity distribution with a good approximation and relatively fast as shown by (Miller et al., 1993), and later on (Lauritzen et al., 2019). Atlas image synthesis usually consist in registering the image with a set of co-registered image pairs along with gold-standard segmentation masks.
Other methods include regression methods (Jog et al., 2013) where the synthesis is produced by a regression forest algorithm trained with paired data of both input and target modality.

2.1.2. Deep Learning Techniques

Since the introduction of Generative Adversarial Models by (Goodfellow et al., 2014), image synthesis shifted towards the adversarial paradigm (Xiang et al., 2018), (Nie et al., 2018), (Hiasa et al., 2018) as an example. This because of the privacy issues related to medical protocol and due to imbalanced dataset (because of the lack of positive cases for each pathology). Although not medical imaging related, Snell et al. (2017) replaced the pixelwise losses (Mean Absolute Error - Mean Squared Error) with perceptual losses such as structural similarity index (SSIM - multi scale SSIM) proving to achieve better results when reconstructing the images. In Yi et al. (2019) an exaustive review on the usage of GANs in medical imaging is presented. In particular, it was shown that to tackle the cross modality problem, researchers tend to develop architectures based on the well know pix2pix framework for co-registered data, and on the CycleGAN framework for unregistered data.

Hiasa et al. (2018) used the Gradient Correlation similarity metric as a Gradient-consistency loss between real and generated images to improve the accuracy at the boundaries, while on the other hand Zhang et al. (2018), tackled the volumetric shape problem by adding a shape-consistency loss in order to constrain the geometric invariance of the generated data, using two networks to segment each modality and provide the necessary semantic labels for each modality.

GAN can be used for generating T2-Weighted scans from T1-Weighted scans as proved from Dar et al. (2019), through the usage of cGAN and pGAN, while Yang et al. (2018) uses cGAN for generating FLAIR scans from T1-Weighted images. These works are based on the original work of Zhu (2017) and both provide 2D implementation like most of the cited papers. Yu et al. (2018) provides a method for a 3D conditional GAN which aims not only to eliminate discontinuities in the sagittal and coronal direction due to the synthesization sliceway, but also try to improve the generation

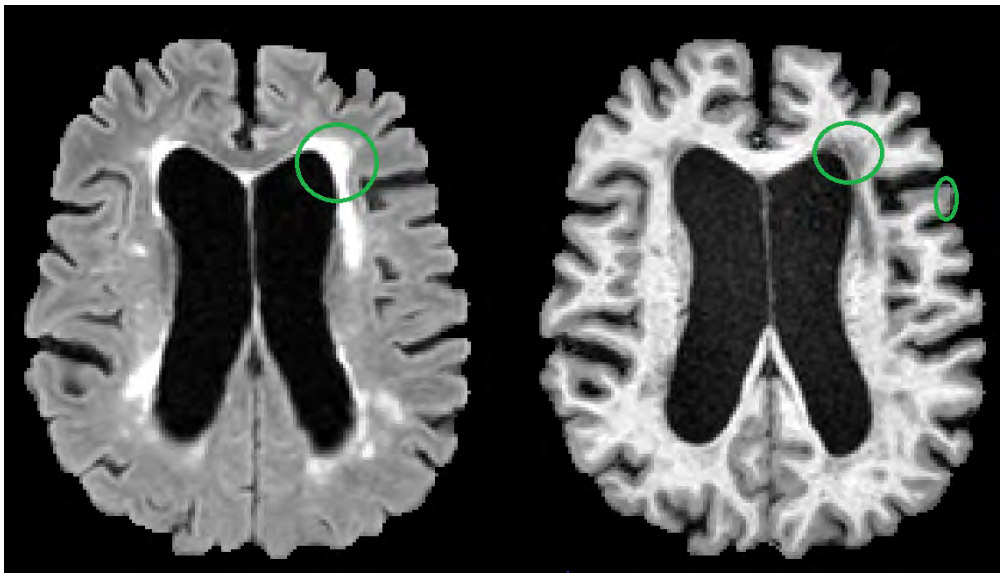


Figure 1: Example of FLAIR modality (on the left) and T1-Weighted modality (on the right). As is possible to see, in FLAIR, the lesions present around the ventricles (green circle) are really bright compared to the rest of the image, while in T1-Weighted image, these brightness is not found. Instead, the lesions appears to have a similar intensity value to the gray matter (green circle)

of synthetic FLAIR images from T1- Weighted images by introducing a global non linear mapping and a local linear mapping from T1-Weighted images to FLAIR. The global mapping determines the similarities from the synthetic image and FLAIR, while the local mapping improves the local details from T1-Weighted images.

3. Materials

The experiments done in this master thesis were conducted on the White Matter Hyperintensities Segmentation Challenge dataset (WMH) and on the Brain Tissue Segmentation Challenge dataset (BraTS) both organized by MICCAI.

3.1. WMH dataset

WMH dataset (MICCAI (2017)) is composed of 60 cases, coming from three different MRI scanners, property of the Vrije Universiteit of Amsterdam.
The first scanner is a 3T Philips Ingenuity, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 9.9ms /4.6ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 4800ms /279ms /1650 ms.
The second scanner is a 3T GE Signa HDxt, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 7.8ms /3.0ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 8000ms /126ms /2340 ms.
The third scanner is a 1.5T GE Signa HDxt, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 12.3ms /5.2ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 6500ms /117ms /1987 ms.

As mentioned, each case presents MRI scan in 3D T1-Weighted and 3D Fluid Attenuated Inversion Recovery (FLAIR) modalities (see Figure 2). In addition, for each patient is given the brain mask as well as the manual annotation of the lesions.

3.2. BraTS dataset

BraTS dataset (MICCAI (2018)) is composed of 285 cases, coming from various scanner from 19 institutions. Each case presents mostly 3T MRI scan in T1-Weighted, T1-Gadolinium (T1Gd), T2-Weighted and FLAIR modalities. To keep the experiments similar between the two datasets, only T1-Weighted and FLAIR modalities were used and Figure 2 as well, shows an example of the dataset.
All the images were pre-processed and pre-registered with a common resolution of $1mm^3$.

4. Analyzed Methods

During the project, several architectures were developed and designed in order to have a qualitative and quantitative analysis of the different techniques for image synthesis. Most of the networks presented here are 3D architectures in which the input usually is a cubic patch of size $32 \times 32 \times 32$ representing a part of the MR volume, while the last 2 are a 2D version in which the input is a square of the size of the whole image. Here we briefly present the various architectures used and give details about the evolution of the project step by step.
There are six different architectures (organized in various setups) that are summarized in Table 1.

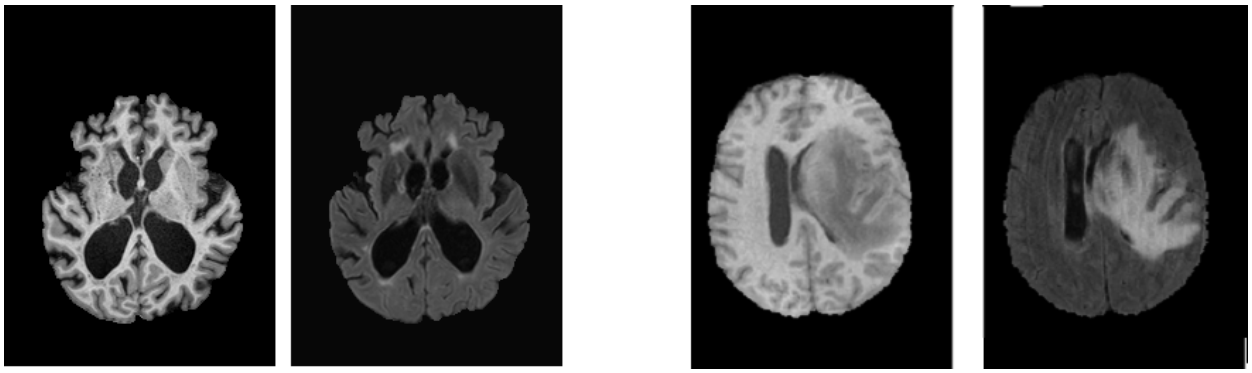


Figure 2: **Left**: WMH dataset, Axial View of MRI scan of one patient. **first image**: T1-Weighted Image, **second image**: FLAIR Image. **Right**: BraTS dataset, Axial View of MRI scan of one patient. **third image**: T1-Weighted Image, **forth image**: FLAIR Image.

Table 1: Summary of the analyzed architectures.

Name	Typology	Dataset	Experiment
Unet	3D	WMH	T1-FLAIR
		BraTS	FLAIR - T1
ResUnet	3D	WMH	T1-FLAIR
		BraTS	FLAIR-T1
GAN	3D	WMH	T1-FLAIR
		BraTS	FLAIR-T1
GAN	2D	WMH	T1-FLAIR
CycleGAN	3D	WMH	T1-FLAIR
			FLAIR-T1
CycleGAN	2D	WMH	T1-FLAIR
			FLAIR-T1

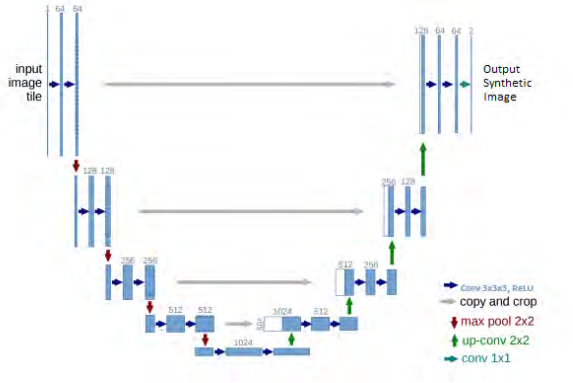


Figure 3: Unet Architecture: as mentioned we can appreciate the U-shape composed by an **encoder** receiving the image in input and extracting features through convolutions and pooling to the latent space and the **decoder** which reconstruct the image.

4.1. Unet shaped architectures

4.1.1. 3D Unet

The first approach which was followed was inspired by the Unet encoder-decoder architecture, proposed by Ronneberger et al. (2015) in the context of biomedical image segmentation (microscopy imaging). The Unet is a Convolutional Neural Network with an Encoder - Decoder shape with the addition of skipping connection between layers. As shown in Figure 3, the Encoder part of the network is composed by 3 Convolutional Layers, each followed by a rectified linear unit (Relu) activation layer and a $2 \times 2 \times 2$ Max Pooling layer with stride 2 for downsampling. In addition, there is a forth Convolutional Layer followed only by a Relu activation layer which drives the input image into the latent space.

The Decoder part is composed by 3 upsampling layers (transpose convolution for upsampling + convolution as pooling) followed by skipping connection layers between the input and the output. For this network each of the filters used is a cube of size 3×3 while the number of filters in each layer varies from 32 to 256 in the encoder part and vice-versa in the decoder part.

4.1.2. 3D ResUNet

Another architecture which was implemented and tested alone in the same set of experiments was an evolution of the 3D Unet, the 3D ResUnet. The main difference is that, to address the vanishing gradient problem, 3D ResUnet uses Residual Blocks in the encoder parts of the Unet in order to exploit the residual connections at each block. This allows a better flow of the gradient through the network layers.

The flow of the architecture is similar to the Unet presented earlier with the addition of Residual Blocks.
A Residual Block is composed by n repetitive layers, each of which presents a 3D Convolutional Layer and a Normalization Layer followed by a Leaky rectified linear unit (Leaky ReLU). The input of the first Convolutional Layer in the block is concatenated to the output of the activation layer.
This architecture has 5 downsampling layers in the Encoder part followed by the same number in the Decoder part as shown in Figure 4. For this network each of the filters used is a cube of size $3 \times 3 \times 3$ while the number of filters (k) varies from 16 to 128 in the Encoder part

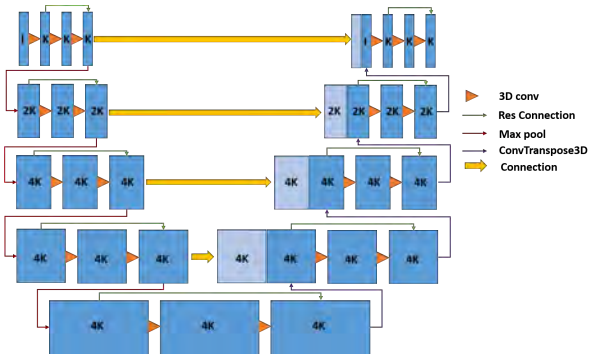


Figure 4: ResUnet Architecture: The shape is similar to the already presented Unet, with the difference of the presence of the residual connection (green arrows). These residual connection help the gradient to backpropagate smoothly so that to make the learning easier for the network.

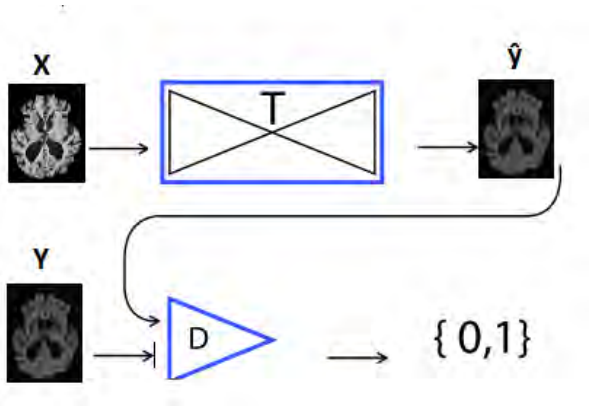


Figure 5: GAN Architecture: T has the shape of an Encoder-Decoder CNN and is one of the two networks presented earlier, while D has the shape of a CNN. The objective of T is to generate images belonging to the target domain, receiving as input images belonging to the input domain. The objective of D is to correctly classify the generated images as "fake" and the real images as "real".

and vice - versa for the decoder part.

4.1.3. Loss Function and Setup

The two networks presented earlier were trained and evaluated in the same modular architecture. The loss function used for training the networks is the L1 Loss, a criterion which measures the Mean Absolute Error (MAE) between each element in the generated set $\hat{y}$ and target y. As shown from Zhao et al. (2015), L1 Loss is usually one of the most used loss functions when it comes to synthesis because it produces less blurried results when compared with L2 (MSE) Loss. It is defined by the following equation:

$$L_1 Loss = \mathbb{E}_{y,\hat{y}}||y - \hat{y}||_1$$

4.2. 3D Generative Adversarial Networks

In 2014 ,Goodfellow et al. (2014) proposed a new type of Neural Network, Generative Adversarial Networks. GANs are composed by two different networks collaborating in an adversarial training. The first network is usually called Generator while the second network is usually called Discriminator. In the first version of GAN, the Generator is so called because it takes as input a noisy distribution and through the training phase produces an output similar to the target.
During the recent years, GANs have been widely used for multiple problems, from Image generation, to Style Transfer, to Domain adaptation. In medical Imaging GANs are used mostly for Segmentation (SeGAN) or for Image translation (MedGAN, cGAN). Since this project is about image translation, we will refer here the Generator as Translator.

As mentioned, GANs can be useful in generating images from random noise distribution, but they can also be used when as input there is an image (see Figure 5). In this setup the generator takes as input the image in the input domain and tries to generate an image in the target domain, as in a standard GAN. A variation of this architecture can appear when the generator takes as input the pair input target, and the output of the network becomes directly conditioned not only form the target, but also from the input. In this case the architecture is the well-known so called Pix2pix, proposed by Zhu (2017).

Figure 6 shows in detail the architecture. The generator takes as input the pair input-target, and according to that, learns a mapping function to translate the input into the target domain. The discriminator instead, tries to understand whether the image it receives as input is the real target or is the generated translation.

4.2.1. Translator and Discriminator

With these preliminaries the Translator is one of the two previously presented networks (Unet/ ResUnet), and it tries to map a source domain image $x \sim p_{source}$ into its ground truth $y \sim p_{target}$.
The Discriminator has the role of a binary classifier, classifying so, if the output of the Translator is an image belonging directly to the target domain (real) or if it comes from another domain and it has been adapted (fake). The training procedure is said to be an adversarial fashion since during each training step, first the discriminator classifies between fake and real samples and then the translator tries to produce a better output.

As shown in Figure 7, the Discriminator D has the shape of a binary classifier, with 3 Convolutional Layers followed each by a rectified linear unit (ReLU) Layer and a Max Pooling Layer. After the Convolutional Layers a series of Fully connected Layers follows each activated by a RelU function.
For the Convolutional Layers the kernel sizes vary from $7 \times 7 \times 7$ in the first layer to $5 \times 5 \times 5$ in the second layer to $3 \times 3 \times 3$ in the third, while the number of filters goes from 64 to 256. The fully connected layers have a

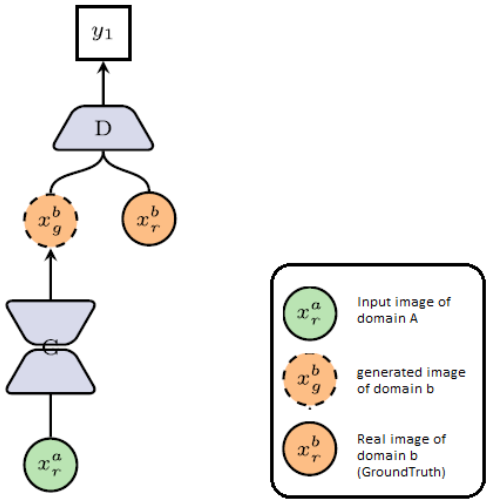


Figure 6: Pix2pix architecture (inspired by the paper Yi et al. (2019)) The difference with a normal GAN is that exists a condition on the input with the target. The generator receives the pair (input - target) as input rather than the only input.

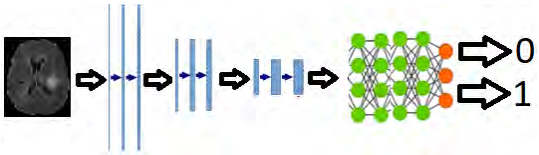


Figure 7: Discriminator architecture: 3 downconvolutional layers + a series of fully connected layers to binary classify the generated image as real or fake.

number of neurons varying from 256 to 1024 in the first three layers, and vice-versa in the last three.

4.2.2. *Loss Function*

Being GANs an architecture composed of two different networks, the Loss Function definition as well as the training setup represents the core of this architecture. The total loss function is composed of two parts: and adversarial part and a feature matching part.

As said, the Discriminator aims to correctly classify the real and the synthetic patches while the Translator aims to fool the Discriminator. Both the networks work in an adversarial fashion following thus, a min - max optimization task on the Adversarial Loss Function:

$$\mathscr{L}_{GAN} = \mathbb{E}_{y \sim p_{data(y)}}[logD(y)] + \mathbb{E}_{x \sim p_{data(x)}}[log(1 - D(T(x))]$$

where the Discriminator D tries to maximize it and the Translator T tries to minimize it.
Since in image to image translation tasks might happen that the translated samples do not produce consistent results but still the Translator fools the Discriminator, a Feature Loss is introduced. This Feature Loss is usually the well known L1 loss (Mean Absolute Error) and it minimizes the differences between the ground truth y and the translated samples $T(x)$ multiplied by a factor λ.

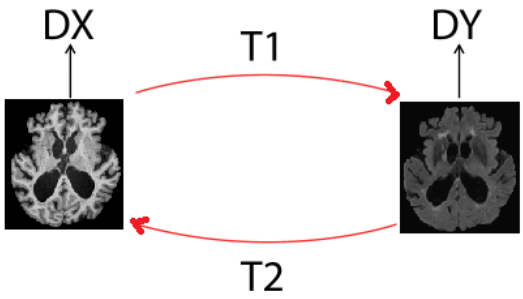


Figure 8: CycleGan scheme: this architecture is formed by 2 Translators (T1 and T2) and 2 Discriminators (Dx and Dy).

The Total GAN loss is therefore composed by a sum of the previous parts as follow:

$$\min_T \max_D \mathscr{L}_{GAN} + \lambda L1$$

with $\lambda = 30$ since it provided a good trade-off in our experimental results. Values of λ smaller than 30 produced less meaningful results, while higher values did not improve with respect to this empiric value.

4.2.3. *Label Smoothing*

Some of the problems that the training of Generative Adversarial Networks can produce, is overfitting and overconfidence. This can occur especially in the Discriminator, during the classification task between the real samples and the generated ones. To overcome this problem, a regularization factor (Wong) during the training of both the Translator and the Discriminator is added to the labels. The Discriminator is originally trained to identify as 1 the patches (or images) coming from real target distribution, while as 0 the patches (or images) coming from the fake (generated) distribution. This is done by replacing the the one-hot encoded label vector 1 as a random distribution vector with values between 0.7 and 1.0 and by replacing the one-hot encoded vector 0 as a random distribution vector with values between 0.0 and 0.3. This smoothing on the labels has the effect of making the Discriminator (in classifying) and the Translator (in generating) less confident when producing its output, reducing so, the possibility of overfitting.

4.3. *CycleGAN*

Zhu (2017) et al. introduced a new method for paired and unpaired image to image translation called CycleGAN. Figure 8 gives a clear view of this CycleGAN architecture. In this configuration there are 2 Translators and 2 Discriminators. The role of the translator is to map the input distribution into the output, while the discriminators have to binary classify the real and the generated samples. Since there are 4 networks, the Translator 1 T_1 will take in input the modality x trying

to generate the modality y, while Translator 2 T_2 does the opposite, generating x from y. Discriminator 1 D_y binary classifies y and $T_1(x)$, while Discriminator 2 D_x binary classifies x and $T_2(y)$.

4.3.1. CycleConsistency Loss and Identity Loss

In addition to this, a cycle consistency criterion (Cycle Consistency Loss) is added. Cycle consistency is based on the fact that one of the two Translators might learn to map the input image into a fixed output distribution, fooling the Discriminator but not producing a representative result. This happens when a Translator (or both) produces a result which is in the target distribution but is not the actual translation of that precise input. To avoid this, Translators are trained to be consistent with each other by imposing $T_2(T_1(x)) \approx x$ and $T_1(T_2(y)) \approx y$.
Furthermore, the goal is also to teach T_1 and T_2 to map the input into the output only when they are different, and to do nothing when in input is the output (Identity Loss). This is done by feeding the images already in modality x to the Translator 2 (T_2) which translates from y to x, because the CycleGAN should understand that the input is already in the correct domain. Therefore, unnecessary changes are penalized. Figure 9 shows the diagram of both the cycle consistency criteria and the identity loss.
Both the Identity Loss and Cycle Consistency Loss are L_1 Losses.

$$\mathscr{L}_{cyc} = \mathbb{E}_{x,\sim p(x)}\|T_2(T_1(x)) - x\|_1$$

$$\mathscr{L}_{id} = \mathbb{E}_{y,\sim p(y)}\|T_1(y) - y\|_1$$

For this configuration, the Total loss is defined as a weighted sum of the aversarial part plus the cyclic consistency part and the identity part as follow:

$$\mathscr{L}(T_1, T_2, D_1, D_2) = \mathscr{L}_{GAN} + \lambda\mathscr{L}_{cyc} + \lambda\mathscr{L}_{id}$$

where $\lambda > 0$.

4.4. Implementation Details

All the codes done in this master thesis work were developed in Python through the Pycharm IDE using Pytorch and the **niclib** library while all the newtorks were trained with an nVidia Titan V GPU available in the Vicorob Lab.

5. Results and Discussion

5.1. Evaluation Metrics

For each experiment the results were evaluated both visually and using different quantitative measures (MSE, SSIM, PSNR), which are usually used when evaluating Image synthesis.

- MSE : Mean Squared Error, defined as

  $$MSE = \mathbb{E}_{y,\hat{y}}\|(y - \hat{y})^2\|_2$$

  where a lower value means better result.

- PSNR: Peak Signal to Noise Ratio, defined as

  $$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

  where MAX_I is the maximum pixel value in the image and a higher value means better result.

- SSIM: Structural Similarity Index, a perceptual difference between two similar images, defined as

  $$SSIM = \frac{(2\mu_x\mu_y + c_1)\cdot(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)\cdot(\sigma_x^2 + \sigma_y^2 + c_2)}$$

  where μ represents the average, σ^2 represents the variance, σ represents the covariance and $c_1 and c_2$ are variables to stabilize division with weak denominators. For SSIM a result of 1.0 means that the two images are identical.

While conducting experiments, these metrics are calculated imagewise, while the average as well as the standard deviation and max/min values are calculated per fold.
To have a correct evaluation of the methods we use a four fold cross-validation strategy on the two datasets presented in Section 3.

The experiments can be summarized in two subsection, one grouping the results obtained synthesizing **FLAIR** from **T1-Weighted** images, and the other grouping the results for the other way around: synthesize **T1-Weighted** images from **FLAIR**. In each of these subsection the results of the different approaches and dataset are analyzed and commented.

5.2. T1-Weighted to Flair

The very first experiment has the goal of synthesizing FLAIR images starting from T1-Weighted Images. For White Matter Hyperintensities dataset are conducted two different experiments, one giving to the networks only one input (the T1-Weighted image) and the other (for the architectures which allow) two inputs, which are the T1-Weighted images plus the binary mask of the lesions. For BraTS dataset, only the T1-Weighted image is given as input to the networks.

5.2.1. Unet 3D and ResUnet 3D

Table 2 show the different setup of configuration for this first experiment.The optimizer used in both the configuration is Adam (lr = 1e-4). We can see that most of the data is similar (i.e. Patience, number of samples, number of Epochs..), although results are different.

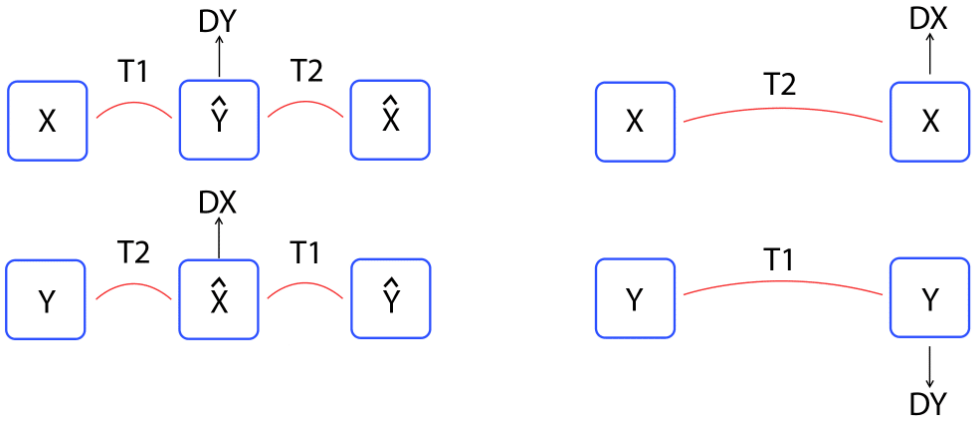


Figure 9: **Left**: Cycle Consistency Loss 1 and 2: The image in the input domain (X) is translated to the target domain ($\hat{y}$) by the first Translator (T1) and then re adapted into the original domain ($\hat{x}$) by the second Translator (T2). MSE is calculated between X and $\hat{X}$. The same process but with the opposite domain is done for the cycle consistency loss 2 (bottom left).**Right**: Identity Loss: The translators (T1 and T2) are trained to not apply any transformation to the input image when this is already in the target domain.

Table 2: Experiments setup for 3D Unet and 3D ResUnet.

Experiment	#Params	#Epochs	#Samples	Patience	Training Time
3D Unet 1 Input WMH	9.14 M	20	2000	**5**	15 m
3D Unet 2 Input WMH	9.14 M	20	2000	5	15 m
3D Unet 1 Input BraTS	9.14 M	20	2000	5	25 m
3D ResUnet 1 Input WMH	3.21 M	20	2000	5	20 m
3D ResUnet 2 Input WMH	3.21 M	20	2000	5	20 m
3D ResUnet 1 Input BraTS	3.21 M	20	2000	5	35 m

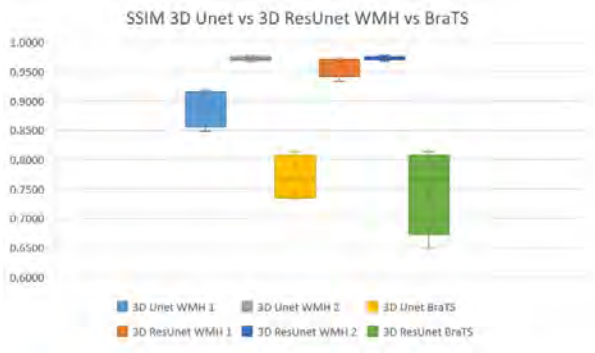


Figure 10: SSIM BoxPlot foldwise calculated on the presented configurations.

Figure 10 represent the distribution **Foldwise** of the calculated SSIM for each of these configurations through boxplot, while Table 3 instead, proposes a more complete comparison of the same configurations by reporting also the qualitative measures MSE and PSNR for both the 3D Unet and 3D ResUnet.

From Table 3 we can see that the best results using this 3D Unet architecture to synthesize FLAIR images from T1-Weighted images are achieved when the

Table 3: Comparison between the three configurations of the two architectures: MSE, SSIM, PSNR

Case	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D Unet 1 Input WMH	0.0019	0.0003	0.8877	0.0323	**56.4474**
3D Unet 2 Input WMH	**0.0017**	**0.0002**	**0.9720**	**0.0034**	56.2893
3D Unet 1 Input BraTS	0.0030	0.0023	0.7699	0.1658	52.4327
3D ResUnet 1 Input WMH	**0.0014**	**0.0002**	0.9611	0.0185	57.3096
3D ResUnet 2 Input WMH	0.0017	**0.0002**	**0.9726**	**0.0034**	55.7748
3D ResUnet 1 Input BraTS	0.0027	0.0022	0.7493	0.0722	**59.3646**

dataset is the WMH and the network is fed with two inputs. Being BraTS dataset with images coming from more scanners, the network seems to not generalize enough well, and in fact, there is a high standard deviation in the SSIM with respect to the experiments done using the WMH dataset.

When comparing the results between Unet 3D and ResUnet 3D from Table 3 we can notice three main objects: when using 2 inputs for WMH dataset, the two architectures are comparable; when passing one input with WMH, the ResUnet can perform better than Unet, but for BraTS dataset, the results are equally bad; about this last configuration we can appreciate the same difficulty that the networks present foldwise to generalize between the different scanners, even though we see a higher variance when using ResUnet (Figure 10).

Figure 11 shows a comparison between 3D FLAIR (on the left), 3D ResUnet (in the middle) and 3D Unet (on the right). From these three images we can appreciate both good things, comparison between the two results and drawbacks of these architectures. The yellow circle represents the lesion around the ventricles which was correctly synthesized by both the networks, but the red circle shows a failure of both the architec-

Table 4: Different Configurations for GAN.

Experiment	#Parameters	#Epochs	#Samples	Patience	Training Time
GAN 1	13.86 M	20	1536	5	45 m
GAN 1 Soft L	13.86 M	20	1536	5	45 m
GAN 1 Soft L BraTS	13.86 M	20	1536	5	65 m
ResGAN 1	7.86 M	20	1536	5	55 m
ResGAN 1 Soft L	7.86 M	20	1536	5	55 m
ResGAN 1 Soft L BraTS	7.86 M	20	1536	5	75 m
GAN 2	10.25 M	20	1536	5	61 m
GAN 2 Soft L	10.25 M	20	1536	5	61 m
ResGAN 2	11.85 M	20	1536	5	59 m
ResGAN 2 Soft L	11.85 M	20	1536	5	59 m
2D GAN	1.85 M	200	1500	20	13 m

tures which confused a piece of gray matter as a lesion. The green circle in the end, shows one of the differences between both the architectures: although the metrics are similar for both of them, and definitely both the synthetic images look synthetic, the one generated by the 3D ResUnet architecture looks more similar to the original FLAIR by being a bit more smooth on the intensities changes with respect to the image generated by 3D Unet.

5.2.2. GAN

The natural further step in developing this family of experiments is to implement both the previous networks in an Adversarial contest and see whether there are improvements or not. As mentioned in Section 3 were implemented two versions of GAN, one 3D and another one 2D. Furthermore, two more 3D architectures were developed in order to have a more extensive comparison between 3D Unet and 3D Resunet, and in the experiments the effect of label smoothing introduced from Salimans et al. (2016) was analyzed as well. Due to the fact that for WMH dataset the results of the networks alone are better when passing two inputs to the networks, the configurations of one input were avoided for this comparison. For all the 3D configurations the Optimizers used are Adam (lr = 1e-6) for the Translator and Adam (lr = 2e-4) for the Discriminator. For the last configuration (2D GAN) the learning rate was linearly decreased after the 100th epoch and the implementation was taken from the github repository provided by Zhu (2017).
Table 4 synthesizes the different configurations tested for GAN.

Figure 12 shows a comparison of the SSIM for all the different tested configurations. Surely the worst performance are achieved by the 2D configuration of GAN (pix2pix). This might be due to the fact that the number of samples was not enough for the network to correctly learn and generalize, even for the WMH dataset. Indeed, it achieved (0.49 ± 0.19) as a mean value for SSIM. Tests for BraTS dataset for this configuration were not done, because the general trend shows that WMH performs better than BraTS, therefore we did not think that this test would be useful. More in detail about the 3D configurations. Some of the tests (GAN 2 and ResGAN 2 wuth BraTS) are missing. This because some of the preliminary experiments on the second dataset produced disappointing results, thus it seemed not useful and redundant to perform further tests. Figure 13 shows a comparison between the proposed GAN using as Translator the Unet described above (GAN 1), and a variation of the Unet (GAN 2). This variation of the Unet uses 4 Convolutional Blocks (a block is formed by conv - pooling - activation - conv - pooling - activation) and 4 UpConvolutional Blocks (a block is formed by upconv - batchnorm - activation - upconv - batchnorm - activation). It is interesting to notice how when using hard labels, the two version of GAN perform in a similar way, but the improvement achieved using Soft Labels in GAN 1 is not visible in GAN 2. Instead, the usage of soft Labels with the second vesrion of GAN worsen the results as shown in Table 5.

Table 5: GAN 1 vs GAN 2 vs ResGAN 1 vs ResGAN 2 vs 2D GAN Hard Labels vs Soft Labels WMH vs BraTS.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	PSNR
GAN 1 Hard	0.0015	**0.0002**	**0.9697**	**0.0120**	57.0659
GAN 2 Hard	**0.0014**	0.0004	0.9608	0.0211	58.4175
ResGAN 1 Hard	**0.0014**	0.0004	0.9356	0.0540	**59.6918**
ResGAN 2 Hard	0.0337	0.0355	0.8007	0.1609	55.5008
GAN 1 Soft	0.0016	**0.0005**	**0.9710**	**0.0107**	57.4216
GAN 2 Soft	**0.0015**	**0.0005**	0.7813	0.1454	**61.0257**
ResGAN 1 Soft	0.0017	**0.0005**	0.8851	0.1267	59.4640
ResGAN 2 Soft	0.0159	0.0064	0.7385	0.1196	48.6462
GAN 1 Soft BraTS	**0.0142**	0.0196	0.7093	**0.0149**	**52.9729**
ResGAN 1 Soft BraTS	0.0310	**0.0064**	**0.7210**	0.1196	48.6462
2D GAN WMH	2.1215	0.2245	0.4858	0.1911	37.2948

Table 6: CycleGAN configuration.

Experiment	Parameters	Epoch	Samples	Patience	Training Time
3D CycleGAN	27.72 M	20	1536	5	79 m
2D CycleGAN	13.2 M	200	1500	20	25 m

Figure 13 shows as well the comparison between the proposed GAN which uses the earlier presented 3D Resunet as Translator(ResGAN 1) and a variation of the ResUnet (ResGAN 2). This variation of the 3D ResUnet is composed by 1 Convolutional Block (presented earlier) and 3 Residual Unit (each unit is composed by a convolution - batchorm - activation) and 3 UpRes-Block (each block is composed by a up convolution - batchnorm- activation) plus a final convolution block. About the BraTS dataset, Figure 14, shows that even though ResGAN 1 achieves a better result (as maximum SSIM), the variance of GAN 1 is smaller.

5.2.3. CycleGAN

The last family of the experiments done for synthesizing FLAIR from T1 Weighted images was the CycleGAN architecture, both in 3D and in 2D, and the set of configurations tested is reported in Table 6.

For the 3D CygleGAN configuration, the experiment was only one, because it was selected the best network among the 3D GANs (which ended up to be GAN 1 with SoftLabels), but with only one input to the Translators. This is due to the fact that cycle

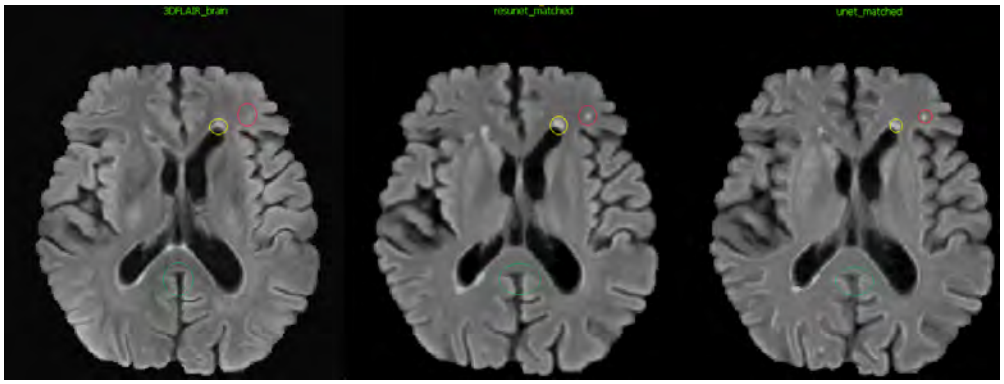


Figure 11: Comparison of two images of WMH dataset. The original FLAIR image is on the left, the 3D ResUnet synthesized image is in the middle and on the right there is the 3D Unet synthesized image.

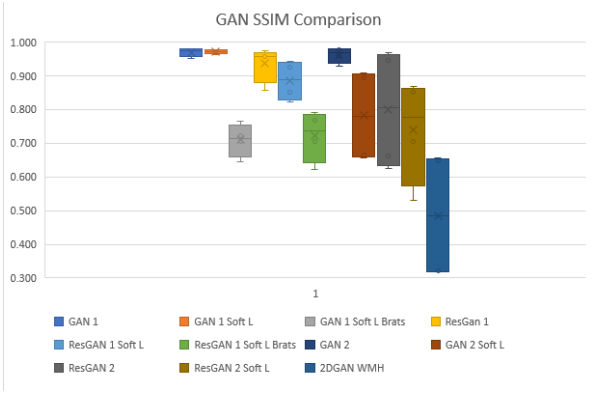


Figure 12: Results of the experiments for GAN using WMH dataset (GAN1, ResGAN1, GAN2, ResGAN2, 2D GAN, GAN 1 Soft Labels, GAN 2 Soft Labels, ResGAN 1 Soft Labels, ResGAN 2 Soft Labels) and BraTS dataset (GAN 1 Soft Labels BraTS, ResGAN 1 Soft Labels BraTS).

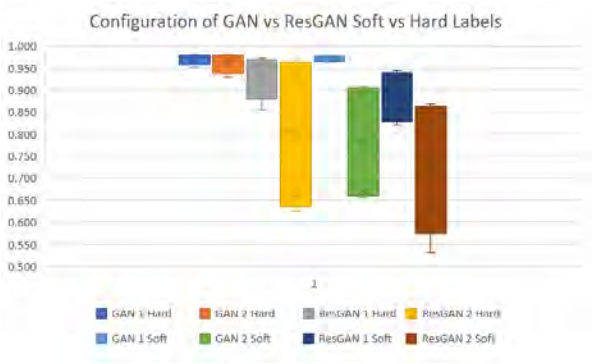


Figure 13: Comparison on WMH dataset between GAN 1, GAN 2, ResGAN 1 and ResGAN 2 while using Hard and Soft Labels.

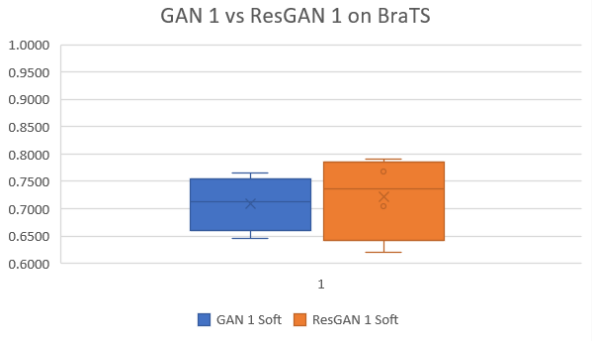


Figure 14: GAN 1 vs ResGAN 1 on BraTS dataset.

consistency needs to be respected. If T_1 receives as input the T1-Weighted images plus the lesion and it outputs only FLAIR, then when calculating the cycle consistency loss ($T_1(T_2(\text{FLAIR}))$, we need T_2 to output two values, but for the way it is designed, this was not a feasible and optimal idea. Instead, reducing the input to only the T1-Weighted images to the first translator was a considered as a more optimal solution from the programming point of view. Figure 16 shows this comparison between the 2D and the 3D implementation of the CycleGAN architecture. As is possible to see (from the numbers in Table 7 as well) the results were not good. This might be due to the fact the this architecture (3D) was too complex for the task and the number of patches was not enough for the networks. More in detail, what was possible to see from the images in result, due to this limited number of samples, and due to the cycle consistency constrain, both the networks were not able to learn any mapping function. In fact (both for 2D and 3D implementation) T_1(T1-Weighted) was really similar to T1-Weighted images. Figure 15 shows a visual comparison of most of these architectures. As is possible to see, GAN1 with the usage of soft labels (top row, third element) outperforms the other architectures, producing a result really similar to the target FLAIR, the lesions around the ventricles are well positioned and the image obtains the same contrast as a real FLAIR image. Right after it is placed ResGAN1 with the usage of soft labels (bottom row, first element), which still produces a good results, although more smoothed than GAN1, therefore details can be less appreciated. GAN2 (bottom row, second element) which from Figure 12 shows good results

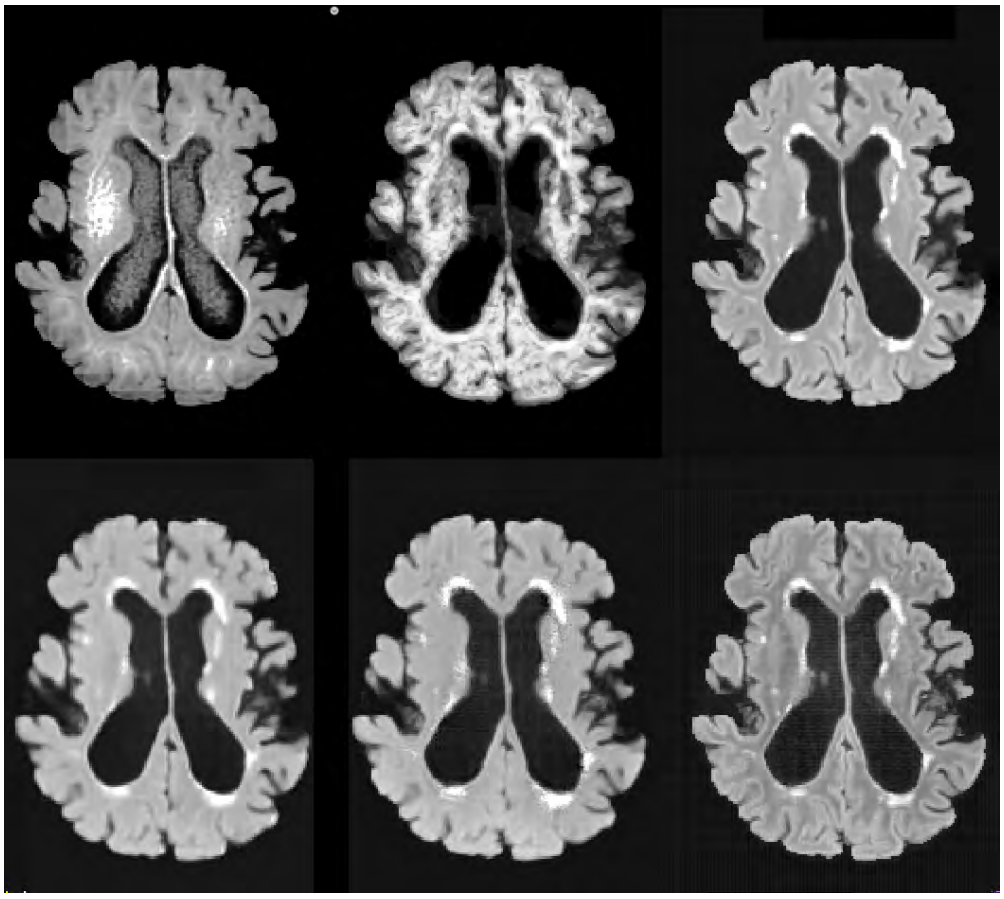


Figure 15: comparison of some of the GANs and CycleGANs on WMH dataset. **Top row**: ResGAN2 (Soft Labels), 3D CycleGAN, GAN1 (Soft Labels). **Bottom row**: ResGAN1 (Soft Labels), GAN2 (Soft Labels), ResGAN2 (Hard Labels). As we can see the Cyclegan cannot map T1-Weighted images into FLAIR, while the others mostly do it. It is interesting to notice how GAN1 outperforms the other architectures, even though ResGAN1 produces a fair result. GAN2 suffers from label smoothing and indeed the image has some artifacts, especially around the ventrivles, while ResGAN2 suffers a bit from the checkboard effect both with Soft Labels and Hard Labels.

Table 7: CycleGAN results.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D CycleGAN	2.6020	4.1941	**0.4686**	**0.0553**	**20.2208**
2D CycleGAN	**1.1121**	**3.6555**	0.3386	0.1002	35.4372

when compared with GAN1, suffers from the effect of label smoothing and produces some artifacts close to the ventricles. Together with ResGAN2 both with hard (top row, first element) and soft labels (bottom row, third element), we notice the presence of the so called "checkboard effect", an artifact which is common when dealing with 3D CNN. The main reason for this effect to occur is because of the transpose convolution in the Decoder part of the network. By replacing it with a block made of upscaling + convolution (as for GAN1 and ResGAN1) this effect disappear. As mentioned above, the 3D CycleGAN (top row, second element) cannot learn a valid mapping function, therefore does almost no operation (a part from smoothing and adding noise) to the input image, therefore this result is similar to the T1-Weighted image.

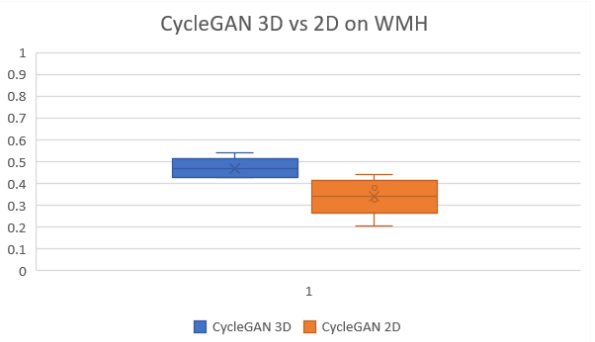


Figure 16: 3D vs 2D implementation of CycleGAN.

5.3. Flair to T1-Weighted

The second group of experimental tests that we performed, examines the previously presented architectures in synthesizing T1-Weighted images from FLAIR images. This is basically a specular analysis of most of the previously cited configurations of the other experiment. However, some of the configurations were avoided due to time consuming and due to the fact

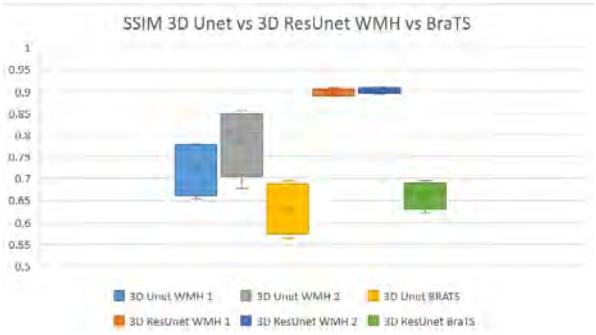


Figure 17: SSIM BoxPlot foldwise calculated on the presented configurations.

Table 8: Comparison between the three configurations of the two architectures: MSE, SSIM, PSNR

Case	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D Unet 1 Input WMH	**0.0022**	**0.0001**	0.7741	**0.0547**	53.7906
3D Unet 2 Input WMH	0.0027	0.0005	**0.7859**	0.0673	**54.9755**
3D Unet 1 Input BraTS	0.0175	0.0065	0.6303	0.0772	50.8214
3D ResUnet 1 Input WMH	0.0010	0.0001	0.8988	0.0071	60.4702
3D ResUnet 2 Input WMH	**0.0009**	**0.0000**	**0.9020**	**0.0056**	**60.7154**
3D ResUnet 1 Input BraTS	0.0033	0.0022	0.6620	0.0193	50.2883

that the obtained results (especially in more complex architectures such as cycleGAN) were not good enough to justify another trial.

5.3.1. Unet3D and ResUnet 3D

Table 2 shows the setup of these configuration of experiments as well, and the results can be seen from Figure 17. These first results show that synthesizing T1-Weighetd images from FLAIR images is a more difficult task with respect to the opposite, even if to the network are fed two inputs as shown in Table 8, in which we actually notice that there is almost no difference in SSIM between the two configurations but instead the Average MSE is better when feeding only one input to the network. This general worsening in the results generating T1 Weighted images from FLAIR happens also when using the BraTS dataset, justifying so the fact that synthesizing T1 -Weighted images form FLAIR is a more difficult task with respect to the opposite.
When we change network an we switch to the 3D ResUnet, we can appreciate this worsening as well, even though the best Average SSIM is still 0.9020 ± 0.0056, so an average result. Furthermore, when analyzing Figure 17 we notice that not only the results of ResUnet are better in general, but also less dispersed from the median value. This means that in general, the network generalizes better than the 3D Unet when it comes to synthesize T1-Weighted images from FLAIR.

Table 9: Gan Configurations FLAIR to T1-Weighted.

Experiment	#Parameters	#Epochs	#Samples	Patience	Training Time
GAN 1 Soft L	13.86 M	20	1536	5	45 m
GAN 2 Hard L	10.25 M	20	1536	5	61 m

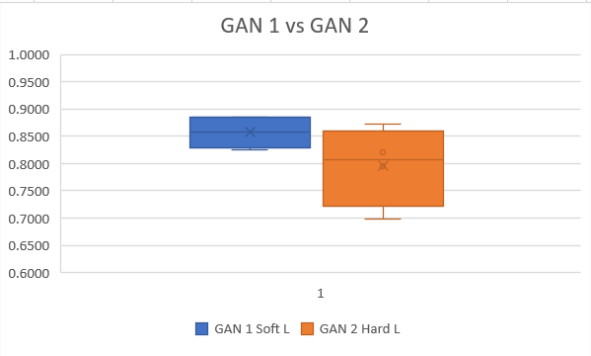


Figure 18: GAN 1 vs GAN 2 FLAIR to T1.

5.3.2. GAN

With respect to the previous experiment (T1-Weighted to FLAIR), this subsection of experiments only shows 2 configurations for synthesizing T1-Weighted images from FLAIR, as shown in Table 9. The fact that only these two configurations are presented is both due to the time consuming, and to the fact that since we generally see a worsening in the performances when synthesizing T1-Weighted images, it was worth to test only the best configurations on the easiest dataset (WMH).

5.3.3. CycleGAN

This last configuration ends the set of experiments done for this master thesis project. Results are presented in Table 11.

When comparing the 2D implementation to the 3D implementation, we notice that in general the 3D is better, yet the results are meaningless since the networks (as mentioned for the other experiment) do not learn any mapping. Another reason for this can be the fact that the discriminator outplays the translators quickly, but this happens no matter the architectures used.

6. Discussion

According to what has been presented in the Experiments and Results Section, we are able to produce some conclusions on the analysis.

Methods - Families. These sets of experiments done allow us to rank the different architectures according to the task itself (through the calculated metrics), but also according to the complexity of the network, training time. The general trend of the results, shows that the best approach to all the test was the **3D GAN** in a configuration with the 3D Unet as Translator and the usage

Table 10: Gan comparison of MSE, SSIM, PSNR.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	PSNR
GAN 1 Soft L	**0.0013**	**0.0003**	**0.8570**	**0.0101**	48.8080
GAN 2 Hard L	0.0036	0.0008	0.7966	0.0335	**54.2240**

Table 11: CycleGAN results.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D CycleGAN	8.26452	6.798724	**0.402639**	**0.053957**	29.97478
2D CycleGAN	**3.644269**	**0.0811**	0.3063	0.118516	**39.7191**

of soft labels. Most probably, this setup is the best to solve this type of problems, according both to the results achieved, and without sacrificing too much time for training. **CycleGAN** (3D and 2D) along with 2D GAN (pix2pix) achieved the worst results overall, this might be due to the high complexity of the networks, combined with few amounts of datasamples (especially in 2D) which do not allow the translators to map the input distribution into the output distribution maintaining a correct cycle consistency within itself. **3D Unet** and **3D ResUnet** achieved as well a good result and **3D ResUnet** even outperformed **3D Unet** when synthesizing T1-Weighted images from FLAIR, and they are the configuration needing the least time to train, but the speedup in time do not compensate the worsen in the performance (although good) when compared to 3D GAN configuration.

T1-Weighted to FLAIR synthesis. As an overall result, we can notice that for each of the presented architectures and for both the datasets, trying to synthesize FLAIR from T1 is a relatively easy task. When comparing the numbers, we also notice that, when is possible to add two inputs to the network, as the lesions in T1 can be confused with the White matter itself, producing more realistic FLAIR (with also correct position of the lesions) is possible. Indeed, in the configurations (CycleGAN) and dataset (BraTS) which did not allow this additional input, results were less good.

FLAIR to T1-Weighted synthesis. On the other hand, trying to synthesize T1 from FLAIR it appeared to be a more difficult task than the previous one, for all the architectures and without any distinction of dataset or number of inputs. This might be due to the fact that generating T1-Weighted from FLAIR sequencies alone (without other modalities such as T2-Weighted or PD (Lee et al., 2019)) might actually not be possible.

Datasets. The presented study evidenced that for these architectures, the WMH dataset was easier to work with, even though it presented less cases. The reason why BraTS dataset achieved worse results, might be due to the fact that the images are coming from 19 scanners, therefore it is more difficult for the network to generalize well on so many different scanners, while for WMH dataset, the task is easier since the images are coming only from three scanners.

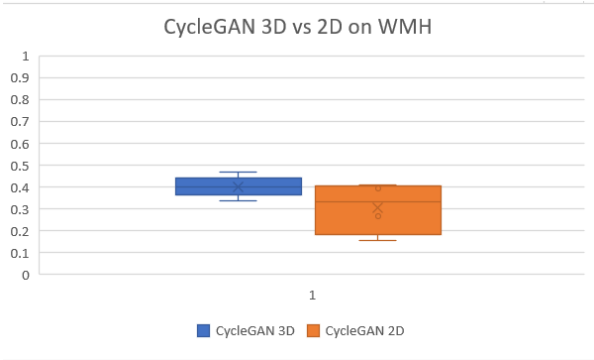


Figure 19: 3D vs 2D implementation of CycleGAN.

Table 12: Performance comparison of the analyzed architectures.

Network	Typology	Training Time	Result rank
Unet	3D	15 m	3
ResUnet	3D	20 m	2
GAN	3D	53 m (as average)	1
GAN	2D	13 m	4
CycleGAN	3D	75 m	5
CycleGAN	2D	25 m	6

1 Input v 2 Input. For the dataset (WMH) where it was possible to test two different configurations for the same problem, it is evidenced that providing the lesion mask as well as the modality in input to the network made the synthesis problem easier, especially for T1 to FLAIR. The fact that for this specific problem, the difference in SSIM when comparing 2 inputs vs 1 input is higher than the rest of the experiments, is due to the fact that the lesions are represented in a very different way in the two modalities. In FLAIR these are white, and easy to spot, while in T1 these can be confused with the White Matter itself, therefore when receiving 1 input, the network is not able to synthesize the lesions correctly.

7. Conclusions and future work

The main goal of this master thesis was to analyze state of the art techniques for synthesis - cross modality problems in Brain MRI images, more specifically, the synthesis from T1-Weighted images to FLAIR and viceversa was examined. To tackle this task, different architectures were developed, in order: 3D Unet, 3D ResUnet, 3D GAN (with variations on Translators and labels), 2D GAN, 3D CycleGAN and 2D CycleGAN. All these architectures were tested on two well known international brain dataset (WMH and BraTS), the first one with images from three sanners, the second one with images from 19 scanners. The analysis done shown that 3D GAN with the usage of label smoothing achieves really good results for WMH dataset when it comes to synthesize FLAIR from T1-Weighted images, along with 3D Unet and 3D ResUnet, this last performing in a more robust way when synthesizing T1-Weighted im-

ages from FLAIR. About BraTS dataset, all the architectures in general were less performing, and as mentioned in the discussion section, this might be due to the big amount of different scanners. In general, 3D Unet, 3D ResUnet and 3D GAN were really good performing in both the tasks. As a drawback, some of the results were not exactly as expected, showing that for this type of problem, the CycleGAN configuration was not able to generalize well and thus produced the worst results, both in 2D and in 3D configuration. A possible solution to this would be to apply some sort of data augmentation in order to have more samples to train with. The problem due to the cycle consistency might indeed be related to this lack of samples. As a further step to this work, the implementation and the testing of the Perceptual Similarity index as a loss function (Snell et al., 2017) which shown good results in other fields of research. A final step might be the implementation and the testing of a new state of the art technique Collagan (Lee et al., 2019) which will produce a more exaustive research and comparison.

Acknowledgments

First of all, I would like to thank my family for all the support I received during these two years away from them. Surely, without them I wouldn't be able to achieve anything in my life. Then, I would like to thank my professors and supervisors Dr Xavier Lladó and Dr Arnau Oliver for all the care they took with me during these months. I would like to thank also the PhD students in the VICOROB lab, Albert, Liliana, Kaisar, for all the hours spent listening to my doubts and for the practical help they provided me. Last, but not least, this is to me. Because at the end, if you really try to achieve something, it happens.

References

Dar, S.U., Yurt, M., Karacan, L., Erdem, A., Erdem, E., Çukur, T., 2019. Image synthesis in multi-contrast mri with conditional generative adversarial networks. IEEE transactions on medical imaging 38, 2375–2388.

Freeman, W.T., Pasztor, E.C., Carmichael, O.T., 2000. Learning low-level vision. International journal of computer vision 40, 25–47.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Hiasa, Y., Otake, Y., Takao, M., Matsuoka, T., Takashima, K., Carass, A., Prince, J.L., Sugano, N., Sato, Y., 2018. Cross-modality image synthesis from unpaired data using cyclegan, in: International workshop on simulation and synthesis in medical imaging, Springer. pp. 31–41.

Jog, A., Roy, S., Carass, A., Prince, J.L., 2013. Magnetic resonance image synthesis through patch regression, in: 2013 IEEE 10th International Symposium on Biomedical Imaging, IEEE. pp. 350–353.

Lauritzen, A.D., Papademetris, X., Turovets, S., Onofrey, J.A., 2019. Evaluation of ct image synthesis methods: From atlas-based registration to deep learning. arXiv preprint arXiv:1906.04467 .

Lee, D., Kim, J., Moon, W.J., Ye, J.C., 2019. Collagan: Collaborative gan for missing image data imputation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2487–2496.

Liu, G., Si, J., Hu, Y., Li, S., 2018. Photographic image synthesis with improved u-net, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), IEEE. pp. 402–407.

MICCAI, 2017. Miccai White Matter Hyperintensities 2017. URL: `https://wmh.isi.uu.nl`.

MICCAI, 2018. Miccai BraTS 2018. URL: `https://www.med.upenn.edu/sbia/brats2018/data.html`.

Miller, M.I., Christensen, G.E., Amit, Y., Grenander, U., 1993. Mathematical textbook of deformable neuroanatomies. Proceedings of the National Academy of Sciences 90, 11944–11948.

Nie, D., Trullo, R., Lian, J., Wang, L., Petitjean, C., Ruan, S., Wang, Q., Shen, D., 2018. Medical image synthesis with deep convolutional adversarial networks. IEEE Transactions on Biomedical Engineering 65, 2720–2730.

Perone, C.S., Ballester, P., Barros, R.C., Cohen-Adad, J., 2019. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. NeuroImage 194, 1–11.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham. pp. 234–241.

Salem, M., Valverde, S., Cabezas, M., Pareto, D., Oliver, A., Salvi, J., Rovira, À., Lladó, X., 2019. Multiple sclerosis lesion synthesis in mri using an encoder-decoder u-net. IEEE Access 7, 25171–25184.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans, in: Advances in neural information processing systems, pp. 2234–2242.

Shin, H.C., Tenenholtz, N.A., Rogers, J.K., Schwarz, C.G., Senjem, M.L., Gunter, J.L., Andriole, K.P., Michalski, M., 2018. Medical image synthesis for data augmentation and anonymization using generative adversarial networks, in: International workshop on simulation and synthesis in medical imaging, Springer. pp. 1–11.

Snell, J., Ridgeway, K., Liao, R., Roads, B.D., Mozer, M.C., Zemel, R.S., 2017. Learning to generate images with perceptual similarity metrics, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 4277–4281.

Vemulapalli, R., Van Nguyen, H., Kevin Zhou, S., 2015. Unsupervised cross-modal synthesis of subject-specific scans, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 630–638.

Wang, L., Chen, W., Yang, W., Bi, F., Yu, F.R., 2020. A state-of-the-art review on image synthesis with generative adversarial networks. IEEE Access 8, 63514–63537.

Wong, W., . What is label smoothing. URL: `https://towardsdatascience.com/what-is-label-smoothing-108debd7ef06`.

Xiang, L., Li, Y., Lin, W., Wang, Q., Shen, D., 2018. Unpaired deep cross-modality synthesis with fast training, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 155–164.

Yang, Q., Li, N., Zhao, Z., Fan, X., Chang, E.I., Xu, Y., et al., 2018. Mri cross-modality neuroimage-to-neuroimage translation. arXiv preprint arXiv:1801.06940 .

Yi, X., Walia, E., Babyn, P., 2019. Generative adversarial network in medical imaging: A review. Medical image analysis 58, 101552.

Yu, B., Zhou, L., Wang, L., Fripp, J., Bourgeat, P., 2018. 3d cgan based cross-modality mr image synthesis for brain tumor segmentation, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE. pp. 626–630.

Zhang, Z., Yang, L., Zheng, Y., 2018. Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9242–9251.

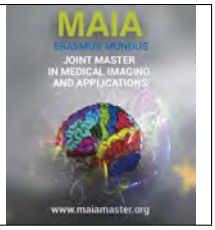
Zhao, H., Gallo, O., Frosio, I., Kautz, J., 2015. Loss func-

tions for neural networks for image processing. arXiv preprint arXiv:1511.08861 .

Zhu, Taesung Park, P.I.A.A.E., 2017. Mri cross-modality neuroimage-to-neuroimage translation. Proceedings of the IEEE International Conference on Computer Vision (ICCV) , 2223–2232doi:arXiv:1801.06940[cs.CV].

Medical Imaging and Applications

Master Thesis, August 2020



Automated 3D DCE-MRI Breast tissue Segmentation and Background Parenchymal Enhancement Classification

Sholpan Zhaisanbayeva, Supervisor: Robert Marti

Universitat de Girona, Girona, Spain

Abstract

The number of women suffering from breast cancer is growing every year. Due to this fact, early detection is one of the main targets, which could be helpful for removing the cancer completely. MRI modality represents high sensitivity imaging technique for detection and recognition of breast abnormality, regardless of breast density. Background parenchymal enhancement (BPE) is the enhancement of fibroglandular tissue (FGT) of the breast in response to MRI contrast agent. As mammographic breast density has been established as an independent risk factor, current researches prove that BPE observed on breast DCE MRI appeared to be strongly predictive of breast cancer risk and it can be used as a biomarker. The aim of this work is to investigate the use of automated tools using deep learning techniques to segment the breast tissues and classify BPE into their respective categories. The database was collected from 405 patients and each of these cases was manually evaluated by 3 professional readers from different countries. The qualitative assessment provided by radiologist was adopted as ground-truth for the automatic method. Breast tissue segmentation, as fundamental task for breast density estimation, was obtained using patch based 3D U-Net architecture with ResNet backbone. Segmented fibroglandular tissues were classified into 4 BPE classes using ResNet model. The proposed segmentation method was evaluated using the dice similarity coefficient and the best-obtained result for FGT is **0.76** and **0.82** is for the fat tissue. Overall accuracy obtained from BPE classification is **61.3%**.

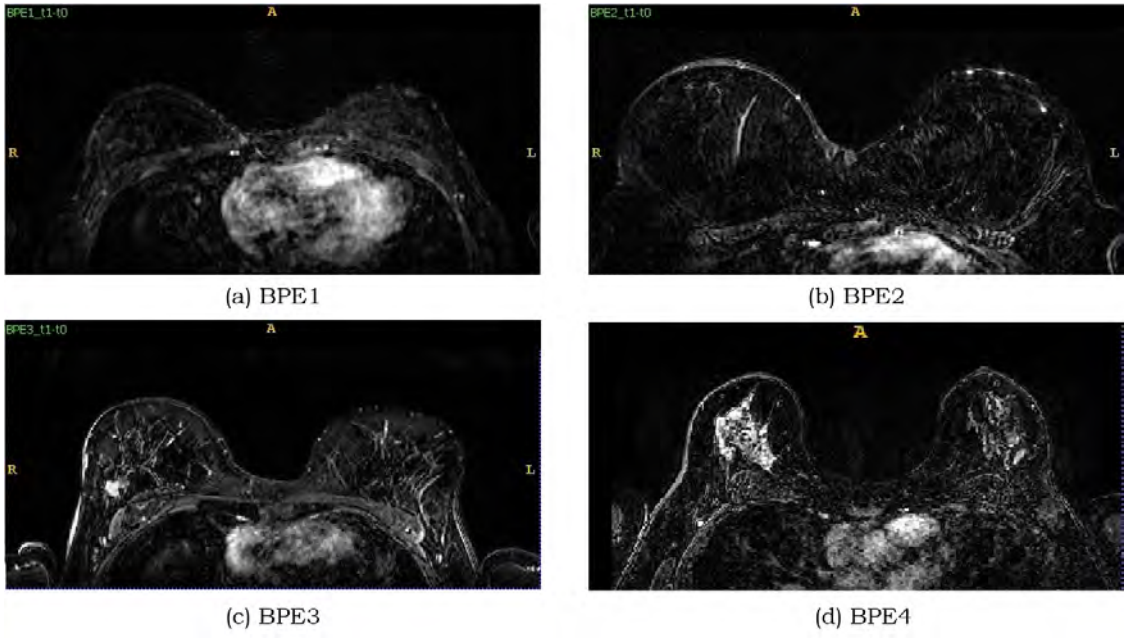
Keywords: BPE, U-Net, Fibroglandular tissue, Segmentation, Classification, ResNet

1. Introduction

Breast cancer is the most frequent cancer among women and the second leading cause of death among women worldwide. According to World Health Organization (WHO) breast cancer impacts 2.1 million women each year, and also causes the greatest number of cancer-related deaths among women. In 2018, it is estimated that 627,000 women died from breast cancer, that is approximately 15% of all cancer deaths among women. While breast cancer rates are higher among women in more developed regions, rates are increasing in nearly every region globally (WHO, 2020).

The reason of appearance of cancer cells is unknown nowadays, including breast cancer as well. Different factors are involved such as age, genetic history, hormonal changes, lifestyle, exposure to radiation. In order to improve breast cancer outcomes and survival, early detection is crucial. There are two early detection strategies for breast cancer: early diagnosis and screening. Early diagnosis strategies focus on providing timely access to cancer treatment by reducing barriers to care and/or improving access to effective diagnosis services. Screening consists of testing women to identify cancers before any symptoms appear (WHO, 2007).

The most common early stage diagnostic tool for breast cancer is a mammography (X-ray), which is considered as the most acceptable and cost-effective method for patients and generally used as the primary screening. Other types of detection of breast changes are ultrasound and MRI. These devices should be used when X-ray mammogram imaging is not conclusive and the breast has a large amount of dense tissue, or for the patient who already has been diagnosed with breast cancer, to help to measure the size of the cancer lesion, look for other tumors in the breast, and to check for tumors in the opposite breast (Hubbard RA, 2014). Nowadays,



(a) BPE1 (b) BPE2

(c) BPE3 (d) BPE4

Figure 1: Subtracted axial DCE-MRI slices of the first post-contrast and pre-contrast images (t1-t0) of 4 different patients and categories of BPE: (a) BPE 1 - minimal enhancement, (b) BPE 2 - mild enhancement, (c) BPE 3 - moderate enhancement, (d) BPE 4 - marked enhancement

American Cancer Society (ACS) recommends to use ultrasound and MRI for breast cancer as an extra modality to mammography for screening (Wu et al., 2016).

Breast MRI is an important modality for high-risk women and for clinical treatment process. Dynamic contrast-enhanced MRI (DCE-MRI) consists of an MRI as examination with the administration of contrast agent (gadolinium), acquiring several 3D images at different time periods. It has main abilities for the diagnosis, detection, and monitoring of malignancy. Additionally, it allows visualization of the stage of disease and visualization of lesion heterogeneity, detection of changes in angiogenic properties before morphological changes, and the possibility to predict the overall response either before the start of therapy or early during treatment (Turnbull, 2009). The physical meaning of the DCE MRI is a reflection of the dynamic signal intensity variations induced by uptake of contrast agent over a period of time and can be described by contrast enhancement kinetics (Kuhl et al., 1999).

MRI Breast tissue segmentation is needed to perform an automatic analysis of those images. Many applications in MRI such as multimodal breast image registration, computer aided analysis of DCE MRI, and breast density estimation require segmentation as an initial step. Related to last one, breast density has been identified as an important risk factor for developing breast cancer. Automated segmentation of breast tissues in breast DCE MRI is one of the major aspect and initial step of this work. To obtain breast density estimation from MRI we need to segment the breast body first and then the fibroglandular tissue (FGT). The segmentation of the breast is initially performed to exclude other tissues that does not belong to the breast, such as pectoral muscle. Next step is segmentation of the FGT after delineating the breast body from other chest organs on the image.

Background parenchymal enhancement (BPE) is the enhancement of fibroglandular tissue (FGT) of the breast in response to MRI contrast agent. It characterizes the risk of breast cancer similar to the evaluation of breast cancer and breast density on mammography (Arslan G, 2017). BPE is evaluated in 4 classes determined in the breast imaging reporting and data system (BI-RADS): minimal enhancement (25% enhancement of the glandular tissue); mild (26–50% enhancement); moderate (51–75% enhancement) and marked enhancement (75% enhancement) (Satoko et al., 2012). Figure 1 presents examples of DCE-MRI axial slices of the 4 different BPE categories.

Current researches prove that BPE can be used as a biomarker to predict the breast cancer risk, like mammoghaphic breast density estimation was taken on the base as an independent risk factor evaluation (Pike and Pearce, 2013).

Nowadays, BPE quantification is an important research topic as high BPE shown to be linked to increased breast cancer risk. As BPE evaluation is performed by radiologists, it may be affected by the subjectivity of manual evaluation together with intra- and inter-observer variability, which ranges widely from fair to substantial. This variability can be a reason of various factors like level of experience, specialization area and many others, also this assessment is tiresome and time consuming. Therefore, automated approaches to analyze BPE in DCE MRI images are in demand (Pujara

et al., 2017a).

In this work we present an automatic breast tissue segmentation and BPE classification method using Deep Learning techniques. We obtain segmentations for breast body (fat) and FGT (dense) tissues using an architecture inspired by patch based 3D U-Net with ResNet backbone. The obtained results are used as input to a classification network based on ResNet architecture.

2. State of the art

Deep learning approaches have become a dominant technique in medical imaging field, because they started to demonstrate their considerable capabilities in different challenges (Hesamian et al., 2019). One of the first object detection systems using convolution neural networks (CNNs) was proposed in 1995. In this research, the authors used a CNN with four layers to detect nodules in X-ray images (Lo et al., 1995). However, traditional approaches are also widely used for the analysis and processing of medical images. Since our work consists of two parts, segmentation and classification, we discuss state-of-the-art approaches in two different subsections below.

2.1. Breast tissue segmentation

Automated breast outer and inner tissue segmentation was implemented by Sehrawat et al. (2018), where research was done with two steps. In the first step, they segmented breast body using morphological operations to remove background noise and any leftover pectoral muscle; landmark points were detected to find outer breast borders and a B-Spline curve was applied to remove non-breast tissues from the lower part of the image, which mainly contains other body parts (liver, heart etc.). In the next step, the inner breast tissue was segmented after subtraction of the different MRI modalities and Otsu thresholding.

Another approach based on 3-D probabilistic atlas was proposed by Gubern-Mérida et al. (2015) to obtain the borders of breast from the image background and the chest wall. They used expectation–maximization(EM) algorithm to obtain a threshold intensity value to distinguish FGT from adipose tissue. Bias field correction, sternum landmark detection and intensity normalization were applied in the preprocessing step.

With the recent popularity of Deep Learning techniques, especially convolutional neural networks, methods based on them started to appear in literature. One of the typical convolutional neural networks that was developed for biomedical image segmentation is the well-known U-Net architecture presented by Ronneberger et al. (2015). This proposed method was used for 2D images and very rapidly 3D implementation arised Çiçek et al. (2016). The implemented architecture consists of 2 parts: contraction path (encoder) and expansion path (decoder) and is similar to autoencoders. The encoding part compresses the input image into the latent space and the decoder part reconstructs the image back.

One of the last publications on breast tissue segmentation proposed by (Zhang et al., 2019) is based on a deep learning approach using U-Net architecture. In this paper authors utilize U-Net for separating three-class labels on each MR image, including fat tissue and FGT inside the breast and all nonbreast tissues outside the breast. The first U-Net was used to segment the breast from the entire image. Then, within the obtained breast mask, the second U-Net was used to differentiate between fat and FGT. The left and right breasts were separated using the centerline, and a square matrix containing one breast was cropped and used as the input.

A method that was proposed by Piantadosi et al. (2018) for fully automated breast-mask extraction in DCE-MRI data, was based on a U-Net architecture also. Authors considered the 3D volume as a composition of slices and performed a slice-by-slice segmentation. They modified the original U-Net proposal: 1) the output feature map of the network was set to one channel to speed up the convergence during the training phase; 2) zero-padding with a size-preserving strategy was applied for preserving the output shapes; 3) batch normalization layers after each convolution was applied to improve the training phase.

In the work done by Dalmş et al. (2016), researchers experimented with 2 ways of the U-Net architecture. In first case, they trained two consecutive U-Nets: first one for segmenting the breast in the whole MRI volume and the second one for segmenting FGT inside the segmented breast. In the second method, a single 3-class U-Net was used, which performs both tasks simultaneously by segmenting the volume into three regions: nonbreast, fat inside the breast and FGT inside the breast. The advantage of convolutional networks with U-Net architecture is that it is possible to use entire images of arbitrary sizes, without dividing them into patches (Ronneberger et al., 2015). This results in a large receptive field that network uses while classifying each voxel, which is important in segmentation of large structures like the breast. Therefore, we selected this architecture to investigate the use of deep-learning methods for breast tissue segmentation in our work.

2.2. BPE classification

Qualitative and quantitative assessment of BPE on breast DCE-MRI was analyzed by Pujara et al. (2017b). In order to make qualitative evaluation, four radiologists graded BPE at 90 seconds and 180 seconds after contrast injection on a 4-point scale. A phantom-validated segmentation approach was obtained to generate FGT masks and they were co-registered to pre- and post contrast fat suppressed images to get the region of interest (ROI) and to calculate a quantitative BPE measure. Receiver operating characteristic (ROC) analyses and kappa coefficients (k) were used to compare subjective

BPE with quantitative BPE. Based on ROC analyses, the authors concluded that BPE at 90 sesonds was best predicted by the quantitative BPE approach compared to subjective assessment. However, at higher levels of quantitative BPE, agreement between subjective BPE and quantitative BPE significantly decreased for all four radiologists at 90 seconds and for 3 out of 4 radiologists at 180 seconds.

One of the latest research works was published by Borkowski et al. (2020), where authors proposed to train a deep convolutional neural network (dCNN) for standardized and automatic classification of BPE categories. They used consensus of the 2 radiologists as golden standard for classification. The proposed deep learning architecture represents 2 densely connected layers on top of the convolutional part of the VGG16 and the approach relies on the use of a transfer learning method. The authors claim that the neural network is at least as accurate as an experienced radiologist and their predictions are standardized and unaffected by the effect of the intra-reader discrepancy, due to the convolutional part of the VGG16 network, which is able to supply as a valid feature extractor for breast MRI, even though it was not trained on medical images.

Ha et al. (2016) published quantitative FGT measurement tool for breast MRI. They proposed a region-based active contours segmentation algorithm for the whole breast and FGT contours on T1-weighted pre-contrast sagittal images. Quantitative measures of FGT were computed with data derived from breast MRI and correlated significantly with conventional qualitative assessments. Later, the same authors Ha et al. (2018) obtained a fully automated convolutional neural network (CNN) method, where the ground truth was established using previous work. The aim of their work was a quantification of breast MRI fibroglandular tissue (FGT) and background parenchymal enhancement (BPE). In this case, a 3D CNN architecture was modified from the standard 2D U-Net to implement voxel-wise prediction of the whole breast and FGT. They successfully quantified FGT and BPE within an average of 0.42s per MRI case, but authors claimed that their approach could still be improved using larger training data.

Yang et al. (2015) proposed a method for quantitative image analysis using DCE-MRI images by integrating BPE features into the decision making process. At first step of their work, breast region segmentation was performed using computer-aided detection scheme. After they computed 18 kinetic features from which 6 were computed from the segmented breast tumor and 12 were BPE features from the background parenchymal regions. They used Support Vector Machine (SVM) based statistical learning classifiers which were trained and optimized using different combinations of features that were computed either from tumor only or from both tumor and BPE. Each SVM was tested using a leave-one-case-out validation method and assessed using an area under the receiver operating characteristic curve (AUC). They concluded that quantitative BPE features perform useful knowledge to the kinetic features of breast tumours in DCE MRI and their integration to computer-aided diagnosis techniques could improve breast cancer diagnosis based on DCE-MRI examinations.

Another quantitative BPE evaluation technique was developed by Klifa C. (2011), which was tested on 16 healthy volunteers and on high risk patients who already underwent 3 months of tamoxifen therapy. Their results showed that high-risk patients had 37% fewer BPE after treatment and suggested that obtained methods are robust.

In this study, we present an automated deep learning method for breast tissue segmentation and background parenchymal enhancement classification into their categories. All DCE MRI studies were independently assessed for BPE by three expert readers and assigned a category from 1 to 4. The qualitative approach was taken to establish golden standard for the automatic quantitative methods.

3. Material and methods

3.1. Database and data preparation

The database includes DCE-MRI Breast images which were taken from clinical database of the Radboud University Medical Centre (Nijmegen, the Netherlands). It contains 491 studies from 405 patients. The dataset was collected from 3 different MRI machines produced by Siemens with 1.5 and 3 Tesla magnetic field (Magnetom Vision, Magnetom Avanto and Magnetom Trio) and we have different set of image volume sizes of $256 \times 128 \times 112, 384 \times 192 \times 160, 448 \times 448 \times 160, 448 \times 448 \times 176$ and $512 \times 256\times$ 120 with pixel spacing between 0.7 and 1.3mm and voxel size between 1 and 1.5 mm. Each examination has one pre-contast and four post-contrast images.

Three expert radiologists from three different countries manually annotated BPE levels of the dataset. Using the guidelines defined in the ACR BIRADS MRI lexicon classification categories, each reader rated the level of BPE independently into the 4 ordinal categories. For this task, each reader visualised the maximum intensity projection images at time point t0 and t1 only to rate the BPE level.

Unilateral mastectomy, bilateral mastectomy and breast implant cases were excluded from the dataset because those cases are likely to corrupt final results. The pre-contrast and post-contrast time points were used because they are assumed to be representative for the BPE level within the entire volume although it is still not clear which time point yields useful information for breast cancer risk stratification and prediction of response to treatment. All readers annotated most of the

Table 1: Information about Readers

Readers	Country	Experience (year)	Specialization
Reader1 (R1)	Netherlands	8	Breast
Reader2 (R2)	Germany	3	Breast
Reader3 (R3)	Spain	25	Prostate

dataset as mild or minimal with few cases classified as moderate or marked. The majority voting ensembling technique was used as golden standard. The rates of the 3 readers were fused together to establish groundtruth labels for the automated approach. Information about readers is shown in Table 1. Cases without any agreement with at least 2 readers were also filtered out. Finally, after applying those restrictions, we ended up with 310 studies out of 491. Figure 2 illustrates excluded cases from database.

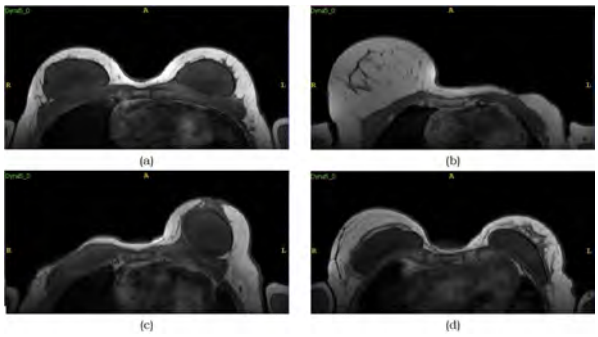


Figure 2: Examples of the excluded cases from dataset: (a) and (d) - double implant cases; (b) - mastectomy case; (c) - implant + mastectomy case.

3.2. *Pre-processing and breast mask generation*

Breast mask, which we assume as the region of interest (ROI), together with pre-contrast (t0) image were given as inputs to the proposed architecture. This ROI is used in the approach to restrict the patch extraction pipeline to breast area only.

A simple binarization method was used to obtain a mask together with morphological opening and closing operations. Another important role of the breast mask is removal of the coil artifact, which in some cases can corrupt the result. So, as background intensities in MRI can be nonzero, after the performed operations background artifacts might still take place. To remove them, we used connected component analysis to leave only the largest connected component, which at the end will provide us with required breast mask.

As our ground-truth (GT) has 8 labels of chest body, we replaced unnecessary labels (labels>3, such as heart, lung, liver, bone and background), which represent body organs not related to breast, as 0. Figure 3 shows an example of GT before and after excluding unnecessary labels.

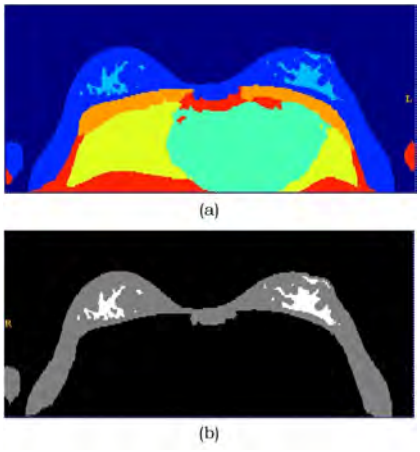


Figure 3: Ground-truth images before and after excluding unnecessary labels. (a) shows the initial ground-truth image with all labels in the chest area; (b) is the processed ground-truth image with only needed labels of fat and FGT.

3.3. *Patch sampling*

Patch sampling is an important aspect to take into account, because the way we sample patches may affect the final segmentation results. Since most medical images have high resolution, patch-based approach is commonly employed for segmentation, where images are divided into small patches with a specified size as the input of the neural network (Zhang et al., 2019). Patch sampling method can fully utilize the local information of the focused area. In our work, patch sampling was done using balanced sampling technique as Valvano et al. (2019) was used on their challenge. Balanced patch sampling means that we extract equal number of patches from all represented classes in the dataset. Balanced sampling can be helpful in problems with class imbalance, like in our case. We decided to use this approach in order to avoid unbalanced patching for each labels, as the number of pixels of FGT on the image is less than fat tissue pixels. We experimented with different patch sizes, the best results obtained with patch size of $(32 \times 32 \times 32)$.

3.4. *Proposed method for breast tissue segmentation*

To segment breast body (fat) and FGT (dense tissue), we used patch based 3D U-Net model, which was proposed by Çiçek et al. (2016). U-Net is a type of fully convolutional network (FCN) with a contraction path (encoder) and expansion path (decoder). The contraction path consists of consecutive convolutional blocks followed by a maxpool downsampling to encode the input image into feature representations at multiple different levels (Ronneberger et al., 2015). It is used to extract features while limiting the size of feature maps. The expansion path (decoder) performs upsampling and has convolutional blocks to recover the size of the segmentation map. Additionally, skip connections are used to share localization information from the contraction

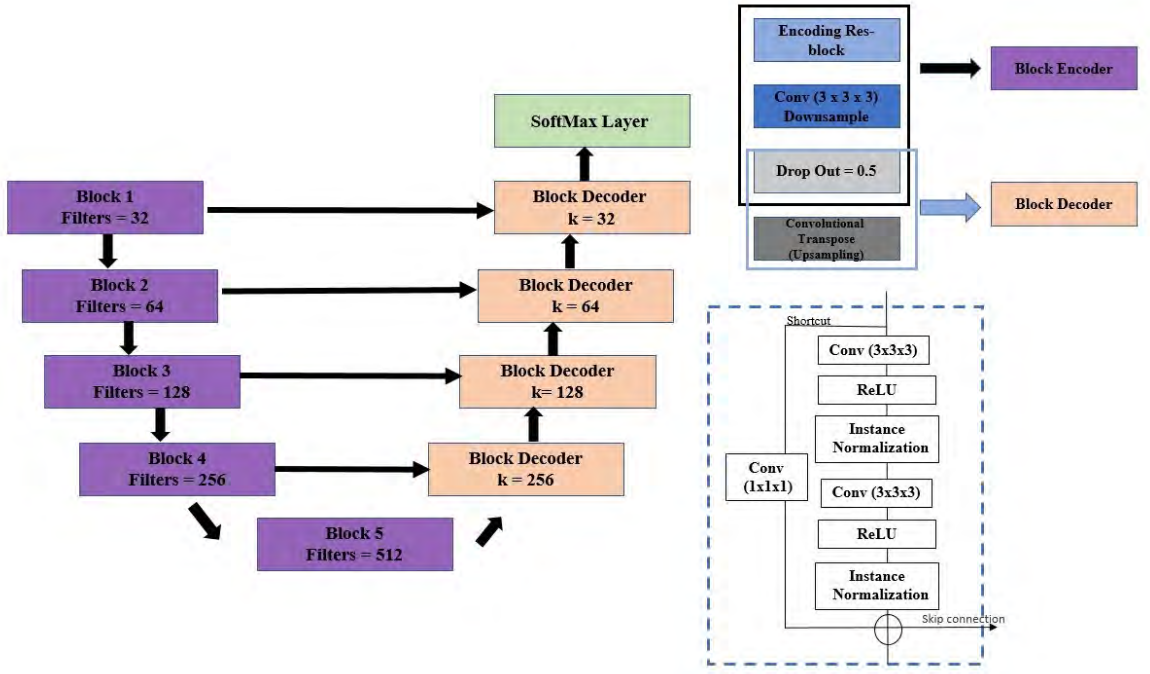


Figure 4: The architecture used in the proposed approach. The network is inspired by 3D U-Net with ResNet backbone.

level to the corresponding expansion level. These are parallel connections allowing signals to propagate directly from one block of the network to another without adding any computational complexity. At the end, a convolutional layer maps the features vector to the required number of target classes in the final segmentation output.

However, we made changes to original architecture. The basic U-Net blocks were replaced with ResNet blocks. ResNet blocks have skip connections, they connect some layers with deeper ones, skipping several layers in between. That helps to avoid gradient vanishing and hence avoiding overfitting, which is a common issue in training deeper networks. Each level of encoder (contraction path) includes: ReLU activation; maxpooling ($2\times2\times2$) with stride 2 for downsampling; droupout and instance normalization. Symmetrically to encoder, decoder (expansion path) consists of: up-convolution ($2 \times 2 \times 2$) with stride 2; concatenation with feature-map from the same level of the encoder; basic ResNet blocks and ReLu activation. At the end, we used an output convolution layer ($1\times1\times1$) with two output channels followed by a softmax layer which returns probabilities for each class. The architecture for the proposed 3D ResUNet model is illustrated in Figure 4.

Instance normalization plays an important role in training the Neural Network effectively and enables faster convergence. Instance Normalization technique was utilized in the encoder part of our architectures. Instance Normalization is similar to Batch Normalization with the difference being that instead of normalization across the entire feature in the mini batch, normalization is applied across each channel.

In order to address the problem of overfitting and better generalization, dropout was added to the model. Dropout helps in the synthesis of different model architectures from the same architecture by randomly dropping certain proportion of nodes from the model. In our case we used a dropout equal to 0.5. Together with the Normalization techniques, the tendency of the net to overfit to the training set was less severe.

Increasing the depth more can lead to overfitting problems while no significant increase in the accuracy. The maximum depth was dictated by the chosen patch size. Due to constrained resources at disposal, the maximum patch size that could be used was 32. This meant that maximum depth that could be utilized was 5. Depth of 5 was fixed for the architecture (Jiang et al., 2017).

3.4.1. Training and testing sets

For the training stage, we composed the training and validation sets from the provided images to train the weights of the network. Following the balanced patch sampling technique, 1000 patches with size ($32 \times 32 \times 32$) were extracted from each image to train our model. The loss function is a critical part of the architecture, since it drives the backpropagation in order to achieve a better performance. In our proposed method, cross entropy loss was used at the final experiments. Adam as an optimizer was used and gave the best performance. To further avoid overfitting, and to take advantage of a net we implemented early stopping in a way that it tracked

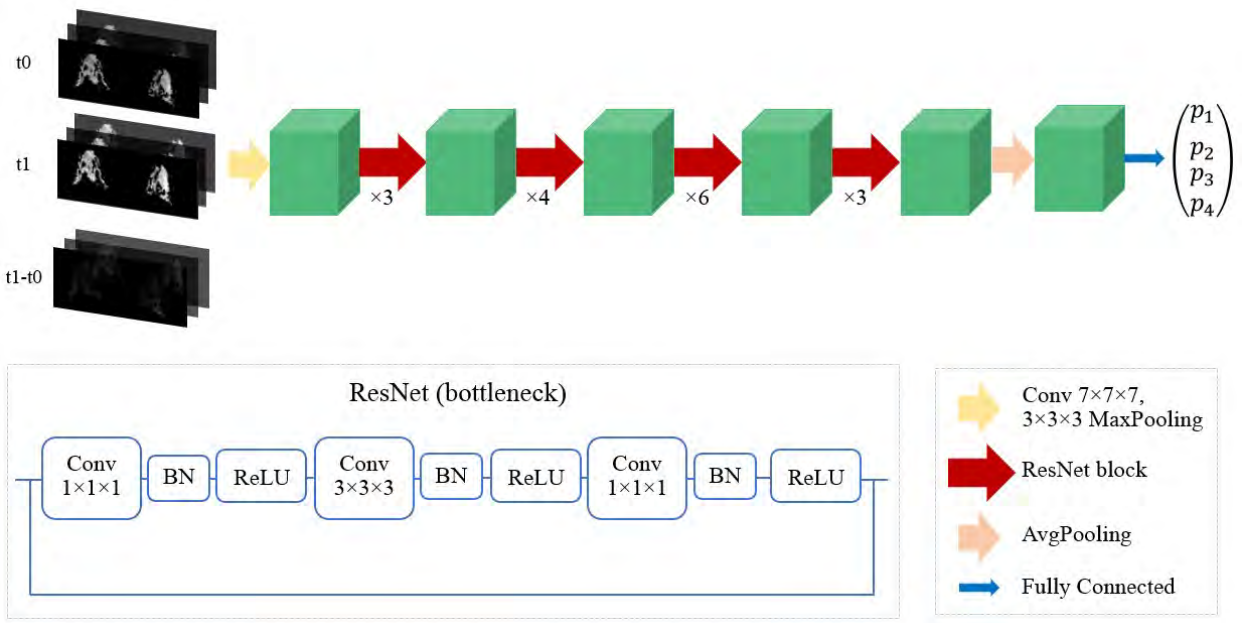


Figure 5: Architecture of proposed 3D Resnet-50. The convolutional layers were pre-trained on ImageNet dataset. The convolutional feature extractors and weights of the hidden layers were frozen and transferred directly to BPE classification.

the validation loss, and allowed for a certain number of epochs (that we called "patience") to improve.

At the testing stage, patches were extracted from each image uniformly with some degree of overlap to improve segmentation results. We extracted patches only from area of interest (breast body), therefore masks for all images were obtained as we explained before. After defining the patch size which is dictated by the memory and GPU resources available at disposal, the overlap between the consecutive samples is defined. The greater the overlap (which means smaller sampling step), larger patches are available for training. This in turn means larger data and more training time. The sampling step of 16 was considered to be appropriate reflecting better performance following by Jiang et al. (2017). In order to test the generalization of network and evaluate all the images in the dataset, 5 fold cross validation was performed. For each of 5 folds different images were taken to the test set. In each fold, the dataset was divided into training (80% cases) and testing (20%) sets automatically, and patches were extracted following the proposed pipeline. Every patch was passed through the network, resulting in a predicted probability for each voxel. The output binary segmentation was produced by assigning the class label according to the maximum probability for each voxel. Finally, the pre-constrast volume (t0) was selected as the input of the network.

3.5. Proposed method for BPE classification

Convolutional neural networks have been widely used for image classification tasks with excellent performance (Krizhevsky et al., 2012), (He et al., 2016). ResNet has been shown to provide a good classification accuracy in many medical applications. Often and due to the limited number of cases available for training, ResNet weights are initialised using transfer learning from other domains. Therefore, we used transfer learning technique with 3D Resnet-50 architecture to classify BPE according to their classes.

3.5.1. Proposed model for BPE classification

The architecture of the ResNet-50 is similar to plain network with 50 layers, but ResNet has a shortcut connection, which turns the network into its counterpart residual version. Each residual block follows the bottleneck design. Bottleneck design is used to increase the network depth while keeping the parameters size as low as possible, it means that three layers are (1×1×1), (3×3×3) and (1×1×1) convolutions, where the (1×1×1) layers are responsible for reducing and then increasing (restoring) dimensions, leaving the (3×3×3) layer a bottleneck with smaller input/output dimensions (He et al., 2016). ResNet's layer is composed of the same blocks stacked one after the other. We perform downsampling directly by convolutional layers that have a stride of 2. Encoder is composed of multiple layers at increasing features size and decoder is a fully connected layer that maps the features learned by the network to their respective classes. In our method, the convolutional layers were pre-trained on ImageNet dataset.

We modified this model by replacing all 2D convolution kernels with the 3D versions and using fine-tuning technique, so that all the hidden layers were frozen and the last fully connected layer was modified by reduc-

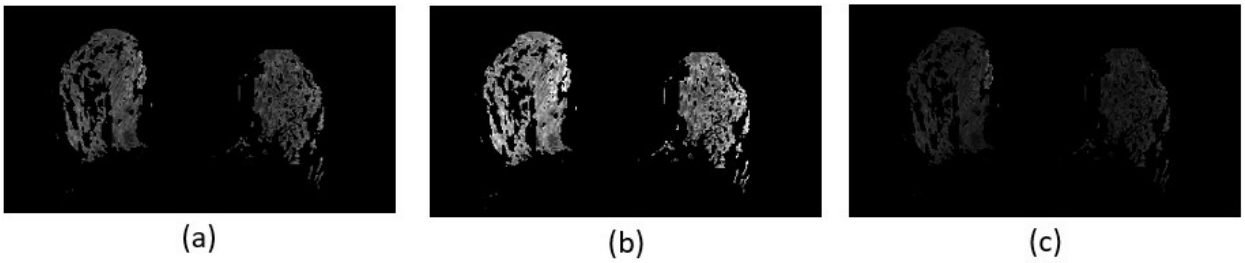


Figure 6: Input images for classification. (a) - pre-contrast (to), (b) - first post-contrast (t1), subtracted t1-t0.

ing the number of features using ReLU weight initialization followed by a dropout layer with probability of 0.5. Finally, last fully connected layer was built up of features classified into 4 classes using the Softmax classifier. Figure 5 illustrates proposed classification architecture.

3.5.2. Training and testing sets for classification

Output segmentation results were used as inputs for classification part. We applied segmented FGT masks on pre-contrast (t0), first post-contrast (t1) and subtracted (t1 - t0) images, which can be observed in Figure 6. These 3 images were used as an input to the BPE classification network. Before training, all these input images were pre-processed because images in dataset are from 3 different machines. An isotropic voxel spacing of $(1 \times 1 \times 1)$ was generated for all input images, after images were rescaled to $(3 \times 240 \times 345)$, based on bilinear interpolation operation.

As in segmentation's training stage, for classification in training stage we optimize network parameters using the cross-entropy loss with the stochastic gradient descent (SGD) method, where the learning rate was set to 0.1, momentum was set to 0.9 and weight decay was set to 0.001. Maximum number of epochs for training set was chosen equal to 100. Like in the segmentation step, 5-fold cross validation was performed until all the entire dataset was once used as a test set. For each fold the dataset was randomly divided into 80% training and 20% testing. Early stopping technique was applied also, which helped to reduce training time and number of training epochs. Evaluations were performed by comparing the indices with the highest probability against the true classes using the softmax classifier.

4. Results and Discussion

This study was performed on Ubuntu with 256 RAM and NVidia GeRore RTX 2080 GPU with 11 GB memory. All experiments were obtained using Python 3.7 based on PyTorch 1.4.0 deep learning library.

In this section we present our experiments and obtained results along with the discussion, for both breast tissue segmentation and BPE classification.

4.1. Breast tissues segmentation

In this subsection we evaluate different aspects of the U-Net segmentation approach and discussion about obtained results. More specifically we test different optimizers, patch sizes and loss functions.

The Dice Similarity Coefficient (DSC) was used for the evaluation of developed segmentation approach and computed using equation(1) :

$$DSC = \frac{2TP}{2TP + FP + FN} \tag{1}$$

4.1.1. Patch size

In the work proposed by (Farabet et al., 2013) they showed that using bigger patches in CNNs may improve segmentation results, as the network can capture more contextual information, which in our case can be also helpful to segment breast tissues, especially FGT. Therefore, an experiment was performed to use different patch sizes. Considering the architecture used and the computational load, two patch sizes were tested: $(16 \times 16 \times 16)$ and $(32 \times 32 \times 32)$. Qualitative comparison results obtained for both patch sizes are shown in the Figure 7.

Indeed, our experiments showed, that increasing of the patch size from $(16 \times 16 \times 16)$ to $(32 \times 32 \times 32)$ helped to increase segmentation results. Especially, the larger patch size increased overall DSC for FGT - from 0.695 to 0.75, while segmentation for the fat tissue was nearly the same. Overall results are shown in Table 2. As expected, network training time was longer with bigger patch size: with patch size $(16 \times 16 \times 16)$ training time per epoch was 7min, while with patch size $(32 \times 32 \times 32)$ - 28.5min. As one can notice in Figure 7 (1st column), result obtained from the smaller patch did not segment part of the FGT tissue. Segmentation obtained in the 2nd case (on the 2nd column), partly performed wrong result, a part of the FGT was segmented as fat tissue, while in the 3rd case (3rd column) part of the FGT was segmented as background. Meanwhile, we can see that patches with larger size gave better performance. As mentioned before, enlargement of patch size should have helped the network to better segment FGT.

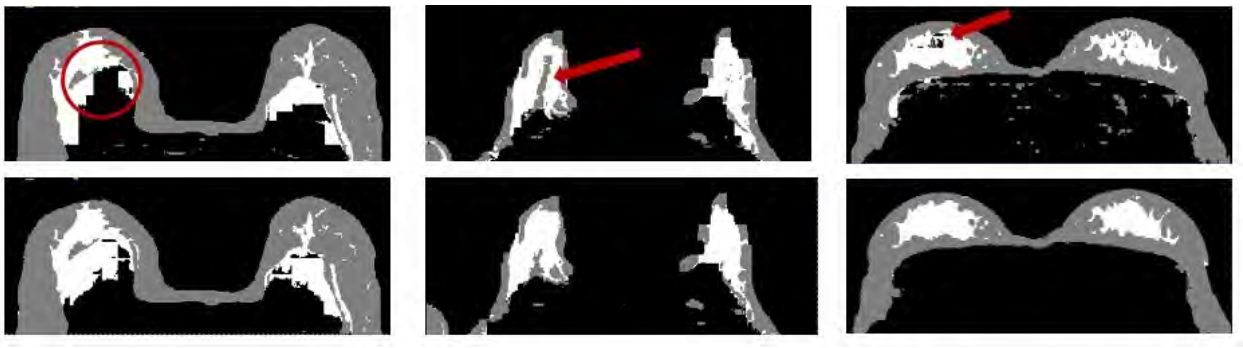


Figure 7: Examples of the segmentation results. On the top row images obtained with patch size ($16 \times 16 \times 16$). On the bottom row examples with patch sizes ($32 \times 32 \times 32$).

4.1.2. Optimizer

Optimizers Adam and Adadelta were studied in the experiments as they are common optimizers for breast tissues segmentation task following the proposed methods by Zhang et al. (2018) and Piantadosi et al. (2018).

Experiments with optimizers showed that Adam optimizer performed better results than Adadelta, even if the learning rate of Adadelta optimizer is adaptive to pixel coordinates. Adam Optimizer in our configuration took more time to converge as compared to Adadelta. Adam optimizer with learning rate 0.001 gave us best performance. Overall average dice score was increased from 0.695 to 0.7 for FGT, while for the fat tissue optimizer did not improve result so far, it changed from 0.81 to 0.82. Also, we could assume that model trained with Adam optimizer did well with background artifacts, while results with Adadelta optimizer, represent noisy background. Segmentation examples of experiments are illustrated in Figure 8. In Figure 8 (a) on the top row, we see artifacts related with non-breast body, while segmentation using Adam optimizer (on the bottom row) represents better performance. On the example displayed on Figure 8 (b) top row, background artifacts, labeled parts on the image (look at the red range and arrow) were segmented as FGT, which is not correct. Segmentation obtained with model trained with Adam optimizer (on the bottom row) performed generally better.

4.1.3. Loss function.

Two different loss functions were used in our experiments: Cross Entropy (CE) Loss and Dice Loss. Cross Entropy loss examines the pixels individually and comparison is done with the target labels. Pixel wise log loss is calculated and summed over all the classes. This is computed over all the pixels and averaged. The following Equation 2 shows the expression for the log loss:

$$CELoss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i)\cdot log(1-p(y_i)) \tag{2}$$

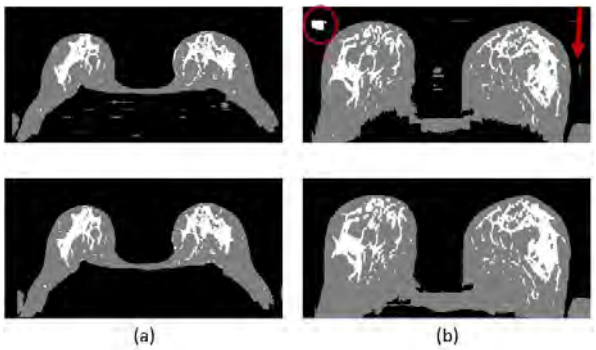


Figure 8: Examples of the obtained segmentation results. On the top row images obtained using model trained with Adadelta optimizer. On the bottom row same cases segmented with model trained with Adam optimizer.

Where y is the label and $p(y)$ is the predicted probability for all the N points (voxels). Cross Entropy loss leads to equal learning to each voxel in the volume. But in case of unbalanced classes it could be a problem to train network. In this case, balanced patch sampling might be useful like we explained before. Also, following the proposed method by Ronneberger et al. (2015), loss weighting scheme could be used for each pixel such that a larger weight is assigned to pixels at the contour of segmented objects.

Dice Loss is essentially the overlap between the predicted segmentation and the ground-truth segmentation. This overlap is then scaled by the sum of the total number of pixels in the predicted and the actual segmentation. This can be represented by the following equation 3:

$$DiceLoss = -\frac{Predicted \cap Actual}{\sum_{pixels} Prediction + \sum_{pixels} Actual} \tag{3}$$

Experiments with Dice loss function did not gave us improvement in results. Dice score was little less than with experiments with Cross Entropy Loss. Results are shown in Table 2.

Table 2: DSC for all experiments.

	Adam Patch ($32 \times 32 \times 32$) CrossEntropy Loss	Adam Patch ($32 \times 32 \times 32$) Dice Loss	Adadelta Patch ($16 \times 16 \times 16$) CrossEntropy Loss	Adadelta Patch ($32 \times 32 \times 32$) CrossEntropy Loss
Fat tissue	**0.76**	0.69	0.67	0.71
FGT	**0.82**	0.792	0.79	0.81

Final results were obtained using patch size (32×32× 32), Adam optimizer and Cross Entropy Loss, which represent the best DSC. Average DSC for the fat tissue was equal to **0.82**, for the FGT - **0.76**. The median results among all the patients for each 5-fold are presented in Table 3. Qualitative results of final segmentation for proposed approach are shown in the Figure 9, while results from all experiments are shown in Table 2.

Table 3: Average DSC of final proposed performance for each of 5 folds and each segmented class.

	Fold 0	Fold 1	Fold2	Fold3	Fold4	Avg. DSC
Fat tissue	0.744	0.788	0.729	0.775	0.785	**0.764**
FGT	0.819	0.792	0.8208	0.834	0.826	**0.82**

During the experiments we found out that in each fold we have cases with very low DSC, such as **0.102, 0.286, 0.250** for fat tissue and **0.0019, 0.0015** for FGT segmentation, while remaining cases have scores from 0.7 to 0.95 for FGT and from 0.8 to 0.95 for fat tissue. Also, we have case with dice 0, obtained during experiment with smaller patch size. In Figure 10, we show example cases for which we obtained a low Dice score. If we visually analyse these cases, there are areas inside the breast assigned to class 0 (non breast tissue class) in GT. However, by observing the image we assume that those intensities correspond to FGT and/or fat classes and not background class as in the GT. Therefore, we assume that this case were miss-classified in the ground-truth. These cases affect to our overall DSC.

In the end of this section, we compared our obtained final segmentation results with already published work in Table 4 , which proposed methods we explained in the section before.

4.2. BPE classification

Next part of our work was related to BPE level classification the 4 ordinal categories defined by ACR. First, we calculated intra-observers agreement for further comparison with our obtained results, after we discuss about results obtained from BPE classification.

4.3. Inter-Observer Agreement

As we explained in Database section, 3 radiologists assessed each case and we used majority voting ensembling technique to establish ground-truth for automated classification. Figure 11 represents individual annotations of each professional readers and ensembling rates (majority voting ensembling technique results).

We calculated inter-observer agreement (k) - kappa value between 3 professional readers (R1, R2, R3) and agreement all of them with the ensembled rates. The agreement was measured using the quadratic weighted Cohen's kappa coefficient (k) given by Equation 4 :

$$k = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}m_{ij}} \tag{4}$$

where n is the number of classes and w_{ij}, x_{ij}, m_{ij} are elements in the weight, observed, and expected matrices respectively.

$(k) < 0.0$ was interpreted as poor agreement, $0.0 \leq (k) \leq 0.20$ as slight agreement, $0.20(k) \leq 0.40$ as fair agreement, $0.40(k) \leq 0.60$ as moderate agreement and $(k)\,0.60$ as substantial agreement (Landis and Koch, 1977). The Table 5 shows agreement (k value) between the readers and readers with ensembled rates (E).

4.3.1. Classification

Metrics to evaluate the classification results were overall accuracy ($Acc\%$) and accuracy per each BPE class ($Acc(BPEn_n)\%$). Accuracy was calculated using Equations 5 and 6:

$$Acc = \frac{TP}{TP + FP + TN + FN} \tag{5}$$

$$Acc(BPE_n) = \frac{TP(BPE_n)}{TP(BPE_n) + FP(BPE_n)} \tag{6}$$

Experiments for automatic BPE classification was obtained using two different classifier: softmax and sigmoid. Best results were obtained using softmax classifier with overall accuracy 61.3%, while with sigmoid classifier we obtained 52.00 %. Experiment shows, that accuracy for BPE4 was nearly same with both classifiers, while results for the BPE2 increased from 40 % to 54 % and for BPE3 from 53 % to 64 % . Results for other classes were increased also. The results obtained for the accuracy per class and overall accuracy are shown in Table 6.

The hyperparameters to train the network such as the batch size, type of the loss function, optimizer, learning rate and number of epochs were set to obtain optimal results as in the publications proposed by (He et al., 2016), (Chen et al., 2019) and (Herent et al., 2019).

Table 4: Comparison of results of our approach and published works in Section 2.

Reference	proposed approach	FGT DSC	Fat tissue DSC	overall DSC
Piantadosi et al. (2018)	U-Net	-	-	0.959
Zhang et al. (2019)	U-Net	0.95	0.91	-
Dalmş et al. (2016)	U-Net	0.85	0.933	-
Gubern-Mérida et al. (2015)	Atlas-based	0.80	0.94	-
Sehrawat et al. (2018)	Landmark detection	0.915	0.977	-
Our method	U-Net	0.76	0.82	-

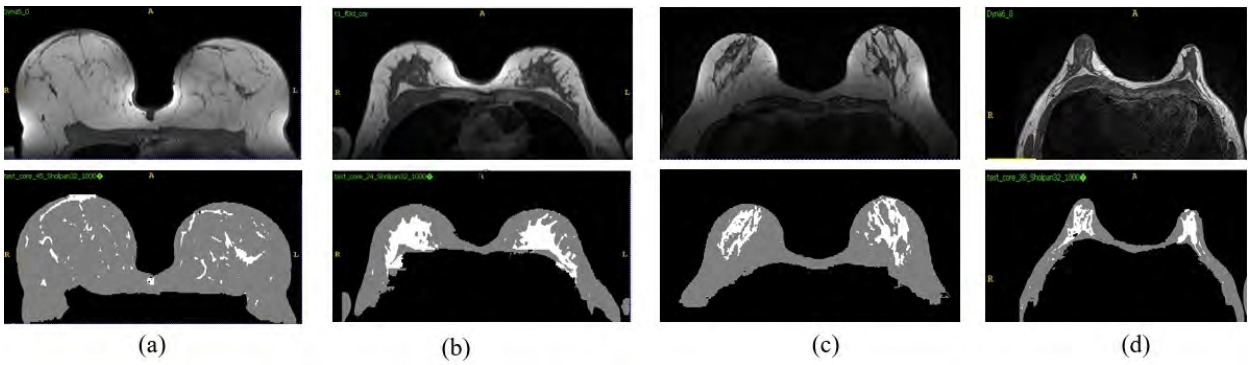


Figure 9: Segmentation results for different BPE categories. On the top input pre-contrast (t0) images, output segmented breast tissues on the bottom. (a) - BPE1, (b) - BPE2, (c) - BPE3, (d) - BPE4.

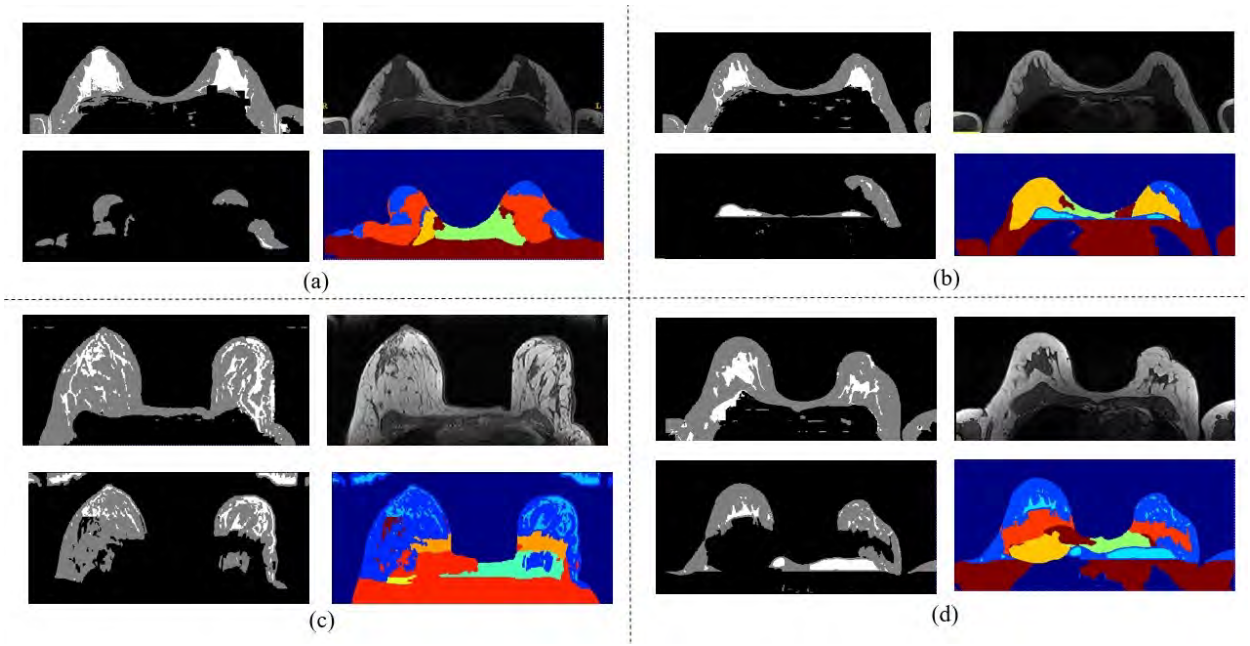


Figure 10: 4 examples (a, b, c, d) of obtained low DSC. Top left - output segmentation; top right - input pre-contrast image (to). Bottom left - GT with only fat and FGT classes; bottom right - GT with all given classes.

Nowadays, manual annotation currently assessed by radiologists suffers from large intra- and inter-observer variability, as we show in Table 5, inter-observer agreement varies from slight to moderate agreement. Agreement between R1 and R2 was moderate with k= 0.4. However, agreement between R1 and R3 was slight with k = 0.13. Since, the agreement between R2 and R3 with k = 0.18. This difference between the readers was mostly caused by Reader 3, we assume that is because of his field of speciality was not oriented in the breast or a slightly different criteria was applied. This disagreements highlight the need of quantitative tools of

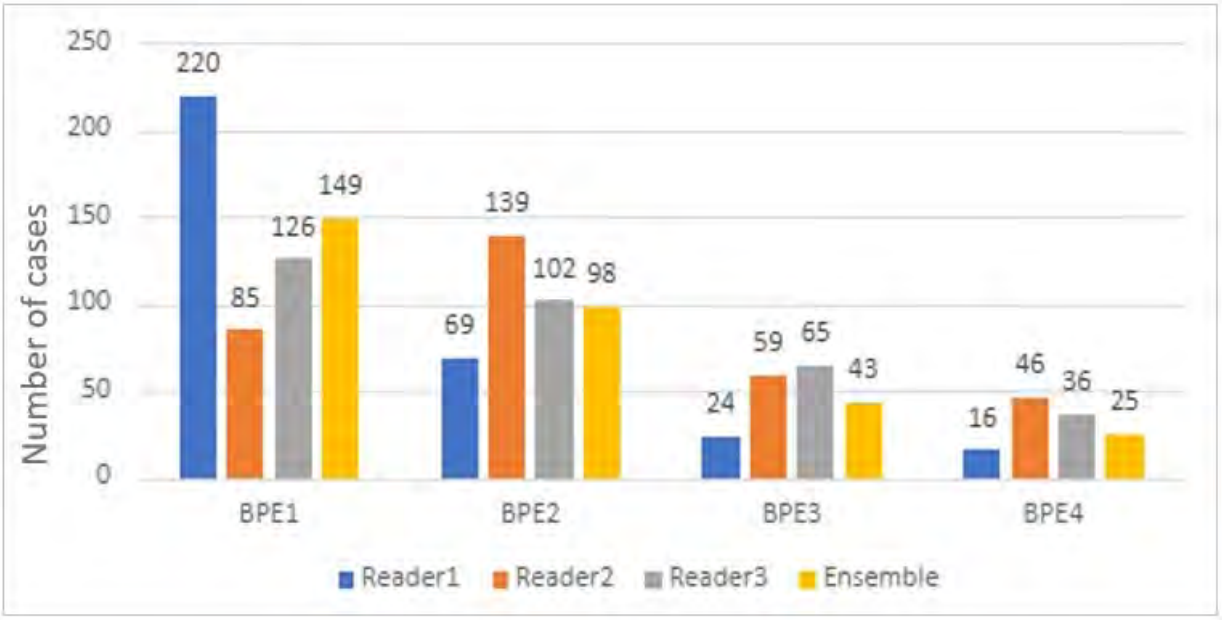


Figure 11: Histogram of BPE annotations for each of the readers and majority voting ensembling technique results (Emsemble).

Table 5: Inter-observers agreement

Readers	k value	Agreement
R1 and R2	0.41	Moderate
R1 and R3	0.18	Slight
R2 and R3	0.13	Slight
R1 and E	0.65	Substantial
R2 and E	0.66	Substantial
R2 and E	0.46	Moderate

assessment of BPE as the one developed in this project with the aim to potentially reduce the tedious, subjective nature of BPE classification process. Additionally, such tools could help in training beginner radiologists in BPE classification or even the reduction of inter- and intra-reader variability. In order to solve these problem many researchers like Klifa C. (2011), Ha et al. (2018), Yang et al. (2015), Pujara et al. (2017a), suggested approaches for a quantitative analysis of the BPE. But their methods still require intervention of the professional readers to delineate the ROI (FGT) manually, which may introduce potential subjective bias. Methods proposed by Borkowski et al. (2020) and Ha et al. (2016) are a fully automated approaches based on convolutional neural network (CNN) for FGT quantification and BPE classification and shows very high accuracy.

To obtain ground-truth labels for this task, we performed majority voting between the individual rates of three radiologists, as this information was provided within the dataset. Considering this technique, we assign final label based on the agreement between at least two raters, therefore, if there were cases without agreement between at least two raters, we had to exclude them from the final dataset used for this task. We also checked if the final label we obtained agrees with the provided individual labels and we can conclude that majority voting can be used for our purposes, as we observed a moderate agreement for each case. In the training pipeline, we fed three input sequences to the network: t0, t1 and (t1-t0). In the segmentation step we obtained masks only for t0 images, however, we used them for all time points. This could be done because the provided dataset did not report about patient motion between different scans, so we assumed the absence of patient motion. As we were sticking to a 3D approach in the segmentation step, we continued with it for the classification task, therefore the input images were 3D. 5-fold crossvalidation experiment was performed, with 80% of dataset used for training and remaining 20% for testing in each fold. We performed such step to guarantee that our neural network learns effectively, as in some cases it could happen that the subtraction images (t1-t0) could be insignificant, because they have almost zero intensities.

In the end, we compared our results to already published works on BPE classification as shown in Table 7. Borkowski et al. (2020) and Ha et al. (2016) received the best overall accuracy. The high performance observed in their approach may be related to to the high baseline used to establish ground truth as well as the robust segmentation tool used in segmenting FGT prior to the classification tasks.

Table 6: BPE classification accuracy (%) per class and overall accuracy (%).

Classifier	BPE1(%)	BPE2(%)	BPE3(%)	BPE4(%)	overall Accuracy %
Softmax	61.4	54.8	64.02	77.00	**61.3**
Sigmoid	55.22	40.14	53.00	76.00	**52.00**

Table 7: Comparison of our approach and published works in section 2.

Reference	*Acc*BPE1(%)	*Acc*BPE2(%)	*Acc*BPE3(%)3	*Acc*BPE4(%)	overall *Acc*(%) -
Pujara et al. (2017a)	20.20	25.20	50	50	-
Klifa C. (2011)	23.00	22.0	17.00	23.00	-
Ha et al. (2018)	4.61	8.74	18.10	37.40	-
Ha et al. (2016)	-	-	-	-	**82.90**
Borkowski et al. (2020)	**84**	**80**	48	**88**	
Our method (Softmax)	61.4	54.8	**64.02**	77.00	**61.3**
Our method (Sigmoid)	55.22	40.14	53.00	76.00	**52.00**

5. Conclusions

We presented a deep-learning approach based on the 3D U-Net architecture for breast tissue and FGT segmentation on MRI. This method showed good results that could be used for further BPE classification. In the classification part of our work, we obtained automatic method for BPE prediction in 4 categories, based on 3D ResNet architecture. Our proposed methods were tested on the clinical dataset with 310 DCE MRI studies.

The aim of these kind of methods is to generate a tool, which can help to reduce inter- and intra-reader variability observed during manual BPE classification and it could also be helpful for the beginner radiologists or medical students as they could be trained on the annotations of a more experienced radiologists.

Since segmentation of the FGT is a major aspect and it affects to BPE classification results, manual segmentation of dataset (ground truth) could be improved in future work, because from our obtained results we could observe cases, in which some ground truth voxels might be inaccurately segmented.

Because the accuracy of the proposed automatic BPE classification is directly dependent on the manual annotation by professional readers, more radiologists could evaluate dataset regarding BPE rate.

In addition, a larger training dataset is estimated to improve the accuracy of the model.

6. Acknowledgments

I would like to express my very great appreciation to my supervisor Robert Marti. Advice given by him has been a great help to perform master thesis project. I would like to offer my special thanks to all MAIA academic and administrative staff for the opportunities provided to gain new knowledge, skills and irreplaceable experience. I am particularly grateful for the assistance given by Valeria Abramova.

References

Arslan G, Celik, L.C.R.C.L..A.M.M., 2017. Background parenchymal enhancement: is it just an innocent effect of estrogen on the breast? Diagn Interv Radiol , 414–419.doi:10.5152/dir.2017.17048.

Borkowski, K., Rossi, C., Ciritsis, A., Marcon, M., Hejduk, P., Stieb, S., Boss, A., Berger, N., 2020. Fully automatic classification of breast mri background parenchymal enhancement using a transfer learning approach. Medicine 99, e21243. doi:10.1097/MD.0000000000021243.

Chen, S., Ma, K., Zheng, Y., 2019. Med3d: Transfer learning for 3d medical image analysis.

Dalmş, M., Litjens, G., Holland, K., Setio, A., Mann, R., Karssemeijer, N., Gubern-Mérida, A., 2016. Using deep learning to segment breast and fibroglanduar tissue in mri volumes. Medical Physics 44. doi:10.1002/mp.12079.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1915–1929.

Gubern-Mérida, A., Kallenberg, M., Mann, R., Martí, R., Karssemeijer, N., 2015. Breast segmentation and density estimation in breast mri: A fully automatic framework. IEEE journal of biomedical and health informatics 19, 349–57. doi:10.1109/JBHI.2014.2311163.

Ha, R., Chang, P., Mema, E., Mutasa, S., Karcich, J., Wynn, R., Liu, M., Jambawalikar, S., 2018. Fully automated convolutional neural network method for quantification of breast mri fibroglandular tissue and background parenchymal enhancement. Journal of Digital Imaging 32. doi:10.1007/s10278-018-0114-7.

Ha, R., Mema, E., Guo, X., Mango, V., Desperito, E., Ha, J., Wynn, R., Zhao, B., 2016. Quantitative 3d breast magnetic resonance imaging fibroglandular tissue analysis and correlation with qualitative assessments: A feasibility study. Quantitative Imaging in Medicine and Surgery 6, 144–150. doi:10.21037/qims.2016.03.03.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, pp. 770–778. doi:10.1109/CVPR.2016.90.

Herent, P., Schmauch, B., Jehanno, P., Dehaene, O., Saillard, C., Arfi-Rouche, J., Jégou, S., 2019. Detection and characterization of mri breast lesions using deep learning. Diagnostic and Interventional Imaging 100. doi:10.1016/j.diii.2019.02.008.

Hesamian, M.H., Jia, W., He, X., Kennedy, P., 2019. Deep learning techniques for medical image segmentation: Achievements and challenges. Journal of Digital Imaging 32. doi:10.1007/s10278-019-00227-x.

Hubbard RA, Kerlikowske K, F.C.Y.B.Z.W.M.D., 2014. Cumulative probability of false-positive recall or biopsy recommendation after 10 years of screening mammography: a cohort study. pub-

lished correction appears in Ann Intern Med doi:10.7326/0003-4819-155-8-201110180-00004.

Jiang, L., Hu, X., Xiao, Q., Gu, Y., Li, Q., 2017. Fully automated segmentation of whole breast using dynamic programming in dynamic contrast enhanced mr images. Medical physics 44. doi:10.1002/mp.12254.

Klifa C., PhD S. Suzuki BS S. Aliu PhD L. Singer PhD L. Wilmes PhD D. Newitt PhD B. Joe MD, P.N.H.P., 2011. Quantification of background enhancement in breast magnetic resonance imaging. journal of Magnetic Resonance Imaging 33, 1229–1234. doi:10.1002/jmri.22545.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems 25. doi:10.1145/3065386.

Kuhl, C., Mielcareck, P., Klaschik, S., Leutner, C., Wardelmann, E., Gieseke, J., Schild, H., 1999. Dynamic breast mr imaging: Are signal intensity time course data useful for differential diagnosis of enhancing lesions?1. Radiology 211, 101–10. doi:10.1148/radiology.211.1.r99ap38101.

Landis, J., Koch, G., 1977. The measurement of observer agreement for categorical data. Biometrics 33, 159–74. doi:10.2307/2529310.

Lo, S.C., Lou, S.L., Lin, J.S., Freedman, M., Chien, M., Mun, S., 1995. Artificial convolution neural network techniques and applications for lung nodule detection. Medical Imaging, IEEE Transactions on 14, 711 – 718. doi:10.1109/42.476112.

Piantadosi, G., Sansone, M., Sansone, C., 2018. Breast segmentation in mri via u-net deep convolutional neural networks, in: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3917–3922. doi:10.1109/ICPR.2018.8545327.

Pike, M., Pearce, C., 2013. Mammographic density, mri background parenchymal enhancement and breast cancer risk. Annals of Oncology 24, viii37–viii41. doi:10.1093/annonc/mdt310.

Pujara, A., Mikheev, A., Rusinek, H., Gao, Y., Chhor, C., Pysarenko, K., Rallapalli, H., Walczyk, J., Moccaldi, M., Babb, J., Melsaether, A., 2017a. Comparison between qualitative and quantitative assessment of background parenchymal enhancement on breast mri: Qualitative versus quantitative bpe. Journal of Magnetic Resonance Imaging 47. doi:10.1002/jmri.25895.

Pujara, A., Mikheev, A., Rusinek, H., Gao, Y., Chhor, C., Pysarenko, K., Rallapalli, H., Walczyk, J., Moccaldi, M., Babb, J., Melsaether, A., 2017b. Comparison between qualitative and quantitative assessment of background parenchymal enhancement on breast mri: Qualitative versus quantitative bpe. Journal of Magnetic Resonance Imaging 47. doi:10.1002/jmri.25895.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

Satoko, N., Ishigaki, S., Satake, H., Kawamura, A., Kawai, H., Naganawa, S., 2012. Background parenchymal enhancement in preoperative breast mri.

Sehrawat, S., Chatterjee, S., Singhal, M., Gupta, R., Singh, A., 2018. Automatic outer and inner breast tissue segmentation using multiparametric mri images of breast tumor patients. PLOS ONE 13. doi:10.1371/journal.pone.0190348.

Turnbull, L.W., 2009. Dynamic contrast-enhanced mri in the diagnosis and management of breast cancer. NMR in Biomedicine 22, 28–39. doi:10.1002/nbm.1273.

Valvano, G., Santini, G., Martini, N., Iacconi, C., Chiappino, D., Della Latta, D., 2019. Convolutional neural networks for the segmentation of microcalcification in mammography imaging. Journal of Healthcare Engineering 2019, 1–9. doi:10.1155/2019/9360941.

WHO, 2007. Cancer control : knowledge into action : Who guide for effective programmes / world health organization. SERBIULA (sistema Librum 2.0) .

WHO, 2020. Who position paper on mammography screening. 1.mammography. 2.early detection of cancer. 3.breast neoplasms – prevention and control. i.world health organization. isbn 978 92 4 150793 6 , 82.

Wu, S., Berg, W., Zuley, M., Kurland, B., Jankowitz, R., Nishikawa, R., Gur, D., Sumkin, J., 2016. Breast mri contrast enhancement kinetics of normal parenchyma correlate with presence of breast cancer. Breast Cancer Research 18. doi:10.1186/s13058-016-0734-0.

Yang, Q., Li, L., Zhang, J., Shao, G., Zheng, B., 2015. A new quantitative image analysis method for improving breast cancer diagnosis using dce-mri examinations. Medical physics 42, 103. doi:10.1118/1.4903280.

Zhang, J., Saha, A., Soher, B., Mazurowski, M., 2018. Automatic deep learning-based normalization of breast dynamic contrast-enhanced magnetic resonance images.

Zhang, Y., Chen, J.H., Chang, K.T., Park, V., Kim, M.J., Chan, S., Chang, P., Chow, D., Luk, A., Kwong, T., Su, M.Y., 2019. Automatic breast and fibroglandular tissue segmentation in breast mri using deep learning by a fully-convolutional residual neural network u-net. Academic Radiology 26. doi:10.1016/j.acra.2019.01.012.

Çiçek, , Abdulkadir, A., Lienkamp, S., Brox, T., Ronneberger, O., 2016. 3d u-net: Learning dense volumetric segmentation from sparse annotation, pp. 424–432. doi:10.1007/978-3-319-46723-8_49.