Segmentation of Arteries in Whole Body Magnetic Resonance Angiography Datasets Using Deep-Learning Techniques

Student: James Neill
Supervisors: Mr. Andrew McNeil, Prof. Emanuele Trucco

Abstract—The accurate segmentation of arteries is key to the success of automated software tools aimed at assisting clinicians with their daily workflow. This paper builds upon a recently published research study investigating convolutional neural networks for the purpose of vessel segmentation in whole-body magnetic resonance angiography images. Attempting to improve upon the experimental techniques used, three different approaches are evaluated. These include a selective sub-sampling algorithm, data augmentation, and a fully convolutional architecture. Experimentation has shown both the sub-sampling algorithm and fully convolutional architecture approaches to be effective techniques in increasing efficiency. With the fully convolutional architecture improving performance significantly.

1 Introduction

Deep learning has seen tremendous success as of late, superseding many conventional machine learning approaches to establish itself as the new state-of-the-art for many applications. In computer vision, one of the key areas of deep learning research is that of semantic segmentation—the process of partitioning a 2D image, video or 3D volume into distinct regions or classes. Accurate semantic segmentation algorithms are of increasing relevance to a variety of fields, with applications including autonomous vehicles [1], satellite imagery [2], [3], augmented reality [4], and medical imaging [5], [6], [7], [8].

Deep learning techniques have been successfully applied to a range of different medical imaging modalities consistently producing state-of-the-art results. In this project, we take a closer look at the novel application of deep learning for the segmentation of arteries in whole-body magnetic resonance angiography (MRA) datasets.

1.1 Background and Motivation

The assessment of vascular structures is a crucial step in diagnosing a range of cardiovascular-related conditions. Currently we rely on the evaluations of expert clinicians whom manually dissect each image, often using somewhat crude, qualitative measurements [9]. This process is not only time-consuming and expensive but also prone to error due to factors such as inter-operator variability and operator fatigue. Image analysis through deep learning then stands as the best candidate to improve detection and diagnosis to support the clinician with their workflow. The first step in building these tools is the implementation of an effective vessel segmentation algorithm.

1.2 Problem Statement

In a recently published study [10] investigating vessel segmentation in whole-body MRA, a deep convolutional neural network (CNN) was narrowly surpassed by conventional segmentation techniques. The study attributes this to the lack of available training data, consisting of only 3 annotated patients, alongside the adopted network architecture of a voxel-classifier which proved slow in training and testing, making optimisation and experimentation difficult. Taking the results presented in [10] as our starting point, this project has three key aims:

1) Improve the training speed of the voxel-wise classifier by optimising the extraction of the training data.
2) Attempt to improve the performance of the network using common data augmentation techniques.
3) Compare the results with other *fully convolutional* networks which are shown to be more computationally efficient and have been successfully applied to other imaging modalities.

2 Background

2.1 Neural Networks

Much of the core concepts within deep learning share their origins in simple linear models such as the McCulloch-Pitts model [11]. This linear model is a binary classifier, capable of differentiating between two classes of inputs by calculating the weighted sum of each of the inputs passed through an activation function along with a bias. These weights are set manually and are task dependent. The structure of this model can be seen in Figure 1.

Following that the perceptron [12] was later outlined and became the first method capable of learning and adjusting the weights through a new supervised learning approach. This was achieved by calculating the difference between the output and target values, and scaling the result according to a newly defined learning rate. After each forward pass the weights were then simultaneously adjusted, this process then guarantees that a perceptron will converge for any two given classes which are linearly separable.

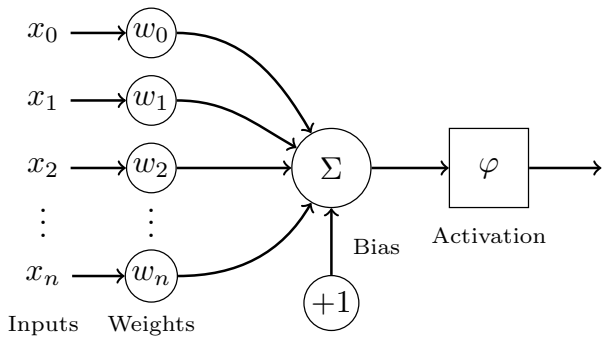


Fig. 1: The structure of an artificial neuron accepting n inputs each with an associated weight. The result of which is summed along with a bias before being passed through an activation function φ.

The multilayer perceptron (MLP) is the network architecture most commonly referred to as a neural network, consisting of at least three layers of multiple artificial neurons. Each neuron is connected to the neurons in the subsequent layer of the network and applies a non-linear transformation in the form of an activation function. The intermediate layers, known as the hidden layers of the network, are what afford MLPs the ability to approximate any continuous function [13], meaning it can differentiate between data which is not linearly separable.

The weights between each neuron are adjusted according to a loss function, with the output error computed, the gradient between each neuron is then calculated using back-propagation and through the process of gradient descent, the weights are then adjusted individually and step towards a local minimum of the loss function.

2.2 Deep Learning

Deep learning expands upon the structure of the MLP, adding many intermediate hidden layers each capable of performing different functions or operations on the inputs. A popular application of deep learning today is that of convolutional neural networks (CNN). The primary layers used within a CNN are the convolutional, pooling, drop-out, and fully connected layers.

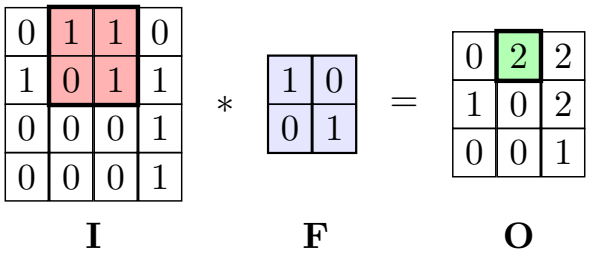


Fig. 2: Depicts the process carried out by a convolutional layer by which a 2D input I is convolved with a filter F to produce an output O. A stride of 1 is used with no padding being applied to the input. The highlighted regions follow one step of the operation.

Convolutional layers convolve an input with a series of small, learnable filters and calculate the dot-product between the input and the filter to produce an activation map. This process can be seen above in Figure 2. The number of filters typically increases as data is passed through the network, increasing the number of local features extracted. Pooling layers are employed to reduce the dimensionality of the data and prevent overfitting with the most popular approach being max-pooling which uses the value of maximum response in a given region. Dropout layers are used to randomly discard a portion of the data in an attempt to prevent the network overfitting to the training data. Fully-connected layers are used to capture global features, with each neuron being fully connected to the neurons in the previous layer.

2.3 Segmentation

Semantic segmentation—the process of partitioning a 2D image, video or 3D volume into distinct regions or classes—is an active area of research. The application of deep convolutional neural networks has proven extremely effective when dealing with natural images, with significant progress being made in medical imaging. There currently exist two primary approaches from which most implementations are derived. The first—and hereby referred to as *pixel-wise* classification for 2D or *voxel-wise* for 3D—is the process of extracting a patch focussed around a central pixel or voxel from which the class or label is then derived. Models trained using this approach will generate a dense segmentation mask by combining its predictions for each pixel in the input. These networks typically consist of a series of convolutional and pooling layers before being passed to a fully-connected layer. Examples of this approach include [14].

The second approach—hereby referred to as *fully-convolutional*—was first defined in [15] which proposes a fully-convolutional network (FCN) architecture. Substituting fully-connected layers for additional convolutional layers, FCNs were capable to then accept inputs of arbitrary sizes and later, through a variety of up-sampling techniques, output a one-to-one pixel-wise classification of the input. The output is therefore of the same spatial size as the input, with a number of channels equal to the number of classes. By applying a Softmax probability function to the output you then obtain the probability of each pixel belonging to a given class.

One technique commonly used for upsampling is the transpose convolutional layer[1] which, much like normal convolutional layers, convolve an input with a set of learnable filters to produce an output. It lends its name from mathematics where it is defined as the transpose of the convolution operation when defined as a matrix operation. Prior to transpose convolutions, simple upsampling layers were used instead which upsample an input through a predetermined algorithm with no learnable parameters.

2.4 Medical Imaging

Medical imaging refers to a variety of techniques and processes used to visualise internal tissue or organs for

1. The transpose convolution operation can also be referred to as a fractionally-strided convolution or a deconvolution.

use in clinical diagnosis and treatment. Within the literature, the most prominently studied modalities for applications in deep learning include X-ray computed tomography (CT), optical microscopy, and structural magnetic resonance imaging (MRI)—with the majority of research focussing on segmentation [16]. Part of deep learnings recent success can be attributed to the increased availability of large annotated datasets for training. A suitable example being ImageNet [17], a dataset which contains in excess of 14 million natural images each with multiple associated bounding boxes and labels. This is in stark contrast to medical imaging datasets which rarely exceed sample sizes in the hundreds due to the time and expense of collecting and annotating such data, in addition to requiring specialised datasets for each imaging modality[2]. This lack of sufficiently high-quality training data is one of the biggest challenges faced in trying to apply deep learning to medical imaging today and thus an important area of research.

3 Literature Review

As previously stated, the use of deep learning within medical imaging is a popular area of research spanning multiple modalities and applications. One approach which has proven effective for segmentation in 2D electron-microscopy images is the U-Net [7] architecture. U-Net employs an encoder-decoder like structure, consisting of a contractive and expansive path. Each block of the contractive path consists of a series of convolutional layers followed by a max-pooling layer. The output of each contractive block halves the spatial resolution of the input whilst doubling the number of features. An expansive block upsamples the input which is then cropped and concatenated with the feature map from the corresponding contractive block and is followed by a series of convolutional layers. The output of each expansive block doubles the spatial resolution of the input whilst halving the number of features. Rectified Linear Unit (ReLU) is used as the activation function, alongside a cross-entropy loss function. Relying on the use of elastic deformations for data augmentation, U-Net was capable of extending their dataset to attain state-of-the-art performance.

Later an extension of this architecture for volumetric data was proposed, which extends the original U-Net architecture with some choice optimizations. Named V-Net [8], it focusses on the segmentation of the prostate in 3D MR images from the PROMISE2012 [18] dataset. All of the networks convolutional layers apply *same* padding, resulting in an output the same spatial size as the input. This alleviates the need to crop the feature maps before concatenation with the results of the upsampling layer. Parametric ReLU (PReLU) is used over ReLU to avoid ReLU saturation. All max-pooling layers are substituted for additional convolutional layers with a kernel size and stride of 2, resulting in the spatial resolution being reduced by a factor of 2. Upsampling layers are then replaced with transpose convolutional layers. Both of these changes introduce more learnable parameters into the network, removing the use of untrainable predetermined algorithms. A novel Dice-coefficient based loss function is used during training, the details of which are further described in Section 5.3. These factors combined resulted in a network which attained better results at a decreased inference time.

The V-Net paper is an example of one of the few implementations which exist in the literature using a 3D architecture, with most approaches which use volumetric data instead opting to process it slice-by-slice due to the decreased computational complexity. However, this approach is less than ideal as it results in the loss of valuable 3D contextual information. Some however then use this decrease in computational complexity to supply their networks with the extracted slices at multiple scales, one such paper [19] looks at this very application for brain tumour segmentation using the BraTS [20], [21] dataset. For their experiments data was fed into a network consisting of three different pathways, one for each scale of the input. The output of each pathway is then combined and passed through an additional two fully connected layers to produce a classification. Their proposed three-pathway architecture outperforms each of the individual networks it is comprised of and is capable of attaining results on-par with the previous state-of-the-art implementations.

Taking a closer look at segmentation for MRI, there is a focus on tumour, lesion, and tissue segmentation with an overwhelming majority of applications focussing on the brain. To our best knowledge, the study by McNeil et al. [10] is the first application of deep learning to whole-body MRA for the purpose of vessel segmentation.

4 Materials & Methods

4.1 Technology

There exist a great number of technologies which could have been employed throughout the duration of the project. This section will cover the technologies used and the rationale behind these choices. Special consideration was taken to account for the shorter time-scale of the project and the authors' relative inexperience with much of the technology required.

4.1.1 Python

Python is a general-purpose programming language which has seen great success in the field of scientific research. This can be attributed to the languages ease-of-use as well as the eco-system, which provides a number of rich purpose-built scientific libraries. With these two considerations in mind, Python was chosen as it offered the author the path of least resistance in achieving the goals of the project.

4.1.2 Framework

Keras is a popular high-level neural network API for Python which acts as a front-end for a variety of deep learning frameworks. Keras was chosen because of its simplicity and flexibility, with Tensorflow being chosen for the backend for 3D transpose convolution operation support. Keras allows for models to be defined quickly whilst retaining full compatibility with raw Tensorflow, allowing the author to leverage additional Tensorflow code snippets and examples where necessary.

2. Transfer Learning has proven an effective method for mitigating this challenge however it falls outside the scope of this project.

4.1.3 Hardware

For the duration of the project, access was granted to the Computer Vision and Image Processing (CVIP) groups compute server, containing two Nvidia GTX Titan X 12GB GPUs using CUDA 8.0 with CuDNN 5. The server was accessed remotely through SSH, connecting through an internal VPN. This allowed for all of the patient data to remain server side in accordance with the conditions outlined for the ethics committee. It is worthwhile noting this particular combination of CUDA and CuDNN only allowed for Tensorflow 1.2.1 to be used as later versions were compiled against CuDNN 6 and above.

4.1.4 Miscellaneous

Excluding those which Keras & Tensorflow depend on such as NumPy, a number of additional third-party libraries were also employed throughout development. The SciPy library was used solely for its `ndimage` class for image manipulation, alongside scikit-image & scikit-learn for importing and performing basic processing on the raw data, which was supplied in the form of stacked tiff images. The graphing library matplotlib was used for plotting figures as well as cPickle for serializing and saving Python objects.

4.2 Data

The data used in this project was collected at Ninewells Hospital in Dundee as part of a recent study [10]. The data has previously been approved for use in research by the East of Scotland Research Ethics Committee. All pre-processing including any corrections and labelling was carried out prior to the beginning of this project.

The dataset is comprised of three patients each split into four stations: 1) head and neck 2) thorax and abdomen 3) pelvis and thighs and 4) the lower legs and feet, each accompanied by a manually segmented ground truth. In keeping with the approach outlined in [10] the lungs for each patient remain masked off. A view of each patient and their corresponding segmentation mask can be seen in Figure 3. From there it is easier to see the challenges faced by the network, where, due to a natural consequence of the acquisition method, organs and other microvasculature are clearly present to varying degrees.

4.3 Experiment Methods

As stated in Section 1.2, a number of different approaches were developed and evaluated for this project, which are described in the following sections.

4.3.1 Voxel Classifier

The voxel classifier used is of the same implementation of the network originally described in [10], with the code defining the network structure along with the training and testing processes being obtained from the author. The network structure was not modified however the number of training epochs was adjusted accordingly to ensure convergence.

4.3.2 Fully Convolutional Classifier

As a consequence of the voxel-wise classification approach, inference of a whole volume can be a lengthy process as each voxel has to be individually classified in order to produce a dense segmentation map. In an attempt to mitigate this, alternative architectures were explored, reproducing the architecture outlined in [7] with some of the improvements suggested in [8], a 2D fully-convolutional network was implemented whose architecture can be seen in Figure 4. Fully-convolutional architectures can produce a dense segmentation map equal to the size of the input volume therefore significantly improving both training and inference times.

The V-Net [8] 3D fully convolutional network structure was originally recreated with the intended purpose of processing 3D volumes to produce a dense 3D segmentation map. However, the network proved untrainable, consistently classifying all inputs as background.

4.4 Training

The training process is split on a station-by-station basis using two patients data, with a validation split of 0.05 and reserving the last remaining unseen patient for testing. Initially as a result of having limited computational resources, this approach was later determined to be the most appropriate as each station presents its own unique challenges best resolved by splitting the training process into distinct regions.

4.5 Evaluation

For the purpose of evaluating the performance of the output segmentation the Dice-similarity coefficient (DSC), seen in Equation (1), was used as it a popular metric for measuring the performance of segmentation algorithms.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

The DSC score ranges between a value of 0 and 1, comparing two samples according to their intersection (true-positive) and union (false-positive), with a score of 1 representing a perfect segmentation. This metric is preferred over simpler ones such as accuracy which can give a skewed measure of performance for highly unbalanced datasets, such as in our case.

5 Experiments

As previously stated, due to time constraints experiments were carried out and tested against patient two as they obtained the lowest average score and contained the worst performing station. An assumption was therefore made any improvements observed would generalise to the other patients.

5.1 Selective Sub-Sampling

Looking at the original implementation, volumes were extracted at random with the number of voxels being bound to a pre-defined upper limit. This random sampling would result in volumes being extracted which had significant

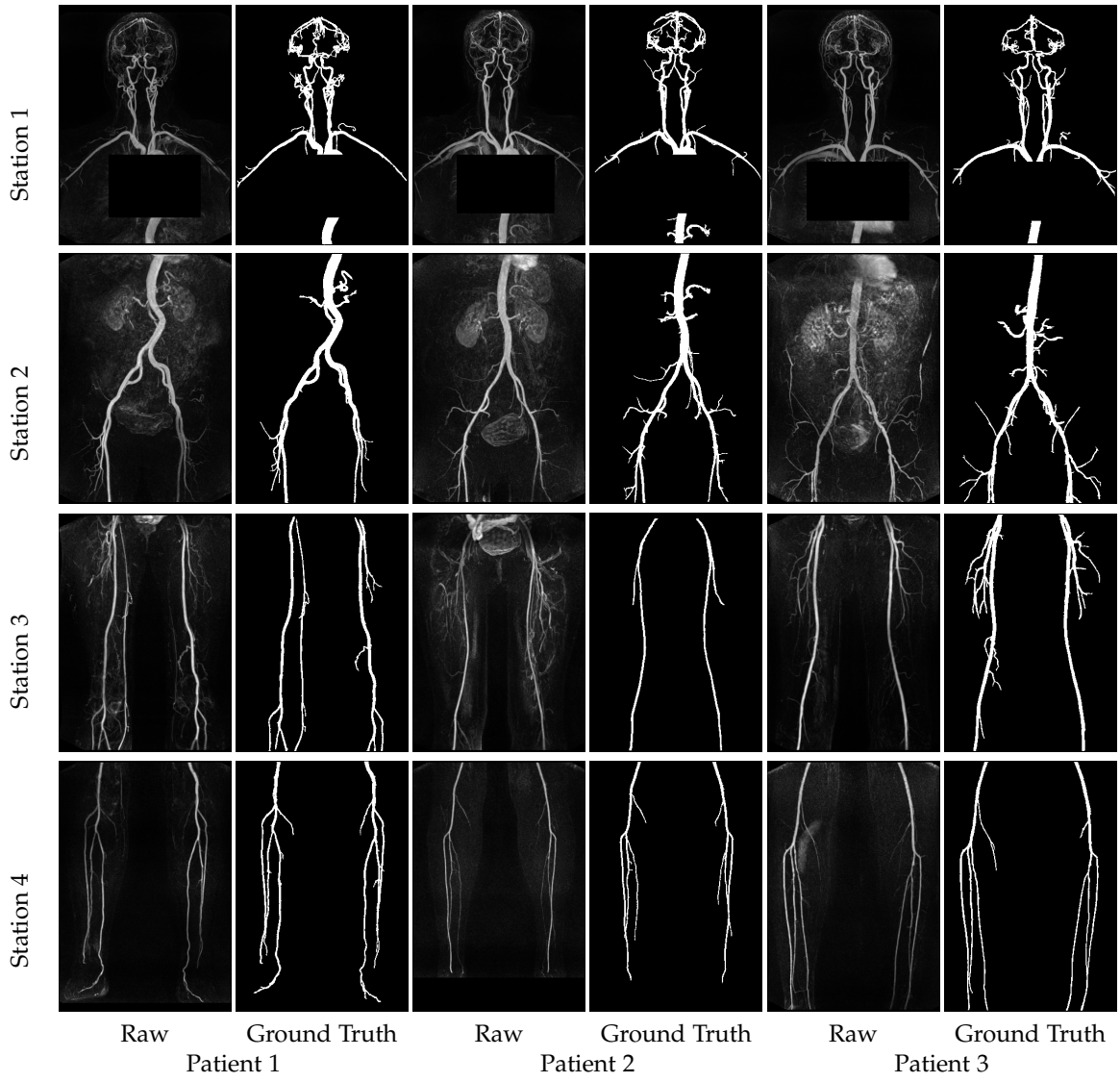


Fig. 3: A comparison of the maximum intensity projections of all patients from both the raw and ground truth volumes.

overlap, introducing an amount of redundancy to the data being fed into the network. This approach resulted in long training times, making it impractical to implement on a large scale and also difficult to assess new architectures or parameters.

This experiment attempts to improve upon the sampling of volumes and instead opts to sub-sample the data in a more regular, structured manner. It is hoped that giving the network fewer samples in a more structured way as to reduce redundancy will offer similar performance for a much smaller footprint in terms of both memory & training times. Doing so would allow for more rapid development and evaluation of different network structures and parameters alongside making a large scale implementation more feasible.

The strategy used for extracting training samples can play a significant role in the resulting performance of a network. A standard approach involves sampling an equal number of patches from each class represented within the data. In this instance the ground truth is a binary segmentation mask of vessels and therefore consists of two classes; vessel and background.

For the voxel-wise classifier, however, during the extraction process voxels are considered to be one of three categories: vessel, near-vessel, or background, with near-vessel samples being defined as background samples located within 3 voxels of a vessel. Voxels are then extracted using the appropriate extraction map which dictates the central locations of voxels for extraction.

The background extraction map is derived as the inverse of the dilated ground truth. The near-vessel extraction map is then obtained by subtracting the ground truth from the dilated ground truth, leaving a region of background voxels surrounding any vessels. The vessel extraction map is then defined simply as the ground truth. All extraction maps are then sub-sampled according to the step parameter. A demonstration of this process and the sub-sampling effect can be seen in Figure 5. The sub-sampling algorithm offsets each axis from one another accordingly, maximising the context of each extracted voxel. As the sub-sampling step increased there became significantly fewer vessels which were being extracted. Therefore, to maintain a balanced

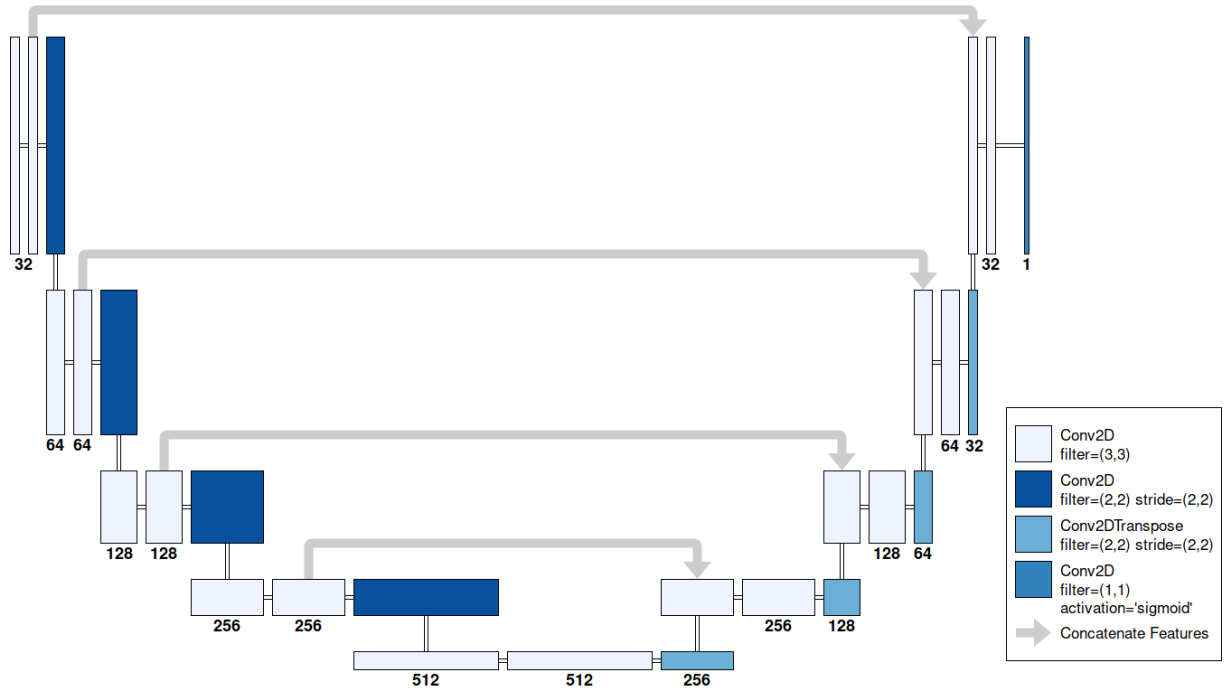


Fig. 4: The network architecture employed for the fully-convolutional classifier. All layers use same padding, ReLU activation function and a stride of (1,1) unless otherwise stated. The number of features is indicated beneath each layer with no fixed input dimensions being specified.

a) b) c) d)

Fig. 5: Visualisation of the extraction process, with a sub-sampling step of 2. **a)** Ground truth slice **b)** Sub-sampled Vessel extraction map **c)** Sub-sampled Near-vessel extraction map **d)** Sub-sampled Background extraction map

dataset, an equal number of background samples were chosen randomly, comprising of 2/3 near-vessel samples and 1/3 background samples.

5.2 Data Augmentation

5.2.1 Augmentation by Orientation

The dataset being using for this project is quite limited, consisting of only three patients. This experiment aims to extend the existing data by exploiting the natural symmetry present in the human body, applying rotations and reflections to the data and evaluating the resulting performance. Rotation is carried out along the axial plane as this best represented the natural orientation of vessels. In addition to this, reflection of volumes is performed along the sagittal plane, making a volume then appear as if it originated from the opposite side of the body.

5.2.2 Augmentation by Noise

Following on from the orientation augmentation experiments, a noise augmentation model was explored in an attempt to emulate the noise and variations in intensity observed in the data originating from changes in the magnetic field of the scanner. To achieve this a Gaussian distribution was generated with decreasing values of sigma then summed to the original volume.

5.3 Fully Convolutional Architecture

Seeking to speed up training and inference times a fully convolutional architecture was explored. For the experiments, patches are extracted according to two parameters: the volume size and the stride. The volume size indicates the size of the volume to be extracted with the stride then dictating how far the window shifts each iteration. This process is commonly referred to as a sliding window approach. Once all of the patches have been extracted their corresponding ground truths are evaluated. Any patch for which no vessel samples are present in the ground truth is then discarded.

When training the FCN classifier, a Dice-coefficient based loss function was employed as seen in [8], however, their implementation is not fully expanded upon. Our interpretation of their method can be seen in Equation (2), with the specifics of the Dice-coefficient itself being discussed in Section 4.5.

$$L(DSC) = -(\frac{2|X \cap Y| + 1}{|X| + |Y| + 1}) \tag{2}$$

A smoothing factor of 1 is included in both the numerator and denominator, this is done to avoid calculation errors when the network correctly predicts a volume as being entirely background i.e. division by zero. By using a Dice-coefficient based loss function the network is then optimising its internal parameters in order to produce results

which will attain a maximum DSC score. In addition to this, because it doesn't factor in true-negatives, it's able to deal with unbalanced datasets where one class is significantly over-represented, such as in our case.

6 Results & Discussion

6.1 Selective Sub-Sampling

A comparison of the performance of different sub-sampling levels against the original sampling technique can be seen in Table 1, with a side-by-side comparison against the ground truth shown in Figure 8. In all instances, the network was trained until convergence. Immediately a significant reduction in the epoch duration can be observed, with an initial increase of the step size to 2 resulting in epoch durations dropping by as much as a factor of 4. Increasing the sub-sampling step further continues this trend at the expense of some loss in performance. This can be seen in Figure 6.

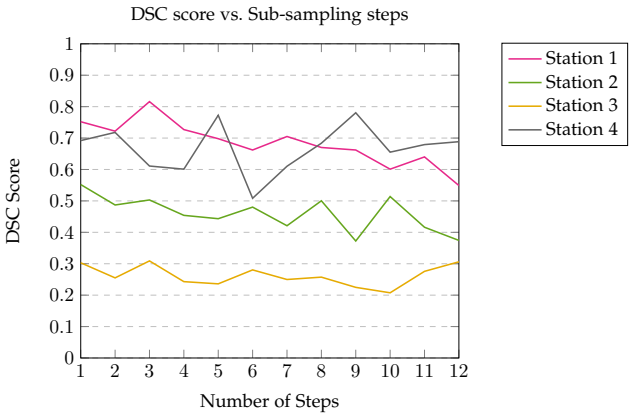


Fig. 6: Comparison of DSC scores at differing sub-sampling steps over each station.

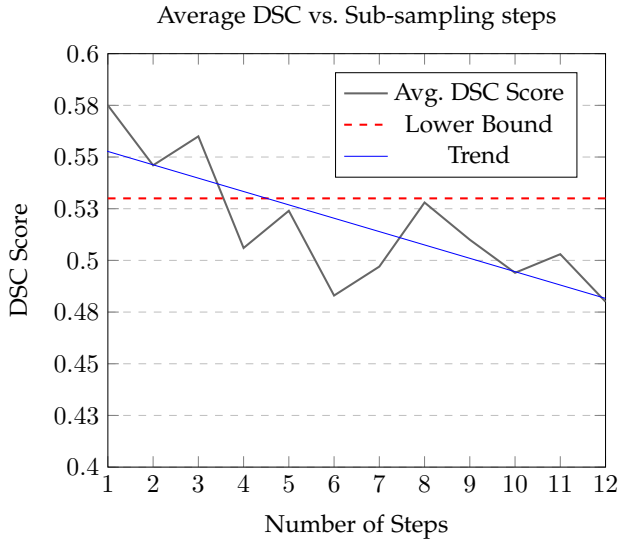


Fig. 7: Average DSC score over increasing sub-sampling steps. Dotted red line indicating the lower limit of the confidence boundary of the original sampling technique.

In some instances higher sub-sampling steps may obtain better results than that of lower sub-sampling steps, this can be easily seen when comparing the DSC scores in Figure 6. This may at first seem counter-intuitive that a network being supplied with significantly fewer samples would perform better, however, it is instead assumed that the data being extracted better represents the volume which the model is later tested against. Given enough time, the author would have liked to have performed cross-validation with the other patients but this had to be put aside in the interest of time.

Assessing the results presented in Table 1, a sub-sampling step of 3 was determined to be optimal as it achieved an average within the variance, or confidence interval, of the original sampling technique. On average stations trained using this sub-sampling step required approximately 10 times fewer samples and whilst maintaining an overall, albeit arbitrary, average relative performance level of 98% when compared to the original sampling technique. Therefore, by extrapolating these numbers, the same network could conceivably be trained using 20 patients data in approximately the same time-frame as the original approach which used 2 patients.

Breaking down the DSC score into its two primary components we can gain a better picture of the performance of the network as the step size increases. As shown in Figure 9 the intersection of the predicted volume and the ground truth remains fairly constant as the step size increases however the union steadily grows. This indicates that at higher sub-sampling steps the loss in performance can be attributed to the network misclassifying background voxels. This is expected as the structure of a background volume is much more variant than that of one which contains a vessel, with the background containing almost any region of the volume and having no obvious structure unlike a vessel. This effect can be seen when comparing the predicted segmentation maps from Figure 8, noting the increase in observable noise across station 1 corresponding with increases in the value of the union in Figure 9.

6.2 Augmentation by Orientation

Looking at then improving the performance of the network, some basic data augmentation techniques were explored and compared which changed the orientation of volumes through rotations and reflections along a given axis. Initially it was intended to use the optimal sub-sampling step established in Section 6.1, however due to time constraints a step of 8 was used instead because of the significant decrease in the number of volumes. The results of this experiment can be seen in Table 2.

Assessing the results the only effective technique was the rotational augmentation, which rotated each sub-volume 90, 180, and 270 degrees along the axial plane. The author is surprised at the ineffectiveness of the standalone reflection augmentation and anticipated it to achieve the greatest relative performance gain. After assessing the results further it is believed that rotational augmentation will introduce some new data which is not anatomically correct. Whenever this is reflected the effect is then compounded, leading the network to become optimised to data which is not representative of real-world situations. As a result of using a high sub-sampling step the number of samples used for

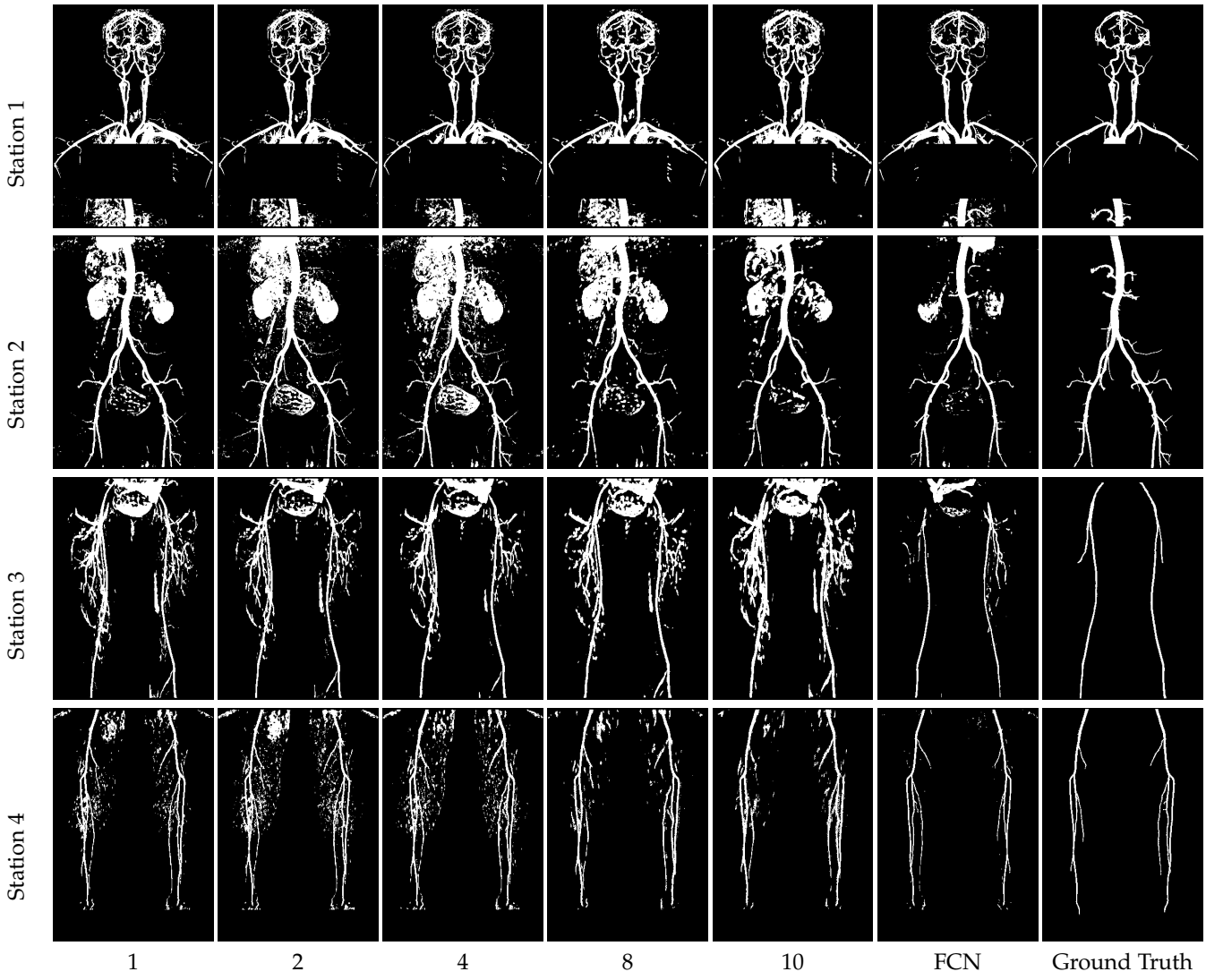


Fig. 8: Comparison of predicted volumes of networks trained with differing levels of sub-sampling dilation, a fully-convolutional architecture and finally the associated ground truth.

augmentation was quite low, decreasing the diversity of the samples and possibly decreasing the effectiveness of the data augmentation.

Whilst rotational augmentation proved effective in this instance, it may be the case the data supplied was better suited to being augmented in this manner, producing data which remained representative of the test data. However, given different samples, it is possible that another augmentation technique might have proved more effective, with rotation maybe even reducing the performance.

6.3 Augmentation by Noise

Due to time limitations this experiment was carried out using a sub-sampling step size of 8 alongside rotational augmentation. Initial results from the noise modelling experiment can be seen in Table 3 using three different values of sigma. A significant drop in performance is immediately observed, with performance steadily decreasing as sigma increases.

Taking a closer look then at the applied augmentations in Figure 10, it is clear that the noise model used is too simplistic, resulting in data which is not representative of real-world conditions. The author would have liked to have been able to revise their implementation and carry out further experiments however due to time constraints this had to be skipped.

6.4 Fully Convolutional Architecture

Using the fully-convolutional architecture outlined in 4.3.2, experiments were carried out using different combinations of volume and stride sizes. The network consists of a series of 2D convolutional layers, each supporting inputs with up to three channels typically for handling RGB images. Since our data contains only a single channel for intensity, multiple slices were able to be processed at once giving the network some, however very minimal, 3D contextual information. For this experiment patches were extracted along the coronal plane to maximise the area occupied by vessels, however, sagittal would have also been suffice.

Assessing the results presented in Table 4 a significant increase in performance can be seen across the board when

Step	Station	DSC	Mean (σ^2)	Samples	Epoch Duration
1	1	0.752	0.575 (0.0398)	272,000	2m19s
	2	0.552		320,000	3m8s
	3	0.303		56,000	53s
	4	0.692		48,000	37s
2	1	0.722	0.546	76,626	35s
	2	0.487		100,814	46s
	3	0.255		41,778	19s
	4	0.718		19,842	9s
3	1	0.816	0.560	22,580	10s
	2	0.503		29,842	14s
	3	0.309		12,418	6s
	4	0.611		5,872	3s
4	1	0.727	0.506	9,592	4s
	2	0.454		12,418	6s
	3	0.243		5,218	2s
	4	0.601		2,404	1s
5	1	0.643	0.524	4,982	3s
	2	0.443		6,432	6s
	3	0.236		2,720	1s
	4	0.773		1,256	1s
6	1	0.662	0.483	2,796	2s
	2	0.480		3,730	2s
	3	0.280		1,572	1s
	4	0.508		722	1s
7	1	0.705	0.497	1,798	1s
	2	0.421		2,360	1s
	3	0.250		1,000	1s
	4	0.610		466	0s
8	1	0.670	0.528	1,202	1s
	2	0.500		1,614	1s
	3	0.257		646	1s
	4	0.684		268	0s
9	1	0.662	0.510	862	1s
	2	0.372		1,116	1s
	3	0.225		472	0s
	4	0.781		196	0s
10	1	0.601	0.494	666	0s
	2	0.514		806	1s
	3	0.208		316	0s
	4	0.655		168	0s
11	1	0.640	0.503	412	0s
	2	0.416		590	1s
	3	0.276		230	0s
	4	0.679		96	0s
12	1	0.549	0.480	358	0s
	2	0.374		468	1s
	3	0.306		198	0s
	4	0.688		90	0s

TABLE 1: Results of sub-sampling experiment with different steps compared across stations. Epoch duration measurements were obtained using Keras during the training process and were only available to the nearest second. A step size of 1 is representative of the original sampling technique.

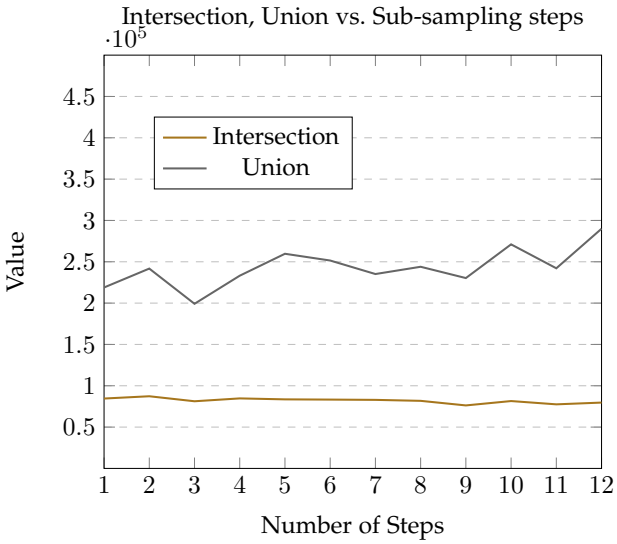


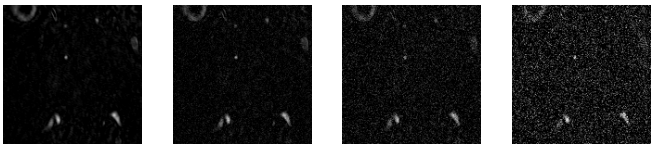
Fig. 9: Comparison of intersection and union values at differing sub-sampling steps for Station 1.

Augmentation	Scaling	Station	DSC	Mean
None	1x	1	0.670	0.528
		2	0.500	
		3	0.257	
		4	**0.684**	
Reflection	2x	1	0.647	0.465
		2	0.506	
		3	0.258	
		4	0.449	
3 90° Rotations	4x	1	0.693	0.547
		2	**0.508**	
		3	**0.337**	
		4	0.651	
3 90° Rotations, All Reflected	8x	1	0.685	0.490
		2	0.477	
		3	0.306	
		4	0.491	
180° Rotations	2x	1	**0.715**	0.443
		2	0.448	
		3	0.289	
		4	0.319	
180° Rotations, All Reflected	4x	1	0.685	0.490
		2	0.477	
		3	0.306	
		4	0.491	

TABLE 2: Results of orientation data augmentation, table shows the additional data being added to the dataset each time. In the scaling column *n*x refers to the amount the dataset was scaled or expanded upon through data augmentation e.g. given a sample size, s = 1000, and a scaling factor, n = 3, the resulting dataset would contain 3000 total samples.

compared to the voxel-wise classifier. Noting that even the poorest performing configuration attained performance above that of the original implementation. This can be seen further by comparing the predicted segmentation maps shown in Figure 8, with the fully convolutional approach containing much less noise across all four stations.

The first set of experiments increased the channel depth of the extracted volume from 1 to the maximum supported of 3. A loss in performance can be seen when 2 channels are used, whereas extending the patch further to 3 channels results in an overall performance gain. Although the amount of additional data is small these results may indicate there is value in supplying the network with further 3D contextual information. From here two different stride values were tested, assessing how the network responded to receiving additional overlapping data. In both instances changing the



None 0.05 0.1 0.2

Fig. 10: Visualisation of noise being applied to a volume slice with increasing values of sigma.

Sigma	Station	DSC	Mean
0.05	1	0.703	0.434
	2	0.445	
	3	0.249	
	4	0.337	
0.1	1	0.717	0.402
	2	0.420	
	3	0.243	
	4	0.227	
0.2	1	0.708	0.388
	2	0.389	
	3	0.248	
	4	0.208	

TABLE 3: Results of noise modelling data augmentation.

stride results in an overall decrease in performance. Cross-validation with the remaining two patients would help further understand the effects of these experiments however due to time constraints and restrictions within the software implementation this was not possible.

Lastly, the inference times of the fully-convolutional and voxel-wise classifiers were compared. The duration was measured using Python's built-in `time` class, with the code being wrapped around only the relevant Keras method. The results are presented in Table 5 demonstrate clearly the efficiency of the fully-convolutional approach over that taken in the voxel-wise classifier.

7 Conclusion

The aim of this project was to extend the experiments carried out by McNeil et al. [10], seeking to improve the training time and predictive performance of the proposed network, as well as comparing it to other deep learning architectures. An effective sub-sampling technique was presented which lead to data being sampled in a more structured manner. Resulting in a significant reduction in required training times at the expense of a minimal performance loss. Data augmentation techniques were explored however the proposed model proved to be too simplistic, leaving further work to be carried out in this area. Inspired by a state-of-the-art network architecture in the literature, a fully convolutional network was implemented. Resulting in reduced training and inference times in addition to more accurate segmentation masks being predicted.

8 Future Work

The results produced from this project prove promising however it is recognised there remains significant room for improvement. Some of the suggestions were either initially considered or partially implemented but, for a variety of reasons, results could not be produced.

Volume Size	Strides	Station	DSC Score	Mean
(64, 64, 1)	(32, 32, 1)	1	0.871	0.681
		2	0.571	
		3	0.397	
		4	0.883	
(64, 64, 2)	(32, 32, 1)	1	0.862	0.676
		2	0.538	
		3	0.417	
		4	0.887	
(64, 64, 3)	(32, 32, 1)	1	0.859	**0.707**
		2	0.662	
		3	0.415	
		4	0.890	
(64, 64, 3)	(16, 16, 1)	1	0.834	0.657
		2	0.657	
		3	0.389	
		4	0.747	
(64, 64, 3)	(64, 64, 1)	1	0.851	0.665
		2	0.453	
		3	0.448	
		4	0.906	

TABLE 4: Results produced by the fully-convolutional architecture with differing number of channels and stride sizes.

Station	Voxel-wise	FCN
1	155.62s	3.88s
2	378.40s	4.43s
3	212.55s	2.78s
4	81.67s	3.11s

TABLE 5: Comparison of inference times in seconds between the voxel-wise classifier and the fully-convolutional classifier, excluding data extraction and any pre-processing.

Before any further extensions to this project are to be considered cross-validation should be performed for each approach to verify its validity. This could take the form of running the experiments again using the remaining combinations of patient data or, in the case of the sub-sampling experiment, changing the starting position of the algorithm. Until that has been complete these results should be considered preliminary. From there a more realistic noise model and other data augmentation techniques could be implemented. With papers such as U-Net [7] showing just how effective semantically correct augmentations can be. The training process used for the fully convolutional classifier is very basic, a better sampling technique could be explored alongside changing the extraction axis. Results shown in Table 4 indicate there is possible value in providing the network 3D contextual information. With much of the code already being in place, the next improvement therefore would be to extend the architecture to allow for true volumetric data to be processed as seen in [8].

9 Appraisal

Looking back over the project I am proud of what I was able to achieve, especially considering the time-scale and

having no prior experience with even basic machine learning. The initial 3 months of the project were consumed with familiarising myself with the necessary deep learning concepts, alongside Python and the broad range of libraries which would later be employed such as Keras and NumPy. However I never truly stopped learning with each passing week presenting new challenges.

One of the biggest challenges faced throughout the project was interpreting some of the literature, either due to the level of assumed knowledge, variations in definitions or even seemingly vital details of the implementation being concealed behind vague terms such as "appropriately". The successful application of deep learning often lies within the specifics of the implementation itself, so as someone new to the field I found it frustrating when a key part of an implementation wasn't expanded upon. A consequence of the long training and testing times, concepts had to be well thought out before being applied. As someone who relies heavily on trial and error as part of their learning process, this, combined with the inherent delay between executing an idea and seeing the result, made learning all that more difficult.

Given the benefit of hindsight however it is clear a number of these challenges were a direct result of inexperience which ultimately led to naive assumptions being made and mental models needing to be re-evaluated. This lead to a significant amount of time being misspent having to re-run experiments. A further example of this can be seen in certain aspects of the software design, where unforeseen changes needed to be made late into the project but couldn't be due to the structure being built around certain assumptions. Given the opportunity to start over there are a number of things which I would choose to do differently. The first of which being to prototype functions locally on a smaller scale. Doing this at an earlier stage would have been much more productive and lead to more effective algorithms being established earlier on. Another aspect which I could improve upon is project management. I often found that due to my inexperience I didn't know what a task truly entailed or what impediments it might present until I had completed it. The biggest change, however, would be to start writing the report much earlier. Having to commit ideas and concepts to paper forced me to not only understand them but also how they fit into the bigger picture.

With having no experience in any of the subject areas, choosing this project did present an element of risk. However, I am glad to say the risk has most certainly paid off with this being the most interesting and enjoyable project I have had the opportunity to work on yet.

Acknowledgements

I would like to thank both of my project supervisors Mr. Andrew McNeil and Prof. Emanuele Trucco for both their time and guidance over the course of this project. Andrew provided me with the dataset and code-base from which this project is based, setting aside a significant portion of his time during the final year of his PhD to advise me on many aspects of the project. In addition to this, I would like to thank Manuel for making me feel so welcomed within research at Dundee, with the VAMPIRE project meetings standing out as one of the highlights of the project.

I would also like to thank a very core group of friends who have kept me sane through many assignments and our time at university.

Finally, I would like to thank my parents for their unfailing moral and financial support throughout my degree.

APPENDIX A

Source Code

APPENDIX B

Mid-Project Report

APPENDIX C

Meeting Minutes

APPENDIX D

Poster

APPENDIX E

Fully-convolutional network predicted volumes on the remaining two patients. Due to misaligned padding a DSC score could not be produced.

REFERENCES

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[2] M. Volpi and V. Ferrari, "Semantic segmentation of urban scenes by learning local class interactions," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on*, pp. 1–9, IEEE, 2015.

[3] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of cnns," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, p. 473, 2016.

[4] M. Roxas, T. Hori, T. Fukiage, Y. Okamoto, and T. Oishi, "Occlusion handling using semantic segmentation and visibility-based rendering for mixed reality," *arXiv preprint arXiv:1707.09603*, 2017.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[6] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, pp. 2843–2851, 2012.

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

[8] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*, pp. 565–571, IEEE, 2016.

[9] E. L. Yuh, S. R. Cooper, A. R. Ferguson, and G. T. Manley, "Quantitative ct improves outcome prediction in acute traumatic brain injury," *Journal of neurotrauma*, vol. 29, no. 5, pp. 735–746, 2012.

[10] McNeil, A., Degano, G., Poole, I., Houston, G., Trucco, and E., "Comparison of automatic vessel segmentation techniques for whole body magnetic resonance angiography with limited ground truth data," in *Proc. 21st MIUA Medical Image Understanding and Analysis*, vol. 723, pp. 144–155, 2017.

[11] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[12] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[13] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[14] W. Thong, S. Kadoury, N. Piché, and C. J. Pal, "Convolutional networks for kidney segmentation in contrast-enhanced ct scans," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–6, 2016.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[16] G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *CoRR*, vol. abs/1702.05747, 2017.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[18] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang, *et al.*, "Evaluation of prostate segmentation algorithms for mri: the promise12 challenge," *Medical image analysis*, vol. 18, no. 2, pp. 359–373, 2014.

[19] L. Zhao and K. Jia, "Multiscale cnns for brain tumor segmentation and diagnosis," *Computational and mathematical methods in medicine*, vol. 2016, 2016.

[20] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[21] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data*, vol. 4, p. 170117, 2017.