Automatic Segmentation of Brain Tumours in 3D MRI

An adaptation of 3D U-Net++

Neil Micallef

Supervised by Mr Dylan Seychell

Co-supervised by Dr Claude Julien Bajada

Department of Artificial Intelligence

Faculty of ICT

University of Malta

January, 2021

A dissertation submitted in partial fulfilment of the requirements for the degree of M.Sc. (Hons.) Artificial Intelligence.

L-Università
ta' Malta

WWW.UM.EDU.MT

First edition, January 16, 2021

Acknowledgements

I would like to take this opportunity to thank my supervisor and co-supervisor Mr Dylan Seychell and Dr Claude Julien Bajada. Their support throughout this project and the conference submissions was invaluable. Sincere thanks also go to my family and close friends for their support throughout the development of this project. Finally, I would like to thank the staff at MFED who manage communications related to the Endeavour scholarship for their continued communication throughout the dissertation.

EU funds for Malta 2014 2020

The research work disclosed in this publication is partially funded by the
Endeavour Scholarship Scheme (Malta). Scholarships are part-financed
by the European Union - European Social Fund (ESF) -
Operational Programme II – Cohesion Policy 2014-2020
"*Investing in human capital to create more opportunities and promote the well-being of society*".

European Union – European Structural and Investment Funds
Operational Programme II – Cohesion Policy 2014-2020
"Investing in human capital to create more opportunities
and promote the well-being of society"
Scholarships are part-financed by the European Union -
European Social Funds (ESF)
Co-financing rate: 80% EU Funds;20% National Funds

Abstract

In recent times, the capabilities and availability of deep learning techniques for image segmentation have increased considerably. U-Net is an example of a popular deep learning model designed specifically for biomedical image segmentation. U-Net++ combines the parameter efficiency of U-Net with the benefits of dense networks; namely the usage of more refined global and local feature contexts. We propose a model which adapts the U-Net++ architecture for brain tumour segmentation, with some of the modifications to the architecture resulting in an even smaller model. The proposed approach obtained Dice Coefficient scores of 0.7192, 0.8712, and 0.7817 for the Enhancing Tumour, Whole Tumour and Tumour Core classes when compared to the expert ground truths of the BraTS 2019 Challenge validation dataset. Whilst these results are surpassed by some published works discussed in Section 4.3.2, the results obtained are still fairly positive.

The proposed approach differs from the standard U-Net++ model in a number of ways. Firstly, the optimisation function used in this experiment is a multiclass Dice Coefficient loss rather than the binary cross entropy. The number of convolutional blocks was halved to produce a much smaller model. U-Net++ also produces full-resolution secondary segmentation maps, which the proposed model combines using element-wise additions rather than averaging as in the original work. The model also makes use of $3 \times 3 \times 3$ segmentation kernels instead of $1 \times 1 \times 1$ as some initial tests showed that this configuration produced slightly better results.

The experiments performed throughout the project's development demonstrate the importance of using data augmentation and post-processing techniques, both of which substantially improved the model predictions. Furthermore, additional experiments were carried out including a change in optimisation function to the original binary cross entropy function used for U-Net++; using the standard number of convolutional blocks, and the use of dropout regularisation. These experiments were selected on the basis of them being either standard experiments (dropout) or properties of the original model (binary cross entropy, convolutional blocks). Nonetheless, this latter set of experiments did not merit sufficient improvement to be included in the proposed model architecture.

Thus, this dissertation presents a novel adaptation of the U-Net++ architecture, which is lightweight and produces good quality segmentation results. A variant of the model was also submitted to the Organisation for Human Brain Mapping (OHBM) for a poster presentation, and to the IEEE MELECON 2020 Conference as a conference paper under the 'Advances in Medical Informatics for Healthcare Applications' track. Both submissions were accepted by the respective bodies and presented using electronic methods due to the ongoing global pandemic situation.

Contents

List of Figures

List of Tables

1

Introduction

The GLOBOCAN 2019 cancer statistics have not been published at the time of writing, however statistics for the US reported by Siegel et al. (2019) exhibit projections of over 1.7M new cases of cancer in the US for 2019, where brain tumours accounted for over 23,000 of these diagnoses. Cancer was also the second highest cause of death following heart disease, with a death rate approaching 33%. A brain tumour may be defined as an abnormal growth of cells within the brain (Anitha and Murugavalli, 2016). Low grade gliomas (LGG) are benign and may be removed through medical intervention, whilst high grade gliomas (HGG) such as glioblastoma multiforme (GBM) are dangerous and require treatment using radiotherapy. Therefore, care must be taken to prevent benign tumours from reaching later stages (Rajini et al., 2012).

Magnetic Resonance Imaging (MRI) is an imaging technique for diagnosing and monitoring brain tumours, as it provides detailed maps of the brain (Guo, 2012). One of the traditional approaches for analysing MRI involves manually segmenting salient regions within the brain (Gordillo et al., 2013; Liu et al., 2014). This is a tedious process which requires an analyst to draw over a stack of 2D slices from the scan using specialised software. Compared to an automatic method, manual segmentation was found to be as much as several hours slower on a 20-patient study (Kaus et al., 2001).

Manual segmentation is also heavily reliant on the skill level and attentiveness of the analyst. This introduces inter and intra-operator variability (Mazzara et al., 2004; White et al., 1999), as individual analysts may produce substantially different segmentations. One should also consider the effects of external factors such as the analyst's fatigue which may impact segmentation quality. This emphasizes the benefits of an automatic system capable of producing consistent medical grade results. The amount of operator time saved by forgoing manual segmentation is already a sizeable benefit, as operators would have extra time to perform other tasks as necessary.

1.1 | Motivation

Gordillo et al. (2013) state that clinical acceptance (and adoption) of segmentation techniques generally depends on the simplicity of the approach and the level of interaction a system has with the user. This statement highlights how some medical institutions may opt to use manual segmentation methods in favour of techniques which may at least appear overly complex and require extensive training. Liu et al. (2014) also claim that manual segmentation is often considered for clinical use. Manually delineating brain tumours from MR images is a tedious process which also requires specialised software to view the scan and manually draw on the rendered images. This implies that apart from the procedure of manual segmentation being in itself a slow operation, users would also require specific training to make use of the software to carry out the task. Ideally, tumour segmentation is unhindered in terms of processing time and operator bias.

This is also a skill-based problem, since experienced operators or analysts would take a lesser amount of time as opposed to someone who is newer to the domain. Naturally, a hospital or any other facility specialising in this practice would have new entrants who would require an appropriate window to learn how to perform the segmentation. This also creates room for errors to take place during the learning curve, which is less than ideal since such diagnoses have a severe impact on the patients' lives. Additionally, training sessions of this manner come at a cost of the experienced operator's time to teach the process, apart from the extra time taken on the actual segmentation.

The implementation and adoption of fully automatic segmentation techniques into clinical environments provides a number of considerable benefits. A model pre-trained on a large number of MR images would only require a short duration to process unseen observations. Automation of the process also effectively nullifies human errors which may arise due to external factors such as fatigue or distractions. Similar to most solutions implementing artificial intelligence, the quality of the data is an essential factor in the model's performance. Efficient preprocessing techniques minimise noise in the images and also any intensity gradients introduced through the use of different scanners. The use of a robust dataset also allows for stronger predictions on real-world cases, allowing the model to be capable of performing well on actual institutional data. This is why we propose a model that automatically segments brain tumours from multimodal MRI using the BraTS 2019 datasets. More details on the competition and dataset are provided in Sections 2.1.2 and 3.1.2.2. An example from the BraTs training data is shown in Figure 1.1, depicting a slice from the MR image and the ground truth of the tumour segmentation.

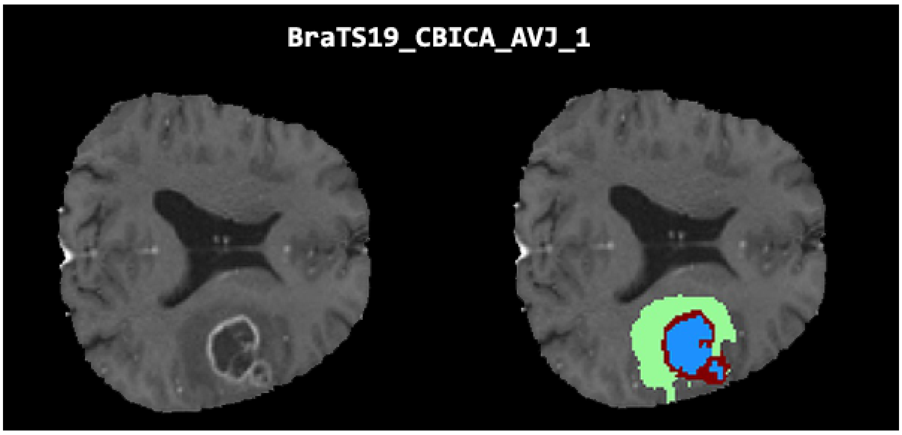


Figure 1.1: Left to Right: T_1ce slice from the BraTS training data and corresponding ground truth for the tumour segmentation. More details on MR image modalities and segmentation classes will be provided in the sections to follow.

1.2 | Aims and Objectives

The goal of this research is to create an automatic computer vision system capable of performing brain tumour segmentation on multimodal MR images. To achieve this goal, the following objectives were considered:

- To create a model using multimodal 3D MR images as input and generate a prediction of the corresponding brain tumour, compatible with standard MR viewers.
- To validate the work through a peer-reviewed process during development and receive constructive feedback for improving the project.
- To work independently, not requiring any input or feedback from the user during training and prediction.
- To leverage the U-Net++ model and investigate how changes to the architecture affect performance.
- To survey state-of-the-art methods to produce a unique approach with results adequate for a clinical setting and assess the approach against these methods, as discussed in Section 4.3.2.

1.3 | Publications

Two of the most notable contributions made by this project are the small size and GPU efficiency of the proposed model which do not compromise segmentation performance. An adaptation of the model which also embodies these properties was the main focus of publications in the international peer-reviewed conferences listed below:

- An electronic poster presentation[1] for the OHBM 2020 Conference with the following description: Neil Micallef, Dylan Seychell, and Claude Bajada. A nested U-Net approach for brain tumour segmentation. In *OHBM 2020 Conference (OHBM 2020)*, Montreal, Canada, June 2020.

- A conference paper (Micallef et al., 2020) for the IEEE MELECON 2020 Conference with the following description: Neil Micallef, Dylan Seychell, and Claude Bajada. A nested U-Net approach for brain tumour segmentation. In *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON) (MELECON 2020)*, Palermo, Italy, June 2020.

1.4 | Proposed Solution

The proposed model leverages a Convolutional Neural Network to adapt the U-Net++ architecture developed by Zhou et al. (2018c).The latter is an encoder-decoder structure which was itself an adaptation of the U-Net CNN built by Ronneberger et al. (2015) for the 2015 ISBI Cell Tracking Challenge. U-Net++ builds upon U-Net by making use of a dense network of skip-connections throughout the model. This allows the model to maintain a detailed network of feature maps from the training process. The original system by Zhou et al. (2018c) was developed and tested on microscopy, RGB video, and 3D CT scans. Our approach is built around multimodal 3D Structural MRI, effectively working with 4-Dimensional Input. The BraTS 2019 Training dataset was obtained as training data for the model, with the 2019 Validation set used for the evaluation on the official CBICA Image Processing Portal (IPP)[2].

[1] https://files.aievolution.com/prd/hbm2001/abstracts/34599/1232_Micallef.pdf

[2] https://ipp.cbica.upenn.edu/

1.5 | Document Structure

Chapter 2 elaborates on the history and minutiae of the problem being solved. The background section provides an overview of MR imaging and the BraTS competition. A description of brain tumour segmentation techniques including clinical approaches and machine learning methods also forms part of the background. The literature review explores peer-reviewed systems performing brain tumour segmentation, presented as a timeline ranging from classical machine learning methods to modern techniques.

Chapter 3 provides a comprehensive look into the data and approach used for this project. This includes a description of the project pipeline and the BraTS challenge datasets used throughout development. Following an analysis of the data's acquisition and definition, data cleaning and preprocessing techniques are explained. Finally, the model architecture and training are addressed, including the hardware deliberations and procedure used for model training.

Chapter 4 expands on the results obtained throughout the project and the evaluation. All major tests and experiments are outlined in this section, including which of the changes were maintained, and which were inconsequential to the final results on the BraTS'19 Validaton set. Visualisations of the model's predictions on unseen samples from the training set against the expert ground truths are also provided. Finally, the proposed model is evaluated against two models built internally and externally against some peer-reviewed state-of-the-art approaches.

Chapter 5 is the final chapter of this document, discussing the conclusions derived from the results and evaluation of the proposed model. The critique and limitations of the project are described in this section. This is followed by a future work section which discusses potential improvements to the model, using the state-of-the-art as a reference where applicable. The end of this chapter consists of the final remarks on the entire project, including my personal experience and comments.

2

Background & Literature Overview

2.1 | Background

A detailed timeline of brain tumour segmentation techniques is provided in this section. Section 2.1.1 introduces MRI as an imaging method. All descriptions within Section 2.1.1 outside of personal observations related to machine learning/computer vision are adapted from 'Picture to Proton' by McRobbie et al. (2017). This publication is a field guide for novices to the MR room which explains both the fundamentals of MR imaging as well as the physics behind the technique. A short description of the MICCAI BraTS challenge will also be provided in Section 2.1.2. Following the imaging method and data source being introduced, an explanation of different methods for performing segmentation is provided in Section 2.1.3. Section 2.1.4 describes traditional machine learning techniques used in a number of related works. Section 2.1.5 introduces Convolutional Neural Networks and a timeline of important models which contribute to the proposed method in this study. Finally, the evaluation criteria used to assess the performance of this project are introduced in Section 2.1.6.

2.1.1 | MRI

MRI is an imaging method that works upon sensitivity to the properties of water, which makes up a high percentage of most tissues in the human body. MRI of the brain is a commonly performed examination in most clinical environments, exhibiting the usefulness of the technique and the visual information it provides about the brain. MRI may be performed across multiple planes, depending on which axis the scanner acts perpendicularly to in 3D. Axial scans are perpendicular to the Z-axis, sagittal scans are perpendicular to the X-axis, and coronal scans are perpendicular to the Y-axis. Figure 2.1 shows an example of each view on a sample from the BraTS 2019 Training set.

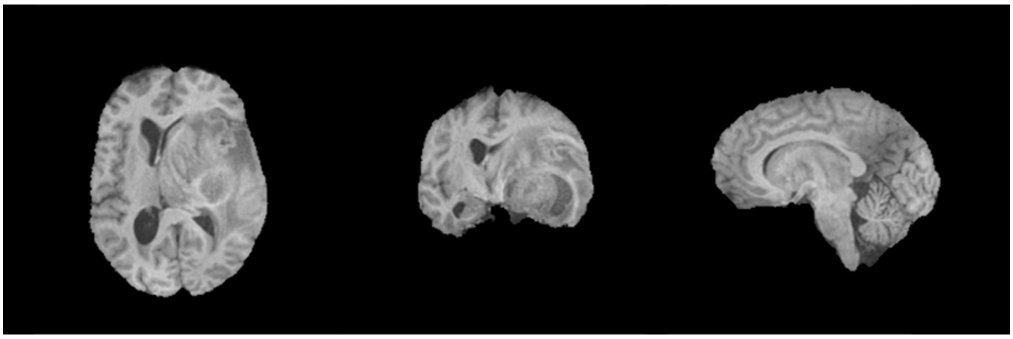


Figure 2.1: Left to Right: axial, coronal, and sagittal slices, selected from the training set

MR signals are propagated from patient tissues in response to radiofrequency (RF) pulses, which are generated from a transmitter coil within the equipment. The localisation of MR signals in the body may be performed through the use of gradient pulses, which are spatial variations in the magnetic field strength across the patient. There are two principal sequence types known as spin echo (SE) and gradient echo (GE). SE sequences generate higher quality images by using two RF pulses to create the echo. GE sequences use a single RF pulse followed by a gradient pulse to create the echo. The main difference between the two is that SE sequences generally create higher quality scans but take minutes to produce rather than seconds. The values of the echo time (TE) and repetition time (TR) are used when describing SE and GE sequences as the timings affect the image contrast.

Separate tissue types have a different level of brightness in MR, enhancing the boundaries between tissues. The natural properties of hydrogen come into effect when we consider the spin-lattice relaxation time (T_1) and the spin-spin relaxation time (T_2). T_1 and T_2 determine how long the tissues will take to return to equilibrium following an RF pulse (or gradient for GE), and change depending on whether the tissue is fat-based or water-based. It is noteworthy that the T_2 timing value is always shorter than T_1. Within the brain, fluids have the longest T_2, followed by water-based tissues and fat. Proton Density (PD) is another property of hydrogen, however PD images are generally used for specific procedures which are outside the scope of this study.

T_1-weighted (T_1w) images may be produced using both GE and SE sequences, as long as a short TR and TE are used. The latter enhances the T_1 differences between tissues, showing fluid and water-based tissues as darker and emphasising fat. Myelin is a fatty tissue often found in white matter, resulting in the latter having a high intensity in T_1w scans. This implies that T_1w images would be optimal for correctly segmenting white matter in the brain, suitable for anatomical studies such as brain segmentation.

Contrast enhancement agents may be applied during an MRI. One example is gadolinium, a metallic element injected into the body or administered orally near the end of an examination. Gadolinium creates a different contrast distribution in the scan, giving the greatest signal intensity for pathologies such as brain tumours. Since pathological tissues are brighter in this modality, it is optimal for the detection and characterisation of diseases. Other potential benefits include identification of both tumour growth and boundaries between healthy and pathological tissues. These properties of gadolinium-enhanced T_1w images (T_1ce) were also observed by Liu et al. (2014). The increased sensitivity to unhealthy tissues makes T_1ce an excellent modality to use for detection/segmentation of abnormalities, and thus very relevant to this study.

T_2-weighted (T_2w) images may also be generated using either SE or GE. Contrary to T_1w , T_2w scans have a long TR and TE. These characteristics make T_2w images almost visually opposite to T_1w. Fluids are now the brightest feature, with water and fat-based tissues being mid-grey. T_2w scans may also be useful for pathology scans as abnormal fluid will have a higher signal intensity compared to healthy tissues.

A variant of T_2w images known as an inversion recovery (IR) sequence exists, termed Fluid Attenuation Inversion Recovery (FLAIR). IR sequences use a different timing parameter known as inversion time, which is used by an extra RF pulse propagated by the system prior to the original pulse. The main difference between FLAIR and T_2w scans is that the Cerebrospinal fluid (CSF) within the brain and any tissues with a similar T_1 value are suppressed from the scan. This is beneficial for identifying pathologies close to ventricles. Figure 2.2 shows the visual differences for each image modality on a random subject selected from the BraTS 2019 Training dataset. One may notice the tumour being particularly visible in the bottom-left corner of the T_1ce slice.

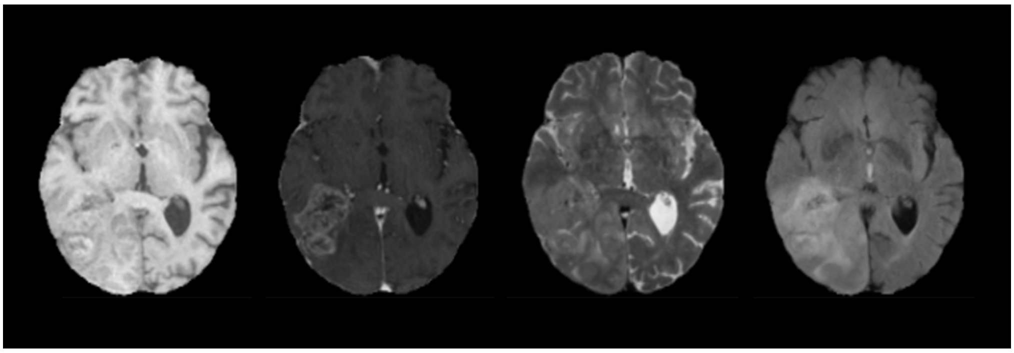


Figure 2.2: Left to Right: T_1, T_1ce, T_2, and FLAIR

2.1.2 | MICCAI BraTS

The MICCAI MultiModal Brain Tumour Segmentation Challenge (BraTS [1]) (Bakas et al., 2017a,b,c, 2018; Menze et al., 2014) is a competition focused on evaluating state-of-the-art techniques for brain tumour segmentation. The organisers maintain the quality of the competition data by providing multi-institutional routine MRI scans, manually segmented by multiple experts. The scans are provided in T_1, T_1ce, T_2, and FLAIR modalities. The data being multimodal allows competitors to create segmentation approaches which are robust to the MRI sequence type. Since the data are obtained from real-world cases, these are very beneficial datasets to use for this task. For this experiment, we will be using the 2019 datasets. Further details on dataset acquisition and the changes applied over successive years of the challenge will be provided in Section 3.1.

2.1.3 | Brain Tumour Segmentation Categories

A survey by Gordillo et al. (2013) categorises segmentation techniques depending on the amount of user interaction involved, a method also used in studies by Foo (2006) and Olabarriaga and Smeulders (2001). At the simplest level, segmentation techniques may be classified as manual, semi-automatic, or fully-automatic.

2.1.3.1 | Manual Brain Tumour Segmentation

Manual brain tumour segmentation is a traditional technique in the domain, where the analyst traces outlines of abnormalities observed in a scan. Such systems are heavily dependent on the operator's knowledge; in extracting the region of interest (ROI), the user is tested on their anatomic expertise and mastery of brain tumour segmentation (Liu et al., 2014). An example of manual segmentation is shown in Figure 2.3.

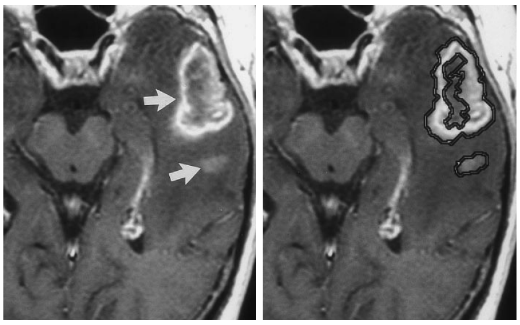


Figure 2.3: An axial MRI slice before and after manual segmentation (Joe et al., 1999)

[1] https://www.med.upenn.edu/cbica/brats2019/data.html

One must consider the fact that tumours do not have a fixed morphology and vary even within the same growth stage/grade. Soltaninejad et al. (2018) state that manual segmentation requires the operator to have a robust knowledge of different tumour shapes and textures, including discrepancies introduced when changing MR scanners. This could however be alleviated if adequate preprocessing steps such as inhomogeneity correction are taken. An observation by Kermi et al. (2018) also shows that there is a level of information loss in all three directions due to how manual segmentation is carried out. The reason behind this is that the practitioner has essentially stacked a set of contours to model the 3D representation of the tumour. Since this depends entirely on the operator being consistent for the annotation of each slice, there is an increased probability of mislabelling.

Studies by Mazzara et al. (2004); White et al. (1999) have shown that there is also a degree of inter and intra-operator variability when extracting a ROI such as a glioma from a medical image. Mazzara et al. (2004) discovered that there was a 28%±12% inter-operator variation in quantified volume between separate individuals performing the same brain tumour segmentation task. There was also a quantified 20±15% intra-operator variation for the same individual performing a task three times over one month intervals. A sample of ROI segmentation variability involving different analysts may be observed in Figure 2.4.

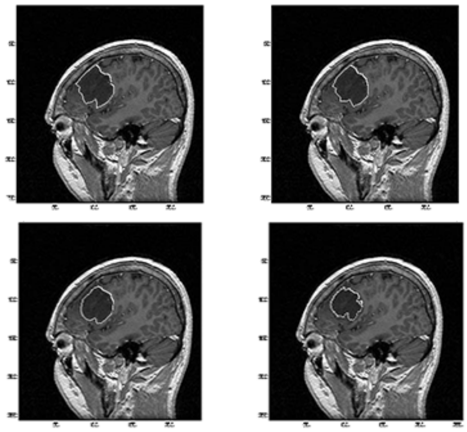


Figure 2.4: Manual segmentation by four experts on the same glioma (Luo et al., 2003)

In spite of these problems, the survey by Liu et al. (2014) claims that manual segmentation methods see considerable use in clinical environments. This is not ideal since medical institutions making use of manual segmentation techniques may be impeded by the restrictions discussed thus far.

2.1.3.2 | Semi-Automatic Segmentation

Semi-automatic approaches act as a middleman between expert knowledge and computation. Rather than the operator carrying out the entire analysis manually, there is an exchange of input and feedback between the operator and software performing the task. Whilst semi-automatic systems work well in combining medical professional knowledge with algorithms, the skill level of the analyst is still an important factor in the quality of the segmentation. A comparison between manual and semi-automatic segmentation was carried out by Joe et al. (1999). The first process evaluated was the contour generation, comparing an analyst using a track ball against the semi-automatic algorithm's contours given certain seed points. The researchers also took note of the time spent loading and saving files for both methods. In total, it was noticed that the semi-automatic algorithm was completing the process roughly 5 minutes faster than the manual method.

An example of a more recent, successful semi-automatic approach is the winner of the BraTS 2015 challenge GlistrBoost (Bakas et al., 2015). This system made use of a modified version of the GLISTR tumour segmentation software (Gooya et al., 2012), taking as input the seed point and radius for a brain tumour. Selection of these parameters constitutes the semi-automatic part of the approach. The preprocessing step involved using the SUSAN corner detector, originally proposed for MRI segmentation by Rezai-Rad and Aghababaie (2006). A gradient-boosting model was then used to generate probability maps for multiple tumour class segmentation. These posterior probabilities were used as input with the image intensity, derivative, geodesic information, and texture features to train the GlistrBoost model. This model obtained accuracies which outperformed the other techniques for that year.

In terms of efficiency, the combination of a computerised approach and human feedback for brain tumour segmentation is a sizeable improvement from manually tracing MRI volumes. However, since there is still a considerable dependency on the input and feedback of the user, such systems may still suffer from inter and intra-operator variability, which was an observation made by Gordillo et al. (2013) as well. This disadvantage of bias and potential skill gaps between operators could be averted by using a system which performs consistently well with minimum to no user intervention. This is the motivation behind fully automatic segmentation methods.

2.1.3.3 | Fully Automatic Segmentation

Fully automatic methods carry out the entire segmentation process with no human interaction involved. Some of the earliest fully automatic methods date back as early as

1998 (Clark et al., 1998) and 2001 (Kaus et al., 2001). The latter claimed to have reduced operator segmentation time from 3-5 hours to 5-10 minutes. Naturally, such systems would require domain-specific expert knowledge to be supplied *a priori*, such that the program can emulate a human operator. This knowledge may be fed into a system in a multitude of ways, one of which would be following rules used by experts for feature extraction from MR images. A more convenient approach would involve obtaining labelled data from control and pathological groups, and feeding the data into a model such as a CNN which performs feature extraction internally. Gordillo et al. (2013) states that automatic segmentation is slightly simplified owing to the brain being well quantified structurally within an MRI. Another point is that the behaviour of different tissue types within the brain is also well known.

Data have also become more accessible for automatic segmentation; 2D slices mitigate the requirement of performing inference of depth, which would be simpler for a human to perform than a computer. In addition, most current datasets are already in 3D, which naturally mitigates the problem. Finally, current hardware and artificial intelligence models are also very well equipped to process segmentation tasks beyond two dimensions. A further challenge for fully automatic systems would be to handle intensity gradients across images obtained from disparate institutions. Literature in this domain has been explored, and techniques designed for correcting intensity inhomogeneities were reviewed by Vovk et al. (2007). These methods could assist the segmentation by applying preprocessing to the image to correct intensity gradients.

2.1.4 | Classical Machine Learning Techniques

Years before the current capabilities of deep learning models for brain tumour segmentation, researchers leveraged machine learning methods available at the time to automate the process. Two main disciplines of techniques will be explored in this section; Support Vector Machines and clustering. There are still a number of recent publications implementing these techniques, with some instances combining the two into a multistage approach, such as Padmanaban et al. (2019).

2.1.4.1 | K-means Clustering

Clustering is a popular unsupervised machine learning technique, used for grouping data belonging to unknown classes. For K-means clustering, the process starts by assigning k observations to be cluster centroids. Selection may be carried out either arbitrarily or through some basic filtering such as sorting and splitting. The other data points in the dataset are then mapped to the 'closest' centroid depending on the distance function being used. The centroids are updated using the mean of the current clusters, hence the name 'K-means'. The process is repeated at each iteration until there are no further changes in class membership. Clusters are mutually exclusive; no data points may be simultaneously a member of two or more clusters at any time. Figure 2.5 is a simple example showing the discreteness of K-means membership.

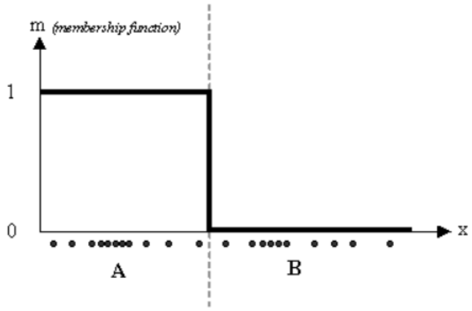


Figure 2.5: K-means Clustering membership function example[2].

One of the preliminary studies exploring the behaviour of K-means clustering is the study by MacQueen et al. (1967). One advantage mentioned in this work is the computational efficiency of K-means, which is where the capability to process larger datasets comes in. One of the cited applications for the technique is 'finding qualitative and quantitative understanding of large amounts of *N*-dimensional data by providing [...] reasonably good similarity groups'. The datasets in biomedical image segmentation are generally sizeable and may also be unlabelled, creating a use-case for K-means.

It is noteworthy that there are other variants of K-means clustering, including Lloyd's least square quantization (Lloyd, 1982) and Hartigan's K-means (Hartigan and Wong, 1979). The latter two methods make use of different heuristics and conditions which may lead to separate clustering results from the conventional K-means clustering. Further analysis of these techniques is beyond the scope of this document. Examples of K-Means clustering implementations for tumour detection/segmentation are discussed in Section 2.2.1.1 in Wu et al. (2007) and Juan-Albarracín et al. (2015)'s research.

2.1.4.2 | Support Vector Machines

Support Vector Machines (SVM) are a commonly used classifier in many domains of machine learning. Some concepts behind the technique were developed from older works of Vapnik and Chervonenkis (Vapnik, 1963) which started development of 'VC Theory'.

[2] https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html

This original paper does not present SVM as we know the technique today, rather focusing on distinction and recognition learning concepts. The SVM was later developed through further collaborations between Vapnik and other researchers (Boser et al., 1992; Cortes and Vapnik, 1995), when the previous research conducted in the VC Theory period gained more attention. The first of the two cited works focused on presenting a classifier maximising the margin between data points and the line of separation between classes. This margin of maximisation is determined by the decision boundary, also referred to as a hyperplane in linear SVMs, as shown in Figure 2.6.

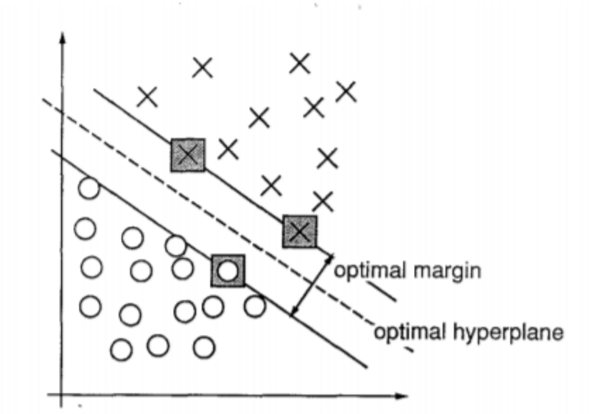


Figure 2.6: Hyperplane example (Cortes and Vapnik, 1995).

The data points at the edge of the decision boundary (in squares in Figure 2.6) are called support vectors. Boser et al. (1992) claim that maximising the distance between opposing support vectors minimises the maximum loss of a classification function. Hence, the optimal hyperplane results in the maximum margin between classes, and best overall results. Equation 2.1 (Boser et al., 1992) shows the base formula used to generate the decision function $D(x)$ for the optimal margin:

$$D(x) = w.\phi(x) + b \tag{2.1}$$

w and $\phi(x)$ refer to vectors of equal dimension, and b is a bias constant. $D(x)$ may be divided by $||w||$ to get the distance between the hyperplane and data point x. It is also noteworthy that the hyperplane is in ϕ space. This equation is still used to date to determine the hyperplane for Linear SVMs. The paper by Boser et al. (1992) provides another contribution in the application of kernel functions such as Radial Basis Functions to transform the input data.

Cortes and Vapnik (1995) later proposed 'Support Vector Networks', essentially presenting a binary classifier now also applicable to non-linearly separable training data and using polynomial decision spaces. In the study, the process is broken down into a number of steps. Feature vectors are provided as input to the support-vector-network and a high-dimensional feature space using non-linear mapping. Within the feature space, a linear decision space is constructed with several properties which enhance the classification capabilities of the network. Due to the capability of the support-vector-network to handle non-separable spaces, the technique was compared to the potential of neural networks at the time.

SVMs may be implemented for biomedical image segmentation as a number of useful features can be extracted from an image, such as signal intensities, geometric information, and textures. A feature vector may be created from a subset of these attributes both for pathological and control cases. The SVM would then attempt to create the optimal hyperplane between each class from the given features to classify the inputs into the respective classes. Nonetheless, manually carrying out feature selection and extraction is a more cumbersome approach compared to more recent technology such as Convolutional Neural Networks which implicitly perform the process.

2.1.5 | Convolutional Neural Networks

The use of Convolutional Neural Networks (CNN) for images, speech, and time-series was explored by LeCun et al. (1995) as a comparison with popular techniques at the time, such as fully-connected systems. The authors claimed that such an architecture could eliminate the trend of pattern recognition systems at the time to use hand-crafted feature extractors. Rather, the CNN would learn feature extraction through backpropagation over a number of iterations. The example of handwriting classification was used, where convolutional networks would have the upper hand against fully-connected systems, as the latter would have to be quite comprehensive to learn how to be shift-invariant, which could also in turn affect the visibility of the features to be gleaned.

Another argument in favour of CNN made by LeCun et al. (1995) was the usefulness of a 'local' receptive field. The authors used the example of images, where pixels which are in close proximity to each other generally share some form of related context. As mentioned by LeCun et al. (1995), fully-connected systems at the time ignored the topology of the input, posing the threat of losing such local information which would be useful for classification. Putting this into the scope of this project, consider an MRI slice with a glioma. Voxels or pixels surrounding the tumour core may be correlated to each other as they form part of the pathology. Convolutional networks force the hidden units to have local receptive fields which would prevent such correlations from being ignored. An example of a CNN from this era is Le-Net5 by LeCun et al. (1998), a network capable of identifying hand-written characters for document recognition. A visualisation of the network architecture may be observed in Figure 2.7.

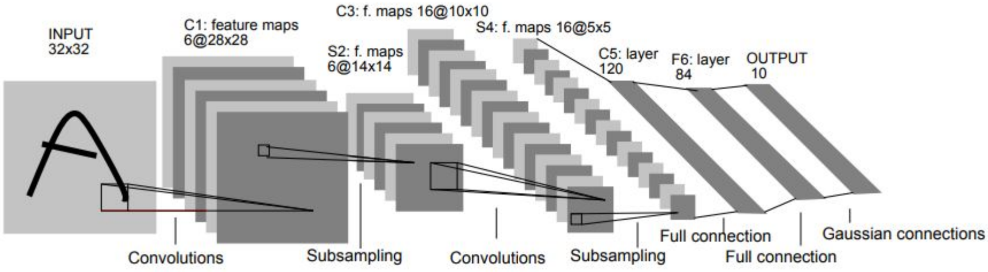


Figure 2.7: Architecture for LeNet-5 (LeCun et al., 1998).

Following the pipeline, the input starts as a 32 × 32 2D image. The initial convolution uses 6 filters and pads the image, adding a depth of 6, with the spatial resolution dropping to 28 × 28. A pooling operation is applied which takes salient features from the convolution, and halves the spatial resolution once again to 14 × 14. The process is repeated up to the full connection *C5* where a kernel with the same spatial resolution as the processed image (5 × 5) is used. As mentioned previously, the network described above is a fairly primitive approach by today's standards, however the techniques used lay the foundation for the current state-of-the-art CNNs. Whilst network architectures have become much more robust, the use of an encoder pathway to extract feature maps, and pooling/subsampling to obtain salient features are still commonly used. Some modern iterations of CNN relevant to this project will be described in the sections to follow; namely ResNet, DenseNet, U-Net, and U-Net++.

2.1.5.1 | ResNet

With further improvements in hardware since the first iterations of CNNs, the possibility of deeper neural networks became more of a reality. For problems in visual recognition like semantic segmentation, deeper networks such as those proposed by Girshick et al. (2014); Long et al. (2015) showed positive results. However, experiments in Girshick et al. (2014)'s proposal of R-CNN showed that when they removed 94% of the parameters at one point, the drop in accuracy was inconsequential. This is noteworthy as such a reduction in network size reduces hardware constraints, thereby also improving performance. This is also bearing in mind that R-CNN is still a much slower architecture than its improvements Fast-R-CNN by Girshick (2015) and Faster-R-CNN proposed by Ren et al. (2015).

He et al. (2016) went a step further and observed that stacking layers would eventually lead to a decrease in network performance. Their observations showed that the network loss would stagnate, before beginning to decrease. He et al. (2016) state that the reasons behind the decrease involved the network attempting to fit some form of underlying mappings between successive layers. Ideally, deeper layers would continue learning from their shallow counterparts, however this was not the case since the accuracy was saturating or even dropping. The authors propose making the network fit a residual mapping; observe Figure 2.8 which shows one such residual block.

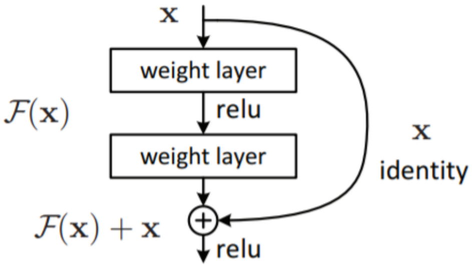

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.8: Example of a residual block (He et al., 2016).

With $H(x)$ as the 'underlying mapping' mentioned previously, the authors propose the mapping shown in Equation 2.2.

$$F(x) = H(x) - x \tag{2.2}$$

The formula may then be re-arranged as in Equation 2.3 to obtain $H(x)$:

$$H(x) = F(x) + x \tag{2.3}$$

Translating the block in Figure 2.8 to CNNs, the mapping for $H(x)$ may be obtained through the use of skip-connections, such as the identity link in the image. An element-wise addition may then be used to combine the identity with the output from the non-linearity and weight layers. He et al. (2016) state that optimising the residual function in this way was found to be simpler than attempting to optimise a network using conventional stacked layers. Their results were also impressive as these networks allowed for a reduced number of parameters in the model despite having very deep architectures. The results were competitive enough to achieve first place in the ILSVRC 2015 classification task. A comparison between training ResNet and a plain stacked network may be observed in Figure 2.9, showing that ResNet converged to a lower error with additional layers in place.

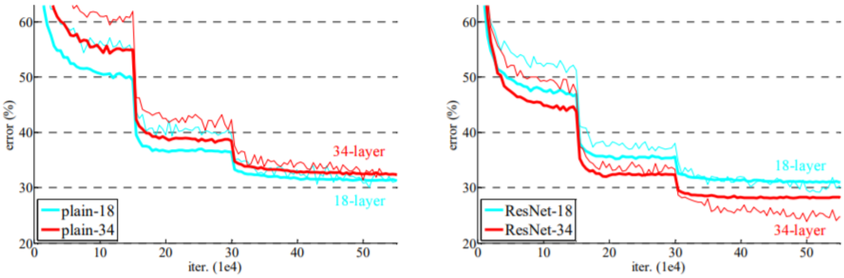


Figure 2.9: Deep neural net training - stacked layers vs. residual layers (He et al., 2016).

2.1.5.2 | DenseNet

In Szegedy et al. (2016)'s proposal of the Inception v3 model, one of the important factors addressed in the paper is efficient grid size reduction, accomplished through layer-to-layer concatenations. Inspired by the performance of networks such as ResNet (He et al., 2016) on very deep architectures, Huang et al. (2017) would later propose their dense convolutional network, DenseNet. Whilst ResNet (He et al., 2016) employed element-wise additions at different points in a model, an alternate approach is to make use of concatenations between layers, as shown in Figure 2.10.

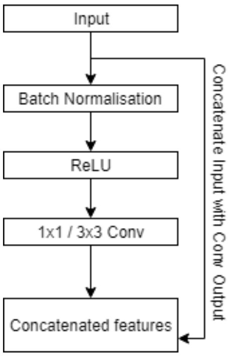


Figure 2.10: Composite Function in a Dense Block.

Every layer in the network is connected to the other layers in a feed-forward fashion. This allows for the feature maps of a node to be moved to every other layer in the system. As an example, a layer in the middle of the network would have received every output from the shallower layers, and would propagate its own output to every successive layer. Figure 2.11 from DenseNet's CVPR2017 presentation shows how the feature maps are concatenated for a forward pass within a dense block.

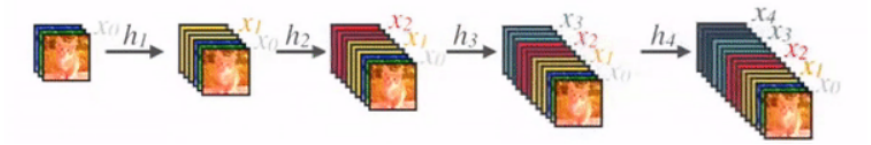


Figure 2.11: Visualisation of feed-forward propagation in DenseNet block.

The network is divided into multiple dense blocks as in Figure 2.11 which share the same spatial dimensions. Huang et al. (2017) claim that they decided on concatenating the feature maps rather than element-wise additions as in ResNet (He et al., 2016), as the latter may impede gradient flow during network training. Huang et al. (2017) also claim that concatenations create a diverse feature space, allowing for superior performance. Reusing feature maps allowed for smaller filter resolutions, boosting the parameter efficiency of the network. The final new feature in DenseNet was the growth factor, represented by the parameter k. The growth rate was a hyperparameter which controlled the expansion of the filter resolution during training.

Whilst the standard DenseNet was already comparable to the state-of-the-art at the time, Huang et al. (2017) also proposed DenseNet-B which made use of a bottleneck layer. A 1×1 convolution was used here as a means of dimensionality reduction in the feature space of the network. This operation was carried out normally following batch normalisation and the ReLU nonlinearity. The bottleneck was placed before every 3×3 convolution to reduce the number of feature maps before the convolutions with the larger kernel. For $k = 12$ the number of parameters from standard DenseNet (40 layer) to DenseNet-B (100 layer) dropped from 7M to 0.8M. Despite being 60 layers deeper, DenseNet-B reduced parameters by close to a factor of 10 and also obtained superior results to standard DenseNet. Regarding overall performance, the network obtained state-of-the-art results on the competitive datasets CIFAR10, CIFAR100, SVHN, and ImageNet.

2.1.5.3 | U-Net

Compared to other popular image processing techniques such as face recognition, biomedical image segmentation presents some additional challenges due to the nature of the data. Training images are generally limited to select sources, owing to the smaller number of public datasets for problems such as tumour segmentation. Whilst there are a number of repositories and challenge datasets available, the annotation strategies and sequence types used tend to vary substantially. The nature of segmentation as a task is also difficult, as predicting solely a class label for a given image is sufficient only for simpler tasks, such as detecting whether an MR image contains a pathology. Segmentation requires that every pixel in the image is assigned a classification, with many problems having multiple foreground classes. Due to the spatial resolution of these images, care must be taken not to encumber a network with too many parameters.

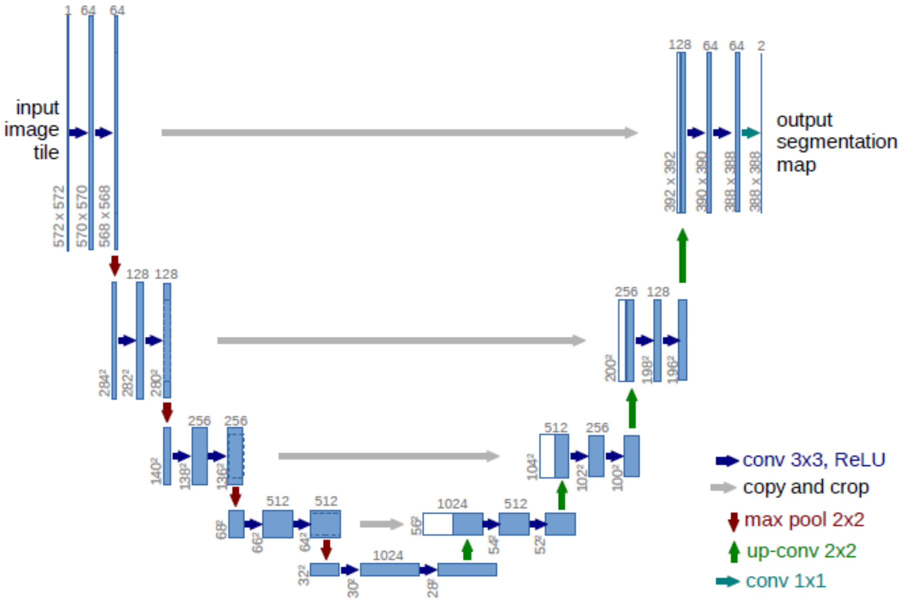


Figure 2.12: U-Net architecture (Ronneberger et al., 2015).

For this purpose, Ronneberger et al. (2015) designed U-Net, shown in Figure 2.12. The network is composed of two sequential pathways, one intended for aggregating context information, and the other for decoding this context with respect to the input image. The initial pathway resembles the architecture of a standard CNN; an input image is downsampled, with features being extracted using convolution and pooling operations to refine the filter maps. The resolution of the feature maps is actually quite high in this model, allowing U-Net to obtain a sizeable context from the input image.

The second half of the 'U' initiates the decoding pathway. The images are upsampled, and information from the encoding pathway is propagated using concatenation via skip connections. Skip connections aid the network by providing context to the filter maps in the decoder since features from the encoder are being re-introduced at this stage. Contrary to standard CNN architecture there is no fully connected layer, as the output is generated via a segmentation map which contains the combined data from the entire pipeline. As Ronneberger et al. (2015) state, the segmentation maps contain only those pixels which have full context in the input image, a useful feature for segmentation tasks. The symmetric encoder-decoder architecture produces the "U-Shape" shown in Figure 2.12.

An additional problem which frequently occurs in biomedical image segmentation is when classes overlap, such as a non-enhancing and enhancing tumour core section in an MR image which are touching. Since these are members of overlapping classes, they may be mislabelled if the boundaries between them are not considered. For this reason, Ronneberger et al. (2015) made use of a weighted loss function which gives high values to separations. This forces the network to learn separations such as cell boundaries, avoiding the aforementioned problem. The generalisation capabilities of the network were also enhanced through use of data augmentation techniques. Ronneberger et al. (2015) implemented elastic deformations to make U-Net invariant to shifting and rotation. Elastic deformations follow a process in metallurgy where the shape of the metal changes, generally by a shear (Stráský et al., 2018). Ronneberger et al. (2015) claim that this is especially useful for biomedical images since abnormalities would often naturally take on forms similar to these deformations. Thus, the network could potentially be learning how to classify abnormalities whilst also avoiding overfitting.

2.1.5.4 | Residual U-Net

At the time, U-Net was very competitive and Ronneberger et al. (2015) claimed that U-Net won the ISBI 2015 Cell Tracking competition by a large margin. Since it was tailor-made network for biomedical image segmentation, U-Net would see a multitude of adaptations. One of these adaptations would be the residual U-Net by Isensee et al. (2017), shown in Figure 2.13.

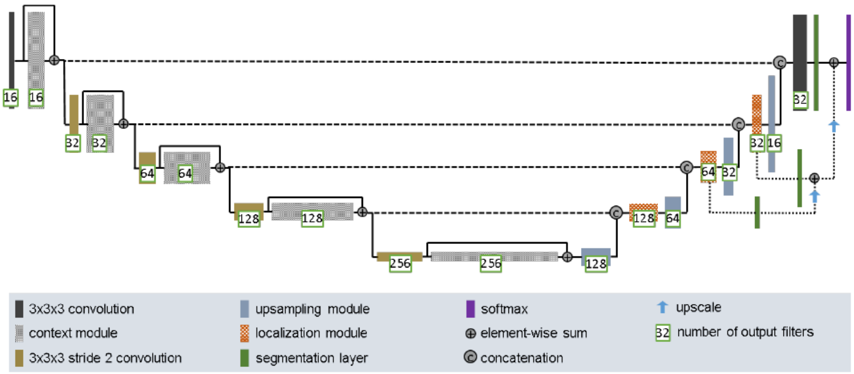


Figure 2.13: Residual U-Net architecture (Isensee et al., 2017).

This model makes a number of modifications to the standard architecture, which will be discussed further in Section 3.1.5.2; this explanation will focus on the changes brought about by the residual blocks. Element-wise additions are used throughout the encoder portion of the network to combine features passed pre-convolution and activation with those at the end of a convolutional block. In theory, this should provide the benefits mentioned by He et al. (2016) in refining the feature context of the network. Further details about this specific model's performance and history will be provided in Section 2.2.

2.1.5.5 | U-Net++

As previously discussed in Section 2.1.5.2, an additional means of enhancing feature map propagation in a network is through the use of dense blocks. Zhou et al. (2018c) propose a U-net architecture making use of dense blocks, as shown in Figure 2.14.

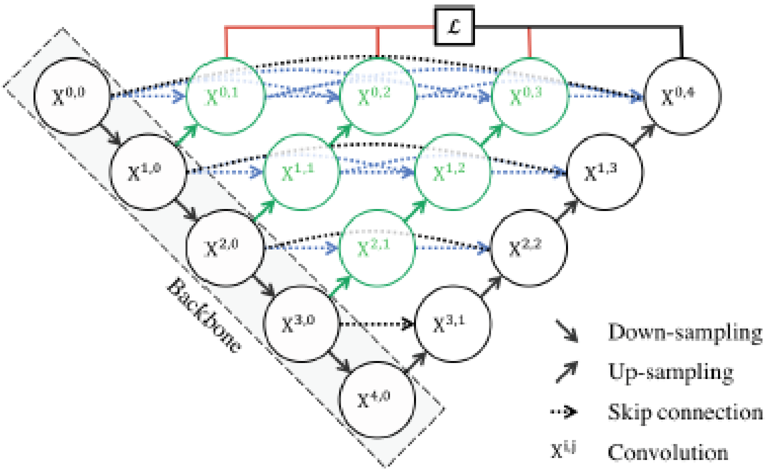


Figure 2.14: U-Net++ architecture (Zhou et al., 2018c).

The standard encoder-decoder structure of U-Net was maintained, however this was combined with additional upsampling layers along the skip-connections between the encoder and decoder halves of the network. This builds upon the convention of standard U-Net where a concatenation connects the encoder to the decoder at each level. The benefit of making the concatenations increasingly richer as the network gets shallower is that the final prediction would be produced from a highly refined feature context. Rather than just having a concatenation from the first convolutional block of the encoder and the previous decoder layer upsampled, the concatenation now includes upsampled and concatenated features from every layer of the encoder. In addition, the skip-connections past the first level also receive feature maps using information from the previous levels. This results in nested structures similar to smaller U-Net networks within the model.

This merger of architectures effectively combines the rich feature space of a dense network with the low-parameter requirements of U-Net. Whilst the number of layers is definitely higher in this architecture, Zhou et al. (2018c) claim that the number of parameters is quite similar to Ronneberger et al. (2015)'s U-Net and the wide variant of U-Net which uses larger feature channels. This comparison is also made on the grounds that the same number of convolutional kernels are used in both models. Zhou et al. (2018c) used a standard U-Net and a 'wide U-Net' as baseline models to compare with U-Net++. The latter structure implements a conventional U-Net architecture with the base number of filters starting from 35 rather than 32. The Jaccard Index (also known as Intersection over Union or IoU) was used to evaluate U-Net++, outperforming standard and wide U-Net by an average of 2.8 to 3.3 points of IoU.

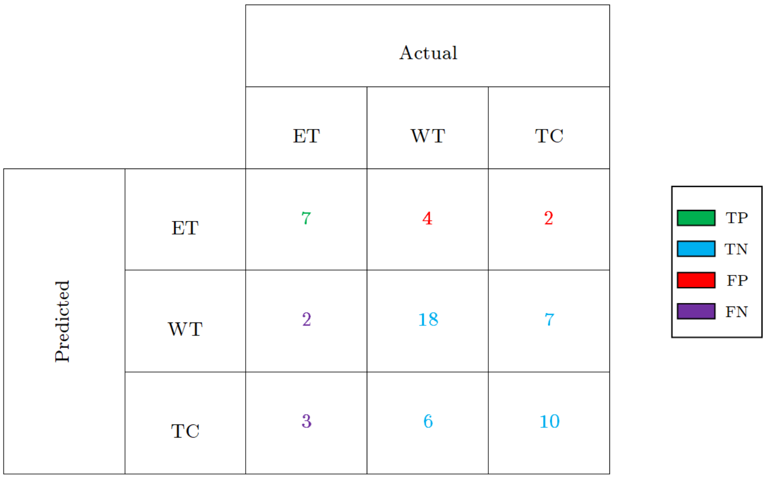
2.1.6 | Evaluation Criteria

The evaluation for this project was conducted on the CBICA online portal. At the time of writing, the site allows users registered for the challenge to upload predictions for either the training dataset or the separate validation dataset for 2018 or 2019. The output file is a spreadsheet with computed measurements for each of the uploaded samples, and the means, median, standard deviation and quartile information for all uploaded samples. This evaluation is fairly comprehensive, and includes the Dice Coefficient, Sensitivity, Specificity, and Hausdorff Distance for the whole tumour (WT), tumour core (TC), and enhancing tumour (ET). The Dice Coefficient acts a direct comparison between the predictions and ground truths, and is also the most used metric in validating medical volume segmentations (Taha and Hanbury, 2015). The formula for the Dice Coefficient computes the similarity between a prediction and expert ground truth as shown in Equation 2.7.Miller et al. (2020)

$$DSC = \frac{2(Y \cap \hat{Y})}{(Y + \hat{Y})} \tag{2.4}$$

Where Y and $\hat{Y}$ represent the ground truths and predictions respectively. The score fluctuates between 0 and 1, where a score of 1 represents the output and ground truth being equivalent.

Sensitivity and specificity are useful metrics for brain tumour segmentation since they evaluate a system's capability to correctly predict true positives and negatives. To understand the context of positive and negative predictions for brain tumour segmentation, one may consider the example confusion matrix shown in Figure 2.15.



		Actual		
		ET	WT	TC
Predicted	ET	7	4	2
	WT	2	18	7
	TC	3	6	10

Figure 2.15: Confusion Matrix for multiclass tumour segmentation. Note that the TP, TN, FP, FN markings are for the ET class.

Minimisation of false positives and false negatives avoids the possibility of a wrong diagnosis. A good example would be healthy voxels wrongly classified as the enhancing tumour, or the opposite case where pathological voxels are predicted as control samples. Sensitivity and Specificity may be calculated using Equation 2.5 where TP and FP refer to true and false positives, and TN, FN true negatives and false negatives.

$$Sensitivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \tag{2.5}$$

Whilst both sensitivity and specificity are important evaluation metrics for medical data, one should also go a step further and differentiate between the two. A low specificity implies an underlying inclusion of false pathological tissues in a prediction. Whilst this is a considerable issue, once a pathology is brought to light there will generally be follow-up examinations to check the patient's condition further. However, it is arguable that false negatives are more dire since it is an omission of pathologies in a scan. False negatives in a multiclass segmentation problem could result in scenarios where diseased tumour voxels are predicted as a different class. The latter may result in less than ideal patient follow-up such as surgery or radiotherapy treatment. Thus, if a decision which has to involve a trade-off between specificity and sensitivity has to be made, sensitivity should be given a higher priority. Sensitivity scores are also generally much lower than specificity in this task, as discussed further below in Section 4.2.

The Hausdorff Distance is a measure showing the maximum shortest distance between subsets, measured in *mm*. Thus, fluctuations in Hausdorff distance across different models may be much larger than deviations for the other measurements. A straightforward definition would be that it is the measure of the largest class segmentation error (Karimi and Salcudean, 2019). The one-sided Hausdorff distance between two sets A and B may be calculated as per Equation 2.6, where points a and b are members of A and B respectively.

$$D_H(A, B) = max_a min_b ||a - b|| \tag{2.6}$$

The minimum distance from each point of the first set is computed over all points in the second set. The maximum value of these minimum distances is then used to represent the largest error, in this case for the worst segmentation predictions in relation to the expert ground truths. Since this is a distance measure, a lower Hausdorff Distance score indicates an improvement. It is also important to mention that the Hausdorff Distance used by the BraTS portal takes the 95^{th} percentile rather than the maximum distance. A possible reason for this is that taking a non-fractional Hausdorff Distance may heavily punish the final score if even one severe outlier is present.

2.2 | Literature Review

This section will discuss a number of brain tumour segmentation techniques, ranging from 'classical' machine learning approaches such as SVM and clustering to the current state-of-the-art techniques, including 3D CNN, U-Net, and ensemble methods. The structure follows categories of techniques as a timeline of how the state-of-the-art has developed in the domain. Section 2.2.1 will discuss older ML techniques, mainly studies making use of various forms of clustering and SVM. Very often, these approaches implement custom feature extraction pipelines based on raw image properties, leading to interesting solutions. The ability to construct models without the need for labelled data in some instances is also a very strong advantage. Section 2.2.2 describes several deep learning techniques used for biomedical image and brain tumour segmentation, leading up to the more recent models which were inspired by these works. High priority has been given to studies which were evaluated on BraTS datasets, as this not only makes the results more sound, but also provides a good baseline for uniform comparison with other models.

2.2.1 | Conventional Machine Learning

As described by Lundervold and Lundervold (2019), initial research on brain tumour segmentation consisted of a number of contributions making use of unsupervised machine learning approaches. Open data had not yet become the phenomenon it is at the time of writing, and the BraTS challenge did not exist. Whilst brain tumours are not an uncommon form of cancer, labelled data (especially online) was definitely not as available as at the present time. Unsupervised techniques proved useful in this scenario as they did not rely on having expert ground truths alongside an MR image dataset, rather intrinsically creating relevant groups of salient features. An example of popular unsupervised techniques in the past is clustering, which will be discussed briefly in Section 2.2.1.1.

2.2.1.1 | Clustering

Fuzzy C-Means (FCM) clustering is one such unsupervised method which could prove useful for finding unknown substructures such as tumours in unlabelled MR images. Phillips II et al. (1995) explored the use of FCM for segmentation of GBM, a form of HGG. The intensity values for the T_1w, T_2w, and PD modalites for each pixel were used to construct feature vectors. It is noteworthy that neighbourhood information outside of these individual values was not included, as at the time this had 'increased computation time but not yielded useful results' (Phillips II et al., 1995). The authors claim that FCM was chosen for the segmentation task as fuzzy clustering is compatible with the overlapping gray-scale intensity gradients caused by various forms of noise in the MR image. Whilst there were some setbacks in this study attributed to the imperfections in the dataset, the method displayed both healthy and pathologic tissues in the data successfully.

Wu et al. (2007) later explored the use of K-means clustering for tumour detection in MRI. The approach involved a number of operations spanning multiple stages. Initially, grayscale MR images were converted into the RGB colourspace, and then converted into CIELAB format. The CIELAB representation is defined by the variables $L^*a^*b^*$, where L^* is a luminosity coefficient, and a^*, b^* are chromaticity along the red and blue-yellow axis. All three of these variables were leveraged; luminosity was used as a feature for histogram clustering, and the chromaticity coefficients were used for K-means clustering.

Chaira and Anand (2011) also found the use of colourmap conversions from grayscale to CIELAB format to be useful in their study on brain tumour detection and segmentation. This work also made use of an 'intuitionistic' FCM approach. The pipeline is very similar to the study by Wu et al. (2007), with the colourspace conversion being the first step in the process. In Chaira and Anand (2011)'s study however, the next step was to generate multiple regions of interest using intuitionistic FCM. This variant of FCM builds upon the principles of standard FCM where the degrees of membership and non-membership are not discrete. A hesitation coefficient (or intuitionistic degree) may then be calculated by subtracting the membership and non-membership function from 1. This theory was then applied to generate intuitionistic fuzzy images. Histogram thresholding was applied to the generated images, reducing the number of regions to those containing clusters with tumour tissues. Finally, a number of edge templates were applied to perform edge detection on the ROI to be superimposed on the original image. The results show that the technique segmented clot and haemorrhage regions in the brain well compared to standard edge detection approaches.

Another approach examining multiple unsupervised techniques was explored by Juan-Albarracín et al. (2015). K-means, Fuzzy K-means and Gaussian Mixture Model were all evaluated on the GBM samples from the BraTS 2013 Test dataset. A Gaussian Hidden Markov Random Field (MRF) was also evaluated separately as a structured unsupervised approach. The samples were initially fed through a preprocessing pipeline consisting of denoising, skull-stripping, and bias field correction. A super-resolution (Manjón et al., 2010) technique was also used to regularise any intensity gradients in the images. Features were extracted from the difference image between the T_1ce and T_1 images. Histograms were computed in the local 3D neighbourhoods of each voxel, from which metrics such as the mean, skewness, and kurtosis were extracted to be fed into the clustering and Gaussian models. The results obtained from histogram clustering and K-means were then combined to produce the final segmented image. The results shown illustrate that the method was capable of extracting the lesions from the input images.The best results were obtained by the MRF approach, scoring 0.72, 0.62, and 0.59 for the whole tumour, tumour core, and enhancing tumour. The Gaussian Mixture Model was a close second, obtaining scores of 0.69, 0.60, 0.55 for each category.

Spectral clustering is a method which categorises data into clusters through the eigenvectors and eigenvalues of the Laplacian matrix. In the context of MR images, one would first generate a graph representation of samples from the dataset from which the Laplacian could be calculated. Angulakshmi and Priya (2018) explored the use of spectral clustering combined with superpixels for segmentation of brain tumours from MRI. The pipeline for Angulakshmi and Priya (2018)'s approach started with a non-local mean filter to redistribute image pixels with mean neighbouring intensity values. For this study, superpixels represent local salient regions within the image with valuable information for segmentation.

Localisation of these superpixels was performed by partitioning the images into blocks and using central tendency values such as the mean, mode, and median, representing central points in the filtered image distribution. This part of the process is essential, as spectral clustering could then be performed only on the extracted superpixels as the ROI. The use of this smaller, more salient set of data for spectral clustering was done with the intention of minimising a drawback of the technique which is the creation of dense similarity matrices (Angulakshmi and Priya, 2018). The final step involved identifying the tumour blocks within the superpixels by using Local Binary Patterns (LBP) as a feature extractor, applying a binary threshold to pixels depending on their value relative to the superpixels. The approach was tested on the BraTS 2012 dataset and obtained moderately positive scores on the synthetic part of the dataset, with poorer scores on the real patient data.

As mentioned above, one of the advantages of unsupervised segmentation is the ability to localise and delineate brain tumours using raw features from MR images rather than relying strictly on labelled images. One should bear in mind, that although datasets such as the BraTS datasets are verified by multiple experts, there may be other data sources which have more lax enforcement of annotation quality. Thus, it may be sensible to use unsupervised approaches in these cases.

Conversely, there are also a number of disadvantages to consider when using unsupervised segmentation techniques in medical imaging. The morphology of tumours as well as the number of pathologies in a scan are not properties which can be predetermined. Tumours may appear anywhere within the brain and have almost any form, size, and contrast (Havaei et al., 2017). Since these methods do not have prior knowledge about the texture/shape of the tumour through labels, one would have to generalise a pathology's structure. Tumours may also manifest within multiple regions of the brain, which would present another issue when performing unsupervised classification. Having the ground truths available as in supervised methods such as deep neural networks could mitigate this issue since the exact tumour labels would be used to extract features corresponding to each tumour section whilst training the model.

One should also mention that unsupervised methods have a greater sensitivity to noise, especially methods relying solely on signal intensity. Mitigating these disadvantages is possible to an extent, using techniques such as MRI intensity correction (Vovk et al., 2007) to highlight salient intensities. Another commonly used approach is to standardise the input dataset by mean and standard deviation. Finally, separation of the brain from the skull (skull-stripping) using a mask as in Juan-Albarracín et al. (2015); Menon and Ramakrishnan (2015) is also a viable solution to reduce misclassifications, if it has not been performed already prior to dataset distribution.

2.2.1.2 | Support Vector Machines

Moving on to supervised ML approaches, a one-class Support Vector Machine (SVM) approach was proposed by Zhou et al. (2006) for segmenting brain tumours from clinical MR images. The data for this study consisted of 24 total 512 × 512 slices and was obtained from 5 patients. Zhou et al. (2006)'s technique is a semi-automatic approach as a user would first select a tumour sample using a bounding box for input to the SVM. The hyperplane and SVM parameters were then trained on the pre-selected tumour regions, using an RBF kernel. At the tail end of the process, the learned hyperplane and SVM parameters were applied to test images to localise the tumour region.

Zhou et al. (2006) claim that no explicit feature extraction was required in this approach due to the generalisation capacity of the SVM. In truth, this is a claim which is quite specific to this study. Whilst it is true that the seed points provided by the users should localise the tumour quite well, feature extraction could still potentially provide additional information for operations such as eroding/dilating the user selection. It is noteworthy that as the title of the study states, this approach performed one-class segmentation solely of the whole tumour, and did not segment the mass further into substructures. The results were compared to semi-supervised fuzzy clustering and found to perform an overall better segmentation. Manually traced ground truths were provided by radiologists for evaluation of the method. The one-class SVM obtained an average percentage match and correspondence ratio of 83.5% and 78% respectively.

Luts et al. (2007) later implemented a Least-Square SVM (LS-SVM) for the task of multi-class brain tumour segmentation in MRI and MRSI (Medical Resonance Spectroscopy Imaging). Luts et al. (2007) claim that the motivation for combining MRI and MRSI was that this had yielded positive results for other classifiers. Additionally, the observations related to feature selection in the previous study by Zhou et al. (2006) were contrasted here as a specific pipeline purely for feature selection was created. The latter was carried out using a combination of three statistical tests and an automatic relevance determination algorithm (ARD), which is a method capable of deriving a relevant subset of features from a large set of irrelevant variables (Wipf and Nagarajan, 2008).

Once the feature selection process had been established, the classification task was the next step. It was decided to split the tumour groups into a total of 10 classes, which included healthy tissue, CSF, and the rest separate tumour types including Grade IV gliomas. It was decided to use pair-wise classification for each pair of classes, resulting in a total of 45 classifiers. Whilst this combination resulted in a massive number of models, Luts et al. (2007) claim that they were simple enough to remain computationally efficient (compared to techniques like exhaustive search and grid search). A comparison of the results obtained using all of the features vs. only a subset showed that the selected set obtained better scores. The approach was then evaluated against Linear Discriminant Analysis (LDA), and found to surpass LDA when discriminating between individual tumour types and different tumour growth stages.

Zhang et al. (2009) later proposed a multi-modal, multi-kernel SVM approach for brain tumour segmentation. The sequence types used as input were T_2, PD and FLAIR. The process was initiated by selecting a number of points chosen inside and outside the tumour region as training samples. Separate feature selection processes and multi-kernel SVMs were then curated for each of the three sequence types. Zhang et al. (2009) claim that the reason exclusive feature selection pipelines and SVMs were used was that each sequence had specific parameters which a single-kernel SVM could not accurately reflect. The enrichment of kernels used in multi-kernel learning was also stated to improve the result estimation (Zhang et al., 2009).

Once all three pipelines had been processed, the combined intersection would then produce the final tumour class. However, it was observed that some parts of the tumour were missing following the intersection operation. This was likely due to noise in the original image, and Zhang et al. (2009) proposed a region growing approach to pad the missing pixels. The distance between the missing points and tumour contour, as well as maximum likelihood measures were used for this step. The restoration of the tumour was processed by iteratively applying morphological dilation operations until the tumour contour stopped changing. The work was evaluated using true positives, false positives, false negatives and the sum of false positives and false negatives as a 'total error' metric. The results obtained were reported by the authors as being satisfactory.

One may notice that most of the studies described above which made use of SVM are quite dated. A more recent approach combining SVM and FCM was proposed by Sriramakrishnan et al. (2019). The pipeline made use of probabilistic local ternary patterns (PLTP), which are a probabilistic variant of LTP, a ternary variation of LBP which thresholds data into -1, 0, or 1. The SVM was used in the first part of the pipeline; performing feature extraction and classifying the images into a normal or pathological block using FLAIR slices.

The second step used T_2w and FLAIR slices which were classified by the SVM as diseased. Both modalities were then passed through FCM to obtain the ROI which could contain a tumour. Two paths followed thereafter; FLAIR was split solely into tumour components, whilst T_2 was split into CSF and tumour. The CSF was voided, however the T_2 image tumour appeared to have parts missing, as observed in the previous study by Zhang et al. (2009). The solution used to fix these missing parts was extracting the union of the T_2 and FLAIR tumour components using a logical OR operation. The largest connected regions of the union were used to extract the tumour.

With the tumour region obtained, the final step was to segment substructures within the tumour. This is where the PLTP section of the pipeline began. Rather than using intensity values directly, the probability distribution of the image was used. Upper and lower boundaries were computed by using a 3×3 mask on the histogram and applying thresholding. Intensity values were then calculated for the respective center bin in the histogram, with this process repeated for all bins. Sriramakrishnan et al. (2019) showed side-by-side results of LTP and PLTP, where LTP only extracted outlines of the entire brain, unsuccessfully delineating the tumour. PLTP however applied a more finely-grained intensity distribution and the tumour was visible even when unsegmented. The project was evaluated on the BraTS 2013 and 2015 training datasets, making use of T_1ce, T_2, and FLAIR sequences. The WT, TC, and ET Dice Scores obtained were 0.76, 0.53, and 0.58 for 2013, and 0.81, 0.49, 0.47 on the 2015 dataset.

2.2.1.3 | Random Forest

Random Forests are another popular classifier for MR image segmentation. Tustison et al. (2015) made use of a Gaussian Mixture Model combined with a 2-stage Random Forest for segmenting brain tumours. The feature sets used for the experiment were the intensity, geometry, and asymmetry of each image. The random forests were used to refine MRF segmentation, effectively combining supervised learning with probabilistic segmentation. Tustison et al. (2015)'s approach was evaluated on the BraTS 2013 dataset using the Dice coefficient. The random forest obtained Dice scores of 0.87 and 0.78 for the whole tumour and tumour core evaluations, with a score of 0.74 for the enhancing tumour segmentation.

Another approach by Soltaninejad et al. (2018) combined Random Forests with texture features for supervoxel classification. This approach used Gabor filters to generate histograms of textons, which are salient features in an image. This feature set was then used as input for the random forest to categorise each supervoxel into either an edema, tumour, or healthy brain tissue. Soltaninejad's technique was also evaluated on the BraTS 2013 dataset, with positive results on the multimodal set of 30 MR images.

2.2.2 | Deep Learning

As one may observe when browsing literature on brain tumour segmentation, the popularity of traditional ML methods and unsupervised approaches has waned in recent years, with the current trend shifting towards robust deep networks (Lundervold and

Lundervold, 2019). This has also been observed when examining the past years' submissions to the BraTS challenge, where the main methods are mostly CNN and U-Net variations (Ghaffari et al., 2019).

2.2.2.1 | Standard CNN Approaches

The departure from traditional Machine Learning techniques being the gold standard came with the rise in popularity of CNNs, which presented many possibilities for medical image segmentation. Ciresan et al. (2012) developed a sliding-window CNN which outperformed every other competitor for the ISBI 2012 EM Segmentation Challenge. The approach took raw intensity values as input from a window centered about individual pixels in the input. This window was also designed to cater for pixels which were close to an image border, with the window's pixels mirrored across the border to pad the otherwise empty data.

Ciresan et al. (2012)'s model made use of a stack of convolutional layers, with max pooling and fully connected layers, generating a 1D feature vector as output. A softmax activation function was used to generate the probabilities of whether individual pixels in the image belonged to a 'membrane' or 'non-membrane' class. Data augmentation was also viable for this implementation, since changing the rotation of the target classes would create realistic synthetic samples. Thus, the authors mirrored and rotated samples by 90 degrees in either direction to create additional training examples for the network.

Ciresan et al. (2012) claimed that another defining feature of this network differing from others at the time was the number of feature maps per layer being much larger, as well as the number of weights and connections. Naturally, this also presented some drawbacks since the size of the network would increase. Another setback of the deep neural net used for this experiment is that the model had a characteristically high variance in its output. This meant that networks with varying architectures would potentially have disparate results. To stabilise this variance, Ciresan et al. (2012) averaged the evaluation of multiple networks, which ended up having overall greater results than any one individual model.

Similarly to how Ciresan et al. (2012)'s approach was a great contribution to biomedical image segmentation, Havaei et al. (2017) proposed one of the most influential experiments using deep neural networks for brain tumour segmentation. Their architecture consisted of two pathways, making use of 7×7 and 13×13 feature map resolutions respectively. The reasoning behind the two paths was to simultaneously capture a 'local' and 'global' context from the input images taken from BraTS 2013 dataset. Havaei et al.

(2017) also ensembled two CNNs, where the output probabilities from the first network were used as input for the second model using concatenation layers.

The pre-processing pipeline made use of bias-field processing and standardisation by mean and standard deviation to counter intensity inhomogeneities in the dataset. Havaei et al. (2017) also removed the top and bottom 1% of intensities from the input images. The data imbalance present in brain tumour segmentation was also observed by Havaei et al. (2017). For the BraTS 2013 dataset, it was observed that 98% of the voxels were healthy. The remaining 2% of pathological voxels were split into 1.1% of edema, 0.12% for non-enhancing tumour, and 0.38% for the enhancing tumour. To counter this, Havaei et al. (2017) constructed the patch dataset using equiprobable labels. A second training phase was then applied solely on the model's output layer, keeping the other layers' kernels fixed. The goal behind this technique was to capture the class diversity and re-calibrate solely the output probabilities correctly. L2 and L1-regularisation and dropout were used to counter overfitting. A noteworthy observation from this experiment is that increasing the number of layers in the model showed no considerable improvements. The project was evaluated on the BraTS 2013 test dataset, with very competitive WT, TC, and ET Dice scores of 0.88, 0.79, and 0.73.

Another formidable implementation for brain tumour segmentation was proposed by Pereira et al. (2016). Separate CNN pathways were employed, separating high grade and low grade gliomas, with different architectures and normalisation configurations for each path. The authors claim that the reason for this was the notable disparities between the morphology of each grade of tumour. The pathway for low grade gliomas is slightly shallower as the addition of further convolutional layers provided no improvements to the output of the CNN, much like the observations noted by Havaei et al. (2017). This research is also notable for its use of small convolutional kernels, inspired by Simonyan and Zisserman (2014)'s research on VGGNets. Simonyan and Zisserman (2014) showed that using small convolutional kernels (3×3) would pave the way for deeper networks which could perform at higher levels. Pereira et al. (2016) reiterate this by stating that using smaller kernels allowed for the inclusion of additional layers. This implies that the receptive field between small and large kernel setups can be equivalent without introducing as many weights/parameters as through using larger kernels.

The preprocessing methods used in this study are also noteworthy. Pereira et al. (2016) implemented Tustison et al. (2010)'s N4ITK to correct intensity inhomogeneities between identical classes of brain tissue in the data. Another preprocessing problem Pereira et al. (2016) address is heterogeneity in MRI sequences which are taken by different scanners. There is an underlying problem as the same MRI sequence may have varying intensity values both intra and inter-patient. This creates inconsistencies in data being inserted into a model since sequences which should be uniform across all patients may be displaced. Pereira et al. (2016) claim that this variance may also occur on a single patient if the image is acquired at different time points. To standardise values across all sequences, an algorithm developed by Nyúl et al. (2000) was considered. The method learns 'intensity landmarks' from a sequence and after training, applies a linear transformation which standardises each sequence. This system achieved first place in the 2013 iteration of the BraTS challenge with WT, TC and ET Dice scores of 0.88, 0.83, 0.77, and second place in BraTS 2015 using the same model with Dice scores of 0.78, 0.65, and 0.75.

DeepMedic by Kamnitsas et al. (2016) leveraged a 3D CNN for brain tumour segmentation. This study is notable as it performed experiments on the BraTS 2015 and 2016 dataset. Preprocessing involved the use of standard feature scaling on each scan, with every image being subtracted by its mean and divided by the standard deviation. The CNN used was 11-layers deep, making use of two parallel-processing pathways at different resolutions, a similar concept to the dual paths in Havaei et al. (2017). The authors claim that this allowed for a large receptive field being available for final classification without excessive computational overhead. Another technique to minimise training time was once again the use of small convolutional kernels, which allowed for the expansion of CNNs whilst keeping the number of model parameters at runtime low, much like Pereira et al. (2016).

As discussed in Section 2.1.5.1, the use of residual blocks in a CNN may allow for improved results through refinement of the feature maps. Kamnitsas et al. (2016) made use of residual connections as an extension to DeepMedic, with the new model named DMRes. The authors claim that residual connections should be able to ease learning for biomedical applications to an even greater extent. The reason stated is that such implementations generally require fewer layers than standard image recognition problems. The connections in DMRes employed residual blocks using an element-wise addition between the output of a layer and the input to the preceding layer. This was avoided for the first two layers for each of the parallel pathways to minimise the influence of the initial raw scans on the network.

The performance of DeepMedic and DMRes was evaluated on BraTS 2015 and 2016, and for 2015 obtained a Dice coefficient of 0.89 for the whole tumour, 0.75 for the core, and 0.72 for the enhancing tumour. DMRes performed better for the Dice and sensitivity metrics, but saw a slight decrease in precision. The 2016 challenge shared the same dataset, and the authors pre-trained three models using the 11-layer CNN architecture mentioned previously. A post-processing technique using Conditional Random Fields (CRF) was considered. The CRF was fed a probability map composed of the merged multi-class predictions of the CNN. This provided a segmentation mask for the tumour which would keep the ROI intact. The variation of DeepMedic with residual connections achieved the top Dice scores for the TC and ET classes of images for the 2016 challenge.

2.2.2.2 | U-Net and Other Encoder-Decoder Models

Departing from standard CNN architectures, a large number of successful submissions to the BraTS challenge have made use of the U-Net architecture discussed in Section 2.1.5.3. As stated previously, this technique is recognisable from the U-shape structure of the network, combining a contracting and expanding path for extracting salient information from input data and localising the features within the input image. Kayalibay et al. (2017) proposed a model which adapted U-Net for brain tumour segmentation. This project was fairly comprehensive as it evaluated a vast number of techniques which tackle several issues in biomedical image segmentation.

Firstly, 3D kernels were used here in place of the standard U-Net model which makes use of 2D convolutions. The kernels used were also small in size as in the work by Kamnitsas et al. (2016); Pereira et al. (2016); Simonyan and Zisserman (2014) so as not to grow the network immensely. Zero padding was enforced throughout all convolutions for the output to maintain the resolution of the input image, as opposed to the padding applied in the original U-Net (Ronneberger et al., 2015). Kayalibay et al. (2017) also made use of residual blocks which were placed in the context pathway, with an element-wise addition being applied between convolutions at the same depth to maintain gradient flow in the network. Downsampling in the context pathway was carried out using strided convolutions rather than max-pooling. The former were found by Kayalibay et al. (2017) to yield superior results whilst still achieving the objective of halving the spatial resolution of the image.

The localisation part of the network made use of deep supervision, a concept explored by Chen et al. (2016); Dou et al. (2016). At each level in the decoder half, a secondary segmentation map using a softmax activation is generated from the convolved features at that level. These segmentation maps were combined with their successor at the level above using upsampling and an element-wise addition. Kayalibay et al. (2017) claim that deep supervision refined the segmentation process of the network, with the intermediate losses being weighted and combined with the final segmentation to produce the loss of the network. It is also noteworthy that the skip connections joining both pathways made use of an element-wise addition aside from the usual concatenation operation. Kayalibay et al. (2017) claim that the reason for this is that summing information extracted from the context path with the localisation path would insert information obtained earlier on to later stages in the network. This would lead the feature maps nearing the end of the pipeline to be refined with global features from the input data.

The choice of cost function for this model was fairly interesting, as the Intersection-Over-Union (IoU) was considered in place of the Dice similarity. It is noteworthy that IoU does suffer from maximising loss for images which are completely background or have a near complete absence of foreground pixels, as the loss increases substantially in these cases (Kayalibay et al., 2017). The IoU also suffers from the same drawback as the Dice similarity; without modifications both loss functions are only suitable for binary classification problems. To counteract these drawbacks, Kayalibay et al. (2017) made use of an adaptation of the IoU formula which catered specifically for multi-class segmentation.

Kayalibay et al. (2017) inspired another noteworthy U-Net model proposed by Isensee et al. (2017) as a submission for BraTS 2017. The architecture of both models was very similar however Isensee et al. (2017)'s approach varied slightly as the number of feature maps in the expanding pathway of the network were doubled. The network also made use of instance normalisation in place of batch normalisation since the batch sizes used were small. The loss function was also changed from the IoU to a multi-class Dice Coefficient, which also catered for multi-class segmentation. Overfitting was minimised through use of data augmentation techniques such as random rotations, scaling, elastic deformations, and gamma-level correction. This model obtained competitive results for BraTS 2017, with WT, TC, and ET scores of 0.896, 0.797 and 0.732 on the validation set, and 0.858 for whole, 0.775 for core and 0.647 on the testing dataset.

An improvement over the 2017 model was later submitted for BraTS 2018 by the same authors under the alias "No-New-Net" (Isensee et al., 2018). The team co-trained the 2018 BraTS data with images from the medical image decathlon, which uses older datasets from the challenge. Preprocessing was also fine-tuned for this approach as the authors discovered that localising only brain voxels (not including the background) led to better results. The architecture implemented was very similar to Isensee et al. (2017), maintaining the same input patch size and small batch size for training. Separate segmentation layers were used to cater for the decathlon BraTS data which had different label definitions. The combination of these improvements boosted No-New-Net to second place in the BraTS 2018 challenge, with Dice Coefficient scores of 0.88, 0.81, and 0.78 for the whole tumour, tumour core, and enhancing tumour respectively.

Whilst the previous models made use of residual connections, McKinley et al. (2018) later made use of dense blocks (Huang et al., 2017) in a U-Net style network with encoding and decoding pathways. This project was an adaptation of the team's previous semantic segmentation approach termed DeepSCAN (McKinley et al., 2017). DeepSCAN was also a dense network, making use of dilated convolutions with no pooling or transition layers. However, the resource requirements of DeepSCAN were substantial since all feature maps were retained during training. This is the main reason McKinley et al. (2018) integrated U-Net with DeepSCAN, allowing for a lower spatial resolution within the dense portion of the network to keep the model size reasonable. Dilated convolutions were still retained from the former approach, and the authors employed a loss function combining label noise and uncertainty. This approach performed competitively in BraTS 2018, placing directly below No-New-Net (Isensee et al., 2018) in third place.

The original authors of U-Net++ (Zhou et al., 2018c) also proposed two additional models, the first of which was a One-Pass Multi Network 'OM-Net' (Zhou et al., 2018b). OM-Net integrated segmentation of the tumour into three classes as a single task. The network itself was a CNN making use of residual blocks. Initially, the complete tumour was localized by a singular network and dilated to obtain a refined segmentation mask. Training patches were then obtained from the dilated mask and a final network was trained specifically on these patches to segment the enhancing tumour. OM-Net performed very well on the BraTS 2015 and BraTS 2017 datasets.

This network would later be used in Zhou et al. (2018a)'s proposal of an ensemble approach evaluated on the BraTS 2018 datasets. This study made use of three identical encoder-decoder CNN models, used to precisely segment the whole tumour, whole tumour and tumour core, and enhancing tumour classes respectively. In Zhou et al. (2018a), OM-Net was extended in multiple ways. Firstly, the network was deepened through the use of an additional residual block after every already existing residual block. The network also received an additional series of nested and dense-skip connections to correlate features between the encoder and decoder better, inspired by U-Net++. A 'Squeeze-and-Excitation" (SE) block common to Inception networks was then used to recalibrate features across all of the input channels. Zhou et al. (2018a) claim that this block adaptively modified features across the input modalities to increase the sensitivity of salient features in the data and dampened the impact of irrelevant features.

Finally, another network with a dual pathway for processing input patches at two different resolutions was also introduced to consider contextual information from both patch sizes, with the alias MC-Net. This was done to achieve the benefits of using larger input patch sizes whilst retaining ROI information from the smaller resolution patches. An ensemble of OM-Net, MC-Net, and Deeper OM-Net with the SE blocks and post-processing resulted in very competitive Dice coefficient scores. The ensemble achieved

0.88, 0.79, and 0.78 for the WT, TC, and ET Dice scores respectively on the BraTS 2018 test dataset. Zhou et al. (2018a) state that this experiment came in third place for the 2018 challenge, tied with McKinley et al. (2018).

Whilst the aforementioned U-Net/CNN based approaches performed well enough to secure runner-up positions in the competition, Myronenko (2018) implemented an encoder-decoder CNN with a variational auto-encoder (VAE) branch to secure first place. This model works in a similar way to U-Net, where the model first aggregates context information before applying it to input images. The difference lies in the connection between the encoding and decoding parts of the network. The output of the encoder was split halfway into the mean and standard deviation, which were then used to generate samples from a Gaussian distribution to reconstruct the images prior to the localisation process beginning. Skip-connections transmitted from the encoder to the decoder were not passed on to the central bottleneck. Some other differences of note are the large patch size of $160 \times 192 \times 128$ and the loss function, which used a soft Dice-loss combined with standard VAE loss terms. The Dice Coefficient scores obtained on the BraTS 2018 test dataset for the whole tumour, tumour core, and enhancing tumour were 0.88, 0.82, and 0.77.

In summary, methods for brain tumour segmentation have developed substantially from their earlier counterparts. The need for manual feature extraction has been reduced greatly, since CNNs perform the process automatically. CNNs and systems which follow encoder-decoder architectures or are ensembles of proven methods have been shown to perform very competitively in the BraTS challenge. This is very evident considering that 70% of submissions for BraTS 2018 were either U-Net models or ensembles of U-Net and CNNs (Ghaffari et al., 2019). The top ranking techniques are also generally U-Net-like architectures or ensembles of CNNs as in Zhou et al. (2018a). It is also noteworthy that whilst the top entry in BraTS 2018 (Myronenko, 2018) made use of a CNN/VAE hybrid approach, there were also several other differences in the implementation such as the input image patch size used during training which was the largest among all of the top-ranking submissions.

3

Materials & Methods

This chapter will elaborate on everything related to the data and pipeline for the project. Selection of the dataset and how it was acquired will be discussed in Section 3.1.2. Data cleaning in preparation for model training will then be explained in Section 3.1.3. The training process used for the proposed approach and reference models will be explained in 3.1.4. Section 3.1.5 provides visualisations and a discussion of the architecture for each of the individual models.

3.1 | Methodology

3.1.1 | Project Pipeline

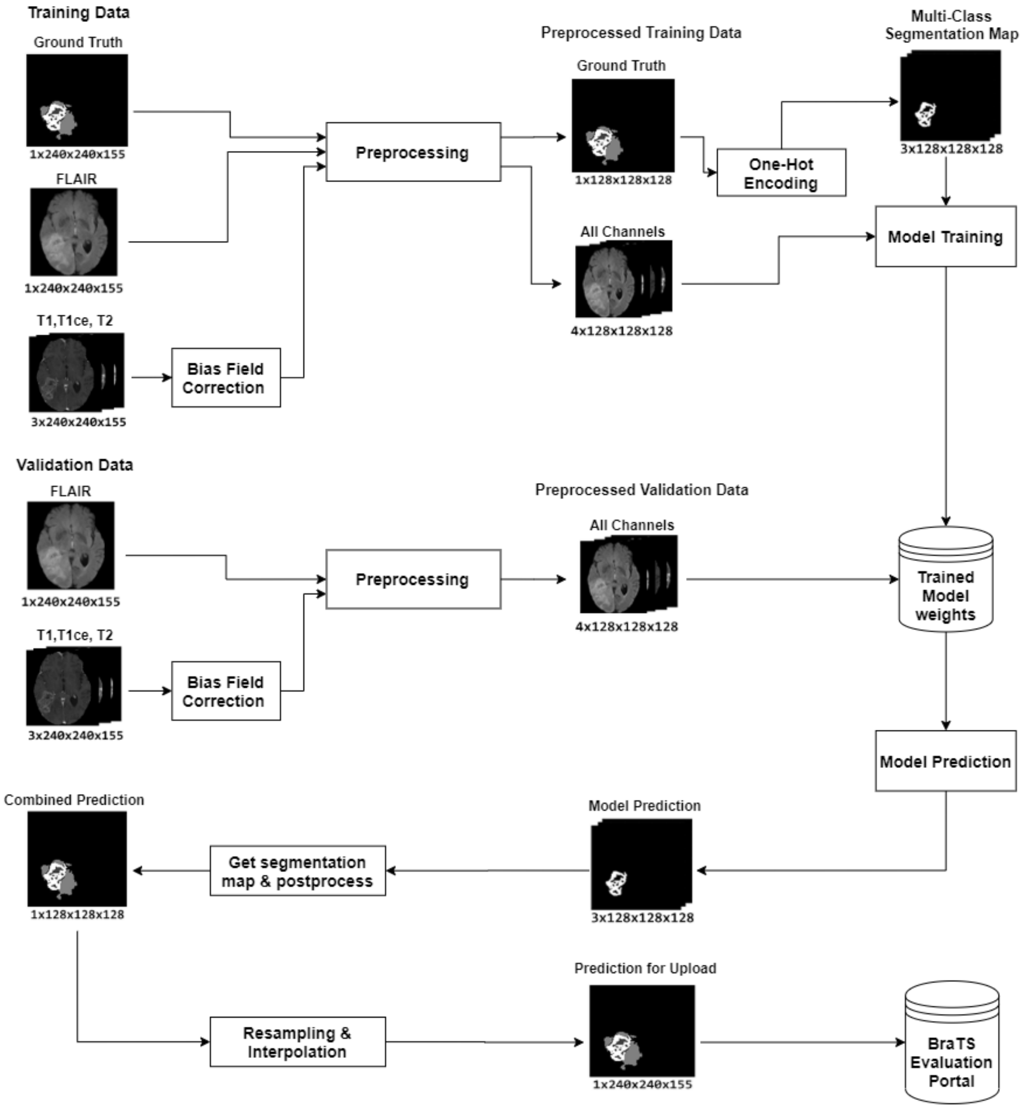


Figure 3.1: The project pipeline from data acquisition to uploading to the CBICA IPP.

Figure 3.1 shows the proposed project pipeline from preprocessing the input data to submitting the predictions to the CBICA portal for evaluation. Initially, N4 bias-field correction and preprocessing techniques are applied to the raw training data. This is carried out in an equivalent way on the validation data to create identical conditions to those used during training. This step is also necessary since the model requires $4 \times 128 \times$

128 × 128 input for generating predictions. The reasons for this resolution being selected for downsampling will be discussed shortly in Section 3.1.3. The preprocessing pipeline only needs to be carried out once for each set of data as the files are then stored on disk due to the large size of a whole dataset. As an example, individual multimodal samples from the BraTS 2019 training dataset in compressed Nifti format occupy close to 19MB on disk, with the whole dataset occupying 4.5 GB when compressed in a *pytables .h5* file. Further explanation regarding the Nifti file format will be provided in Section 3.1.2.1. Following preprocessing, training batches are generated from image generators and fed to the model. The ground truths are one-hot encoded for each of the classes throughout this step, creating three binary segmentation maps; one for each of the peritumoural edema, non-enhancing and necrotic tumour core, and enhancing tumour labels.

Once training has been completed for the selected model, the preprocessed validation data may be used to generate predictions. Predictions have the same dimensions as the one-hot-encoded inputs, consisting of three 1 × 128 × 128 × 128 binary segmentation maps. The predicted segmentation maps are then merged into a single 1 × 128 × 128 × 128 image containing all three sets of labels. This process is a prerequisite of the IPP which only accepts data which has the same dimensions as the original dataset. This not only involves resampling and re-interpolating each prediction to 240 × 240 × 155, but also using the same stride direction across each axis. To simplify this process, the *MRtrix3* (Tournier et al., 2019) library for MR analysis was used. The *mrtransform* function allows a template image (in this case a ground truth from the training set) to be specified to remap the dimensions and stride directions accordingly.

It is important to mention that the process displayed above was adapted from a popular Github repository[1] aiming to replicate the implementation by Isensee et al. (2017). The repository in question falls within an MIT licence, making the proposed approach also within these licence terms. The methodology the Github code aims to mirror follows the thought process of one of the best teams in the BraTS challenge both in 2017 and also 2018 (Isensee et al., 2018), although it is based purely on the 2017 implementation. It is quite well rated and shares a similar pre-processing pipeline with some peer-reviewed works (Havaei et al., 2017; Kamnitsas et al., 2016; Pereira et al., 2016). Thus, the majority of the pipeline performing preprocessing and generation of samples for the models follows this repository very closely, including one of the models used for internal comparison with the proposed approach. Having said that, this project's code was mostly written from scratch, using the repository as a close reference throughout the process to maintain the same workflow.

[1] https://github.com/ellisdg/3DUnetCNN

3.1.2 | Data Definition

3.1.2.1 | MR Image format

The datasets provided for the challenge are available in a compressed Neuroimaging Informatics Technology Initiative (Nifti) format. The Nifti file format was proposed as a replacement to the previously-used ANALYZE files, which lacked MR image header information such as the orientation of features within the voxel space[2]. Nifti allows storage of important geometric features of the data as well as patient metadata. Since ANALYZE had already been adopted by a number of software suites, similar interfaces for the Nifti format also exist. Thus, the *.nii* files following this format are compatible with a large number of libraries in programming languages such as MATLAB and Python. The Nifti files used for BraTS are volumes measuring 240 × 240 × 155, where each dimension represents one of the axial, coronal and sagittal views of the MR image.

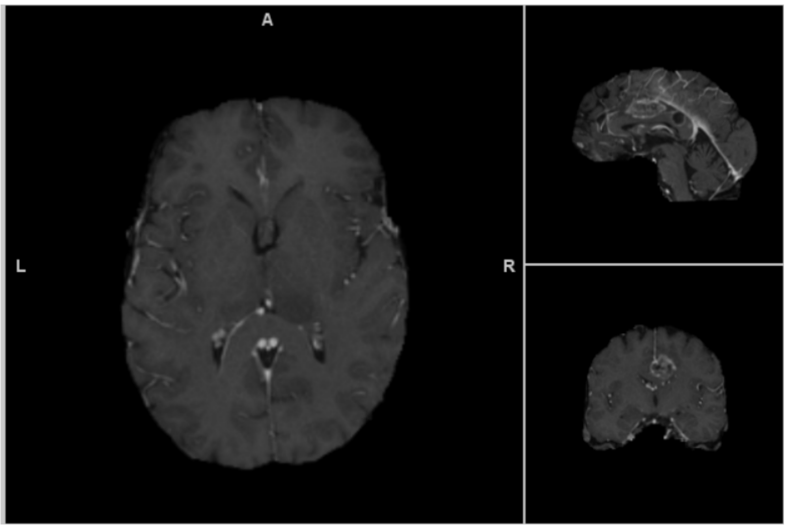


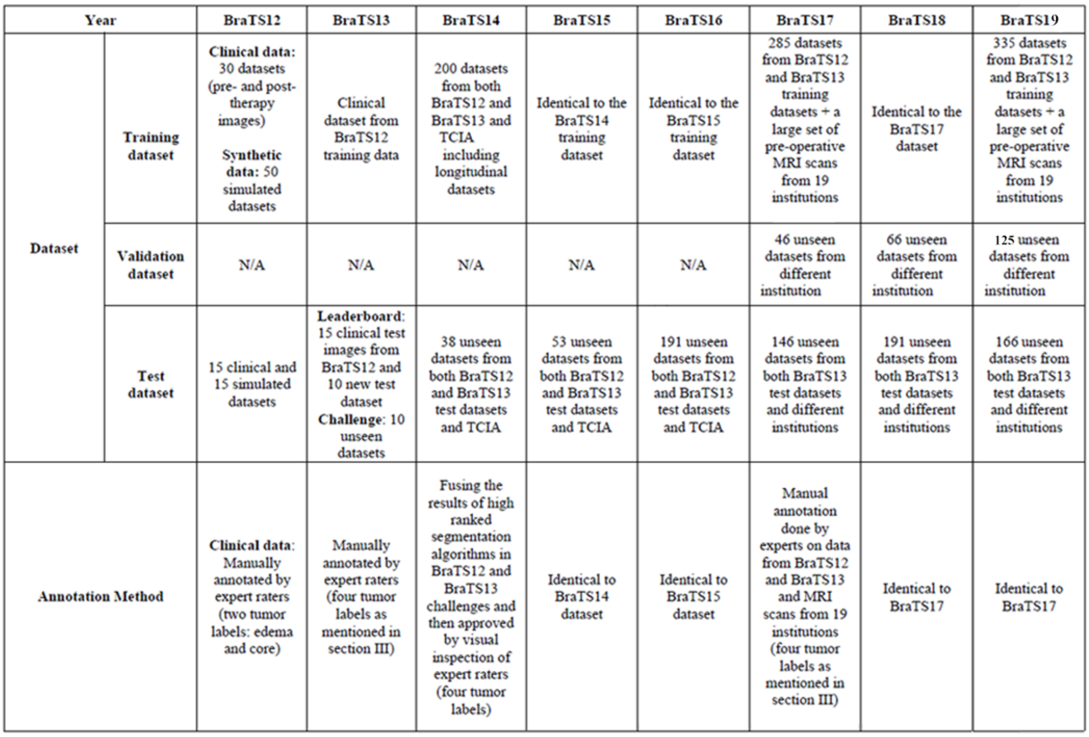
Figure 3.2: Example of a T_1ce training set Nifti sample shown in the *Mango*[3] MR image viewer.

The header information of the provided Nifti files also contains details about the resolution, affine transformation, and strides of each subject. These are important to maintain and update when performing transformation operations such as preprocessing and resampling. The Nifti file header is also checked by the BraTS portal to ensure the dimensions are homogeneous with the original dataset.

[2] https://brainder.org/2012/09/23/the-nifti-file-format/

[3] http://ric.uthscsa.edu/mango/

3.1.2.2 | BraTS Datasets



Year		BraTS12	BraTS13	BraTS14	BraTS15	BraTS16	BraTS17	BraTS18	BraTS19
Dataset	Training dataset	**Clinical data:** 30 datasets (pre- and post-therapy images) **Synthetic data:** 50 simulated datasets	Clinical dataset from BraTS12 training data	200 datasets from both BraTS12 and BraTS13 and TCIA including longitudinal datasets	Identical to the BraTS14 training dataset	Identical to the BraTS15 training dataset	285 datasets from BraTS12 and BraTS13 training datasets + a large set of pre-operative MRI scans from 19 institutions	Identical to the BraTS17 dataset	335 datasets from BraTS12 and BraTS13 training datasets + a large set of pre-operative MRI scans from 19 institutions
	Validation dataset	N/A	N/A	N/A	N/A	N/A	46 unseen datasets from different institution	66 unseen datasets from different institution	125 unseen datasets from different institution
	Test dataset	15 clinical and 15 simulated datasets	**Leaderboard**: 15 clinical test images from BraTS12 and 10 new test dataset **Challenge**: 10 unseen datasets	38 unseen datasets from both BraTS12 and BraTS13 test datasets and TCIA	53 unseen datasets from both BraTS12 and BraTS13 test datasets and TCIA	191 unseen datasets from both BraTS12 and BraTS13 test datasets and TCIA	146 unseen datasets from both BraTS13 test datasets and different institutions	191 unseen datasets from both BraTS13 test datasets and different institutions	166 unseen datasets from both BraTS13 test datasets and different institutions
Annotation Method		**Clinical data**: Manually annotated by expert raters (two tumor labels: edema and core)	Manually annotated by expert raters (four tumor labels as mentioned in section III)	Fusing the results of high ranked segmentation algorithms in BraTS12 and BraTS13 challenges and then approved by visual inspection of expert raters (four tumor labels)	Identical to BraTS14 dataset	Identical to BraTS15 dataset	Manual annotation done by experts on data from BraTS12 and BraTS13 and MRI scans from 19 institutions (four tumor labels as mentioned in section III)	Identical to BraTS17	Identical to BraTS17

Figure 3.3: BraTS dataset changes since competition inception (Ghaffari et al., 2019).

The BraTS datasets provide MR images which are substantial in number and are also annotated by multiple experts, corroborating the data's status as a form of 'gold standard' in the domain. The datasets have been rigorously updated over the years, as shown in Figure 3.3, with a small addendum added to the image for the 2019 challenge dataset. As stated in Bakas et al. (2018), the first challenges for 2012 and 2013 comprised a training and testing dataset of 35 and 15 MRI scans respectively, and generated such a level of interest from the community that the accompanying paper was the most popular and top downloaded work from the IEEE TMI Journal in 2018 (Bakas et al., 2018). At the time of writing, this paper is still immediately listed in the popular papers linked with the TMI journal, showing the substantial reach of both the challenge and the datasets.

The first main changes in the dataset occurred in 2014 and 2015 with the inclusion of data from The Cancer Imaging Archive (TCIA) and the Center for Biomedical Image Computing and Analytics (CBICA) at the University of Pennsylvania (UPenn). The TCIA and CBICA training data were provided pre-segmented by top-ranking algorithms from previous years of the challenge. 2016 followed a very similar route, with the main difference being additional samples provided as test data. The annotations for the years between 2012 and 2016 of the BraTS challenge consisted of four labels, namely the necrotic tumour core (label 1), peritumoural edema (label 2), non-enhancing tumour core (label 3), and enhancing tumour core (label 4). Label 3 was being overestimated by some annotators (Bakas et al., 2018), resulting in it being merged with the necrotic tumour core (label 1) from 2017 onwards, leaving only labels 1, 2, and 4.

In 2017, the next major change took place for the data used in the challenge. The ground truths previously segmented by algorithms were evaluated and revised by experts where necessary. In addition, new clinically-acquired 3T multimodal MRI scans were also provided, with the ground truth labels manually-revised by expert board-certified neuroradiologists. 2017 was also the first year a validation set was included to facilitate training of segmentation models. As discussed in Section 2.1.6, the training and validation are two separate datasets, hence why 'unseen' samples is used in Figure 3.3. The new standards set by the 2017 data would lay the foundation for later iterations of the competition. The training dataset for 2018 appears to be identical to 2017, with the only change being the validation and test datasets growing considerably. The BraTS 2019 data is also very similar to the 2017 and 2018 datasets, with the main change being an increase in the overall number of scans and corresponding expert annotations.

Since BraTS 2019 currently provides the largest quantity of expert data as well as being the most recent revision of the older datasets, it was considered the best year to use data for this project. The data were obtained by enrolling as a contestant to the BraTS challenge on the CBICA site to make use of the Image Processing Portal. The 2019 training dataset was provided shortly after, and the validation set a few weeks following that. The training dataset was split 80-20 during training of the model, whilst the validation dataset is provided without ground truths and evaluated solely on the CBICA IPP. The test data were available for a limited time window (August-September 2019, very early on in development) and only for live uploads to the challenge in August, thus it will not be included in this project. The datasets are co-registered to a single anatomical template, skull-stripped, and interpolated to the resolution of $1mm^3$. Table 3.1 expands on the aforementioned labels and target classes for the challenge.

Table 3.1: Ground Truth annotations for the BraTS 2019 dataset.

Class	Enhancing Tumour Core (Label 4)	Peritumoural Edema (Label 2)	Non-Enhancing & Necrotic Tumour Core (Label 1)
Enhancing Tumour (ET)	✓		
Whole Tumour (WT)	✓	✓	✓
Tumour Core (TC)	✓		✓

From Table 3.1, one may observe how the target classes are overlapping. Whilst ET is solely composed of label 4, TC consists of both tumour core components (labels 1 and 4), and WT is formed from all three labels. However, it is also essential to consider the frequency of each label, as gliomas may have a very large edema region and a minuscule enhancing tumour core (if at all). Havaei et al. (2017) observed that for BraTS 2013, 98% of the MR image voxels were healthy tissues, with the rest still being unevenly distributed in favour of edema tissues. Thus, it is possible that any model trained on these images would obtain higher WT scores compared to the other classes, owing to the large edema segment exclusive to WT.

3.1.2.3 | Data Distribution

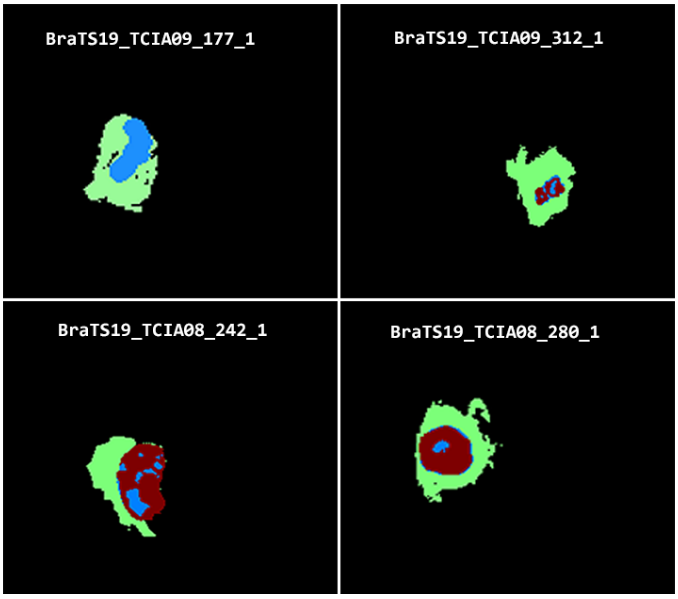


Figure 3.4: Subject ground truths from the BraTS 2019 Training set showing top - LGG, bottom - HGG. The masks correspond to the labels as follows: edema - green, enhancing tumour - red, non-enhancing tumour/necrotic tumour core - blue.

As discussed in Section 2.1.2, the dataset images are provided in T_1w, T_1ce, T_2w, and FLAIR modalities, split into HGG and LGG. For 2019, the 335 samples in the training data are divided into 259 HGG and 76 LGG cases. The imbalance between both categories is a factor to consider as there are more than three HGG cases for every LGG patient. Another observation is how HGG and LGG have both intra and inter-categorical visual differences even within the same slice range. Figure 3.4 shows an example of how the structure of HGG and LGG is different both within and across glioma types.

Furthermore, Bakas et al. (2018) state in the BraTS data description that there is an uncertainty surrounding the segmentation of LGG, which is also reflected in the expert annotations in the BraTS datasets (mostly the ET segments). Having said that, the multimodal nature of the data provides the diverse strengths of each sequence type when training a model. As an example, the property of T_1ce and T_2w sequences exhibit a higher signal intensity for pathologies, which is a definite advantage which may assist with these issues.

3.1.3 | Preprocessing

The initial preprocessing task addressed was intensity inhomogeneity in the raw MR images, where identical tissue types in the brain had different intensities. N4ITK bias field correction (Tustison et al., 2010) is one popular technique used to restore the intensity distribution to brain tissues of the same class. An N4 library is available with the Advanced Normalization Tools (ANTs) (Avants et al., 2009) toolkit for multimodal image analysis. It should be noted that some ANTs functionalities are known to raise issues when set up on a Windows system. Therefore, the process was executed either on native Ubuntu 18.04 or using the Ubuntu 18.04 LTS subsystem available within Windows, depending on the stage of development. Bias correction was applied to all MR sequence types except for FLAIR, following the process in the online repository. Results of the process are shown in Figure 3.5 on a T_2w MRI from the training set.

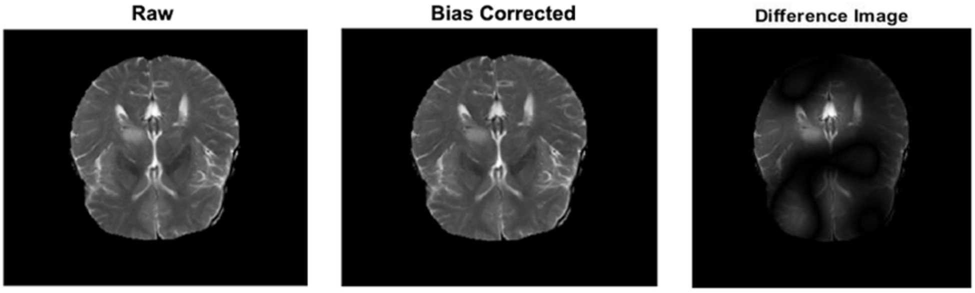


Figure 3.5: Bias-field correction results.

The next step was to perform background removal on the images by using the Python neuroimaging library *nilearn*'s *crop_img* method. It is noteworthy that there are some parts of a scan, notably the initial and ending slices which are close to a total zero intensity, meaning that there is little to no useful information. Representative slices are generally situated near the middle of a volume. As a result, the sections with an overwhelming number of zeros within the volume are better candidates for cropping, reducing the final spatial resolution of the input images more intelligently. The function crops values ranging between zero and a relative tolerance parameter (in this case the default value of $1e^{-8}$). Non-zero values outside this range are untouched, which makes this a good approach to retain foreground voxels whilst also removing 'extra' background voxels from the scans.

Given that each set of images has different non-zero value distributions, the previous step produced cropped images with varying resolutions, which would be infeasible for model training. Furthermore, U-Net CNNs leveraging 2×2 max pooling require even-dimensions on the input for tiling the output segmentation map (Ronneberger et al., 2015). Thus, it was necessary to decide on a final image resolution to be used by all of the input samples during training. As discussed previously, the original resolution of the provided MR images was $240 \times 240 \times 155$. These dimensions are not accounting for the fact that all four image modalities were supplied to the model simultaneously at training time, bringing the total input resolution in this scenario to $4 \times 240 \times 240 \times 155$. Therefore, the target image resolution would also have to consider the fourth axis referring to the MR sequence type.

The final input image resolution decided for the inputs was $128 \times 128 \times 128$, following analysis of GPU memory consumption and the fact that competitive studies such as Isensee et al. (2017, 2018) make use of these input dimensions as well. A combination of *nilearn* and *SimpleITK* functions were used to create the resized version of the dataset. The first step stored the original image as a reference image throughout the resizing process. Calculation of the new affine transformation for the target resolution was then carried out. This affine transform was essential to retain as it kept the preprocessed MRI's new position in space with reference to the original image. The latter also made it possible to easily rescale the image back to the original input dimensions when preparing uploads for the BraTS portal.

Following the above preparations, each MR image was resampled and interpolated, mapping the original image to the new dimensions. *SimpleITK* was used to resample the image using the new affine transformation on the four sequence volumes (T_1, T_1ce, T_2, FLAIR) and ground truth file for each patient. The parameters used were the same for both file types, except for the interpolation. When processing the input volumes, linear interpolation was used as it is considered the best tradeoff between accuracy and computational efficiency[4]. For the ground truths, nearest neighbour interpolation was always used to avoid the introduction of any constants outside the BraTS labels.

There is also the issue mentioned previously that scans acquired from separate scan-

[4] https://simpleitk.github.io/SPIE2018_COURSE/images_and_resampling.pdf

ners across different institutions will produce varying intensity distributions. This was corrected by applying standardisation to the data by subtracting the mean and dividing by the standard deviation as described by Myronenko (2018); Pereira et al. (2016). The equation below is used to apply z-score normalisation to the data to fit these criteria:

$$x_{new} = \frac{x - \mu}{\sigma} \tag{3.1}$$

x_{new} refers to the standardised scans whilst x refers to the original set of MR images, both over all four modalities. μ and σ refer to the mean and standard deviation of the whole dataset. Thus, every input channel is standardised over an averaged mean and standard deviation of the entire dataset, rather than the mean and standard deviation of that patient. This was done to create a more uniform distribution of intensity values throughout the preprocessed data. Jiang et al. (2019); McKinley et al. (2019); Zhao et al. (2019).

3.1.4 | Training

Once preprocessing was completed, the training process commenced. Keras 2.3.1 and Tensorflow-GPU 1.14.0 were used to build the models and facilitate the training process within Python 3.5.6. This specific Python version was used as it had the best compatibility with the combination of libraries used for the project. The parameters used for model training were identical for each of the architectures to be discussed, barring the model mentioned in Section 4.1.6 as this used a different training-validation split (albeit using the same training to validation ratio). Single-patient batches consisting of a multimodal input volume and the corresponding one-hot encoded ground truth segmentation map were passed to the model being trained, totalling 268 training steps when using an 80-20 split. The hyperparameters used for training all models including the experiments and reference models closely follows the process explored by Isensee et al. (2017). Each model was trained for 300 epochs using a learning rate of $5e^{-4}$, with a callback set to drop the learning rate if a plateau was encountered after 50 epochs. The Adam gradient descent algorithm (Kingma and Ba, 2014) was used as the optimisation algorithm for the model. Observing how previous papers had reported on the class imbalance often present in biomedical image segmentation (Isensee et al., 2017; Kayalibay et al., 2017), the loss function used is the multi-class adaptation of the Dice loss as devised by Isensee et al. (2017). Equation 3.2 shows how the negative multi-class Dice loss is calculated.

$$L = -\frac{1}{K} \sum_{k \in K} \frac{2(Y_k \cap \hat{Y}_k) + \alpha}{(Y_k + \hat{Y}_k) + \alpha} \tag{3.2}$$

Here, K refers to the 3 labels and Y, $\hat{Y}$ refer to the ground truth and prediction of the model respectively. The divisor coefficient and summation outside of the main function is the main difference from the standard Dice loss, providing a weighting for the multiclass problem. α is a smoothing constant with a value of $1e^{-5}$. It should be noted that given the large resolution of the dataset, training the models proved to be a long process which had to be moved to Google Cloud for most runs. The input resolution was quite demanding on resources, instantly ruling out the possibility of working using solely the CPU, as the latter would have resulted in a training cycle taking weeks to complete. A small number of tests were runnable on the Nvidia GTX970 GPU with 6GB of GPU memory, with a Google Cloud Deep Learning Virtual Machine used for the others. Initially, processes were carried out using an instance on Google Cloud with a Tesla K80 GPU with 12 GB of GPU memory. This allowed at least two training cycles to be run in parallel when using both local and cloud machines.

When selecting the platform for a test run, the viability of local vs. cloud was mostly determined by the model's size and complexity. As an example, a standard U-Net model could be run locally due to the reduced model parameters and complexity. Table 3.2 shows the estimated difference in training time on an earlier build of U-Net++, as described in Section 4.1.6.

Table 3.2: GPU and training time comparisons for the MELECON U-Net++ model, which is a slightly simpler version of the proposed model runnable on a GTX970 GPU.

GPU	CUDA	CUDA cores	GPU memory	T/step (s)	T/epoch (s)	T/300 epochs (hrs)
GTX 970M	5.2	1,280	6 GB GDDR5	2	589	49
Tesla K80	3.7	4,992	12 GB GDDR5	8	2,185	182
Tesla P100	6.0	3,584	16 GB HBM2	2	520	43

One may observe the inclusion of a Tesla P100 in Table 3.2. Whilst the move to a cloud instance running a Tesla K80 led to most of the tests being runnable, it was a slow and expensive process as the CUDA compute capability of the Tesla K80 is fairly low. The process could take anywhere from 4-7+ days depending on the size and complexity of the model being trained. In light of this, a decision was made to switch to a Tesla P100 GPU which has a superior compute capability and also an increased capacity of 16GB GPU memory. This was more expensive in terms of personal cost, as no funds for Cloud services were provided for this project. Nonetheless, the switch to the Tesla P100 resulted in a reduction of total training time down to 2-3 days for 300 epochs with most models.

3.1.4.1 | Data Augmentation

During initial training, it was noticed that overfitting was very likely to occur when running any of the models. The main observation implying this behaviour was the validation accuracy saturating long before the training accuracy. To minimise this problem, a number of data augmentation techniques were used to introduce synthetic samples in the dataset. Some examples of these techniques are random permutations of rotations and axis flips, which were applied to the training data from the image generators used to provide the input to the model.

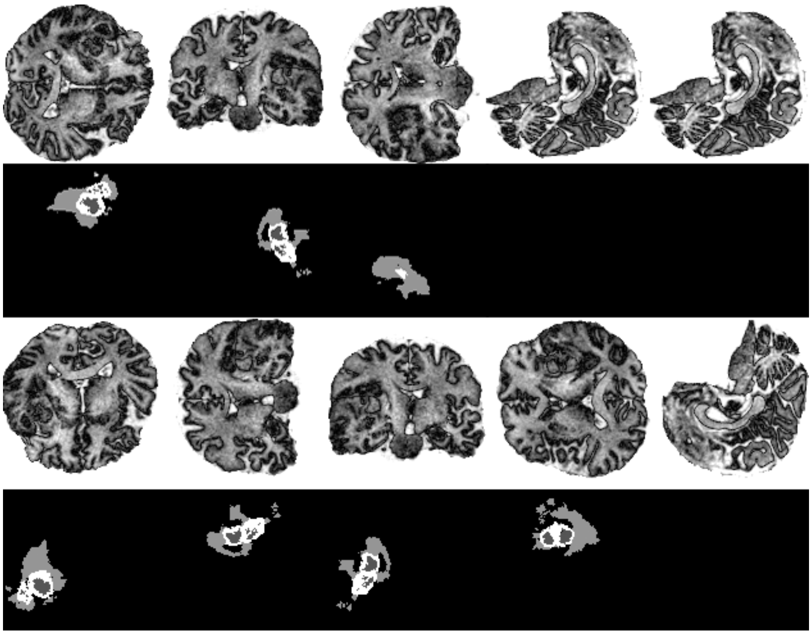


Figure 3.6: Synthetic sampling using random rotations and flips, generated using a sample from the 2019 training set.

The images in Figure 3.6 were generated to exhibit the sample variety shown to the model during training when using random rotations and axis flips. Instances where the ground truth does not appear are due to a flip to the sagittal view where this particular slice does not show the tumour. These cases may be observed in the fourth, fifth, and final slices in Figure 3.6. This process was applied to every multimodal set of input channels and ground truth during training, creating additional examples for the model to improve generalisation. Other techniques which were considered included elastic deformations and b-spline-interpolation via *Gryds*[5], a Python package for geo-

[5] `https://github.com/tueimage/gryds`

metric transformations. Figure 3.7 shows some examples of elastic deformation applied to a sample from the training set.

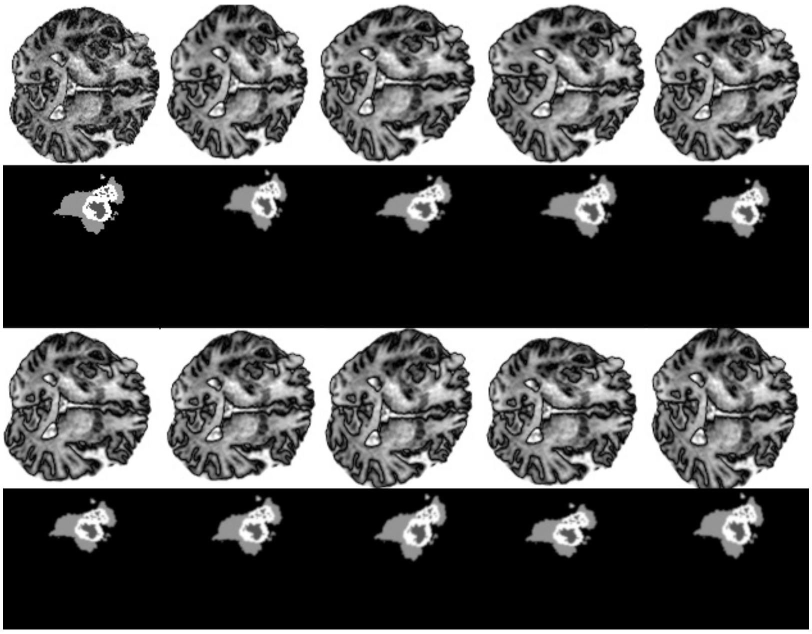


Figure 3.7: Synthetic sampling using elastic deformations on the same BraTS 2019 training set sample.

Figure 3.7 shows the effects of nine elastic deformation transformations with an order of 5, and sigma of 3. The order and sigma control the severity of the deformation, with the values used in Figure 3.7 being considerably low. This was done to avoid overly transforming the image such that it would not be recognisable during training. It is notable that the libraries used also apply a light Gaussian smoothing effect on the image, as one may observe from the blurry truth samples compared to the original ground truth image. Elastic deformations were considered as they were used to good effect by Ronneberger et al. (2015) for the original U-Net. The technique was ultimately scrapped from the pipeline, for reasons mentioned in Section 4.1.1. Similar experiments were also carried out for the implementation of dropout layers, discussed in Section 4.1.4.

3.1.5 | Models

The models explored in this experiment are all based on U-Net (Ronneberger et al., 2015). This decision was made as U-Net architectures are generally very parameter efficient and have been shown to be competitive in the BRaTS challenges throughout the years (Ghaffari et al., 2019). The models to be discussed are the following:

- A 3D adaptation of standard U-Net with some modifications to fit our pipeline.
- A 3D residual U-Net inspired by Isensee et al. (2017).
- The proposed model, a 3D U-Net++ (Zhou et al., 2018c) inspired model.

Each of the above models will be expanded upon in detail in Sections 3.1.5.1, 3.1.5.2, and 3.1.5.3. Of the three models listed, the U-Net++ model was tested as the proposed technique for this project, with the other models being included as comparison tools later on in the development cycle for internal evaluation. Nonetheless, a visualisation and explanation of each of the three models will be provided. Since the models evolve upon each others' architecture, explaining each model facilitates a better context for fair comparison. Before exploring each architecture, it is important to explain the core modules/blocks for each model and how these vary across each system.

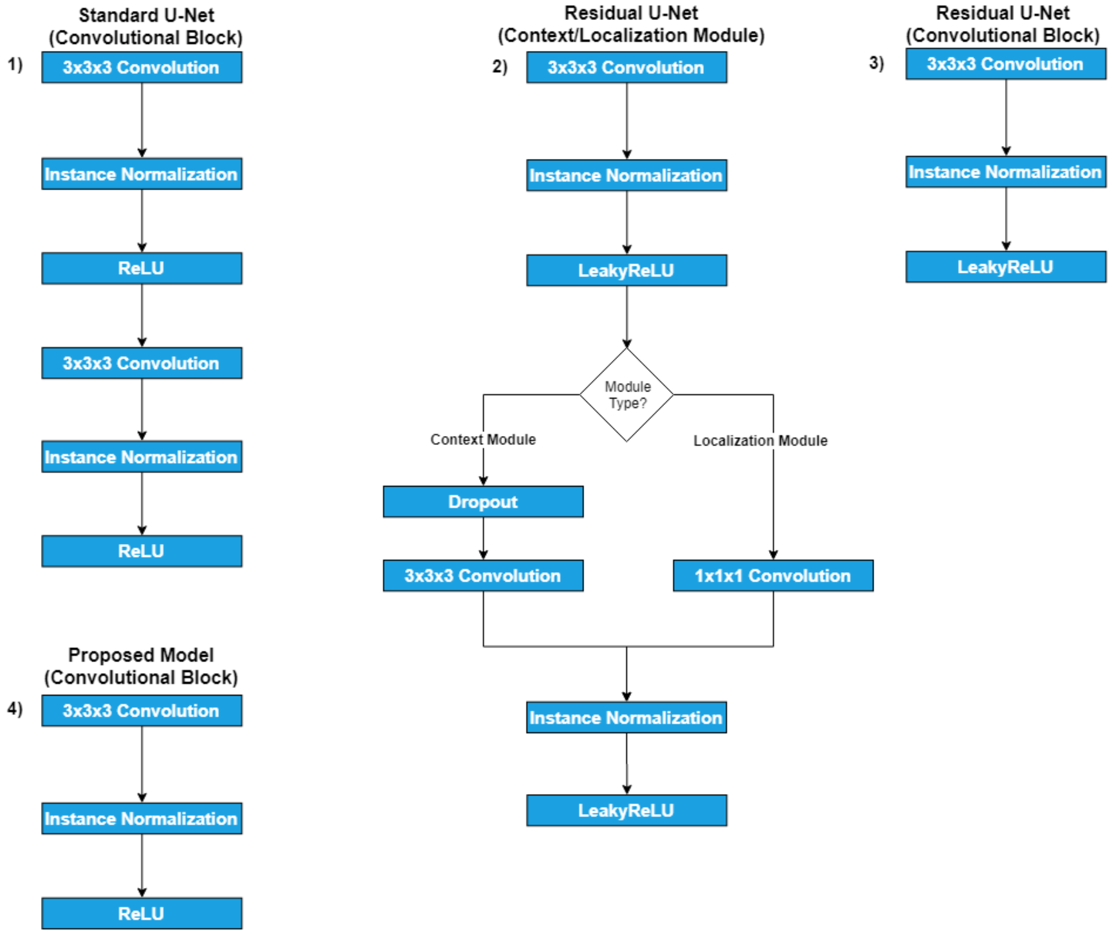


Figure 3.8: Different convolutional block setups used for this project

Each model is composed of a number of convolutional blocks, as shown in Figure 3.8. The standard U-Net and U-Net++ convolutional block setups are straightforward, and uniform throughout the network. The residual 3D U-Net follows a similar structure, however making use of two module types which implement convolutional blocks. Context modules are used in the encoder part of the network and make use of dropout regularisation. Localisation modules are used in the decoder part of the network and remove dropout, also replacing the final $3 \times 3 \times 3$ convolution with a $1 \times 1 \times 1$ convolution layer. Tests were also carried out for U-Net++ using different convolutional block combinations, however the structure shown in Figure 3.8 is the final choice for reasons which will be discussed in Section 4.1.3.

3.1.5.1 | 3D U-Net

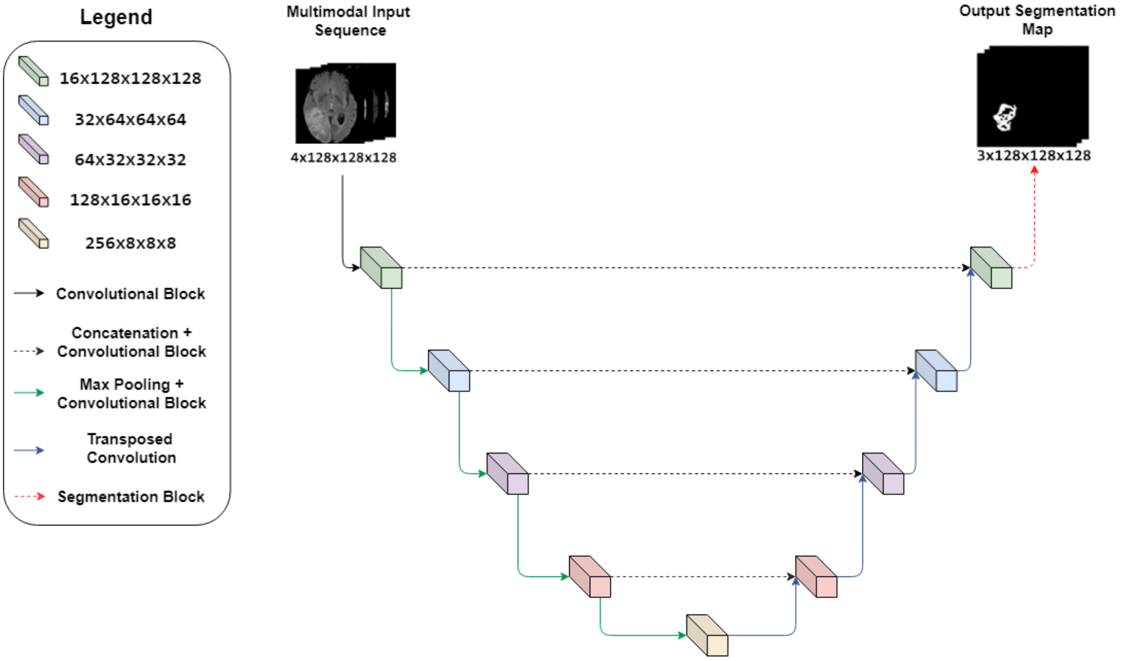


Figure 3.9: 3D U-Net architecture. The 3D rectangular blocks in the diagrams and their dimensions refer to the filter resolution throughout model training. This model diagram and the two to follow are designed specifically for this project and correspond directly to each of the internal evaluation models.

This model's architecture (pictured in Figure 3.9) builds upon the fundamentals of standard U-Net (Ronneberger et al., 2015). The encoder half of the network extracts salient features from the $4 \times 128 \times 128 \times 128$ inputs by reducing the spatial resolution by half and doubling the amount of filter maps at each level. The encoder features convolutional blocks with two 3 × 3 × 3 convolutions and a 2 × 2 max pooling operation. As previously discussed, the small kernels allow the model to maintain a relatively small number of parameters (Simonyan and Zisserman, 2014). There are five levels in the network, with the final level acting as a bottleneck between the decoder and encoder. Concatenation layers connect both halves of the network at each level outside of the bottleneck as skip-connections. Following the experiments in Section 4.1.4 showing that dropout layers with the tested value were not beneficial to the approach, they were omitted from the standard U-Net model as well.

Once the bottleneck layer at the centre of the network is reached, the process moves to the decoder half of U-Net, which is composed of a sequence of 3 × 3 × 3 transposed convolution operations followed by two 3 × 3 × 3 convolutions at each level. Effectively, the decoder is structurally identical to the encoder except that it performs the direct opposite functions. Transposed convolutions with stride two are used to perform the inverse of max pooling and convolution operations; halving the filter space and doubling the spatial resolution of the image. The concatenations at each level combine the upsampled features with features from the encoder at the same level of the network. This sequence is repeated throughout the decoder until the shallowest level is reached. The final segmentation map is then generated through a 1 × 1 × 1 convolution on the final set of convolved features. Segmentation modules have the number of filters set to 3 (a filter for each target label) and a 1 × 1 × 1 kernel for dimensionality reduction, mapping the features to each of the three labels.

The architecture was adapted from the standard U-Net model by Ronneberger et al. (2015) to accept multimodal 3D-input with some minor changes. A necessary change was modifying the convolutional blocks in the model to use ‘same’ padding to keep the output resolution equivalent to the input, as the original U-Net did not zero-pad which resulted in the output changing dimensions. The range of filter map resolutions applied to the convolutional blocks was also changed from [64..1024] to [16..256]. This was done due to the input resolution being much larger than the standard U-Net’s 572 × 572 images. This also made the model use uniform parameters to the residual U-Net and U-Net++ which also used the same filter resolution. Finally, since the project’s setup processes patient samples individually, it was observed that batch normalisation was destabilising training, as noted by Isensee et al. (2017). Instance normalisation was used instead between every convolution and ReLU layer as in Isensee et al. (2017). The combined features above accumulate a total of 6.5M parameters for this model.

3.1.5.2 | Residual U-Net

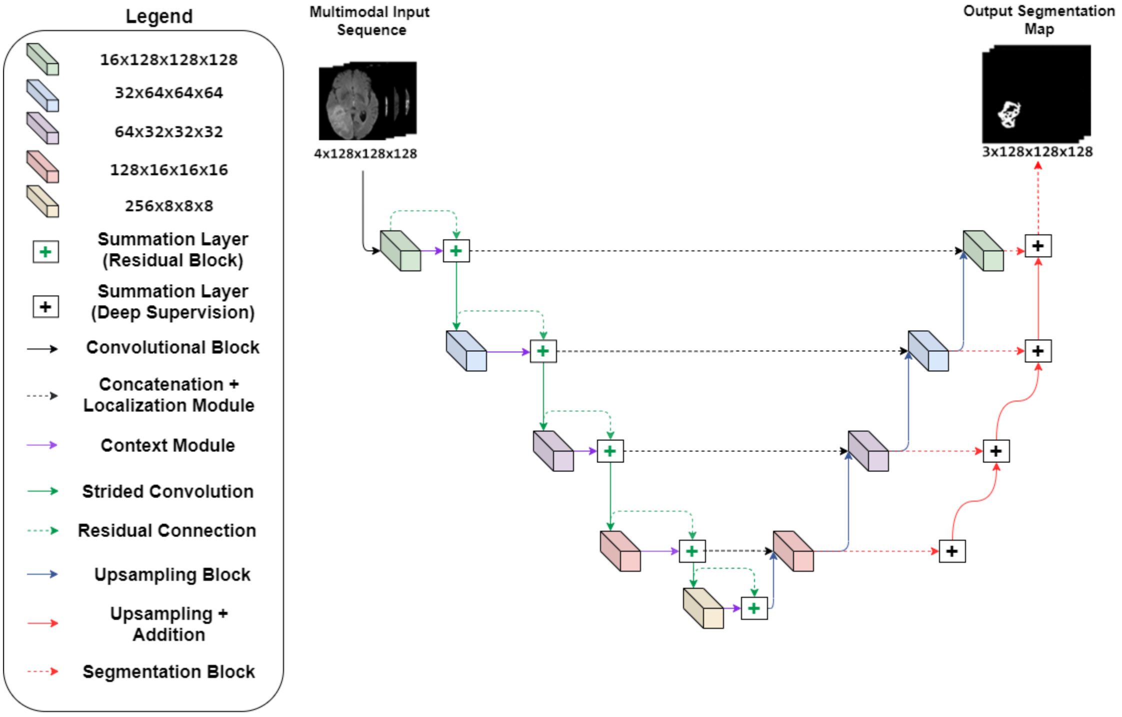


Figure 3.10: 3D Residual U-Net architecture.

The residual U-Net model (shown in Figure 3.10) follows the encoder-decoder pathways standard to U-Net. The major differences in architecture are the addition of residual connections (He et al., 2016) and deep supervision (Chen et al., 2016). Other changes in the implementation of this model compared to standard U-Net are mostly the changes introduced in the study by Isensee et al. (2017). This model was copied directly from the Github repository[6] to ensure the architecture correctly follows the aforementioned work.

As a U-Net adaptation, this 3D residual variant introduces a number of changes in both the encoder and decoder parts of the network. The encoder now performs downsampling by using convolutions with stride 2 in place of max pooling. It is noteworthy that strided convolutions also slightly increase the number of model parameters since they are additional 'learning' layers. In contrast, max pooling always performs the same function in the same way; applying a max filter to sub-regions of the image. A potential benefit of using strided convolutions is the fact that they may allow the model to

[6] https://github.com/ellisdg/3DUnetCNN

learn how to perform downsampling in a more optimal manner. The trade-off between both techniques is mainly between model parameters and a potentially more refined downsampling process. There is no definite answer to which method is superior, as it may still be the case e.g. that max pooling performs a superior selection of features during downsampling, depending on the training conditions. The encoder also features a dropout regularisation layer placed between every pair of $3 \times 3 \times 3$ convolutions in the context modules. The goal behind the use of dropout regularisation in this case was to minimise potential overfitting of the model.

In addition to the above, the biggest change from the standard U-Net to this model are the residual blocks. Residual connections in CNNs have been shown to improve results in previous iterations of the challenge, such as the work by Kamnitsas et al. (2016). In this instance, residual blocks are placed along the encoding pathway down to the bottleneck layer. The blocks take a combined input of features both before and after being passed through the convolution and nonlinearity in each convolutional block. An element-wise addition is then computed with both sets of features before passing on the result to the decoder via skip-connections. In theory, this joins the concepts and benefits of ResNets (He et al., 2016) and U-Net together. The potential of stronger optimisation techniques using residual layers combined with the parameter efficiency of U-Net could allow for better predictions without creating massive deep networks.

The decoder part of the network has also been altered slightly from the standard U-Net; upsampling layers have replaced transposed convolutions, as Isensee et al. (2017) claim this was shown to remove checkerboard artifacts from the output segmentation. Secondary segmentation maps are also generated at every level of the decoder using $1 \times 1 \times 1$ convolutions. These secondary segmentation maps are combined using element-wise additions, with the objective being to generate another set of predictions at each level of the decoder, ultimately refining the final model prediction. A final element-wise addition is performed with the topmost segmentation layer to generate the output. This is the concept of deep supervision mentioned in Section 2.2.

The original repository for this network included the facility to use a smaller input patch-size than the $128 \times 128 \times 128$ leveraged by Isensee et al. (2017). Smaller patches result in the model being less encumbered by parameters, allowing for modifications such as an increased batch size. As a result, some tests were carried out using $64 \times 64 \times 64$ dimensions. This number was chosen for two reasons; first as it is the next most logical size to use since it is half the original dimension, and secondly as the resolution of the input needs to be a cube for some operations in the pipeline. Unfortunately, the training accuracy obtained with the smaller input size was much worse when compared to the original dimensions. In addition, the model converged at a very early point during training. Both of these factors resulted in the patch-wise approach being ultimately scrapped from further testing. With all of this considered, the residual 3D U-Net used for this project consists of 8.3M model parameters.

3.1.5.3 | U-Net++

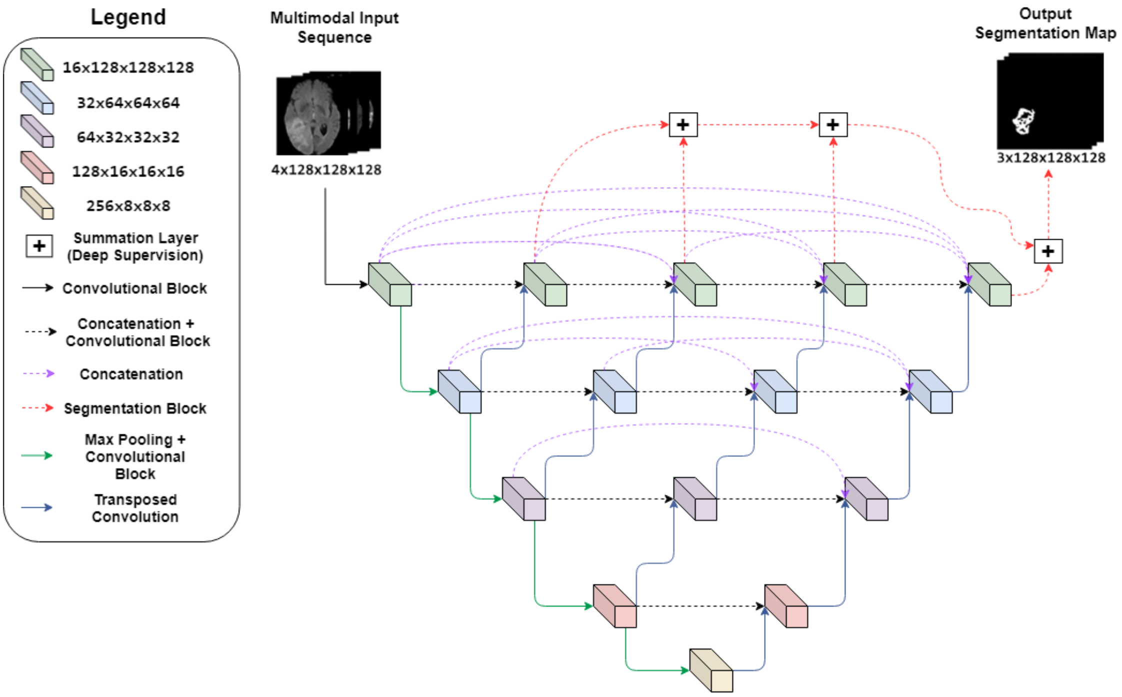


Figure 3.11: 3D U-Net++ architecture.

The final model is a U-Net++ (Zhou et al., 2018c) adaptation. What is instantly evident when observing Figure 3.11 is the convolutional blocks along every skip pathway between the encoder and decoder. The skip-connections present in standard U-Net (Ronneberger et al., 2015) were re-designed here to feature dense blocks as found in DenseNet (Huang et al., 2017); this is one of the main motivations for U-Net++.The dense blocks naturally introduce a much larger number of connections between corresponding layers in the encoder and decoder halves of the model. The intuition behind this densely connected architecture is that the 'semantic gap' between the encoder and decoder portions of the model is reduced, allowing for the optimiser to deal with a simpler learning problem (Zhou et al., 2018c).

As a result, this is the first model observed in this project which features upsampling operations being used in both the decoder and the encoder. The encoder once again makes use of max pooling operations in place of strided convolutions, much like the original U-Net. As the depth of the pathway increases, more upsampling operations occur. This process propagates the deeper feature maps up to the highest levels, combined with the features obtained from the middle of the skip-pathways. The end results of these upsampling operations are full-resolution segmentation maps along the very first skip-pathway (represented by the green blocks in Figure 3.11).

In the original U-Net++ paper, Zhou et al. (2018c) used all of these maps with the final segmentation as a combined output of the model. Zhou et al. (2018c) had averaged the output of each secondary map to obtain the final prediction as a form of 'deep supervision'. Whilst it was reported by Zhou et al. (2018c) that this did not have a massive benefit in terms of scores, in the worst scenarios it still performed comparably to not using deep supervision. As a result, the proposed approach in this project maintains these full-resolution segmentation maps, but applies 'traditional' deep supervision rather than averaging. This is achieved by placing an element-wise addition between each of the segmentation maps as well as the final segmentation. This allows for the full-resolution segmentation maps specific to U-Net++ to be maintained, whilst combining the generated segmentation results using additions rather than using the mean.

Moving on to the decoder, the number of intermediary blocks placed within the skip-connections also increases the level of feature map complexity being passed to the second half of the network. As stated previously, this should once again simplify the learning problem for the optimiser. The decoder's upsampling operations also make use of transposed convolutions, as with all upsampling operations in the model, much like standard U-Net. Since deep supervision is already being applied from the segmentation maps obtained at the top-most level, no secondary segmentation maps were generated from the decoder pathway as in the residual U-Net. This may also be observed from Figure 3.11 where the summation layers are only linked to the first level of skip-connections, with none linked to the decoding side of the model.

A major benefit of this model is that despite the relatively complex architecture, the original U-Net++ does not increase parameters much more than the residual approach. Furthermore, the proposed model in this project takes this a step further as only one convolution-normalisation-activation sequence is used in each block, which halves the model parameters. The reasons for using only a single convolution and no dropout layers will also be explained further in Section 4.1.3. Other slight contributors to the model's parameter efficiency are the fact that max pooling was used for downsampling, and that no dropout regularisation was used. Thus, the proposed approach is the most compact out of the three networks described in this study. The range of filter map resolutions was also maintained at [16..256] as in the previous architectures, as opposed to

Zhou et al. (2018c)'s original approach which used a range of [32..512]. Combined with the benefits of using dense connections with U-Net, the low-parameter requirements of the model are strong benefits of this approach. With all of the modifications in place, the final parameter size of the model is 4.5M.

As a final remark, it is noteworthy that there is a Github repository[7] officially linked to the research by Zhou et al. (2018c), providing simple and detailed variants of U-Net++ with multiple backbones. The simple version of the code was used as a reference to ensure that the network architecture was being followed correctly since the dense connections make it rather complex. With that said, the simple approach only features a U-Net like approach with the addition of the dense concatenations. In the proposed model, convolution blocks make use of instance normalisation, omit the dropout layers, and also use a starting filter resolution of 16 rather than 32. Zhou et al. (2018c) also used a composite BCE Dice Loss rather than the weighted variation of Dice Coefficient used in this study. Furthermore, the proposed use of deep supervision also differs to that implemented by Zhou et al. (2018c), since element-wise additions are being used rather than averaging. Finally, $3 \times 3 \times 3$ segmentation kernels were used in place of $1 \times 1 \times 1$ kernels. Whilst the latter is the usual resolution for such layers for dimensionality reduction, very preliminary tests had shown good results with $3 \times 3 \times 3$ kernels, and these were maintained as a result. These changes along with the use of a single convolution-normalisation-activation per block differentiate the proposed model from the default U-Net++. A large number of the above modifications and other experiments related to the model will be discussed accordingly in Section 4.1.

[7] `https://github.com/MrGiovanni/UNetPlusPlus`

4

Results & Discussion

This chapter will first discuss the experiments performed on the proposed approach. Each experiment will be evaluated using the criteria described in Section 2.1.6. Following the individual experiments, a general summary will also be provided which provides a summarized table with all results. Referring to the results and evaluation, initial plans involved using the BraTS 2019 Test dataset, as discussed in Section 3.1.2.2. In this scenario the BraTS 2019 validation set would be used solely as a basis for model comparison/selection, with the testing set used for the final evaluation. Unfortunately, the test dataset is provided only upon receipt of short papers about solutions for the competition, which had a very early deadline around August 2019. The section on the CBICA portal for uploading the test dataset results is also not available beyond the competition date. Whilst the head organiser of the BraTS challenge was contacted to see if the test data would be provided for academic purposes (specifically this Masters dissertation), the same explanation was given and no progress was made in that regard.

Nonetheless, since the training process uses the BraTS training data and is thus blind to the validation set, it is still viable data for evaluation purposes. Reference is also made to the 2018 data; one should note that the 2018 validation samples are already present in the 2019 dataset. In addition, the 2019 dataset also includes 59 new samples, making it the best source to evaluate the proposed approach at the time of writing. Whilst providing results on the 2018 validation set is possible, this is arguably a redundant process since the proposed approach makes use of the 2019 data which are more robust and already contain the 2018 samples. Therefore, the scores for the experiments, results and evaluation in the sections below will be based solely on the 2019 validation dataset. Section 4.2 will exhibit the final scores obtained by the proposed approach, followed by comparisons against the internal and current state-of-the-art models in Section 4.3.

4.1 | Experiments

This section will describe the thought process and outcomes of the experiments for this project. All of the variations considered for the selected model will be discussed, as well as the reasons behind each decision. The experiments carried out cover a large number of variables, obtaining results from various changes in parameters. These include data augmentation, change in loss function, infrastructure deliberations such as dropout regularisation and use of two convolutions per block, as well as post-processing techniques. As discussed, the final tables for each experiment will be assessed on the BraTS 2019 Validation dataset. Note that calculation of the 'Average Improvement' was performed by first calculating the performance increase/decrease of a measure over each of the WT, TC, and ET categories and averaging the three scores. Due to the lengthy training time of the models, it is noteworthy that the results are not cross-validated over multiple training runs/folds. This is also addressed as future work in Section 5.3.

4.1.1 | Data Augmentation Tests

The original pipeline adapted from the repository made use of data augmentation techniques to generate synthetic samples as discussed in Section 3.1.4.1. One of the initial tests performed was to ensure that synthetic sampling does actually assist a model to generalise unseen samples better for this task.

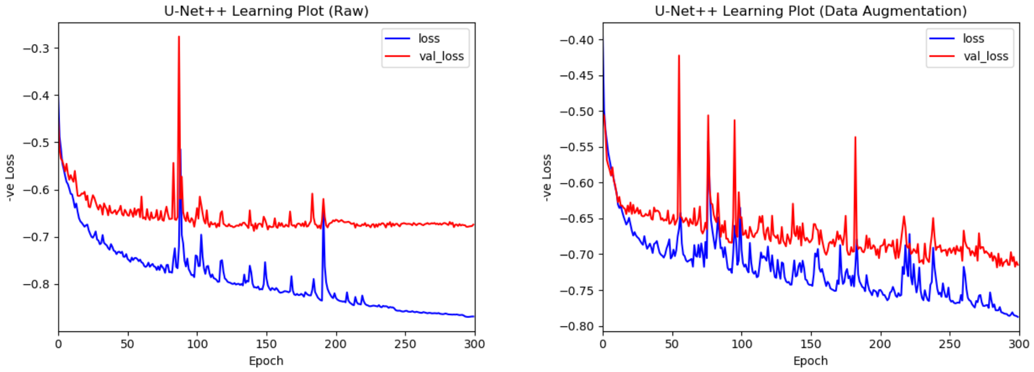


Figure 4.1: Convergence plots shown for: left - original images only, right - images passed through random rotations, axes flips, and transpositions.

The validation loss is the model's weighted Dice score on the unseen split of the training data. The first graph in Figure 4.1 shows how the validation loss of the images without data augmentation saturated very early on, whilst the training loss kept on decreasing substantially. This is possible evidence that the model was overfitting, as the training accuracy was improving without the model learning how to classify the unseen samples. The second sub-figure shows the training and validation loss decreasing more proportionally; we can observe a significant improvement in both the convergence proportion and final validation loss. To validate this claim, we evaluate both models as shown in Tables 4.1 and 4.2.

Table 4.1: Dice Coefficient and Hausdorff Distance for U-Net++ with and without data augmentation. Best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++	0.6505	0.8474	0.7285	8.2741	11.9739	9.6433
U-Net++ (Data Augmentation)	**0.6920**	**0.8709**	**0.7824**	**6.8001**	**8.3279**	**9.4997**
Average Improvement		5.52%			16.58%	

Table 4.2: Sensitivity and Specificity for U-Net++ with and without data augmentation. Best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++	0.7080	0.8606	0.7078	**0.9979**	0.9922	**0.9971**
U-Net++ (Data Augmentation)	**0.7208**	**0.8654**	**0.7655**	**0.9979**	**0.9945**	0.9969
Average Improvement		3.51%			0.07%	

The results demonstrate that data augmentation substantially improved the classification capabilities of the model. As an added note, an attempt was also made to implement elastic deformations, as Ronneberger et al. (2015) had observed that they had a positive effect when training the original U-Net. Unfortunately, the elastic deformations were causing the weighted Dice Coefficient to converge at a very early point in training with low scores. This was also the case when using lower parameters for the filter. As a result, the latter technique was scrapped from the data augmentation pipeline.

4.1.2 | Optimisation Function

The loss function is another factor which heavily influences training of a model. As discussed in Section 3.1.4, a multi-class dice coefficient function was used to train the model. Two other viable loss functions were considered, one of which was the IoU, as implemented by Kayalibay et al. (2017). Starting with the IoU, the function is identified by Equation 4.1.

$$
\begin{aligned}
L &= -\frac{1}{K}\sum_{k\in K}\frac{Y_k \cap \hat{Y}_k}{Y_k \cup \hat{Y}_k} \\
L &= -\frac{1}{K}\sum_{k\in K}\frac{Y_k \cap \hat{Y}_k}{Y_k + \hat{Y}_k - (Y_k \cap \hat{Y}_k)}
\end{aligned}
\tag{4.1}
$$

Equation 4.1 shows why this loss function is known as Intersection-Over-Union, as it is calculating the set intersection of the ground truth Y and prediction $\hat{Y}$ and dividing by their set union. The addition of the $\frac{-1}{K}$ coefficient and summation over K is to cater for the three target classes. Unfortunately, when attempting to use the above equation as a cost function, the learning process was stagnating earlier compared to the other losses. Whilst some other modifications may be applicable to the equation to tailor it to the problem, a decision was made to move on to testing the composite BCE dice loss. The latter had been used for training Zhou et al. (2018c)'s original U-Net++ model. Thus, it was considered beneficial to train a model using the BCE dice loss as in the original work. Equation 4.2 shows the composite loss function adapted from the repository linked to Zhou et al. (2018c).

$$
\begin{aligned}
L &= 0.5 * BCE - DSC \\
L &= 0.5 * \frac{1}{K}\sum_{k\in K}(Y_k * \log(\hat{Y}_k) + (1 - Y_k) * \log(1 - \hat{Y}_k)) - DSC \\
L &= 0.5 * \frac{1}{K}\sum_{k\in K}(Y_k * \log(\hat{Y}_k) + (1 - Y_k) * \log(1 - \hat{Y}_k)) - \frac{1}{K}\sum_{k\in K}\frac{2(Y_k \cap \hat{Y}_k) + \alpha}{(Y_k + \hat{Y}_k) + \alpha}
\end{aligned}
\tag{4.2}
$$

As with the other loss function equations, Y and $\hat{Y}$ refer to the ground truth and model prediction, and K to the three label classes. The standard equations for binary cross entropy and the weighted multiclass dice loss are substituted for the BCE and DSC terms. Implementing the composite loss was very simple; the BCE was first calculated using the Keras *binary_crossentropy* function. The weighted dice coefficient already used as the loss function for the model was then used for the DSC term. Unlike the IoU, the BCE dice function did not plateau during training, and the proposed architecture was trained with this loss function and the standard training parameters.

Table 4.3: Dice Coefficient and Hausdorff Distance for U-Net++ with multiclass Dice loss and binary crossentropy composite loss functions. Best scores in bold.

Model	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++	**0.6920**	**0.8709**	**0.7824**	**6.8001**	**8.3279**	**9.4997**
U-Net++ (BCE Dice Loss)	0.6876	0.8616	0.7656	7.5247	9.5144	11.5140
Average Improvement		-1.28%			-15.46%	

Table 4.4: Sensitivity and Specificity for U-Net++ with multiclass Dice loss and binary crossentropy composite function. Best scores in bold.

Model	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++	0.7208	0.8654	0.7655	**0.9979**	**0.9945**	**0.9969**
U-Net++ (BCE Dice Loss)	**0.7278**	**0.8925**	**0.7910**	**0.9979**	0.9912	0.9940
Average Improvement		2.48%			-0.21%	

The result for the multiclass Dice coefficient and BCE Dice loss comes as a trade-off between segmentation quality and the detection of false negatives through sensitivity. As discussed in Section 2.1.6, both measurements are important as they contribute to a more accurate quantification of the tumour. From the tabulated results, we can observe that the BCE dice loss consistently performed worse in terms of segmentation quality. For sensitivity, there were improvements for every class, whilst the BCE dice loss detected false positives less efficiently than the baseline model. Whilst sensitivity is an important measurement, the final decision was to keep the approach with the highest Dice scores for this study. This was in part substantiated by viewing related work where the main listed measurement is generally the Dice Coefficient. As a result, it was decided to maintain the multiclass Dice as the loss function for the model.

4.1.3 | Doubling Convolutional Blocks

As discussed previously, the proposed model makes use of only one sequence of $3 \times 3 \times 3$ convolution, instance normalisation, and ReLU activation in the convolutional blocks. This architecture amounts to a total of 4.5M model parameters. These changes were mainly done to speed up testing by using a smaller model which still produced positive results, as opposed to the larger model published by Zhou et al. (2018c). However, it was still necessary to evaluate the 9M parameter variant which makes use of two convolution sequences per block, for two reasons; firstly, since the original study follows this architecture, and secondly since most regular U-Net adaptations also make use of at least two convolutions per block if not the whole sequence including normalisation. To compare the two architectures, a model making use of two convolution-normalisation-activation sequences per block was trained using the standard parameters.

Table 4.5: Dice Coefficient and Hausdorff Distance for U-Net++ with and without two convolutional layers per block. Best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++	0.6920	**0.8709**	**0.7824**	6.8001	8.3279	**9.4997**
U-Net++ (Two Convs)	**0.6931**	0.8690	0.7778	**5.2130**	**7.6872**	9.6055
Average Improvement	-0.22%			9.97%		

Table 4.6: Sensitivity and Specificity for U-Net++ with and without two convolutional layers per block. Best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++	**0.7208**	0.8654	0.7655	0.9979	**0.9945**	**0.9969**
U-Net++ (Two Convs)	0.6893	**0.8912**	**0.7957**	**0.9984**	0.9922	0.9956
Average Improvement	0.85%			-0.10%		

Tables 4.5 and 4.6 show the results of the 4.5M vs the 9M parameter model, with the two main improvements being the Hausdorff Distance and the sensitivity measurement. The Hausdorff Distance increase is substantial, showing that the worst case error of the segmentation improved. However, the larger model also obtained a lower average Dice score. The latter shows that whilst the maximum error was minimized, the larger model still performed segmentation slightly worse on average. The sensitivity score had a 0.85% improvement, as the single convolution model only performed the ET segmentation with less false negatives, and obtained a higher specificity for the WT and TC classes.

The Dice score shows that the improvement in Hausdorff Distance is not a sign of improved segmentation on the average. However, the larger 9M parameter model did also have a slightly higher sensitivity. Following the same thought process as previous experiments, a decision was made to consider the smaller model for further testing. Even if one were to favour sensitivity and the Hausdorff Distance and opt for the 9M parameter model, they would have to decide whether the minor performance improvements adequately compensate for the increase in model size, and by extension training time and hardware requirements. As a result, the smaller model was maintained as the size improvements do not come with significant drawbacks to model performance.

4.1.4 | Dropout Regularisation

Another test performed on the model was the use of dropout regularisation. Dropout is commonly used in neural networks as it may avoid overfitting. In the best case scenario, dropout should allow a network to generalise better, much like the data augmentation tests in Section 4.1.1. It is noteworthy that there is no mention of dropout in the original paper by Zhou et al. (2018c). Nonetheless, this test was attempted as an additional check to see if the model's predictions would improve. For this test, a *SpatialDropout3D* layer was used in Keras with a 0.3 dropout rate. Each layer was placed at the end of the convolutional blocks forming part of the encoder half of U-Net++, following the same thought process of Isensee et al. (2017), which only uses dropout in the encoder. Tables 4.7 and 4.8 show the comparison of U-Net++ with and without dropout layers.

Table 4.7: Dice Coefficient and Hausdorff Distance for U-Net++ with and without dropout regularization. Best scores in bold.

Model	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++	0.6920	**0.8709**	**0.7824**	**6.8001**	**8.3279**	**9.4997**
U-Net++ (Dropout)	**0.6940**	0.8589	0.7684	7.7722	8.3574	9.7855
Average Improvement		-0.95%			-5.89%	

Table 4.8: Sensitivity and Specificity for U-Net++ with and without dropout regularization. Best scores in bold.

Model	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++	0.7208	**0.8654**	**0.7655**	**0.9979**	**0.9945**	**0.9969**
U-Net++ (Dropout)	**0.7378**	0.8653	0.7603	0.9978	0.9933	0.9968
Average Improvement		0.56%			-0.05%	

Summarising the tabulated results, the only improvement obtained using dropout is an increased ET sensitivity score. Both the Dice Coefficient and Hausdorff Distance were negatively affected by the introduction of dropout, especially the latter. This shows that the regularisation impacted the segmentation performance of the model on all classes, only improving the possibility of finding false negatives for the ET class. Whilst the latter is a beneficial outcome, the overall improvement to sensitivity is still only 0.56%. Weighing the results presents the conclusion that the improvement in sensitivity is outweighed by the negative contribution to the Dice and Hausdorff scores, and dropout was scrapped from the pipeline as a result. Nonetheless, the tests above are only for one dropout value, this could be tested further as addressed in Section 5.3.

4.1.5 | Post-Processing

Once the predictions have been generated by the model, there may still be further adjustments which can be applied to the output to improve segmentation quality. Apart from the mean scores, the IPP also provides every individual sample's scores. From inspecting the statistics, it was noticed that there were a number of subjects which had a Dice score of 0 for the enhancing tumour. Figure 4.2 shows a box plot outlining how individual samples affected the Dice Scores for the baseline proposed model.

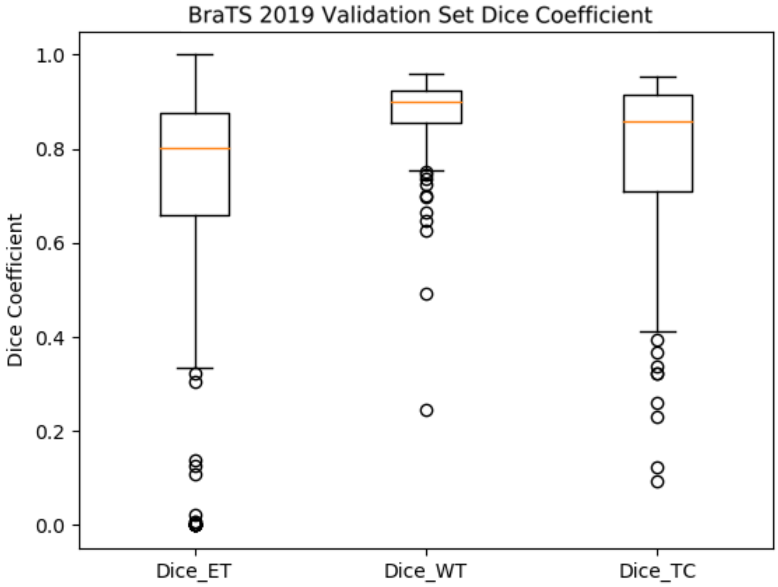


Figure 4.2: Box Plots for Dice Coefficient without post-processing.

One may observe that there is quite a number of outliers for the *Dice_ET* category which have a score of 0. Conversely, we can also see that some of the samples had an ET score of 1, meaning that the absence of an ET segment was detected successfully in some instances. Normally, cases which do have an ET segment will have floating point values between 0 and 1 since getting a voxel-perfect segmentation prediction is rare. A score of 0 is given if e.g. the prediction has 2 ET voxels when there should be none in the original image. Naturally, the opposite logic is also true as a Dice score of 0 will be provided if no ET voxels are predicted by the model when there should be at least one in the scan. To test for these cases, the initial checks involved calculating label frequency ratios between voxels in the NET, ET, and edema classes for the cases with an ET Dice Coefficient of 0.

Table 4.9: BraTS'19 Validation cases with ET Dice score of 0.

Label	Dice Coefficient			Voxels		
	ET	WT	TC	NET	ET	Edema
BraTS19_TCIA09_248_1	0	0.9210	0.5672	16403	83	41162
BraTS19_TCIA10_127_1	0	0.8941	0.8713	10741	19	4975
BraTS19_TCIA10_195_1	0	0.9476	0.7854	50453	1514	82176
BraTS19_TCIA10_232_1	0	0.8965	0.6610	68831	173	71083
BraTS19_TCIA10_609_1	0	0.9514	0.9200	47509	7	25112
BraTS19_TCIA10_614_1	0	0.9301	0.3670	4187	73	21716
BraTS19_TCIA11_612_1	0	0.8155	0.8531	9314	0	14084
BraTS19_TCIA13_619_1	0	0.9033	0.7342	17653	0	62305
BraTS19_TCIA13_648_1	0	0.7823	0.6530	38882	0	26505
BraTS19_TCIA13_652_1	0	0.9258	0.1222	12362	0	19084

When observing Table 4.9, we can see that the occurrences of ET Dice scores being 0 are grouped into two main categories. The first set are cases where a very small amount of enhancing tumour (class 4) voxels were predicted. The Dice Score of 0 in these instances shows that there should have been no ET occurrences for these samples. The next group is composed of the last four entries in Table 4.9; predictions with 0 ET voxels when there are actual ET voxels in the scans. Of these two groups, the first set has the most potential to be rectified, seeing as the minuscule amount of ET pixels may be a detectable pattern in the dataset. To establish that this is not a uniquely occurring pattern on the validation set, we also test the hypothesis on the training set as shown in Table 4.10.

Table 4.10: BraTS'19 Training cases with ET Dice score of 0.

Label	Dice Coefficient			Voxels		
	ET	WT	TC	NET	ET	Edema
BraTS19_2013_29_1	0	0.9339	0.7173	10,938	24	41,162
BraTS19_2013_9_1	0	0.8964	0.7669	9,596	10	4,975
BraTS19_TCIA12_466_1	0	0.9105	0.5029	9,681	567	82,176
BraTS19_TCIA13_630_1	0	0.9145	0.7322	25,012	182	71,083

From Table 4.10 we can observe that the first of the two problem groups was also observable on some instances from the training set. In addition, the ground truths for the training set are available to provide visual aid for problematic samples which may be detected. Figure 4.3 shows the ground truth and prediction examples for one of the samples listed in Table 4.10.

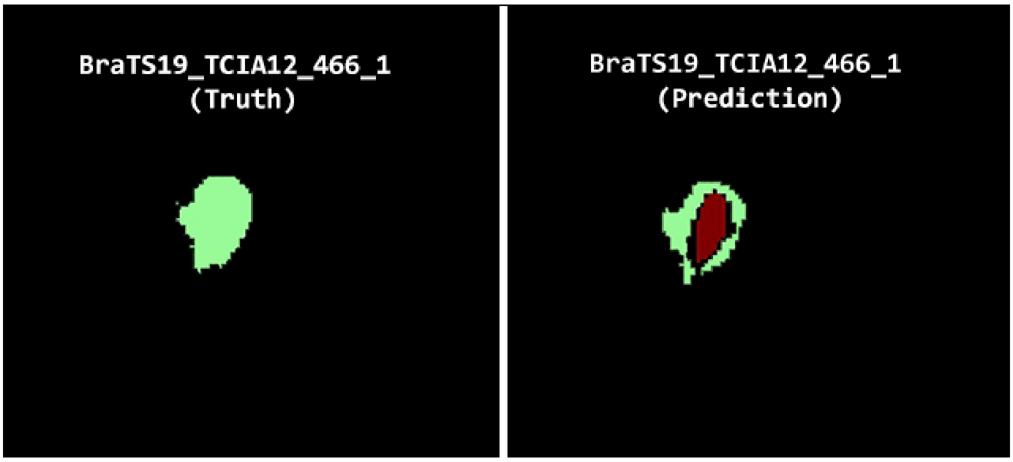


Figure 4.3: Left - Ground truth, Right - Prediction.

Visually, one can already confirm that the enhancing tumour mask on the right is not present in the ground truth. On closer inspection, it appears this should have been classified as an edema segment. This, as well as a check on the volume using *numpy*'s *count_nonzero* function further confirms the hypothesis that in this case, the dice score of 0 is given when there are no ET voxels in the expert ground truth. Transitioning back to the BraTS'19 validation dataset, the ratios shown in Table 3.14 were obtained.

Table 4.11: Ratios of occurrences for label 4 vs. labels 1 and 2.

Label	Dice Coefficient			Pixels	Pixels
	ET	WT	TC	ET/NET	ET/(NET + Edema)
BraTS19_TCIA09_248_1	0	0.9210	0.5672	0.0051	0.0014
BraTS19_TCIA10_127_1	0	0.8941	0.8713	0.0018	0.0012
BraTS19_TCIA10_195_1	0	0.9476	0.7854	0.0300	0.0114
BraTS19_TCIA10_232_1	0	0.8965	0.6610	0.0025	0.0012
BraTS19_TCIA10_609_1	0	0.9514	0.9200	0.0001	0.0001
BraTS19_TCIA10_614_1	0	0.9301	0.3670	0.0174	0.0028

The threshold in this case seems to be around 0.03 for cases where the ET should not have been predicted. This presents the possibility of solving these cases by thresholding instances where the ratio is less than 0.04, with the very slight increase from 0.03 being for contingency. The next step is to check the entire validation set to ensure that such a threshold would not end up destabilising correct predictions. All cases where the label 4 ET/NET ratio is less than 0.04 for the validation set are shown in Table 4.12.

Table 4.12: BraTS'19 Validation set cases which would be erroneously thresholded.

Label	Dice Coefficient			Pixels
	ET	WT	TC	ET/NET
BraTS19_TCIA10_220_1	0.1075	0.9302	0.8091	0.0019
BraTS19_TCIA10_239_1	0.5526	0.9174	0.7849	0.0126
BraTS19_TCIA10_647_1	0.3333	0.9240	0.6086	0.0010
BraTS19_TCIA12_339_1	0.0076	0.8549	0.0936	0.0044
BraTS19_TCIA13_611_1	0.0215	0.7591	0.2607	0.0053
BraTS19_TCIA13_616_1	0.6669	0.9184	0.8399	0.0135
BraTS19_TCIA13_617_1	0.1374	0.8096	0.7085	0.0020
BraTS19_TCIA13_638_1	0.3217	0.8565	0.6007	0.0307
BraTS19_TCIA13_643_1	0.0061	0.8371	0.6632	0.0003

Table 4.12 shows that thresholding by ratio does lead to quite the number of samples having their ET predictions thresholded and thus turned into False Negatives. Alternately, one may consider the possibility of using a constant voxel threshold rather than an ET/NET ratio. From the ET pixels listed in Table 4.10, one may observe that the largest two sets of occurrences for the false positives are 173 and 1514. Thresholding below 1514 creates a far broader window for error compared to 173, thus a number close to 173 would be ideal as a boundary. Table 4.13 shows the number of cases which would be wrongly processed with a constant threshold of under 200 ET voxels.

Table 4.13: BraTS'19 Validation set cases which would be erroneously thresholded (constant threshold).

Label	Dice Coefficient			ET Voxels
	ET	WT	TC	
BraTS19_TCIA10_220_1	0.1075	0.9302	0.8091	65
BraTS19_TCIA10_647_1	0.3333	0.9240	0.6086	6
BraTS19_TCIA12_339_1	0.0076	0.8549	0.0936	8
BraTS19_TCIA13_611_1	0.0215	0.7591	0.2607	148
BraTS19_TCIA13_617_1	0.1374	0.8096	0.7085	74
BraTS19_TCIA13_643_1	0.0061	0.8371	0.6632	31

Fewer false negatives are generated with this post-processing method, and the majority of the cases affected have very low ET Dice scores as well. Thus, the approach making use of a constant rather than a ratio appears to be the superior technique. Nonetheless, to ensure that the WT and TC scores are not negatively impacted by the change, the final step is to apply the constant threshold to 0 on the model predictions and evaluate the results on the online portal.

Table 4.14: Dice Coefficient and Hausdorff Distance for U-Net++ with and without post-processing. Best scores in bold.

Variation	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++	0.6920	0.8709	**0.7824**	6.8001	8.3279	9.4997
Post-Processed (Ratio)	0.7113	0.8709	**0.7824**	5.2052	8.3330	9.4990
Post-Processed (Constant)	**0.7192**	**0.8712**	0.7817	**4.6861**	**8.2157**	**9.4748**
Average Improvement (Ratio)		0.93%			7.80%	
Average Improvement (Constant)		1.29%			10.90%	

Table 4.15: Sensitivity and Specificity for U-Net++ with and without post-processing. Best scores in bold.

Variation	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++	0.7208	0.8654	**0.7655**	0.9979	**0.9945**	**0.9969**
Post-Processed (Ratio)	**0.7248**	0.8653	**0.7655**	0.9979	**0.9945**	**0.9969**
Post-Processed (Constant)	0.7232	**0.8671**	0.7630	**0.9980**	0.9944	**0.9969**
Average Improvement (Ratio)		0.18%			0%	
Average Improvement (Constant)		0.07%			0%	

When observing the results obtained in Tables 4.14 and 4.15, it is clear that the post-processed approaches obtained superior Dice and Hausdorff Distance scores. Between the two, the approach using the constant threshold performed the best as hypothesised. In terms of sensitivity and specificity, there were improvements for the ET and WT categories, and a slight drop-off for the constant threshold variant in TC sensitivity. Nonetheless, from these results one may derive the conclusion that the segmentation quality was improved by post-processing, both in terms of Dice Score and the Hausdorff Distance. Sensitivity and Specificity deviated only slightly, yet still retained a slight improvement on average. Once again, this shows that the post-processing operation used generates a superior segmentation. Whilst the threshold was applied in hindsight of the results on the training/validation set, it was still applicable to unseen samples within the dataset and provided noticeable improvements across all evaluation criteria. It was decided to retain the constant threshold going forward since it obtained the best improvements in segmentation scores.

4.1.6 | Using Upsampled Features Directly in Skip-Connections

The final experiment to be discussed is an initial build of U-Net++, developed late in 2019. The difference between this model and the current proposed U-Net++ is mainly how the skip-connections were propagated. In the present version, Zhou et al. (2018c)'s approach is followed, and all of the dense connections are leveraged normally. The conventional method dictates that blocks in the encoder have upsampling layers which are then passed through a $3 \times 3 \times 3$ convolution, and concatenated along a skip pathway. For this experiment, the upsampled features were concatenated directly along the skip-pathway without the convolution and intermediate concatenation being applied. This essentially reduces some of the density of the feature-map connections and frees up some parameters from the model.

In addition, deep supervision was only applied between the last two segmentation maps in the model rather than all four. The result was a slightly smaller model which was always runnable locally on the GTX 970 GPU, removing the cloud requirement for training the model. The work implementing this model was submitted to two different international bodies; the Organisation of Human Brain Mapping (OHBM) for the 2020 poster submission event, and to the IEEE MELECON 2020 conference as a paper submission. Both the U-Net++ models submitted were accepted and follow the above conventions, with the OHBM model being one training cycle consisting of 100 epochs, and the MELECON model for 300 epochs. Thus, the main difference between the OHBM and MELECON submissions is the number of epochs used. In their respective works, the U-Net++ models were compared to a residual U-Net similar to the model used internally (8.8M parameters rather than 8.3M), and evaluated on the unseen samples from the training set using the Dice Coefficient formula locally, reflecting the development stage of this project at the time. Both models make use of data augmentation and converge following a training pattern similar to the proposed approach.

To compare these models with the 'fully dense' U-Net++ as proposed in this project, we evaluated their results on the BraTS 2019 validation set. The main reason for the validation set being used is that there was a variation in training splits between the conference submissions and the group of models discussed in this project. The reasons behind the change in split were firstly to slightly correct the training-validation split to be an exact 80-20, and secondly, some interim experiments led to a rotation in the datastore's sample order prior to starting off the experiments documented in this project, which all follow the same setup. The validation set allows for a completely 'blind' dataset for either set of experiments. In addition, the validation set is a more robust dataset for comparisons as there are almost double the samples as the holdout set from the 80-20 split. The conference models had been run with data augmentation and no post-processing, thus we use the proposed model without thresholds for the comparison.

Table 4.16: Dice Coefficient and Hausdorff Distance for U-Net++ and the conference models. Best scores in bold.

Configuration	Epochs	Parameters	Dice Coefficient			Hausdorff Distance		
			ET	WT	TC	ET	WT	TC
U-Net++ (OHBM)	100	4.4M	0.6510	0.8642	0.7202	8.4248	8.4060	10.6127
U-Net++ (MELECON)	300	4.4M	0.6711	0.8631	0.7592	7.3657	9.0543	10.0087
U-Net++ (Proposed)	300	4.5M	**0.6920**	**0.8709**	**0.7824**	**6.8001**	**8.3279**	**9.4997**

Table 4.17: Sensitivity and Specificity for U-Net++ and the conference models. Best scores in bold.

Configuration	Epochs	Parameters	Sensitivity			Specificity		
			ET	WT	TC	ET	WT	TC
U-Net++ (OHBM)	100	4.4M	**0.7277**	0.8452	0.6772	0.9970	**0.9954**	**0.9977**
U-Net++ (MELECON)	300	4.4M	0.6804	0.8606	0.7445	**0.9980**	0.9940	0.9967
U-Net++ (Proposed)	300	4.5M	0.7208	**0.8654**	**0.7655**	0.9979	0.9945	0.9969

From Tables 4.16 and 4.17 we may observe that the proposed model leads in terms of segmentation, as it obtained the highest scores for both the Dice Coefficient and the Hausdorff Distance. The sensitivity values are slightly more interesting as the OHBM model actually obtained the highest sensitivity for the ET segment, slightly surpassing even the proposed approach. The implication from this result is that whilst the proposed approach does produce more accurate predictions, it is also slightly more likely to classify the enhancing tumour as a false negative from the increased predictions. Nonetheless, the discrepancy is quite small, and the proposed model obtained higher sensitivity scores for the whole tumour and tumour core. The specificity values were quite similar, with the OHBM model leading for the whole tumour and tumour core, and the MELECON model obtaining very similar values to the proposed U-Net++.

It is noteworthy that the OHBM model actually had a better ET sensitivity compared to the MELECON model. Both models share the same architecture and training parameters, yet the 100 epoch variant managed to obtain a higher sensitivity score than the run for 300 epochs. This may imply that longer training cycles improve segmentation quality, but result in ET false negatives increasing. This is also true for the proposed model which was trained for 300 epochs and obtained a lower ET sensitivity. Otherwise the results obtained are coherent, as the proposed model is the 'comprehensive' version of the three systems, implementing the entire arsenal of dense connections and all secondary segmentation maps. Thus, it is sensible that the proposed U-Net++ performs the most optimal segmentation. Whilst the sensitivity and especially the specificity scores were slightly higher for the other models, the differences are not substantial.

4.1.7 | Summary of Experiments

This section will serve to aggregate all of the results above into one table. The intention behind this is to provide a coherent look at the strengths of each model configuration. Tables 4.18 and 4.19 show the results for all of the experiments.

Table 4.18: Dice Score and Hausdorff Distance for all experiments performed for this project. All models after the first make use of data augmentation. 'Baseline' refers to the U-Net++ model used for the previous comparisons. Best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
Baseline (No Data Augmentation)	0.6505	0.8474	0.7285	8.2741	11.9739	9.6433
OHBM	0.6510	0.8642	0.7202	8.4248	8.4060	10.6127
MELECON	0.6711	0.8631	0.7592	7.3657	9.0543	10.0087
BCE Dice Loss	0.6876	0.8616	0.7656	7.5247	9.5144	11.5140
Double Conv. Blocks	0.6931	0.8690	0.7778	5.2130	**7.6872**	9.6055
Dropout	0.6940	0.8589	0.7684	7.7722	8.3574	9.7855
Baseline	0.6920	0.8709	**0.7824**	6.8001	8.3279	9.4997
Baseline w/ Threshold - Ratio (0.04)	0.7113	0.8709	**0.7824**	5.2052	8.3330	9.499
Baseline w/ Threshold - Constant (200)	**0.7192**	**0.8712**	0.7817	**4.6861**	8.2157	**9.4748**

Table 4.19: Sensitivity and Specificity for all experiments performed for this project. All models after the first make use of data augmentation. 'Baseline' refers to the U-Net++ model used in previous comparisons. Best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
Baseline (No Data Augmentation)	0.7080	0.8606	0.7078	0.9979	0.9922	0.9971
OHBM	0.7277	0.8452	0.6772	0.9970	**0.9954**	**0.9977**
MELECON	0.6804	0.8606	0.7445	0.9980	0.9940	0.9967
BCE Dice Loss	0.7278	**0.8925**	0.791	0.9979	0.9912	0.9940
Double Conv. Blocks	0.6893	0.8912	**0.7957**	**0.9984**	0.9922	0.9956
Dropout	**0.7378**	0.8653	0.7603	0.9978	0.9933	0.9968
Baseline	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
Baseline w/ Threshold - Ratio (0.04)	0.7248	0.8653	0.7655	0.9979	0.9945	0.9969
Baseline w/ Threshold - Constant (200)	0.7232	0.8671	0.7630	0.9980	0.9944	0.9969

From Tables 4.18 and 4.19, it is clear that the two tests which provided substantial improvements to U-Net++ were the data augmentation and constant threshold post-processing. The improved generalisation from data augmentation increased every single metric, and the constant threshold improved the enhancing tumour dice score without heavily compromising the other metrics. For all experiments outside of post-processing, it was noted that the average Dice Coefficient of the baseline model was superior.

The other experiments which merit further discussion are the BCE Dice Loss and larger (Double Conv. Blocks) U-Net++. The BCE Dice Loss approach obtained higher sensitivity scores, but struggled against the baseline U-Net++ in the Dice score and Hausdorff Distance. This resulted in the weighted Dice Coefficient Loss being favoured as the improvements in sensitivity produced by the BCE Dice Loss did not outweigh the drop in segmentation accuracy. The larger U-Net++ showed improvements in the Hausdorff Distance and a slight improvement in average sensitivity. However, a decision was made to maintain the smaller model, primarily since the Dice Coefficient of the large model was still smaller than the baseline U-Net++. Using a model which is twice as large without bearing substantial improvements was not deemed worthwhile.

As a result of the above deliberations, the U-Net++ model with constant post-processing was selected as the final model for evaluation purposes in this project. It was noted that whilst this model had the strongest segmentation performance overall, other models may have their own benefits such as the BCE Dice Loss and larger model having slightly better sensitivity values. A further observation is that the results above may imply that the U-Net++ model is not considerably susceptible to variance across different training runs. Whilst data augmentation did provide a sizeable improvement in scores, this was a sensible outcome since the models would now be coming into contact with many new synthetic samples during training.

Conversely, the fluctuations in scores for all other tests were not as substantial, most likely being produced from the actual changes introduced by each experiment rather than training variance. As an example, the results obtained by the larger U-Net are reasonable as the model would be slightly better than the baseline model, if not for the increased parameters. Thus, it is probable that the architecture's dense nature allows for stable results in separate training runs. This hypothesis would ideally be confirmed via ensemble training, as discussed in Section 5.3 for future work.

4.2 | Results

At the onset of this project, the main objectives were to create a model capable of automatically segmenting multimodal 3D MR images, ready for viewing using standard MR image viewers. The other main objectives of the proposed approach included obtaining results adequate for a clinical setting, submitting conference work for potential feedback, and obtaining results comparable to the state-of-the-art on the BraTS 2019 dataset. The latter objective will be discussed in Section 4.3, where the model is compared to some published works evaluated on BraTS 2019 data.

The first set of objectives will be discussed first, as they are easier to decompose and may thus be answered in a straightforward manner. The input requirements of the model are 4D MR images, with the 4th dimension being the image channel of the 3D volumes (e.g. T_1ce). A reason for this is that as discussed in Section 2.1.1, each sequence has particular contrast distributions and strengths due to the differences in TR and TE and their effects on T_1 and T_2 timings. Whilst one may argue that certain modalities such as T_1w are better suited for strictly segmenting the brain rather than tumours, most of the recent related literature keeps all of the provided sequence types from the BraTS dataset. Therefore, the proposed approach will consider all of the modalities at once, with no extra time investment used for handling individual sequence types.

The system completes the pipeline from preprocessing the raw files to training automatically, provided that the raw dataset is present on disk. There is no interaction between the user and the data such as the tumour region pre-selection in Bakas et al. (2015). The raw files are preprocessed using bias field correction and cropped to the resolution expected by the model. Standardisation is then applied to each set of MR images to normalise the intensity distribution. In training mode, training/validation splits are generated (if not already present on disk), and training starts using the selected model.

If the prediction pipeline is executed, the trained model weights are then used to generate the model's segmentation maps from the selected validation dataset. These files are readily accessible by any standard MR Viewer, tested internally on *Mango* for viewing and *MRtrix3* for analysis and transformations. Regarding publications, the work submitted to the OHBM and MELECON discussed in Section 1.3 and 4.1.6 show that this sub-goal was achieved successfully.

Moving on to the more complex objectives, we may begin with the generation of segmentation maps which are clinically acceptable. The effort discussed in the experiments section was carried out to produce the highest quality segmentation results. The term 'clinical acceptance' encompasses quite a broad definition; in an ideal world medical professionals would have algorithms which are close to 100% accurate. Having said that, we can observe the performance of the proposed approach, including analysis of the score distribution for the validation data. The initial metrics to be analysed are the Dice Coefficient and Hausdorff Distance, assessing the model's segmentation capabilities. The CBICA IPP portal provides a detailed breakdown of the results, as shown in Table 4.20.

Table 4.20: Mean, Standard Deviation, and Quartiles for the Dice Coefficient and Hausdorff Distance for the proposed model on the BraTS 2019 Validation Set.

Measure	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
Mean	0.7192	0.8712	0.7817	4.6863	8.2157	9.4752
Standard Deviation	0.2811	0.0934	0.1914	6.5129	9.8122	12.3579
Median	0.8167	0.8988	0.8592	2.4495	5.3852	6.1644
25th Quantile	0.6849	0.8549	0.7026	1.4142	3.6056	3
75th Quantile	0.8841	0.9234	0.9139	4.3872	8.3066	9.4868

As previously mentioned, Havaei et al. (2017) discovered that from the 2% of pathological pixels in the scan, over half of the distribution were edema pixels. The class imbalance in the 2019 dataset towards the edema is fairly well reflected; the whole tumour is much more well represented, leading to a higher mean Dice Coefficient, as well as a lesser standard deviation to the two other classes. The lesser ET scores may also be in part due to cases such as LGGs which do not have an ET segment. A model may fail to replicate this structure since LGGs have less than 25% representation in the dataset compared to HGG subjects. We may go a step further and have a look at a visual representation of the metrics in the above tables, including outlier patient samples. Box and whisker plots for the Dice Coefficient and Hausdorff Distance are shown in Figure 4.4.

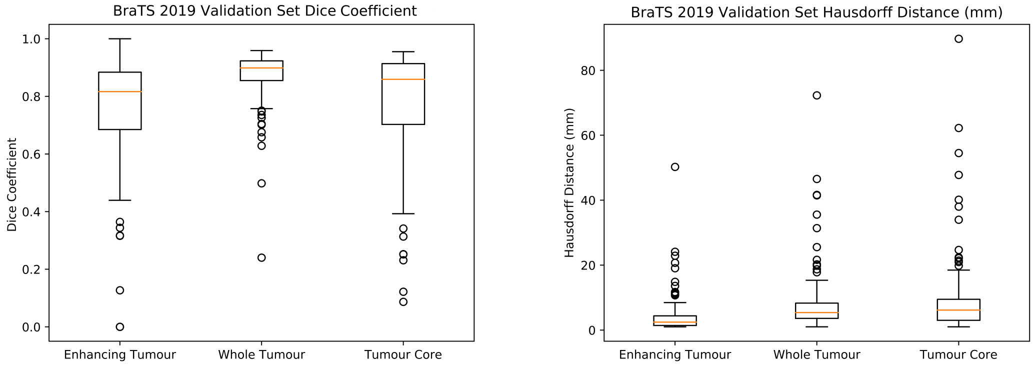


Figure 4.4: Box and whisker plots for the 2019 BraTS validation set Dice Coefficient and Hausdorff Distance.

The Dice Coefficient plots show that the enhancing tumour and tumour core saw quite a variation in the 125 samples present in the validation data. The whole tumour on the other hand, was very well segmented, as expected. The outliers are spread out for all three classes, with a known case being the previously observed samples with an ET Dice Coefficient of 0. The whole tumour outliers appear to be grouped closer together; however, one must consider that the interquartile range of the class is very small, with a small distance between the whiskers as well. Thus, the outliers still cover a considerable WT Dice Coefficient range. Nonetheless, the median Dice scores for each class are above 0.8. Since the Dice Coefficient represents the segmentation accuracy between the model predictions and ground truths, these values exhibit that the median value of segmentations was a fairly high number.

One observation when comparing the box plots for the Dice Coefficient and Hausdorff Distance is that the distributions for ET and WT change considerably. Conversely, the TC plots are similar, apart from the outliers. When weighing these comparisons, one must keep in mind that the Hausdorff Distance represents the largest segmentation error. Whilst the interquartile range is quite small, the number of outliers is much larger. This could partially be due to the nature of the measure which takes into account extreme errors for each sample. This is also substantiated by the WT and TC having long whiskers, which suggests that the range of Hausdorff values varies greatly in both cases. Nonetheless, the interquartiles ranges for both sets of plots are fairly well contained, which implies that the results are reliable.

Analysis of how the outliers in each set of plots appear visually is not straightforward, as the ground truths for the validation data are kept on the online portal and not distributed with the dataset. One test we may attempt is to correlate outliers between the Dice Coefficient and Hausdorff Distance. In this scenario, the whole tumour will be used as an example, as it was found to have the most outlying nonzero points.

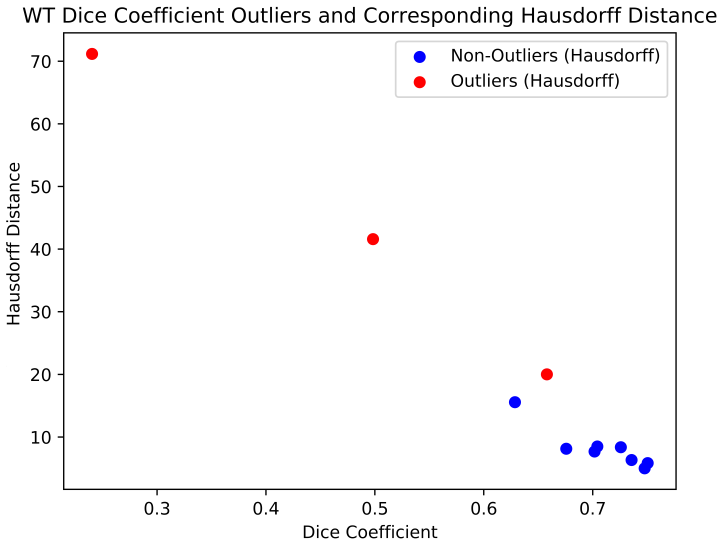


Figure 4.5: Scatter plot showing validation samples with an outlying WT Dice Coefficient and corresponding Hausdorff Distance.

Figure 4.5 shows that the values of Dice Coefficient outliers vary fairly proportionally with the corresponding Hausdorff Distance. The other observation from the plot is that Dice Coefficient outliers do not necessarily translate to Hausdorff Distance outliers, as only three of the Dice outlier samples were also Hausdorff outliers. As a result, we can confirm that whilst the proportion of values for both metrics is maintained, the outlier sample distribution is quite different. Moving on to sensitivity and specificity, the CBICA portal reported the results shown in Table 4.21.

Table 4.21: Mean, Standard Deviation, and Quartiles for Sensitivity and Specificity for the proposed model on the BraTS 2019 Validation Set.

Measure	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
Mean	0.7232	0.8671	0.7631	0.9980	0.9944	0.9969
Standard Deviation	0.2941	0.0965	0.1997	0.0045	0.0063	0.0059
Median	0.8319	0.8916	0.8431	0.9991	0.9963	0.9988
25th Quantile	0.6709	0.8352	0.6613	0.9978	0.9933	0.9968
75th Quantile	0.9351	0.9363	0.9047	0.9999	0.9984	0.9995

The sensitivity values of the model are fairly close to the Dice Coefficient values for each class. The specificity is more complicated to draw correlations with, as the values are extremely high, with a very small standard deviation. An expected correlation observed from Table 4.21 is that higher sensitivity results in a lower specificity value for the particular class. We once again refer to the box plots for both the sensitivity and specificity to analyse each metric more closely.

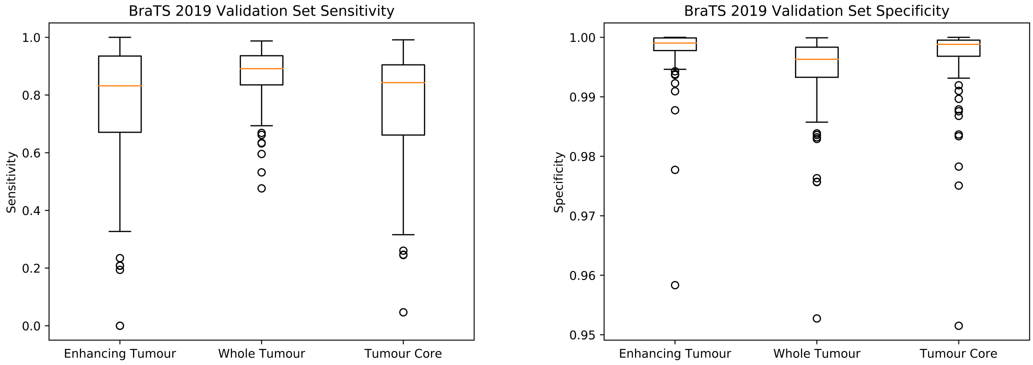


Figure 4.6: Box and whisker plots for the 2019 BraTS validation set Sensitivity and Specificity.

The first observation made is that the interquartile ranges and whisker distance of the sensitivity are considerably larger than the Dice Coefficient. Whilst this results in less outlying points, it does mean that the sensitivity is slightly less reliable as a measurement compared to the Dice scores. The specificity plots' interquartile ranges are near opposite of sensitivity, however one must also bear in mind that the specificity values are mostly between 0.99 and 1, which slightly obscures the results.

Nonetheless, high specificity values show that the model is very good at avoiding false positives. This aids in the assumption that the model would be capable of avoiding erroneous classification across classes. Another important question to ask with regard to false positives outside of the target classes is how the proposed system behaves when classifying healthy brains. This is a bit more difficult to assess; whilst it is arguable that a high specificity would imply that the model is very capable of avoiding false positives, one must remember that the true negatives e.g. for the WT class also consider the false negatives for the ET segment as correct since they were not classified as WT. Since the dataset does not contain any full MRI sequences of healthy brains, this would ideally be tackled as future work on MRI samples from control patients.

For actual pathological samples, an initial test for comparing the predictions and ground truths is shown in Figure 4.7. The image shows predictions across 5-slice intervals of the same MR image on an LGG and HGG patient compared to the corresponding BraTS expert tumour segmentations.

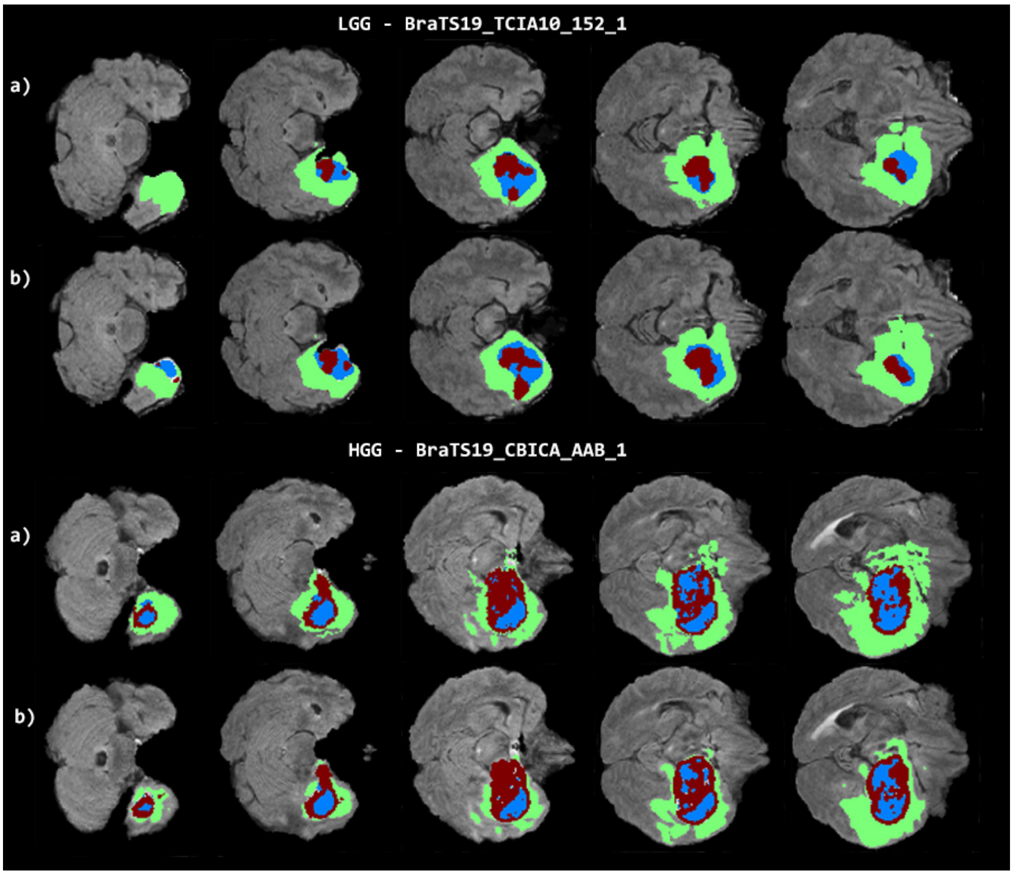


Figure 4.7: Comparison between segmentation by a) expert ground truths and b) model predictions across one LGG and one HGG scan on the BraTS 2019 Training set.

As in the other diagrams using masks, the green, red, and blue masks in Figure 4.8 represent the peritumoural edema, enhancing tumour core, and non-enhancing tumour core respectively. In Figure 4.7, the LGG sequence had a misclassification in the first slice's prediction where the peritumoural edema was predicted as a multi-class segment. It is also observable how the model performed worse overall on the LGG case when distinguishing the tumour core components. The HGG sample on the other hand was reasonably well predicted. The model's worse performance on the LGG sample may be attributed to the large imbalance present in the dataset between HGG and LGG samples. Another test involving the training dataset was conducted to compare the model's segmentation predictions on individual patient slices with some ground truth samples from cases unseen during training.

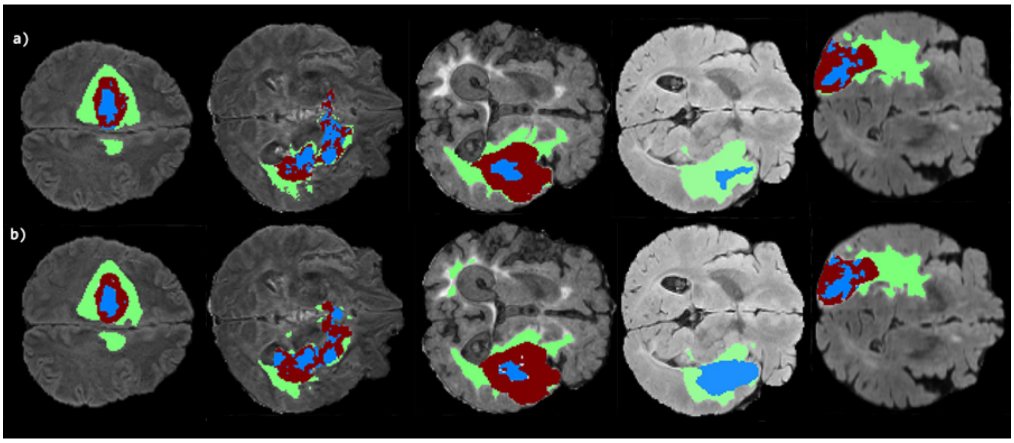


Figure 4.8: Comparison between segmentation by a) expert ground truths b) model predictions on five separate patient scans on the BraTS 2019 Training set.

The second and fourth samples were included in Figure 4.8 specifically as they are fairly particular cases. The second patient's tumour surface area was very spread out and exhibits how much a pathology can vary in shape when compared to the other samples. Nonetheless, the model's prediction was still fairly accurate. The fourth patient in the image was an LGG case without an enhancing tumour core segment. The other segmentations represent average cases of the proposed approach's segmentation accuracy. There are some slight visual differences which may be observed between the ground truths and predictions, such as the third patient's misclassification of the edema in the top left of the predicted slice. Nonetheless, it can be re-iterated from these results that overall the model performs segmentations quite well when compared with the expert annotations from the BraTS dataset.

4.3 | Evaluation

This section will serve to first provide a comparison between the proposed approach and the internal reference models in Section 4.3.1. Furthermore, a comparison of the proposed approach and published models evaluated on the 2019 validation dataset will be discussed in Section 4.3.2.

4.3.1 | Evaluation against Internal Models

Following the experiments performed on the proposed U-Net++, the next test to follow was the internal evaluation against the standard 3D U-Net and Residual 3D U-Net models discussed previously. Since the experiments have shown that the data augmentation and post-processing changes led to superior scores, these properties will be applied to the internal models as well. Whilst the data augmentation is undoubtedly a beneficial feature, we have also included results from the CBICA portal for each model without post-processing. This was performed simply as a check to ensure that the change would not end up negatively affecting the reference models, and biasing the results towards the proposed approach. The models used for this comparison are as discussed in Section 3.1.5. As mentioned previously, the standard U-Net aims to follow the Ronneberger et al. (2015) U-Net with modifications to be compatible with multimodal 3D MR images. The residual 3D U-Net was taken directly from the online repository to ensure correctness.

Table 4.22: All evaluation criteria for the internal models, with and without constant threshold. Best scores in bold.

Configuration	Dice Coefficient			Sensitivity			Specificity			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Standard U-Net (Baseline)	0.6279	0.8737	0.7725	0.7684	0.8859	0.7948	0.9962	0.9934	0.9952	6.7620	8.4077	9.4486
Residual U-Net (Baseline)	0.6428	0.8433	0.7540	0.6840	0.8664	0.7625	0.9981	0.9915	0.9956	9.0067	11.6163	11.9439
U-Net++ (Baseline)	0.6920	0.8709	0.7824	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969	6.8001	8.3279	9.50
Standard U-Net (Threshold)	0.6502	**0.8737**	0.7724	**0.7708**	**0.8859**	**0.7948**	0.9962	0.9934	0.9952	5.9730	8.4096	**9.4494**
Residual U-Net (Threshold)	0.6720	0.8433	0.7537	0.6924	0.8663	0.7621	**0.9981**	0.9915	0.9956	8.3355	11.6119	11.9473
U-Net++ (Threshold)	**0.7192**	0.8712	**0.7817**	0.7232	0.8671	0.7631	0.9980	**0.9944**	**0.9969**	**4.6863**	**8.2157**	9.4752

From Table 4.22, it is evident that the constant threshold was beneficial for each of the internal models to very similar degrees. We may conclude from this that none of the baseline internal models managed to automatically avoid predicting few ET voxels, when there were actually none in the ground truths. Similarly to the other model comparisons, the first metrics to be assessed will be the Dice Coefficient and Hausdorff Distance. For these measurements, the clear top performer was the U-Net++ model. Interestingly enough, the standard U-Net managed to obtain a marginally higher WT dice score, although this is offset by the substantial gap in ET Dice score.

The more fascinating part of the internal evaluation is the sensitivity score. From the tabulated results, we may observe that the standard model managed to obtain the highest sensitivity values. This includes the classes which had lower Dice Coefficient values than the U-Net++ model, namely the ET and TC. The implication from these factors is that whilst the standard U-Net was weaker in terms of pure segmentation, the actual predictions had a lesser number of false negatives. In terms of false positives, the highest specificity values were mostly obtained by the U-Net++ model, with a very slight increase in WT Specificity for the residual approach.

Since these models build upon each other's architecture, it is interesting to see that the standard U-Net for the most part outperformed the residual model. There is a possibility that the standard U-Net training runs are more susceptible to variance than the other models, with good runs having high values, and low runs having much lower scores. We may assess this briefly as we are in possession of another model for each configuration. For the residual U-Net, we have the 8.8M parameter model. This slightly adapted the Github repository implementation as the localization modules use $3 \times 3 \times 3$ kernels rather than $1 \times 1 \times 1$, hence why the original baseline version was used for this comparison. For the standard U-Net, we have a previous training run using a model which had an additional convolution layer at the start of the network (much like the residual U-net) and was missing the final two convolutions prior to the segmentation, which is why it was omitted from the final evaluation. However, we may observe the results obtained by the baseline models and these alternate configurations for posterity.

Table 4.23: All evaluation criteria for the alternate and baseline variants of the standard U-Net and residual U-Net. Best scores in bold.

Configuration	Dice Coefficient			Sensitivity			Specificity			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Standard U-Net (Interim Model)	0.6199	0.8506	0.7324	0.6321	0.8400	0.7131	**0.9988**	**0.9945**	0.9961	8.3572	10.4211	11.6426
Standard U-Net (Baseline)	0.6279	**0.8737**	**0.7725**	**0.7684**	**0.8859**	**0.7948**	0.9962	0.9934	0.9952	**6.7620**	8.4077	9.4486
Residual U-Net (8.8M Model)	**0.6510**	0.8614	0.7640	0.7164	0.8825	0.7503	0.9979	0.9928	**0.9965**	7.5646	**8.0562**	**8.7872**
Residual U-Net (Baseline)	0.6428	0.8433	0.7540	0.6840	0.8664	0.7625	0.9981	0.9915	0.9956	9.0067	11.6163	11.9439

Table 4.23 shows interesting fluctuations in values. The residual U-Net scores did not change too drastically, although the 8.8M model actually performed slightly better. However, the standard U-Net's sensitivity scores are quite different between the interim model and the baseline we have currently set. Whilst this is definitely due to the slight change in architecture, we may recall from experiments on U-Net++ such as doubling the number of convolutional blocks, or even completely changing the skip-connections that the scores did not vary as drastically. Whilst it is only a hypothesis based on a slightly modified architecture, it may be the case for the standard U-Net that this was a run with above average results. The other models, particularly U-Net++, do not seem to be as affected by changes in layers or the model in general, perhaps due to their more

refined infrastructure.

In light of the discussion above, we may first conclude that the U-Net++ model obtained the highest scores on average in terms of Dice Coefficient, Hausdorff Distance, and specificity. The model surpassed both the standard and residual U-Net architectures in this regard. For sensitivity, the baseline standard U-Net performed optimally, although performing multiple training runs on this architecture could lead to potentially different results as discussed above. Interesting ideas for potential future work arise from the results above, such as a model ensemble of U-Net++ and the standard U-Net. In theory, such a combination could produce sound results both in terms of pure segmentation, and also the detection of false positives and negatives.

4.3.2 | Evaluation against State-of-The-Art Techniques

The BraTS 2019 Validation set leaderboard[1] is available from the CBICA website. The scoreboard is unranked, but shows the team names, number of submissions, and all scores of the latest submission uploaded by that team. The scoreboard as at 22nd June 2020 was copied into an Excel worksheet and sorted by average Dice Coefficient across all three target classes, since the experiments gave priority to Dice score improvements. Our approach obtained the 75th place out of 158 total competitors. The only information provided about competitors is the team name, with no links to related work. Nonetheless, a number of published papers evaluated on the 2019 validation set were found and used for evaluation against the proposed model. These models have been selected on the basis of their individual configurations to emphasise the strengths of each approach, and are shown in Table 4.24.

Table 4.24: Comparison between the proposed approach and some state-of-the-art approaches on the BraTS 2019 Validation Set. Best scores in bold, '-' refers to unreported values.

Method	Dice Coefficient			Sensitivity			Specificity			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Amian and Soltaninejad (2019)	0.71	0.84	0.74	0.68	0.82	0.74	1.00	0.99	1.00	10.11	14.00	16.06
Proposed Model	0.72	0.87	0.78	0.72	0.87	0.76	1.00	0.99	1.00	4.69	8.22	9.48
Wang et al. (2019)	0.74	0.89	0.81	0.77	0.90	0.83	1.00	0.995	1.00	5.99	5.68	7.36
Murugesan et al. (2019)	0.78	0.90	0.78	-	-	-	-	-	-	-	-	-
Hamghalam et al. (2019)	0.77	0.90	0.79	0.77	0.91	0.78	1.00	0.99	1.00	4.60	6.90	8.40
Myronenko and Hatamizadeh (2020)	**0.80**	**0.89**	**0.83**	-	-	-	-	-	-	**3.92**	**5.89**	**6.56**

Starting with the first approach by Amian and Soltaninejad (2019), this system makes use of a two-way pipeline. One pathway makes use of the original resolution MR images, whilst the other uses lower resolution versions of the images. The section using the

[1] https://www.cbica.upenn.edu/BraTS19/lboardValidation.html

low resolution images implements a U-Net architecture, whilst the full-size path uses a residual technique similar to the model by Isensee et al. (2017). The results of both paths are then summed using an element-wise addition to obtain the final segmentation. This technique was outperformed consistently by the proposed approach for all three tumour classes, barring specificity, although this may be due to rounding to two decimal places.

The approach by Wang et al. (2019) is the first of the systems used for the evaluation which outperformed the proposed model. The method employed is actually very similar to what is discussed for the residual model in this project. The main difference aside from the actual model is the patching strategy. Wang et al. (2019) make use of a smart patching strategy to maximise the amount of brain voxels whilst disregarding the background. The patch windows are determined by an offset from the brain boundary, later used for two separate patching cycles which extract brain voxels from the image. The use of a patching strategy such as this in place of cropping background values may have been what set this project apart from the proposed model.

Murugesan et al. (2019) proposed an ensemble of multidimensional and multiresolution networks for the segmentation task. Each tumour class was segmented using an individual ensemble, composed of variations of DenseNet, Inception Networks, and Residual Inception Networks. The output of each network in every ensemble was then summed using an element-wise addition. This approach made use of 240×240 axial slices as input, and appears to have followed a similar post-processing methodology to the one implemented in this project. Murugesan et al. (2019) claim that 'clusters of smaller size' were thresholded to reduce false positives, which may allude to the 0 score ET cases discovered previously in the experiments. Otherwise, the ensemble approach, networks used, and input resolution makes this a very different approach from the proposed model.

Hamghalam et al. (2019) implemented a Generative Adversarial Network (GAN) to solve the brain tumour segmentation task using synthetic segmentation. The GAN portion of the network provides synthetic MR images which are combined with actual dataset samples using the FLAIR, T_1ce, and T_2w sequence types. The combined images are then used as input to an ensemble of three 3D Fully Connected Networks (FCN). The reasoning behind having three networks is that an individual model is trained on each of the axial, coronal, and sagittal views. The combination of synthetic and real MR images as well as the use of an FCN for each axis in 3D allowed this model to achieve considerably positive results on all metrics. The omission of the dataset's T_1w MR images is also an interesting choice; as discussed in Section 2.1.1, the modalities which were retained by Hamghalam et al. (2019) are generally ideal for detection of pathologies, whereas T_1w images may be better suited for visualising the anatomy of the brain.

Finally, Myronenko and Hatamizadeh (2020) once again proposed the use of a encoder-decoder architecture for the 2019 challenge. This paper works on very similar principles to Myronenko (2018), which was discussed previously in Section 2.2 and won the author first place in the 2018 BraTS challenge. The input patch size for this experiment was once again quite large, using dimensions of $160 \times 192 \times 128$, with the experiment being run on multi-GPU systems for rapid testing. The model once again follows an encoder-decoder approach, this time making use of a hybrid loss function. The latter is composed of the Dice loss, a focal loss function, and a 3D extension of supervised active contour loss that uses volumetric and length terms from the MR images. It is noteworthy that the large spatial resolution of the inputs used by the experiment makes this model occupy a considerable size. Similarly to the 2018 challenge entry (Myronenko, 2018), this model obtained very high scores across all of the reported metrics, beating every other model discussed in this evaluation. Whilst the sensitivity and specificity for the validation set were not reported in the publication, it is possible that they are also the highest out of the methods in Table 4.24.

5

Conclusions

This chapter will provide an overview of the outcomes of the project, including whether the final model met the initial expectations set at the start of development, as well as possible areas for improvement. Section 5.1 expands on the objectives mentioned in Section 1.2 and how the model performed for each of the listed goals. A discussion of the model's shortcomings and possible reasons for them is provided in Section 5.2. Section 5.3 addresses potential improvements to the model's performance and mitigating the project's limitations. Finally, closing remarks including my personal experience with the project are discussed briefly in Section 5.4.

5.1 | Achieved Aims and Objectives

From the results showcased in Chapter 4, it has been shown that the system succeeds in automatically generating segmentation predictions of brain tumour from a set of multimodal 3D MR images, and produces results of a fairly good quality. The generated segmentation maps are also viewable using many standard MR image viewers since the predicted maps are in the Nifti file format. The more complex objective of the model predictions being clinically accepted has also been addressed; reference is made once again to the Dice Coefficient and Hausdorff Distance, where the proposed approach obtained positive scores. The model's potential in handling false positives and false negatives is shown through the values for the sensitivity and specificity. The sensitivity score is similar to the Dice Coefficient, and the specificity values are very high for all three classes. The sensitivity scores could definitely see the most improvement, and there are a number of ways this could be achieved, as discussed in Section 5.3 below.

The objective to obtain comparable results with the state-of-the-art is definitely the goal which was least satisfied. It is noteworthy that the model did surpass Amian and

Soltaninejad (2019), where one of the authors had previously worked on relatively popular brain tumour segmentation publications, such as Soltaninejad et al. (2017, 2018). The work by Amian and Soltaninejad (2019) is fairly related to this project as it makes use of both a standard U-Net and residual U-Net, two of the internal models used for comparison with the proposed approach. The results follow the observations discussed in this study where the proposed approach outperformed the other models.

However, the other models in the external evaluation did outperform the proposed approach across all metrics, although for the most part this was not by a very large margin. Most of the configurations evaluated vary substantially, either making use of network ensembles (Hamghalam et al., 2019; Murugesan et al., 2019) or very particular training strategies (Myronenko and Hatamizadeh, 2020). The comparison of the largest significance is likely Wang et al. (2019), who made use of a 3D residual U-Net which adapts the Isensee et al. (2017) residual U-Net architecture, much like the model used for internal evaluation in Section 4.3.1. The main difference in Wang et al. (2019)'s work is the use of a dual-patching strategy rather than directly cropping the input data to the final patch size of $128 \times 128 \times 128$. The patching strategy used by Wang et al. (2019) is quite smart as rather than cropping images in a fixed way, an offset from the brain boundaries is used to strictly use salient patches from the brain. The use of an offset is a uniform means to patch within relevant areas, which may have performed better than foreground-cropping, which produces results depending on the distribution of zero-valued pixels in the image. This study was discovered nearing the end of the project and in retrospect, this may have been the defining change which set this approach apart from ours, and will be considered for potential future work.

5.2 | Critique and Limitations

There are a number of factors which may have impeded the model from obtaining even higher scores. It is important to repeat the development process of the pipeline to highlight the limitations of the proposed approach. Initially, the plan was to create a new adaptation of the approach by Isensee et al. (2017), which was also reflected in the Progress Report for this dissertation. With some more research, the work on U-Net++ by Zhou et al. (2018c) was discovered and the approach was less saturated by studies than residual U-Net. Combining this with the results achieved in Section 4.1, U-Net++ was used as a basis for the current proposed approach. Nonetheless, the preprocessing and patch strategies used by the aforementioned Github repository were maintained even after the switch to U-Net++ as it was proven to produce results which were adequate at the time. The project was to be closed once results had been obtained on the holdout samples of the training set. However, it was decided to shift the evaluation to the validation set using the CBICA portal. This was done both to make the approach comparable to other work, and also to ensure no biases are present in the results, since the IPP generates the scores online using the undistributed validation ground truths. Uploading to the portal was a task in itself as it would have unexpected failures from time to time, and successful jobs would take close to an hour to process. Having said that, the majority of uploads were successful and the inclusion of validation set results has provided a much clearer means of evaluating the model both internally and with external studies.

The largest limitation of this project was the training time required. Since training cycles could take anywhere from 3-6 days depending on model configuration and GPU, most of the testing effort was expended on adjusting the model configurations rather than focusing on minor tests like traditional hyperparameter tuning for variables such as the learning rate. Failed tests often could not be cut too short, as some configurations could have a promising convergence gradient but saturate anywhere past 100-200 epochs, which is still a considerable amount of time taken. As a result, it was quite difficult to gauge if a change had positively affected the results on some occasions, which would end up wasting time. In addition, one must also consider the physical cost of running Cloud instances for an extended period of time. The latter came at a personal cost of close to $1,500$ Euros.

Whilst the utmost effort was made to continue development, research, and writing-up the documentation in between training cycles, there would still sometimes be a wait to see the results of certain tests. When the proposed approach had reached its current state, quite some time had passed and thus most of the evaluation model results in Table 4.24 were discovered quite late into development, especially since the competition ran late into 2019, after which publications would slowly start being released. As a result, there was not enough time to attempt to improve the proposed approach any further based on the findings in the evaluation.

Finally, a minor limitation of the model is that at the moment, it requires the entire set of four MR modalities as input, rather than providing the option of using specific sequence types. For the purpose of this study, it was not seen as a good time investment to prioritise minor improvements of this manner over improving the model's performance. Moreover, it could potentially result in the model performing worse if trained on only one sequence type. Nonetheless, it would be interesting to observe how experiments such as training using solely the T_1ce MR image volumes or other combinations of image modalities would perform.

5.3 | Future Work

The potential for future improvements goes hand-in-hand with the discussions in Chapter 4 and the limitations above. Starting with preprocessing techniques, reference is made once again to the research by Wang et al. (2019). Their strategy made use of two-phase training pass which would first use cubic patches within the border of the brain, and then join these patches as well as possible within the actual brain area. It is possible that modifying our patching strategy to this smarter version would improve the quality of the results fed to the model and thus the final predictions. It is also noteworthy that since the validation data is passed through the same preprocessing pipeline, changing the patching strategy may improve the results on that front as well. Furthermore, it is worth performing a test to see if including FLAIR sequences in the bias correction would be beneficial. In the proposed pipeline, we follow the online repository's process of omitting FLAIR by default.

Another potential improvement in the pre-processing pipeline could be using the 'intensity landmark' technique by Nyúl et al. (2000) to process intensity heterogeneities in the data from separate institutions, either in addition to or instead of the z-score normalisation. The normalisation could also be performed in a homogenized fashion i.e. on a per-patient basis, rather than across the whole dataset as performed by Jiang et al. (2019)). Furthermore, it would also make for an interesting study to compare the intensity distributions of the images before and after normalization. Given the size of the data, this may take some time to finish processing.

From the post-processing perspective, an interesting observation by Zhou et al. (2018b) was that given a certain ratio of enhancing tumour to edema pixels, ET pixels are sometimes wrongly classified as WT. These pixels would be the second group of ET segments with a value of 0; as we tackled the first group where the model would wrongly predict ET Voxels for scans without an ET component. According to the post-processing technique mentioned by Zhou et al. (2018b) it is a possibility that the model predicted the ET voxels in this group as edema instead.

It could thus prove useful to follow this approach and cluster the edema pixels based on their assigned label values, and see if any disparate clusters are generated within the edema. This hypothesis would lead to a distinct group of voxels emerging which could then be post-processed into the missing ET voxels. This would lead to a substantial improvement, as those ET samples with a score of 0 lower the average scores substantially. Thus, correct predictions on this group of ET outliers would make the model much more competitive with the state-of-the-art techniques discussed.

Ensembling is the other major change which could provide benefits to the model's performance. When we refer to ensembling, this may be carried out in a number of ways. Let's begin by mentioning the models we currently have in place. From the results of the internal evaluation performed in Section 4.3.1, two main observations were noted in relation to the two reference models. Firstly, the standard U-Net obtained very high sensitivity scores. Secondly, the same model appeared to vary substantially in results with changes in architecture. Verifying the training variance for each model would be a good first step. For U-Net++, the experiments discussed in this documentation show that the results did not vary hugely with changes to the model. However, it is viable to use practices such as training each model a number of times and averaging the results as this may lead to higher scores, or through techniques such as five-fold cross validation. Secondly, if the high sensitivity scores of the standard U-Net are maintained over multiple training runs, this approach could be combined with the proposed model, similarly to the work by Amian and Soltaninejad (2019). As discussed previously, this could then result in higher sensitivity values for the proposed approach.

Another possibility for ensembling would be to use multiple versions of the proposed model for separate parameters in the dataset. As an example, three U-Net++ networks may be used; one for each of the enhancing tumour, whole tumour, and tumour core. Alternately, one may opt for pathways similar to those implemented by Pereira et al. (2016), with one path focusing on LGG and the other on HGG. The work by Hamghalam et al. (2019) also provides an interesting approach where individual networks were trained for the axial, coronal, and sagittal views, which could be another potentially viable experiment.

Finally, there could also be improvements which may be applied to U-Net++ as an architecture as well. It is noteworthy that whilst the major dropout experiment discussed in Section 4.1.4 uses only the dropout value of 0.3, there were other small training runs performed with different values such as 0.2, 0.5, etc. It would interesting to perform full training runs using different dropout values to confirm whether this provides any benefits during training. Another possible experiment would be to add additional deep supervision layers in the decoder, as performed with the residual 3D U-Net. Furthermore, a later paper by Zhou et al. (2019) proposes a number of improvements to the standard U-Net++ architecture, including pruning of the decoder layers to speed up inference time. There is also a lesser known paper by Chen et al. (2018) which describes some modifications to the U-Net++ architectures which could prove useful in this regard.

5.4 | Final Remarks

In summary, this project proposes an approach for brain tumour segmentation which obtained positive results on the BraTS 2019 Validation dataset. The model is small in size owing to the changes made to the architecture, being less taxing on GPU memory than standard approaches. The contribution to the domain made by the proposed approach is mostly attributed to the adaptations performed to the U-Net++ architecture, and the experiments discussed in this dissertation. The significance of the project is substantiated by both the acceptance of the OHBM for a poster presentation, and also a paper publication for the IEEE MELECON 2020 conference. This is especially true since these acceptances were for a lesser variant of the proposed approach which was evaluated on an unseen holdout sample of the training dataset.

To summarise, the proposed approach maintains a complex architecture which combines the benefits of dense networks and U-Net within a more accessible, smaller model. The experiments performed in Section 4.1 present a set of interesting cases which exhibit how certain adjustments to our model adaptation either improved the model performance or ended up being inconsequential. From a personal standpoint, this project proved to be an unmatched lesson in computer vision techniques, model tuning, and mostly learning about biomedical image segmentation. Furthermore, the entire process resulted in learning very important lessons in planning projects of this scale, particularly the organisation component and the prioritisation of experiments. In closing, the proposed model exhibits a unique variant of the U-Net++ model architecture, with the experiments discussed above showcasing interesting conclusions such as how halving the number of convolutional blocks resulted in very minor differences in the model's performance. This result could potentially allow researchers leveraging U-Net++ or similar models to allocate parameters elsewhere, promoting the development of novel techniques.

References

M. Amian and M. Soltaninejad. Multi-resolution 3d cnn for mri brain tumor segmentation and survival prediction. *arXiv preprint arXiv:1911.08388*, 2019.

M. Angulakshmi and G. G. L. Priya. Brain tumour segmentation from mri using superpixels based spectral clustering. *Journal of King Saud University-Computer and Information Sciences*, 2018.

V. Anitha and S. Murugavalli. Brain tumour classification using two-tier classifier with adaptive segmentation technique. *IET computer vision*, 10(1): pages 9–17, 2016.

B. B. Avants, N. Tustison, and G. Song. Advanced normalization tools (ants). *Insight j*, 2(365): pages 1–35, 2009.

S. Bakas, K. Zeng, A. Sotiras, S. Rathore, H. Akbari, B. Gaonkar, M. Rozycki, S. Pati, and C. Davatzikos. Glistrboost: combining multimodal mri segmentation, registration, and biophysical tumor growth modeling with gradient boosting machines for glioma segmentation. In *BrainLes 2015*, pages 144–155. Springer, 2015.

S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific Data*, 4, September 2017a.

S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Segmentation labels and radiomic features for the pre-operative scans of the tcga-lgg collection, July 2017b.

S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos. Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection, July 2017c.

S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

T. Chaira and S. Anand. A novel intuitionistic fuzzy approach for tumour/hemorrhage detection in medical images. *Journal of Scientific and Industrial Research*, 2011.

F. Chen, Y. Ding, Z. Wu, D. Wu, and J. Wen. An improved framework called du++ applied to brain tumor segmentation. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 85–88. IEEE, 2018.

H. Chen, Q. Dou, L. Yu, and P.-A. Heng. Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation. *arXiv preprint arXiv:1608.05895*, 2016.

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*, pages 2843–2851, 2012.

M. C. Clark, L. O. Hall, and D. B. Goldgof. Knowledge-guided processing of magnetic resonance images of the brain. 1998.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3): pages 273–297, 1995.

Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng. 3d deeply supervised network for automatic liver segmentation from ct volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 149–157. Springer, 2016.

J. L. Foo. A survey of user interaction and automation in medical image segmentation methods. *Tech rep ISUHCI 20062, Human Computer Interaction Department, Iowa State University*, 2006.

M. Ghaffari, A. Sowmya, and R. Oliver. Automated brain tumour segmentation using multimodal brain scans, a survey based on models submitted to the brats 2012-18 challenges. *IEEE Reviews in Biomedical Engineering*, pages 1–1, 2019. ISSN 1941-1189.

R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

A. Gooya, K. M. Pohl, M. Bilello, L. Cirillo, G. Biros, E. R. Melhem, and C. Davatzikos. Glistr: glioma image segmentation and registration. *IEEE transactions on medical imaging*, 31(10): pages 1941–1954, 2012.

N. Gordillo, E. Montseny, and P. Sobrevilla. State of the art survey on mri brain tumor segmentation. *Magnetic resonance imaging*, 31(8): pages 1426–1438, 2013.

C. Guo. Machine learning methods for magnetic resonance imaging analysis. January 2012.

M. Hamghalam, B. Lei, and T. Wang. Brain tumor synthetic segmentation in 3d multimodal mri scans. *arXiv preprint arXiv:1909.13640*, 2019.

J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): pages 100–108, 1979.

M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35: pages 18–31, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein. Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge. In *International MICCAI Brainlesion Workshop*, pages 287–297. Springer, 2017.

F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein. No new-net. In *International MICCAI Brainlesion Workshop*, pages 234–244. Springer, 2018.

Z. Jiang, C. Ding, M. Liu, and D. Tao. Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task. In *International MICCAI Brainlesion Workshop*, pages 231–241. Springer, 2019.

B. N. Joe, M. B. Fukui, C. C. Meltzer, Q.-s. Huang, R. S. Day, P. J. Greer, and M. E. Bozik. Brain tumor volume measurement: comparison of manual and semiautomated methods. *Radiology*, 212(3): pages 811–816, 1999.

J. Juan-Albarracín, E. Fuster-Garcia, J. V. Manjón, M. Robles, F. Aparici, L. Martí-Bonmatí, and J. M. García-Gómez. Automated glioblastoma segmentation based on a multiparametric structured unsupervised classification. *Plos One*, 10(5): pages 1–20, May 2015.

K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker. Deepmedic for brain tumor segmentation. In *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, pages 138–149. Springer, 2016.

D. Karimi and S. E. Salcudean. Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE transactions on medical imaging*, 2019.

M. R. Kaus, S. K. Warfield, A. Nabavi, P. M. Black, F. A. Jolesz, and R. Kikinis. Automated segmentation of mr images of brain tumors. *Radiology*, 218(2): pages 586–591, 2001.

B. Kayalibay, G. Jensen, and P. van der Smagt. Cnn-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*, 2017.

A. Kermi, K. Andjouh, and F. Zidane. Fully automated brain tumour segmentation system in 3d-mri using symmetry analysis of brain and level sets. *IET Image Processing*, 12: pages 1964–19717, November 2018. ISSN 1751-9659.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): pages 1995, 1995.

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): pages 2278–2324, 1998.

J. Liu, M. Li, J. Wang, F. Wu, T. Liu, and Y. Pan. A survey of mri-based brain tumor segmentation methods. *Tsinghua Science and Technology*, 19(6): pages 578–595, 2014.

S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): pages 129–137, 1982.

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*, pages 3431–3440, 2015.

A. S. Lundervold and A. Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2): pages 102–127, 2019.

S. Luo, R. Li, and S. Ourselin. A new deformable model using dynamic gradient vector flow and adaptive balloon forces. In *APRS Workshop on Digital Image Computing, Brisbane, Australia*. Citeseer, 2003.

J. Luts, A. Heerschap, J. A. K. Suykens, and S. Van Huffel. A combined mri and mrsi based multiclass system for brain tumour recognition using ls-svms with class probabilities and feature selection. *Artificial intelligence in medicine*, 40(2): pages 87–102, 2007.

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

J. V. Manjón, P. Coupé, A. Buades, V. Fonov, D. L. Collins, and M. Robles. Non-local mri upsampling. *Medical Image Analysis*, 14(6): pages 784–792, 2010. ISSN 1361-8415.

G. P. Mazzara, R. P. Velthuizen, J. L. Pearlman, H. M. Greenberg, and H. Wagner. Brain tumor target volume determination for radiation treatment planning through automated mri segmentation. *International Journal of Radiation Oncology* Biology* Physics*, 59(1): pages 300–312, 2004.

R. McKinley, A. Jungo, R. Wiest, and M. Reyes. Pooling-free fully convolutional networks with dense skip connections for semantic segmentation, with application to brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 169–177. Springer, 2017.

R. McKinley, R. Meier, and R. Wiest. Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 456–465. Springer, 2018.

R. McKinley, M. Rebsamen, R. Meier, and R. Wiest. Triplanar ensemble of 3d-to-2d cnns with label-uncertainty for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 379–387. Springer, 2019.

D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince. *MRI from Picture to Proton*. Cambridge university press, 2017.

N. Menon and R. Ramakrishnan. Brain tumor segmentation in mri images using unsupervised artificial bee colony algorithm and fcm clustering. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 0006–0009, April 2015.

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*, 34(10): pages 1993–2024, 2014.

N. Micallef, D. Seychell, and C. Bajada. A nested U-Net approach for brain tumour segmentation. In *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON) (MELECON 2020)*, Palermo, Italy, June 2020.

K. D. Miller, M. Fidler-Benaoudia, T. H. Keegan, H. S. Hipp, A. Jemal, and R. L. Siegel. Cancer statistics for adolescents and young adults, 2020. *CA: A Cancer Journal for Clinicians*, 2020.

G. K. Murugesan, S. Nalawade, C. G. B. Yogananda, B. Wagner, B. Fei, A. Madhuranthakam, and J. A. Maldjian. Multidimensional and multiresolution ensemble networks for brain tumor segmentation. *bioRxiv*, page 760124, 2019.

A. Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320. Springer, 2018.

A. Myronenko and A. Hatamizadeh. Robust semantic segmentation of brain tumor regions from 3d mris. *arXiv preprint arXiv:2001.02040*, 2020.

L. G. Nyúl, J. K. Udupa, and X. Zhang. New variants of a method of mri scale standardization. *IEEE transactions on medical imaging*, 19(2): pages 143–150, 2000.

S. D. Olabarriaga and A. W. M. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical image analysis*, 5(2): pages 127–142, 2001.

S. Padmanaban, K. Thiruvenkadam, and R. Rangasami. Modified local ternary patterns technique for brain tumour segmentation and volume estimation from mri multi-sequence scans with gpu cuda machine. *Biocybernetics and Biomedical Engineering*, 39: pages 470–487, April 2019.

S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE transactions on medical imaging*, 35(5): pages 1240–1251, 2016.

W. E. Phillips II, R. P. Velthuizen, S. Phuphanich, L. O. Hall, L. P. Clarke, and M. L. Silbiger. Application of fuzzy c-means segmentation technique for tissue differentiation in mr images of a hemorrhagic glioblastoma multiforme. *Magnetic resonance imaging*, 13(2): pages 277–290, 1995.

N. H. Rajini, T. Narmatha, and R. Bhavani. Automatic classification of mr brain tumor images using decision tree. In *Proceedings of international conference on electronics*, volume 31, 2012.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

G. Rezai-Rad and M. Aghababaie. Comparison of susan and sobel edge detection in mri images for feature extraction. In *2006 2nd International Conference on Information Communication Technologies*, volume 1, pages 1103–1107, April 2006.

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*, pages 234–241. Springer, 2015.

R. L. Siegel, K. D. Miller, and A. Jemal. Cancer statistics, 2019. *CA: a cancer journal for clinicians*, 69(1): pages 7–34, 2019.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, and X. Ye. Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in flair mri. *International journal of computer assisted radiology and surgery*, 12(2): pages 183–203, 2017.

M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, and X. Ye. Supervised learning based multimodal mri brain tumour segmentation using texture features from supervoxels. *Computer methods and programs in biomedicine*, 157: pages 69–84, 2018.

P. Sriramakrishnan, T. Kalaiselvi, and R. Rajeswaran. Modified local ternary patterns technique for brain tumour segmentation and volume estimation from mri multi-sequence scans with gpu cuda machine. *Biocybernetics and Biomedical Engineering*, 39(2): pages 470–487, 2019.

J. Stráský, M. Janeček, P. Harcuba, D. Preisler, and M. Landa. 4.2 - biocompatible beta-ti alloys with enhanced strength due to increased oxygen content. In F. H. Froes and M. Qian, editors, *Titanium in Medical and Dental Applications*, Woodhead Publishing Series in Biomaterials, pages 371–392. Woodhead Publishing, 2018. ISBN 978-0-12-812456-7. URL `http://www.sciencedirect.com/science/article/pii/B9780128124567000172`.

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

A. A. Taha and A. Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1): pages 29, 2015.

J.-D. Tournier, R. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly. Mrtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, page 116137, 2019.

N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee. N4itk: improved n3 bias correction. *IEEE transactions on medical imaging*, 29(6): pages 1310, 2010.

N. J. Tustison, K. L. Shrinidhi, M. Wintermark, C. R. Durst, B. M. Kandel, J. C. Gee, M. C. Grossman, and B. B. Avants. Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with antsr. *Neuroinformatics*, 13(2): pages 209–225, 2015.

V. Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24: pages 774–780, 1963.

U. Vovk, F. Pernus, and B. Likar. A review of methods for correction of intensity inhomogeneity in mri. *IEEE transactions on medical imaging*, 26(3): pages 405–421, 2007.

F. Wang, R. Jiang, L. Zheng, B. Biswal, and C. Meng. Brain-wise tumor segmentation and patient overall survival prediction. *arXiv preprint arXiv:1909.12901*, 2019.

D. R. R. White, A. S. Houston, W. F. D. Sampson, and G. P. Wilkins. Intra-and interoperator variations in region-of-interest drawing and their effect on the measurement of glomerular filtration rates. *Clinical nuclear medicine*, 24(3): pages 177–181, 1999.

D. P. Wipf and S. S. Nagarajan. A new view of automatic relevance determination. In *Advances in neural information processing systems*, pages 1625–1632, 2008.

M.-N. Wu, C.-C. Lin, and C.-C. Chang. Brain tumor detection using color-based k-means clustering segmentation. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, volume 2, pages 245–250. IEEE, 2007.

N. Zhang, S. Ruan, S. Lebonvallet, Q. Liao, and Y. Zhu. Multi-kernel svm based classification for brain tumor segmentation of mri multi-sequence. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3373–3376. IEEE, 2009.

Y.-X. Zhao, Y.-M. Zhang, and C.-L. Liu. Bag of tricks for 3d mri brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 210–220. Springer, 2019.

C. Zhou, S. Chen, C. Ding, and D. Tao. Learning contextual and attentive information for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 497–507. Springer, 2018a.

C. Zhou, C. Ding, Z. Lu, X. Wang, and D. Tao. One-pass multi-task convolutional neural networks for efficient brain tumor segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 637–645. Springer, 2018b.

J. Zhou, K. L. Chan, V. F. H. Chong, and S. M. Krishnan. Extraction of brain tumor from mr images using one-class support vector machine. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 6411–6414. IEEE, 2006.

Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018c.

Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 2019.