



*University of St Andrews
School of Computer Science*

Using Deep Learning to Segment Brain Tumours using MRI Data

David Paul Smith
Matriculation Number: 200032966

*MSc in Artificial Intelligence
CS5098 – Group Project and Dissertation*

Group Members: DS291 – 200032966, AG360 – 190031511, JH384 200027630

Word Count: 14981

17th August 2021

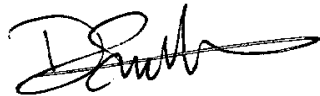
Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year (20/21) except where otherwise stated.

The main text of this project report is 14981 words long and was submitted to the University of St Andrews,

I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Signature

A handwritten signature in black ink, appearing to be 'D. Smith', written over a horizontal line.

Date - 17/08/2021

Acknowledgements

Throughout this project I have received a huge amount of support. I would first like to thank my supervisor, Dr David Harris Birtill, for his advice, good humour, and persistent encouragement. I would also like to give thanks to my group members, Aditi Goswami, Jamel Houd and Ziyu Song. Thank you all for the time spent together working on this project.

To my mentors, Jacqueline, and Christina. Thank you for all the support, for basically kicking me out of the door to go do my MSc and helping me to believe I could get this far. To Shane, for all the work we did together, you gave me the tool kit I needed at the time it mattered most.

To my family. Mum, thank you for your patience. Ben, thank you for keeping me sane. Rebecca, thank you for inspiring me. To Nicola, thank you for making every day an adventure, and ensuring I don't take myself too seriously. I couldn't have done it without you all.

Abstract

In diagnosing and treating brain tumours, the tumour must be segmented, where clinicians sometimes aided by automated solutions, identify the tumour and substructures in a Magnetic Resonance Image (MRI). Performed manually, this is a time-consuming process. The goal of automatic segmentation techniques is to aid clinicians in the process of segmenting a tumour, providing an automatically generated segmentation which can be verified and edited. In recent years, Machine Learning architectures have been deployed to aid in brain tumour segmentation and have begun to be used in the industry. This projects work is divided into two main parts, first is the creation of a 2D U-Net based architecture, capable of automatically segmenting tumours. The model achieved a Dice score metric, measuring the accuracy of the segmentation, of 0.862, 0.792 and 0.745 for the Whole Tumour, Tumour Core and Enhancing Tumour respectively. The second part of the project uses the U-Net model as the basis for experimentation of different loss functions, examining which of 9 different loss functions contributed to favourable performance. The best, using Combo Loss, improved on the original model's performance, achieving Dice scores of 0.869, 0.813 and 0.865 for the Whole Tumour, Tumour Core and Enhancing Tumour respectively. The performance of both models is comparable to or outperforms the existing state-of-the-art research.

Contents

Declaration	2
Acknowledgements	3
Abstract	4
1. Introduction	7
1.1. Background.....	7
1.2. Project Objectives	7
1.3. Data	8
1.4. Ethical Considerations	8
2. Context Survey	9
2.1. Clinical Motivation.....	9
2.2. Cancer Detection and Segmentation	10
2.3. Approaches to Segmentation	12
2.4. Recent Studies	16
2.4.1. BraTS	17
2.4.2. DeepMedic	18
2.4.3. U-Net.....	18
2.4.4. V-Net	19
2.4.5. EMMA	20
2.4.6. Two Stage Cascaded U-Net	21
2.4.7. W-Net.....	22
2.5. Summary and Research Questions	22
3. Implementation Methodology	26
3.1. Development Resources.....	26
3.1.1. Hardware Used.....	26
3.1.2. Languages and Packages	26
3.2. Project Workflow	27
3.1. Group Model.....	28
3.1.1. Data Loading & Pre-processing	28
3.1.2. U-Net Architecture	29
3.1.3. Dice Coefficient.....	32
3.1.4. Evaluation Metrics	33
3.2. Individual Work.....	33
3.2.1. Base Architecture	33
3.2.2. Loss Function Selection.....	34
4. Results	36

4.2. Group Model.....	36
4.2.1. Loss and Accuracy During Training	36
4.2.2. Dice, Precision, Recall and F1 Scores.....	37
4.2.3. Example Overlays	38
4.2. Individual Work.....	39
4.2.1. Summary of Results	39
4.2.2. Mean Squared Error	41
4.2.3. Dice Loss.....	45
4.2.4. Cross Entropy Loss.....	47
4.2.5. Combo Loss	51
4.2.6. Focal Loss.....	55
5. Discussion and Evaluation	57
5.1. Discussion of Results	57
5.2. Evaluation.....	58
5.2.1. Value Alignment.....	58
5.2.2. Summary and Future Work.....	60
6. Conclusion.....	61
7. References.....	62
A. Appendix.....	66
A.I. – Supplementary Tables.....	66
Table A.1. – Division of Group Work.....	66
Table A.2. – Loss Function Dice Scores	66
Table A.3. – Loss Function Dice Scores	67
Table A.4. – Project Dice Scores with Context Survey	68
A.II. – Ethical Approval Letter.....	69
A.III. – System User Guide	70
1. Source Code Directory Listing	70
2. System Setup	71
3. Execution.....	72

1. Introduction

1.1. Background

In the UK, Brain cancer is the 9th most common cause of cancer death, with around 5400 in 2018 with around 12,000 cases of brain tumour each year [1]. A tumour can be defined as a set of unhealthy, abnormal cells, characterised by the way they reproduce in an uncontrolled manner. Brain cancer was found by [2] to have a 18% survival rate after 5 years, [1] found the 10-year survival rate to be around 10%. Early and accurate diagnosis of brain tumours is critical in positively influencing patient outcomes. In order to detect brain tumours, various imaging modalities including Computerised Tomography (CT) and Magnetic Resonance Imaging (MRI) can be deployed, with MRI being the more popular modality.

Segmentation is a critical task to enable accurate diagnosis, treatment planning and treatment itself. During segmentation the tumour region is divided into active tumorous tissue, necrotic tissue and edema. The segmentation is then used as a basis for diagnosis and treatment as well as potentially being used as the basis of targeted radiotherapy. [3, 4]. It is therefore crucial that segmentation be performed with a high degree of accuracy. It can be performed manually by clinicians, automatically by computer imaging systems, or a semi-automated approach can be adopted wherein an automated system performs a segmentation for verification and changes, if necessary, by a clinician. Research into automated approaches to segmentation has been of key focus of Artificial Intelligence (AI) research in recent years.

Machine Learning and Deep Learning (outlined in 2.3) are two popular AI approaches to tumour segmentation. ML applications focus on learning from experience to improve their decision making or predictive accuracy over time. DL models can extract and select features directly, meaning that the developers of the model need not define features for the algorithm. DL has proven to be favourable in recent years both to its improved performance compared to classical ML approaches and because DL's lack of feature definition means developers need not be subject matter experts in medical imaging. Many of the current state of the art applications performing tumour segmentation use Convolutional Neural Networks (outlined in 2.3), a type of DL application which is adroit at processing images, making it well suited for the problem of tumour segmentation [3, 5, 6, 7]. With the performance of CNN architectures at tumour segmentation improving year on year, this project will seek to contribute to this body of work.

1.2. Project Objectives

The first objective of the project was to conduct a Context Survey, exploring the clinical motivation for this project as well as current approaches to cancer detection and segmentation. Then, the Context Survey reviews in depth, a number of state-of-the-art research papers solving the problem of brain tumour segmentation. Particular focus has been given to research which utilises a version of the BraTS dataset, which this project uses (outlined in 1.3.).

This project was conducted partially as a Group project. The group worked together on the next objective, to create an architecture which can be trained on 2D MR images from the BraTS dataset to automatically segment brain tumours. This will include loading in the BraTS dataset and applying any pre-processing steps, then splitting dataset into training, validation, and test sets. Next the Group has developed a U-Net (outlined in 2.4.3. and 3.1.2.) based model architecture which can train on the BraTS dataset, then be used to make predictions for the validation or test sets, which can then be evaluated. The goal was for this model to perform to a similar standard to recent state-of-the-art applications found in the Context Survey when comparing their evaluation metrics (5.1.).

Then, as Individual work, the Group model has been extended, adding experimentation 9 different Loss Functions to the model. The selection of where to extend the Group model develop as part of the Individual work has been informed by the Context Survey, expanding on areas for development identified in 2.5.

1.3. Data

This project uses the BraTS 2020 dataset, an anonymised dataset of 369 records including MRI images and ground truth labels. The use of this dataset has been approved in accordance with [8, 9, 10]. BraTS is one of the largest publicly available datasets for brain tumour segmentation. In subsequent years, the BraTS challenge has included a new version of the BraTS dataset, holding a competition where researchers can develop a segmentation algorithm using BraTS then submit this to the team. The performance of researchers' models are ranked and the top performers research papers included in that years BraTS publication. Researchers wishing to utilise the BraTS dataset are not required to enter this competition. The BraTS dataset is a good choice for this project, as mentioned it is among the largest of its type available. Also, a large amount of research into brain tumour segmentation uses the BraTS dataset, so using the same dataset will aid in comparison between the performance of the model developed in this project and those found in the Context Survey.

1.4. Ethical Considerations

The key ethical concern for this project was the use of sensitive medical data and the possibility of an individual being identified by this. The BraTS is assembled with the consent of those included in it. It is fully anonymised, so it would not be possible for an individual to be identified and no attempt was made to do so. The use of this dataset has been approved in accordance with [8, 9, 10]. Because of the use of personal data pertaining to medical information, full ethical approval has been applied for and granted, included in Appendix A.II.

2. Context Survey

2.1. Clinical Motivation

In the UK, there are around 12,000 cases of brain tumour each year with half of those diagnosed with a cancerous tumour [1]. A tumour may be defined as a set of unhealthy, abnormal cells, characterised by the way they reproduce in an uncontrolled manner. Of those who are found to have a tumour, many are also diagnosed with secondary tumours, whose cells have metastasised to another part of the body. In the UK, Brain cancer is the 9th most common cause of cancer death, with around 5400 in 2018 and a 5-year survival rate of 12%. Incidence rates for brain tumours are projected to rise by 6% between 2014 and 2035 [1, 3]. Tumours can be benign or malignant. Malignant tumours are cancerous, with more aggressive cell structures which can damage surrounding tissue regions and potentially metastasise. Benign tumours are noncancerous and have a less aggressive nature, forming slowly with no chance of metastases [11].

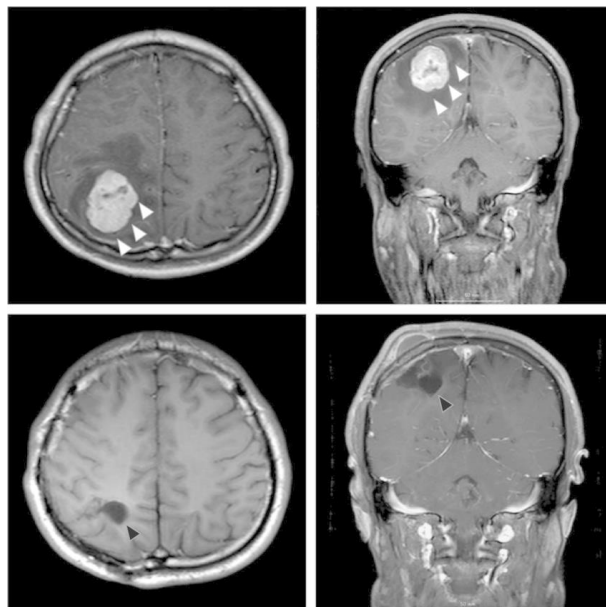


Figure 1 - Magnetic resonance imaging of the brain metastasis. Above, representative axial and coronal T1-weighted Magnetic resonance imaging (MRI) slices of the brain. White arrow: brain tumour. The images reveal a 45-mm mass on the left side of the brain. Below, axial and coronal T1-weighted gadolinium-enhanced MRI slices, taken 6 months post-operation. Black arrow: post-treatment change after operation. [12]

The most common type of brain tumour to occur are gliomas. The World Health Organisation (WHO) implemented a standardised classification system for gliomas, grading them from I to IV according to their severity in terms of growth rate or level of malignancy [13]. Grade I gliomas are less invasive and malignant than Grade II or III gliomas which are faster growing and have a greater level of malignancy. The most severe, Grade 4 gliomas are also known as Glioblastoma Multiforme (GBM) [14]. The grade of glioma has a significant impact on survivability, with median survival of 15-16 months for those with GBMs and receiving treatment [15]. Low grade gliomas are more common in children and young adults [16], which presents a specific danger regarding the likelihood of early diagnosis and treatment.

There is a huge variation in the survival rate between cancer types. A study by [2] evaluated the survival rates of cancer types at 1, 5 and 10 years. They found that of the 21 most common cancers, 12 have a 10-year survival rate of 50% or more. Stomach, brain, oesophageal, lung and pancreatic cancers all have survivability of less than 20% indicating that they are all difficult to diagnose and/or treat [2, 1]. Brain cancer specifically was found by [2] to have a 18% survival rate after 5 years, however [1] found the 10-year survival rate to be around 10%. Although there is certainly some variation to the survival rates found by different studies, all agree that among cancers, brain cancer is particularly difficult. For all types of tumour however, early and accurate diagnosis is key in order to detect and treat the tumour before metastases can occur, improving the prognosis for the patient and treatment [17, 1, 2, 18].

2.2. Cancer Detection and Segmentation

In order to detect brain tumours, various imaging modalities including Computerised Tomography (CT) and Magnetic Resonance Imaging (MRI) scans can be deployed, however MRI is the more popular modality due to the increased image contrast and spatial resolution for soft tissues such as brain tumours [19]. For many cancers, a biopsy can be taken to ascertain for certain whether a group of cells is cancerous. However, biopsies - especially those in and around the brain - present a greater level of risk to the patient than imaging techniques [18]. For this reason, an MRI scan is often one of the first diagnostic tools used. Figure 1 shows an example of an MRI scan.

MRI is a non-invasive, non-ionising, imaging technique which produces high spatial resolution images which give a high level of contrast between different types of soft tissue. It is favourable to use a non-ionising form of imaging since this means avoiding exposing the patient to harmful radiation. The majority of brain tumour diagnosis and the identification of position is done using MRI Images [11]. Usually, the images are in 2D and represent different levels or 'slices' of the brain. Putting a stack of these slices together one can produce a 3D model of the brain [3]. There are four MRI sequences commonly used in diagnosis: T1, which is typically used to isolate healthy tissue; T2, which is better at detection the edges of edemas; T1- Contrast Enhanced (CE), which highlights tumour borders and finally Fluid Attenuated Inversion Recovery (FLAIR), which favours the detection of edemas in the Cerebrospinal fluid [6, 14]. Figure 2 below demonstrates the difference between the MRI sequences. Using different MRI sequences for segmenting tumours leads to greater prediction accuracy since the modalities can be used together to provide a more holistic view of a gliomas sub-regions [3]. Glioma tumours are particularly difficult to detect since their boundary intensities in the image are not easily separable from the normal tissue [11].

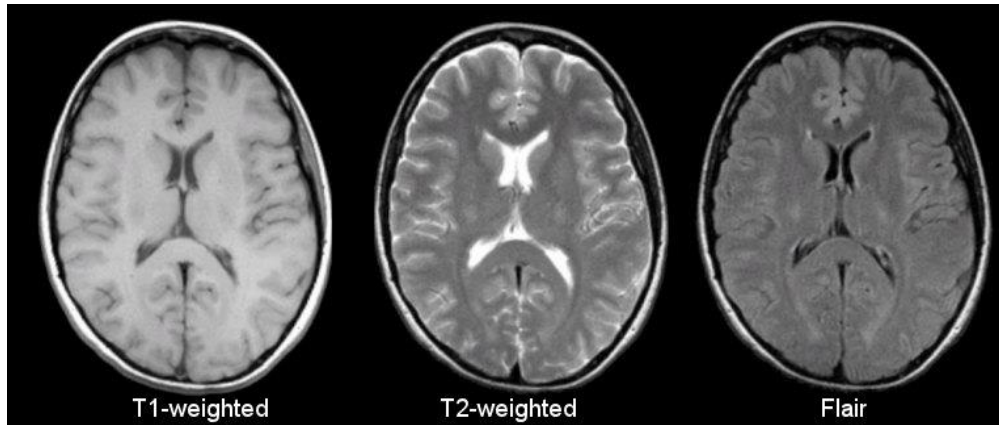


Figure 2 - Comparison of T1 vs T2 vs FLAIR MR Imaging Modalities. Image source: [20]

Segmentation is a critical task to enable accurate diagnosis, treatment planning and treatment itself. During segmentation the location and size of the tumour must be determined with its boundary precisely outlined. The outline obtained through segmentation can be used as the basis of targeted radiotherapy, so it is important that the segmentation exposes as little healthy tissue as possible. The tumour region itself is divided into active tumorous tissue, necrotic tissue and edema. By repeating the process of taking an MR image then performing segmentation, clinicians can track the progress of treatment. [3, 4]

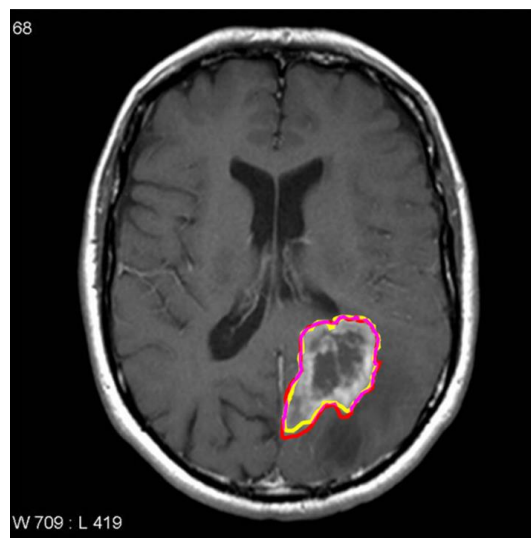


Figure 3 - Manually Segmented MRI. Red and Yellow outlines are the segmentations performed by two radiologists, the purple outline is the intersection. [21]

Different types of brain tumour have varying levels of difficulty in segmentation. Meningiomas can be easily segmented, whilst gliomas and glioblastomas are much more difficult to isolate. This is because as discussed above, their boundaries are not easily separable from healthy tissue due to the low contrast between them, and their shape, which can include root-like structures [22]. For all types of tumours, accurate segmentation is difficult given the large variation in their shapes (which can be very complex), size and diffusion levels at the boundaries with healthy tissue. Figure 3 shows an example of a brain tumour segmented manually, which is to say by a radiologist rather than by automated means.

2.3. Approaches to Segmentation

As discussed earlier, tumours in an MRI will show a large variance in shape, size and smoothness of the boundary with healthy tissue. This makes developing automated techniques for segmentation challenging, since such techniques often segment tumours based on intensity changes relative to healthy tissue. Furthermore, this high degree of variance in features makes it difficult for algorithms to assign strong priors to said features. In difficult cases where tumours have smooth boundaries with healthy tissue, experts and automated systems alike will show variation in their segmentation output [8, 23].

Current approaches to segmentation can be classified into manual, semi-automatic and automatic. Due to the high level of accountability required from the accuracy of the segmentation, current industry methods rely on manual or semi-automatic techniques [11]. Figure 2 shows an example of a manual segmentation technique [21]. In it, two radiologists have manually segmented the tumour and their intersection has been drawn. This is not only a time-consuming task for the human experts to perform, but the segmentation is usually performed on a 2D image rather than the 3D volume of the brain. This means that the expert is limited in their ability to represent the brain in a third dimension. Using multiple slices from different orientations in order to cover the x, y and z axis can help with this but will compound the earlier issue of time requirements for human experts [11, 3]. Developing some form of automated segmentation system is therefore favourable since it would save radiologists time, assuming it could perform the task relatively quickly. It could also contribute to an improvement in the accuracy of segmentation, since compared to manual segmentation, automated approaches have proven the potential to be more accurate and reliable [3, 6].

In order to combat the issues of intensity range variability, contrast and noise present in MR images discussed above, prior to automated analysis, several image processing techniques are performed. These techniques include registration, skull stripping, bias field correction, intensity normalisation and noise reduction [23, 14]. Noise in images is a random variation in the brightness of certain pixels. MR images contain a large amount of noise, making it difficult to delineate between tumorous and normal tissue. Reducing noise in the image, increasing the contrast between regions. Skull-stripping is the process of removing the non-cerebral tissue region such as skull, scalp, and soft tissues. Removal of this region reduces the chances of misclassifying diseased tissues. [23, 14]. Intensity normalisation is a critical pre-processing step in which intensities are mapped onto a standard reference scale. [23]. A Bias field signal is a low frequency, smooth signal that corrupts MRI images, reducing the high frequency contents of the image such as edges and contours. This degrades an algorithm's ability to identify tumour regions for segmentation [24]. Bias field correction uses image processing techniques to account for the presence of bias fields in MRI images. For multimodal images, registration is used to create a common space of reference. In most cases this is performed using a linear transformation model with the Mutual Information (MI) similarity metric and resampling in order to ensure correspondence across all modalities [23].

Machine Learning (ML) and Deep Learning (DL) are two popular approaches to tumour segmentation. ML applications focus on learning from experience to improve their decision making or predictive accuracy over time. One specific type of Machine Learning model is a Neural Network (NN). NNs are comprised nodes arranged in layers, with an input layer, one or more hidden layers and an output layer. Each node connects to another via a weighted connection and threshold. If the output of any node is above the specified threshold value, that node is activated, sending data to the next layer of the network. In training, the weights and threshold values of each node are fine-tuned based on the actual observed value at the final layer through a process called backpropagation [25]. DL is a class of ML, usually involving the use of Neural Networks with two or more hidden layers. In ML, feature selection and extraction are steps human developers perform. DL models can extract and select features directly, meaning that the developers of the model need not define features for the algorithm. This highlights a benefit of DL over ML in the field of medical image analysis, it does not require developers to be domain expertise in the field [18]. Figure 4 gives an example of the output from the automatic segmentation algorithm used in [21].

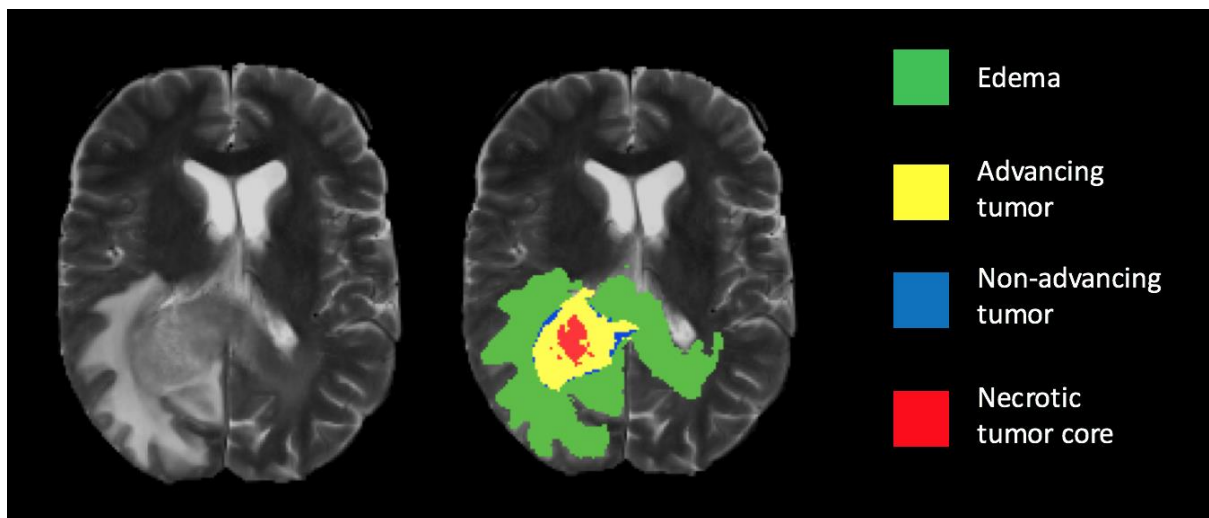


Figure 4 - Automatically Segmented Brain MRI, key on right identifies segmented regions of tumour (edema, advancing tumour, non-advancing tumour, necrotic tumour core). [21]

Machine learning approaches can be categorised as supervised, semi-supervised or unsupervised learning. With supervised learning, the training set includes ground truth labels which the algorithm then uses to establish priors and learn features of the dataset from the provided labels. With unsupervised learning, the algorithm does not require manually labelled training data. Instead, unsupervised learning models group (cluster) similar pixels or regions. This property makes unsupervised learning algorithms poorly placed for tumour segmentation, since they are only able to regionalise or cluster predictions and clinical environment the segmentation must be absolute. Furthermore, in unsupervised learning one must explicitly state the number of regions in the segmentation, however it could be that there are more regions than that required to properly segment the tumour(s) [11]. In the review of state-of-the-art solutions to the task of tumour segmentation, almost all models used utilised supervised learning. For this reason, this project will focus on a supervised approach to this task. Some unsupervised methodologies such as W-Net [26] will be reviewed, including what aspects of their solution could inform future development of supervised approaches.

In recent years, the use of DL has been extended with the use of Convolutional Neural Networks (CNNs), which have proven to outperform other approaches due to their adroitness at processing images [3, 5, 6, 7]. A CNN is an example of a Deep Neural network, typically comprising three types of layers of nodes: convolution, pooling and fully connected layers. Convolution and pooling layers both perform feature extraction and fully connected layers map the extracted features to the final output [7]. Figure 5 below shows an example of a CNN architecture [7].

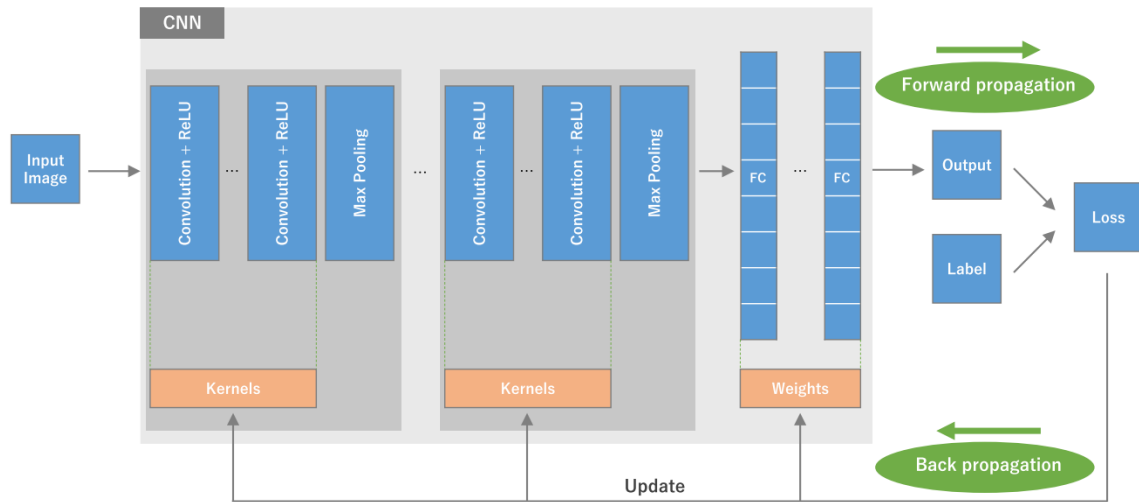


Figure 5 - Example of a Convolutional Neural Network. [7]

The convolutional layer is a fundamental component of the CNN which performs feature extraction, usually consisting of a number of linear and non-linear operations, convolution and an activation function. During convolution, a kernel is applied across the input, an array of numbers known as a tensor. An element-wise product between each element of the kernel and the input tensor is calculated at each tensor location and summed to obtain the output value in the corresponding output tensor, which is known as a feature map. This process is repeated applying different kernels to form a number of feature maps, which represent different characteristics of the input tensor. Figure 6 gives an example of a convolution operation. One can therefore think of different kernels as different types of feature extractor. Thus, two key hyper-parameters for a CNN are the size and number of kernels [7, 27].

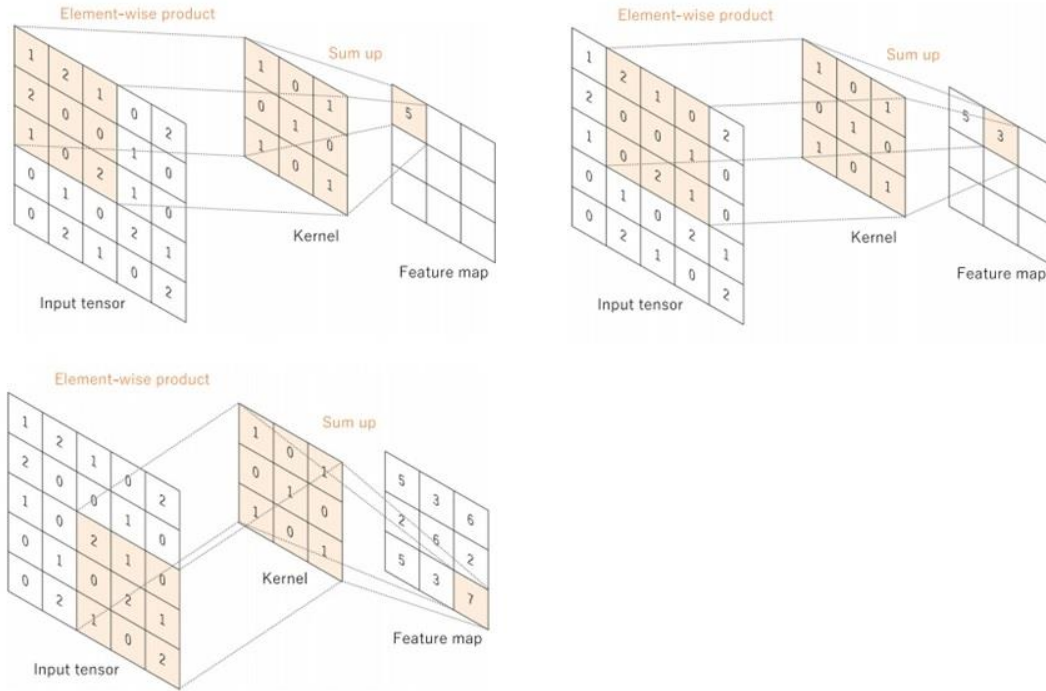


Figure 6 - An example convolution operation with a kernel size of 3x3, no padding and a stride of 1. A kernel is applied across the input tensor and an element wise product between each element of the kernel and the input tensor is calculated for each location and summed to obtain the output tensor, called a feature map. [7]

In order to allow the centre of each kernel to overlap the outermost element of the input tensor, a process called padding must be performed. Padding is the process of adding rows and columns of zeros to the outside of a tensor, such that the kernel used in convolution can move across all elements [7]. Another hyperparameter which must be defined is the stride length, this is the number of times the kernel shifts during each convolution, so a stride length of 1 (commonly used), would result in the kernel shifting one pixel across at a time. The first convolutional layer in the network will capture low-level features such as edges and gradient orientation. As more convolutional layers are used, higher level features can be extracted, building the networks understanding of the images with regard to the task set [27].

The outputs of the linear operation are then passed through a non-linear activation function, the most common activation function used currently is the Rectified Linear activation Unit (ReLU) function which computes the output as $f(x) = \max(0, x)$. The ReLU function will return the value provided if it is greater than 0, otherwise it returns 0. If the value is greater than 0 then the node has activated [7].

The pooling layer, like the convolutional layer is responsible for feature extraction. The goal of pooling is to use a down sampling operation to reduce the size of the feature maps in order to decrease the number of subsequent learning parameters. There are no trainable parameters for pooling layers as there are for convolution operations [7, 27]. There are two main types of pooling: max pooling and global average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel, average pooling returns the average of all pixel values from the portion of the image covered by the kernel [27]. Figure 7 shows an example of pooling including max and average pooling.

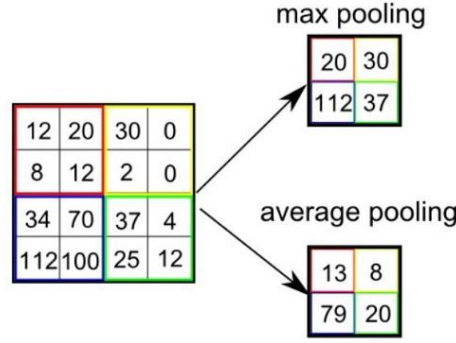


Figure 7 - Example of pooling approaches. In max pooling, the highest value cell from each region is taken. In average pooling, the mean of each region is used. [28]

The output feature maps from the final convolution and pooling layer are typically flattened – transformed into a vector – then passed through one or more fully connected layers, in which every input is connected to every output by a learnable weight. The fully connected layers map the output of the feature maps to the final output of the network such as the probabilities for each class for a classification task [7]. The activation function of the last fully connected layer is often different from other activation functions since an appropriate function needs to be selected according to the task. So, for example, for multiclass classification tasks, a Softmax function which normalises values to target class probabilities between 0 and 1 is commonly used [7].

Training the network is a process of finding kernels in convolution layers and weights in fully connected layers such that the difference between the output prediction and ground truth label is minimised. A model's performance given certain weights and kernels is measured using a loss function which measures the similarity of the output predictions, and the ground truth labels. The particular loss function used is one of the hyper-parameters which developers must decide. Then, learnable parameters, namely kernels and weights are updated according to the value from the loss function through a process known as backpropagation. A commonly used optimiser for backpropagation is gradient descent, which updates the learnable parameters such that the loss value is minimised. The gradient of the loss function gives an indication of the direction in which the loss has an increase or decrease. Using this, the learnable parameters are updated in a negative direction of the gradient with an arbitrary step size known as the learning rate. The learning rate is another hyper-parameter which must be decided. In order to reduce computational complexity – particularly memory usage – of gradient descent, stochastic gradient descent (SGD) can be used, where a subset of the training set is used to calculate the gradient descent. Using SGD, the batch size is another hyper-parameter [7]. Learning is complete when the model has converged, and the loss function has reached a global minimum over training time.

2.4. Recent Studies

This section will explore some of the recent studies in the area of brain tumour segmentation, with specific focus on those which utilised the BraTS dataset. This provides an understanding of state-of-the-art applications in the field, what makes a solution favourable or less so, and where this project can focus its development. First, will be examined the research papers by Menze et. al. which first presented the BraTS benchmark.

2.4.1. BraTS

A 2015 paper by Menze et. al. presented the multi-modal brain tumour image segmentation benchmark (BraTS), in conjunction with the MICCAI 2012 and 2013 conferences. They worked to generate the largest publicly available dataset for brain tumour segmentation and evaluated a number of state-of-the-art applications at that time. The images collected for the dataset include low and high grade glioma patients from a mix of pre and post therapy scans. They were collected by Bern University, Debreen University, Heidelberg University, and Massachusetts General Hospital over the course of several years. The dataset includes scans of different modalities including T1, T1c, T2 and FLAIR. There is also a number of simulated MRI scans created to augment the existing dataset. All images come with ground truth labels defined by either the augmentation software or a medical professional (Figure 8). The BraTS dataset published has had pre-processing performed on its images including skull stripping and registration, but bias field correction, noise reduction and intensity normalisation is left for researchers using the dataset to perform [8]. The BraTS test data uses metrics in the form of a Dice Score, Sensitivity and Specificity for the evaluation of the principle tumour regions; namely the whole tumour - comprising of all tumour components - , the core tumour -comprising of all tumour components except for the edema- and the active tumour -comprising of only active cells [8, 11].

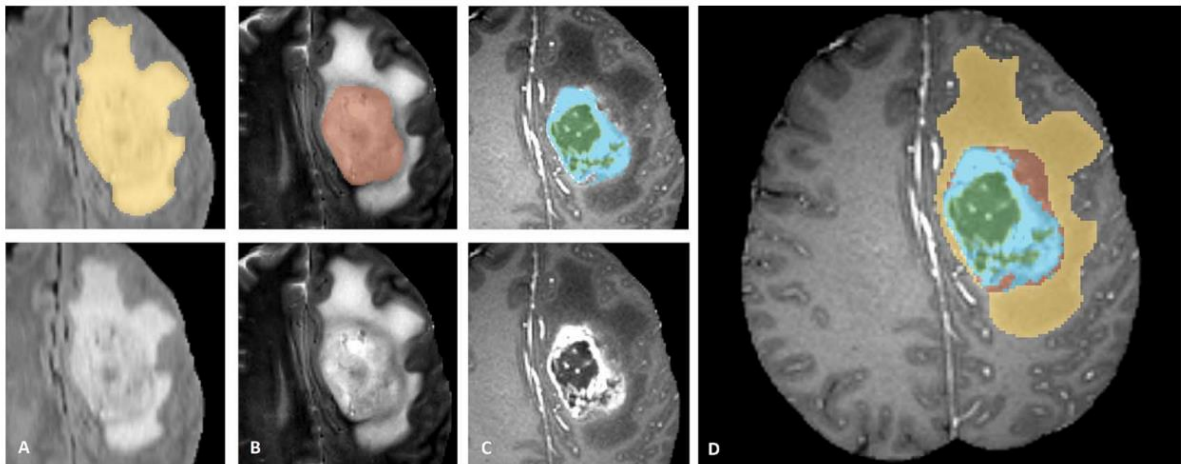


Figure 8 – Example image from the BraTS dataset including manual annotation by experts. Shown are image patches with the tumour structures that are annotated in the different modalities (top left) and the final labels for the whole dataset (right). Image patches show from left to right: the whole tumour visible in FLAIR (A), the tumour core visible in T2 (B), the enhancing tumour structures visible in T1c (blue), surrounding the cystic/necrotic components of the core (green) (C). Segmentations are combined to generate the final labels of the tumour structures (D): edema (yellow), non-enhancing solid core (red), necrotic/cystic core (green), enhancing core (blue).

In their evaluation of 20 different models, the researchers found that Dice scores for most models was between 60 and 82% for the whole tumour, 30% and 61% for the Core tumour and between 40% and 60% for the active tumour [8]. This paper demonstrated the advancement in automatic segmentation algorithms up to that point, since they were able to evaluate a large number of different proposed models against a single common dataset. In the years since the study the BraTS dataset has continued to increase in size, with competitions each year for researchers to create a model which performs best on the new dataset. In studies since, many researches have chosen to use

the BraTS dataset, irrespective of the competition due to it being the largest dataset of its type available [23, 29, 30, 11, 22].

Initially, the tumour subregions defined by BraTS were, as in Figure 8, the necrotic core, enhancing core, non-enhancing core and edema. In more recent iterations of the BraTS project, the non-enhancing core sub-region has been merged with the necrotic core, meaning current BraTS data has three tissue types, the necrotic core, the edema and the enhancing core. Another recent addition is the separate task of survivability prediction, where the task is, based on the MRI data, to predict survivability in months. [31]

2.4.2. DeepMedic

In [32], DeepMedic, a dual-pathway 3D CNN combined with a connected 3D conditional random field (CRF) for automatic brain lesion segmentation was put forward. The project hoped to address the issue of increased memory and computational load caused by the large number of parameters and 3D convolutions present when using 3D CNNs. [11] By processing 3D MRI volumes, they were able to make better use of the 3D contextual information compared to other studies using 2D slices [32, 3]. The researchers found that the DeepMedic model was able to outperform the existing state-of-the-art applications, with top ranking performance on the BRATS 2015 benchmark [32] as well as being computationally efficient. They reported an average of 6 minutes for the DeepScan model to perform the task. However, they found that their model was not able to generalise well when presented with the task of multi-class tumour segmentation, where the type of tumour must also be identified. Because of the variability in tumour substructures, finding a global set of parameters which improved performance for all classes proved challenging [32]. This does speak to a general challenge of multi-class classification however, rather than being a weakness unique to the DeepMedic solution.

2.4.3. U-Net

[33] presented U-Net, a CNN which relied heavily on an augmented dataset. U-Net was used to segment neural structures in electron microscopy images and achieved best results in ISBI challenge [6, 33]. The architecture includes a set of 5 convolutional channels and 5 deconvolutional channels [6]. U-Net implements a U-shaped architecture where each the two sets of channels form two paths in a U shape (Figure 9). The first contracting path comprises the convolutional and max pooling channels, its purpose is to process the context of an image. Each convolutional channel uses a ReLU activation function [6, 11]. The second expanding path creates a high-resolution segmentation map which is used to output a fully segmented image. In contrast to normal CNNs, the U-Net architecture does not include any fully connected layers [6, 33, 11]. [34] used a modified U-Net for tumour segmentation as part of the BraTS 2015 challenge, achieving dice scores of 86% for the Whole Tumour and Tumour Core and 65% for the Enhancing Tumour.

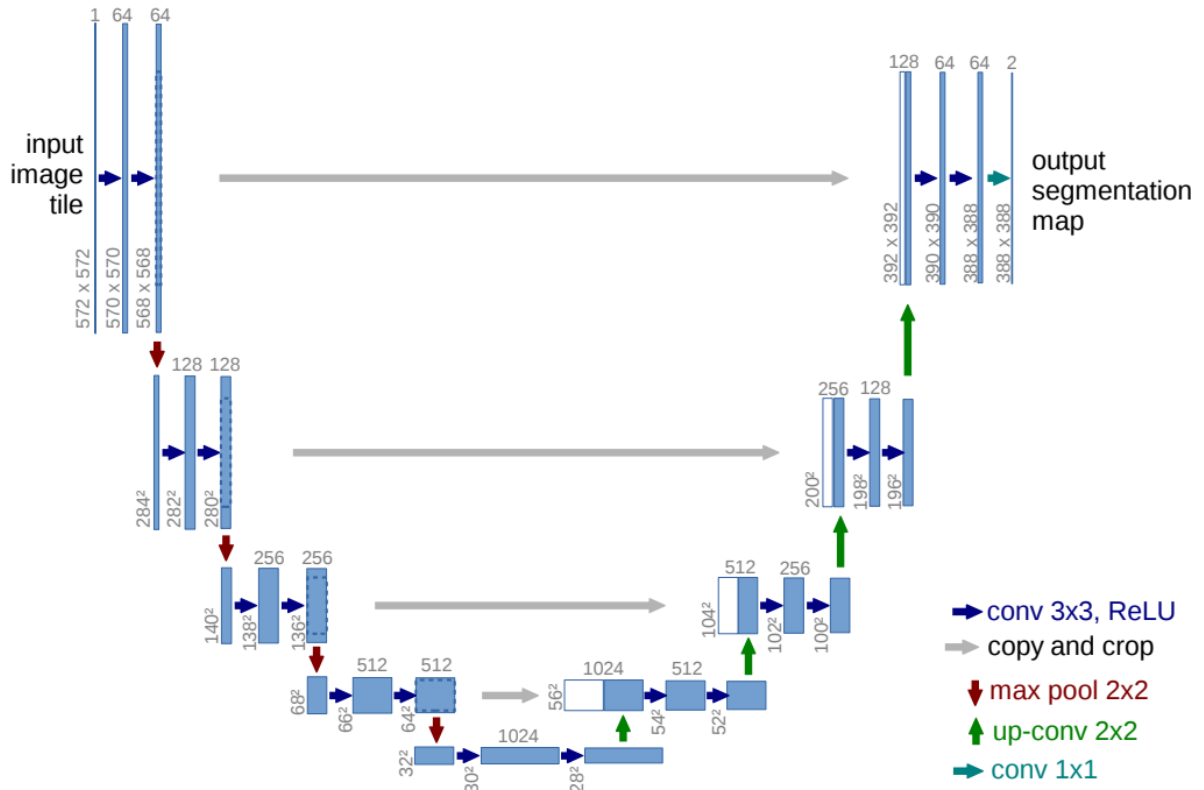


Figure 9 - U-Net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. An x - y -size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. [33]

The U-Net model provides several advantages for segmentation tasks: it allows for the use of global location and context at the same time, it works with very few training samples since it uses augmentation and demonstrated an improved performance over previous work. Finally, U-Net is an end to end pipeline which processes images in a single forward pass, producing segmentation maps. This ensures that U-Net preserves the full context of the input images. [35]

2.4.4. V-Net

Similar to DeepMedic, V-Net uses 3D convolutions to segment 3D volumetric data holistically, rather than stacking 2D image slices to form a 3D volume [36]. The model is trained end to end on MRI volumes of the prostate. Figure 10 shows a schematic of V-Net's architecture. Unlike the down sample, up sample approach shown in the U-Net approach in [35], the V-Net model incorporates high spatial resolution feature maps throughout the network's layers [3]. Researchers in [36] also introduced a new objective function based on the Dice coefficient which could more easily handle the situation where there is a large imbalance between the foreground and background voxels. They found the model achieves good performance on challenging data whilst requiring much less processing time than previous models [36]. The new objective function proposed by V-Net is one of its key advantages, since it does not need samples to be re-weighted when the number of background and foreground pixels is imbalanced [11].

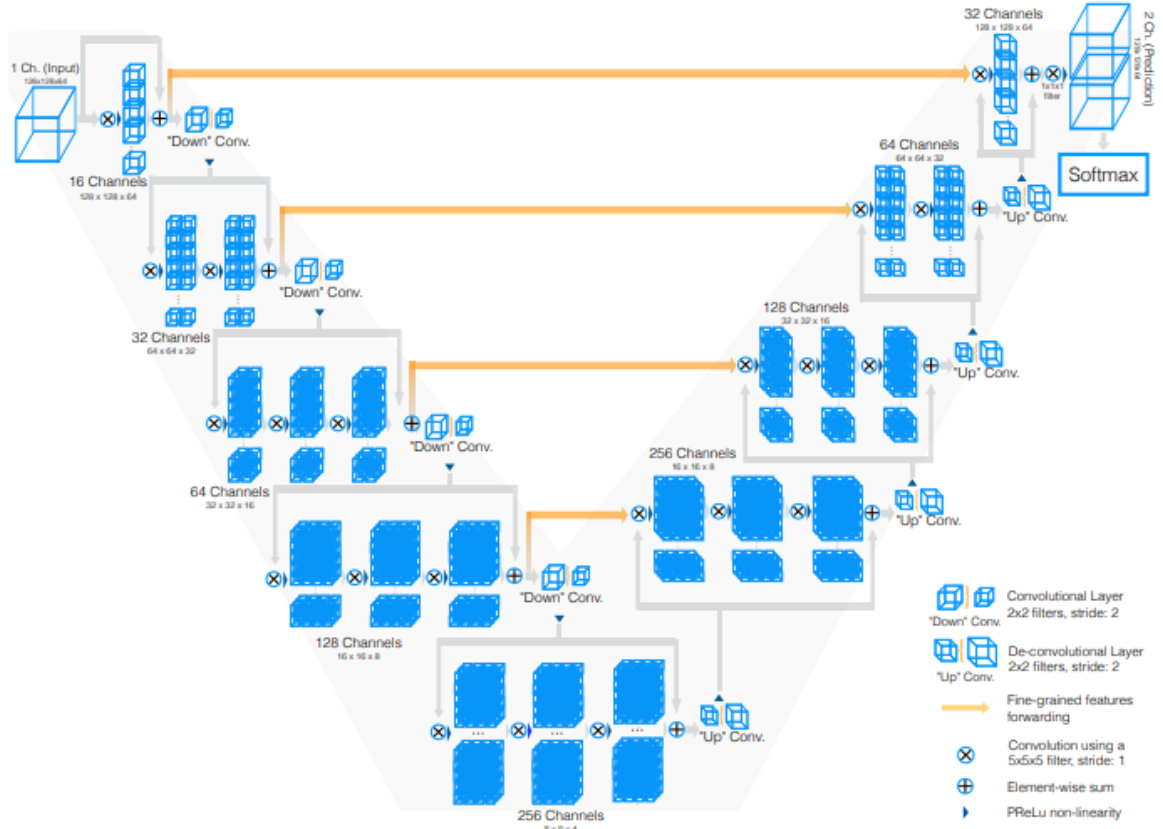


Figure 10 - Schematic representation of V-Net The left part of the network consists of a compression path, while the right part decompresses the signal until its original size is reached. Convolutions are all applied with appropriate padding. [36]

2.4.5. EMMA

Kamnitsas et. al. [30] presented Ensembles of Multiple Models and Architectures (EMMA) for Robust Brain Tumour Segmentation. Their solution used two DeepMedic models [32], three Fully Convolutional Networks (FCN) [37] and two U-Nets. All models are trained separately and at testing, each model individually segments the image. The results are then aggregated by assigning classes with the highest confidence levels from all the models. This approach showed an increase in Dice score over previous work such that it won 1st place in the BRATS 2017 challenge, with a score on the test set of 88.6 for the Whole tumour, 78.5 for the core and 72.9 for the enhancing core [30]. One key advantage of this approach is that it was shown to generalise well to other datasets since using a heterogenous collection of networks is insensitive to independent failures of individual component models [30]. One disadvantage is that this approach requires the models to run independently during training which, despite their being run in parallel, will be far more computationally expensive than other approaches.

2.4.6. Two Stage Cascaded U-Net

The approach put forward by [38] using the BRATS 2019 dataset won 1st place in that years competition with a Dice score on the test set of 88.8 for the Whole tumour, 83.7 for the core and 83.3 for the enhancing core. The model used a 2 stage cascading U-Net. In the first stage, a variant of U-Net was used to train a coarse prediction. In the second stage, the network width is increased, and two decoders are used boost performance. The second stage is added to refine the prediction map by concatenating a preliminary prediction map with the original input. Figure 11 below outlines the architecture of the model. The performance of the model shows a minor improvement in the accuracy of whole tumour segmentation but a major improvement in the scores for the core and enhancing core [38]. Data augmentation is used, which helps to reduce any overfitting. To augment the data, a per channel random intensity shift is applied to existing images as well as cropping and random flipping across the x, y or z axes. A disadvantage of the cascaded U-Net approach suggested is the level of memory requirements. As described by the researchers, the input images had to be cropped in order to overcome a memory limitation in their hardware which had 12 Gb of memory [38]. Also this approach of aggregating the results from multiple models has the disadvantage of not producing repeatable results, with performance variability between each model [38, 11].

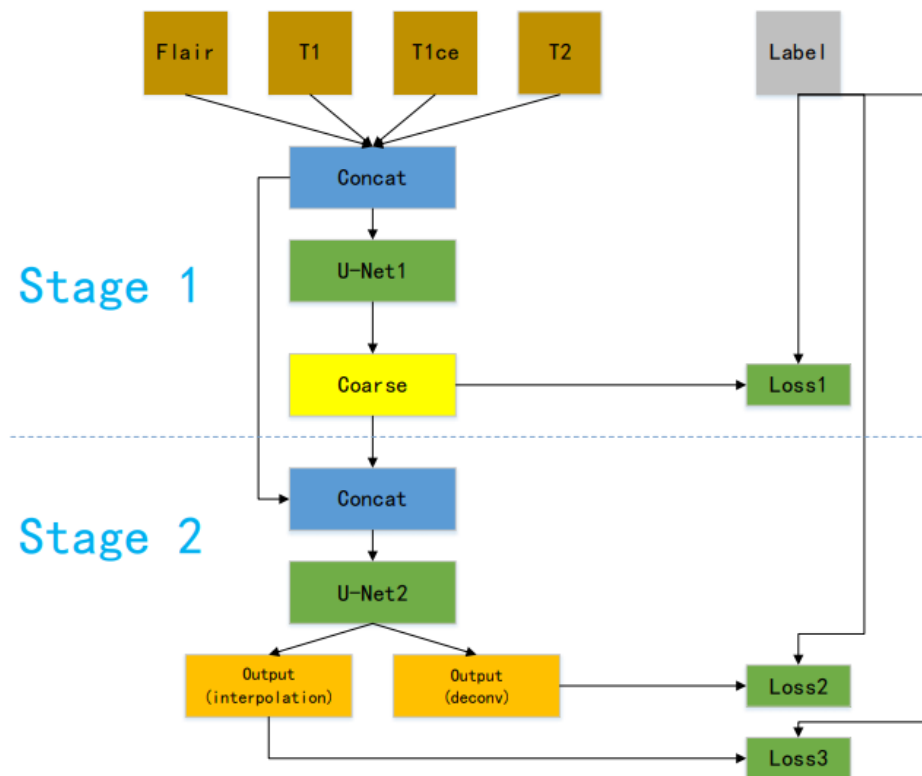


Figure 11 - Two Stage Cascaded U-Net Architecture, four MRI modalities are concatenated into the U-Net which is used to make a coarse prediction. This is then concatenated with the input and fed into second U-Net which makes final prediction [38]

2.4.7. W-Net

W-Net is another architecture built on U-Net, using a two-stage U-Net approach. [26] presented W-Net, trained for unsupervised multiclass segmentation of images. The architecture concatenated two convolutional networks into an autoencoder with one for encoding and one for decoding. Combined with post-processing with conditional random field smoothing and hierarchical segmentation, the resulting architecture outperformed many recent techniques and achieved human level performance [26].

A year later, [39] examined the performance of the W-Net architecture along with two other networks at the task of 2D brain tumour segmentation using the BRATS 2018 dataset [39]. They adapted W-Net for one pass multi-class prediction without using cascaded prediction. The W-Net model achieved a Dice score of 75.70%, 88.98% and 72.53% for the enhancing tumour core, whole tumour, and tumour core respectively [11, 39]. W-Net then, shows similar performance at 2D segmentation to other architectures explored so far. One disadvantage with the W-Net approach is that the increased complexity of its architecture increases the number of learnable parameters, making it more computationally expensive to train the model than for example, base a U-Net implementation.

W-Net served as the basis of the Autofocus Net presented by [3], who tested their model on the BRATS 2018 dataset. The network is comprised of a modified version of W-Net along with a number of autofocus layers replacing the last convolutional layer of the standard W-Net. They found that using autofocus layers with four parallel convolutions performed better than with three, suggesting that an increase in parallel convolutions enables the autofocus layer to handle the variability in brain tumour sizes more effectively [3]. Using this approach, Autofocus Net achieved a Dice Score of 66.88, 55.16, 64.13 for whole tumour, core tumour and enhancing tumour respectively on the BRATS 2018 dataset.

One problem the researchers in [3] found was that NiftyNet, the platform utilised to run W-Net offered fewer options regarding pre-processing such as normalisation and data augmentation. This, along with the increased test size when splitting the dataset, and the need for greater hyper-parameterisation lead to an issue in the replicability of results. Another issue found was that the increase in parallel convolutions by the autofocus layers lead to an increase in computational time and cost [3].

2.5. Summary and Research Questions

To summarise the findings from the wider Context Survey, first was examined the clinical motivations for this project. The initial statistics observed regarding the prevalence and survivability of brain cancer accounts for a large number of deaths each year and that early detection of a cancer is essential in influencing positive outcomes [17, 1, 2, 18]. Part of the diagnostic and treatment process includes segmenting the tumour into subregions which can be used for diagnostic and treatment purposes including targeted radiotherapy [3, 4]. Currently only manual, and semi-automated approaches are implemented in the field due to the high level of accountability needed from such a system [11]. With semi-automated approaches, clinicians work with

automated segmentation tools. However, it was shown in the review of recent implementations that automatic approaches to segmentation have the potential to segment more accurately and faster than manual approaches. Therefore, a well implemented, semi-automated approach, where clinicians verify and if needs be correct an automatic segmentation could save time and segment tumours more accurately, positively impacting patient outcomes.

Model	Dataset Used	Dice Coefficient		
		Whole Tumour	Tumour Core	Enhancing Tumour
DeepMedic [32]	BraTS 2015	89.8%	75.0%	72.1%
EMMA [30]	BraTS 2017	72.9%	88.6%	78.5%
Cascaded U-Net [38]	BraTS 2019	88.7%	83.7%	83.3%
W-Net [3]	BraTS 2017	69.8%	54.9%	63.0%
W-Net [39]	BraTS 2018	87.8%	79.6%	74.7%
U-Net [34]	BraTS 2015	86.0%	86.0%	65.0%

Table 1 - Dice Scores for Context Survey Research which use versions of the BraTS dataset.

Regarding recent approaches to automated segmentation, the Context Survey has outlined the rise of DL, extending to CNNs in recent years [3, 5, 6, 7]. Table 1 above gives a summary of the dice scores for research from the literature which uses some version of the BraTS dataset. The mean scores between these results are 82.5%, 78% and 72.8% for the Whole Tumour, Tumour Core and Enhancing Tumour respectively. The top scoring architecture for the BraTS 2019 challenge was [38], who used a two stage cascaded U-Net and many other architectures also extend the U-Net based approach [3, 26, 38, 11, 36, 30, 39]. For this reason a U-Net architecture will serve as the basis of the solution presented by this project. Exploration of recent studies and architectures has formed a list of desirable features of potential segmentation systems which will inform the evaluation.

- Dice scores and other performance metrics.
These scores will be used to inform the overall performance of the algorithm at the task of segmentation and are therefore critical in assessing the effectiveness of the system.
- Computational Complexity
Thought should be given to the computational resources needed to use the algorithm. Some architectures require extremely powerful GPU machines or large amounts of RAM. It would be desirable to mitigate the computational complexity of the system architecture in order to allow those without access to powerful computer systems to run the code.
- Time Costs
The system should perform tasks in a sensible amount of time. That is, the elapsed time to load in the dataset, train the model, create a prediction set given a validation set and evaluate the results. This should all be performed separately if possible. Whilst it can take varying amounts of time to manually segment a

tumour, it is usually in the order of minutes, so a system architecture should aim for similar or better performance.

- Use of sufficiently large and representative datasets.
Datasets should be representative of the general population, including types of tumours, locations and demographics of patients. This ensures that computer scientists and clinicians can place more trust in the validity of the systems performance. It will also help to ensure that the system can generalise well to its validation and test sets and by extension the real world.
- Addressing class imbalance of training set.
The raw data will include very imbalanced classes, with a much higher occurrence of non-tumorous tissue. A segmentation system architecture should seek to address this class imbalance in some way, by data augmentation, slicing and cropping or other means.
- Generalisability
The system should display similar performance between the test, training, and validation sets. This would indicate that the system displays good generalisability since it performs similarly well on unseen data. If there were increased performance on the training set to the validation or test sets, then the system would be showing overfitting, indicating that it would not generalise well.
- Accessibility
The system should ideally be written in a well-known programming language such as python, with good commenting and object-oriented structure, using standard packages for machine learning such as TensorFlow, sci-kit learn or pytorch. Ideally the system will also be available on a repository such as Github. The accordance to these principles will ensure that other researches can easily adopt the architecture or replicate the results observed.
- Interpretability
The output of the system's predictions should be interpretable and easy to understand by the end user or stakeholder. High quality segmentation maps should be properly labelled and overlay the original image.
- End-to-End architecture.
The system will display an end-to-end architecture, where an MR image or set of images can be given to the model, which will then output a set of predicted segmentation maps as images, with the segmentations overlaid over the original MR image. As well as being end-to-end the system should also ideally be modular, allowing tasks such as data loading, pre-processing, model training and model evaluation being performed separately.

Throughout the Context Survey, researchers have noted the loss functions used by the model, this includes the Dice loss function, Cross Entropy Loss, Focal loss or a combination of Dice and Cross Entropy. In no case, however, have researchers explained the decision making behind the loss function selection, nor explored the performance of a benchmark model using different loss functions. [40] performed a survey of loss functions for semantic segmentation, observing the characteristics of each and the scenarios in which they might be useful. This paper took a more general look at segmentation tasks and highlighted an area of exploration for this project. The individual development stage will therefore examine the performance of different loss functions on a benchmark model developed in in the group phase.

3. Implementation Methodology

3.1. Development Resources

3.1.1. Hardware Used

Model architecture was executed via the GPU using Docker. The system used, accessed remotely through SSH, had the following hardware:

- Intel Core i5-6500 CPU @ 3.2 GHz
- NVIDIA GeForce GTX 1060 6GB
- 32Gb RAM

3.1.2. Languages and Packages

The system was developed in Python with TensorFlow and Keras, using the Visual Studio Code development environment, which allows for easy development on the research machine through SSH. The use of Docker allows for a Docker Image to be created, which packages the Python environment and all the libraries required to run the architecture. Listed below are the libraries and packages used in development.

Library/Package	Use in system
TensorFlow	Contains packages for development of Machine Learning Systems.
Keras	Contains packages for development of Machine Learning Systems.
Numpy	Provides multidimensional array objects, used for dataset.
tqdm	Adds progress bar to loops.
Pickle	Object serialisation package for reading/writing objects to files. Used to save and read class weights.
Nibabel	Used to give read/write access to common medical imaging file formats.
SciKit Learn	Contains packages for development of Machine Learning Systems, used in evaluation to build classification report.
Python Imaging Library (PIL)	Extends Python's image processing capabilities. Used to create overlay images in evaluation.
Matplotlib	Comprehensive library for creating visualisations. Used to create plots during training and generating overlay images in evaluation.

Table 2 - Libraries and Packages used in Development

3.2. Project Workflow

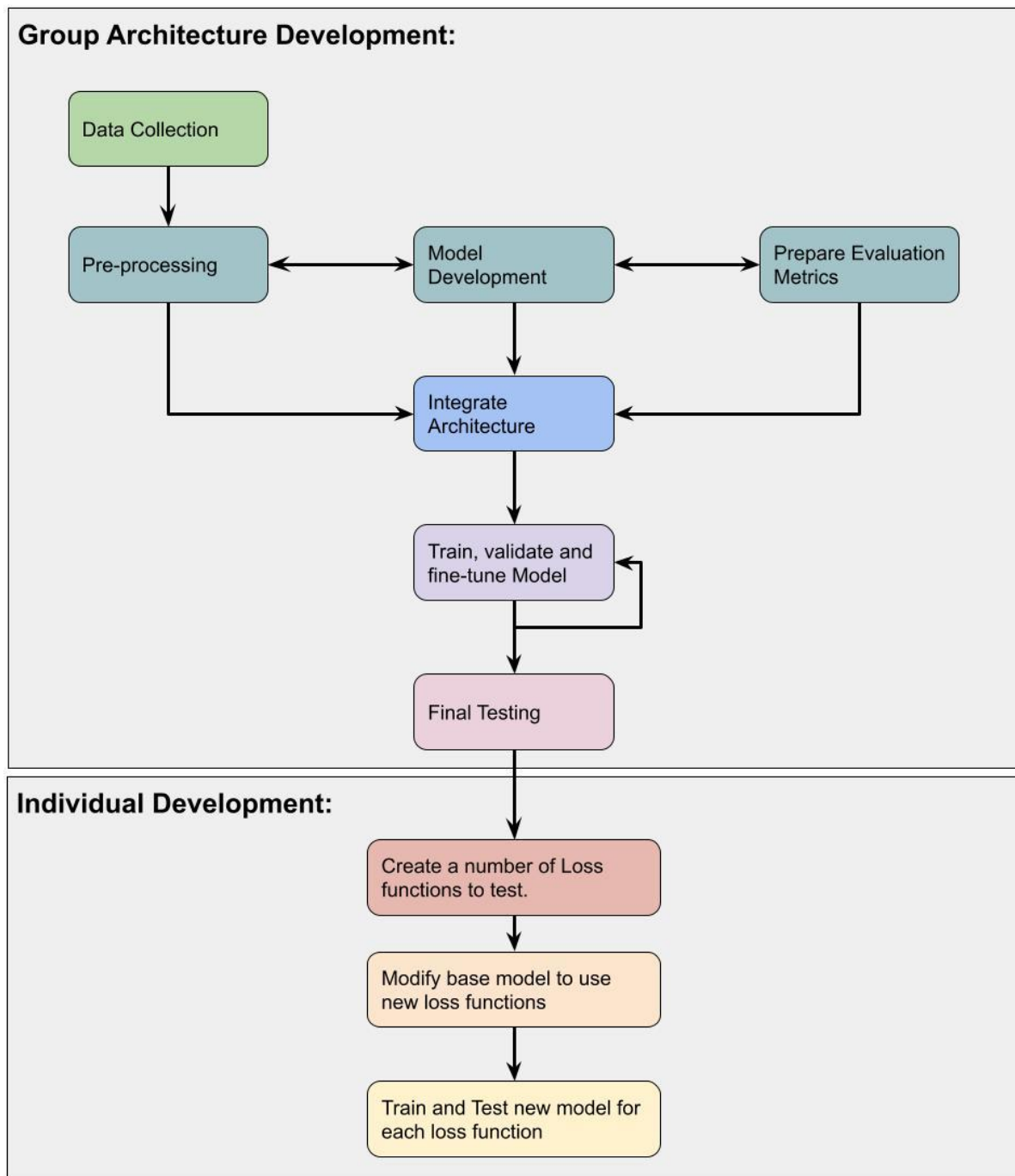


Figure 12 - Project Workflow including Group Architecture and Individual Development

The figure above outlines at a high level the workflow of the project. Initially the group worked together to build a baseline architecture including loading the BraTS dataset, applying any pre-processing steps, and developing a model to train and evaluate. The table in A.I. outlines the distribution of work as part of the project's development.

The architecture has been built with a modular design. As outlined by the system's User Guide (A.III.), the architecture is executed by running the Data Load, Model and Evaluation files. This separates the steps of loading and pre-processing the

dataset, building and training the model and loading the model to make predictions which can be evaluated. This approach allowed for group members to work independently on each module of the architecture, which could then be integrated. Furthermore, it allowed for greater efficiency since when running the Training or Evaluation files, the preceding modules need only be executed once. There are two versions of the architecture, one configured to multiclass segmentation, the other to binary segmentation. This paper, the majority of the literature from the Context Survey and the BraTS challenge focuses on multiclass segmentation. The binary segmentation version was used in the earlier development stages when testing model designs, since a binary segmentation problem is simpler due to the reduced number of target classes.

3.1. Group Model

The Group architecture is available on GitHub via

https://github.com/DSmith537/Brain_Tumour_Segmentation_UNet_Group.

Making the project open source in this way allows other researchers to utilise the work in this project.

3.1.1. Data Loading & Pre-processing

The BraTS dataset includes 369 records, each a separate directory including .nii.gz files of the four MRI modalities (Flair, T1, T1-CE, T2) as well as a ground truth file. The first step in loading the dataset is for each of the records, to load each of the image modalities as arrays and normalise them, setting their cell values between 0 and 1. Normalisation ensures that the input variables have the same treatment in the model and that the weights assigned by the model are not scaled with respect to the units of the input variables. This generally speeds up learning during training, leading to faster convergence [41]. Once the arrays for each image modality are normalised, they are then combined to a single multi-dimensional numpy array and saved as a .npy file in the x directory.

The segmentation maps (ground truth files) are also loaded as arrays. They must then be re-encoded since the ground truth arrays contain the values (0,1,2,4). This is because, as described in the Context Survey, in [31] the class 3 (Non-enhancing Core) was merged with class 4. In re-encoding, the values of 4 in the ground truth arrays are set to 3, making them in the range of [0,1,2,3]. This is a requirement for correct one-hot encoding. Then, the segmentation maps are saved as .npy files in the y directory.

Next, the Class Weights must be calculated. It calculates the number of occurrences of each class in the set of y (ground truth) data. Then, weights for each class are calculated as $\frac{1}{\alpha} * 2\beta$, where α is the number of occurrences of the class within the dataset and β is the size of the dataset. These weights are then saved as a file to be used later to calculate the loss function during training. Next, the ground truth labels are one hot encoded, converting the segmentation arrays from the shape [192,192,1] with a range of values between 0 and 3 to the shape [192,192,4] where the 3rd dimension represents the 4 class values and the values of the array are 0 or 1. One hot encoding

ensures that the model does not place higher importance on higher class values (2 and 3) within the dataset.

Then, both the x and y datasets are reshaped to [90,128,128], which extracts the central 90 slices from each MR volume and takes a smaller patch of the original image, resizing it from 192 x 192 to 128 x 128. The central slices are extracted as they are more likely to have tumorous regions, which will improve the class balance and the Model's chances of learning to identify tumours. This is also done to reduce the tensor size during training due to VRAM limitations.

The final step in pre-processing which must be performed is to split the dataset into a training, validation, and test set. The training set is used to train the model, the validation set is used to evaluate the model's performance. At the end of the project, the models were evaluated on the test set using a modification to the evaluation class, the results of the evaluation class running on the test set form the basis of the results presented in this report. Ensuring that the test set is unseen by the model before final evaluation prevents data leakage, allowing the models generalisability to be assessed. This also ensures that the results presented in this report are scientifically valid, since no changes are made to the model architecture once it has been evaluated with the test set.

3.1.2. U-Net Architecture

The Model architecture developed as part of the group project is an adaptation on the U-Net architecture presented by [6, 33, 42]. U-Net has shown in the Context Survey to be a robust model in many image segmentation applications. Furthermore, it works well on datasets of limited size which the BraTS dataset is, despite being among the largest datasets in this domain. Also, U-Net has been used extensively for the problem of medical image segmentation and has formed the basis of many adaptations including those by [26, 38, 35, 30, 3, 36, 30]. It is also computationally reasonable in its costs, trains in a reasonable time and performs well at this problem due to its ability to preserve context within the input image.

The Context Survey (2.4.3.) gave a high-level description of how the U-Net architecture functions. It includes a set of 5 convolutional channels and 5 deconvolutional channels, forming a U-shaped architecture. The first contracting path processes the context of an image, the second expanding path creates a high-resolution segmentation map. Figure 13 shows a graphical representation of the U-Net architecture developed as part of this project. It also has a set of 5 convolutional channels and 5 deconvolutional channels which form a U shape.

Group U-Net Architecture

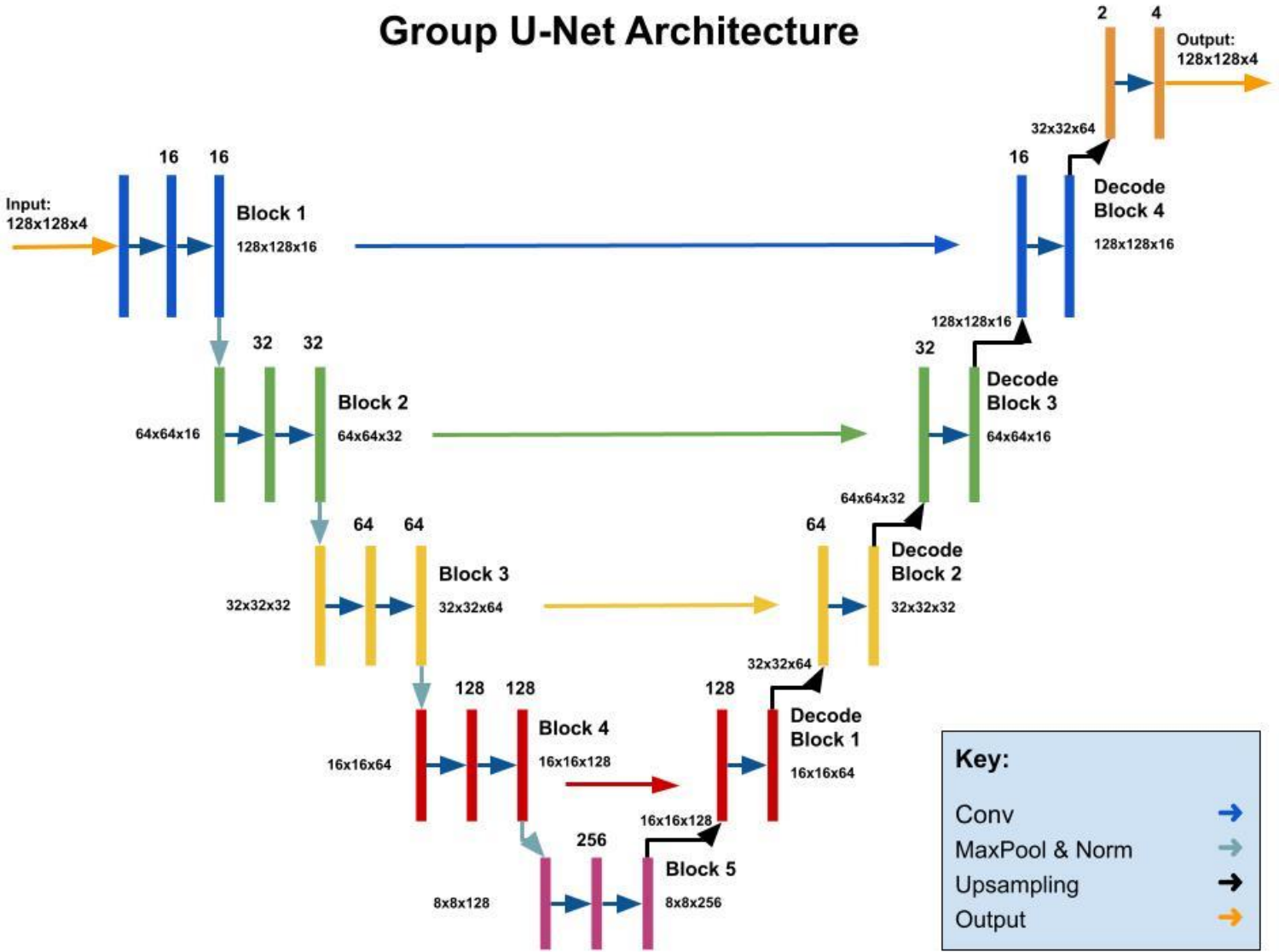


Figure 13 - Group U-Net Architecture. Shows path and shape of data as it moves through the model architecture. Each block contains a number of different layers. Coloured arrows as illustrated by the key, denote layer types (Convolutional, Max Pooling and Normalisation, Up-sampling), numbers above coloured blocks represent number of filters at each level. First section in each decoding block represents concatenation of up sampling from previous block and final output after pooling from equivalent block in decoding path.

In the encoding path, each block comprises two 2D convolutional layers followed by a normalisation and pooling layer. As described in 2.3. convolution layers perform feature extraction, applying a kernel across an input matrix (tensor). In development, the hyperparameters which must be selected are the filter size, kernel size, padding method, activation function and kernel initialiser. Table 3 gives a summary of the hyperparameters used by the model. The filter size represents the number of output filters in the convolution, this expands by an order of two as the block number increases. In the original U-Net presented by [33], the number of filters was $2^{(\text{block number} + 5)}$ for the encoding path and $2^{(11 - \text{block number})}$ for the decoding path. This has been reduced to $2^{(\text{block number} + 3)}$ for the encoding path and $2^{(5 - \text{block number})}$ for the decoding path due to VRAM limitations since increasing the number of filters increases the tensor size and thus the number of learnable parameters.

However, having an increased number of learnable parameters could increase model performance. The values for the activation function (ReLU), kernel size (3), padding method ('same') were all the same as used in previous research from the Context Survey [33, 42] and were found through experimentation to lead to the most favourable performance. In the case of kernel size, an increased size could lead to improved performance but would have been too computationally expensive during training.

	Parameter	Value
Training Parameters	Epochs	60
	Learning Rate	5*1e-4
	Optimiser	Adam
	Loss Function	Weighted Dice
Convolution Layers	Filter Size	Encoding Path: $2^{(\text{block number} + 3)}$, Decoding Path: $2^{(5 - \text{block number})}$
	Kernel Size	3
	Activation function	ReLU
	Padding	same
	Strides	(1,1)
	Kernel initialiser	glorot_uniform

Table 3 - Hyper-parameters used by Group Model

Using Batch Normalisation applies normalisation across a mini-batch of data and has the effect of stabilising the learning process, reducing the number of training epochs to lead to convergence [43]. In the project architecture, the default hyperparameters used by Keras are taken as they have been selected to work well with no parameterisation necessary. Similarly, default parameters are used for the Max Pooling layer in each encoding block (described in 2.3.). Max pooling takes the maximum value across each pooling kernel, the parameters which can be chosen in this later are the size of the pooling kernel, the stride size and padding method. By default, a 2 x 2 pooling kernel is used with an equal stride size, this effectively down-samples the image by an order of two. Increasing either of these parameters would reduce the number of learnable parameters and speed up training but would likely lead to reduced performance due to data degradation from the modified sample rate [44].

Each encoding block consists of a transposed Convolution layer, a Concatenation layer, and a Convolution layer. The transposed Convolution layer is an up-sampling operation, which transforms the input tensor from having the shape of the output of some convolution to something that has the shape of its input, this has the effect of reconstructing the original image [45]. The number of filters must follow the equation given above of $2^{(5 - \text{block number})}$. Next the output of the transposed convolution layer is concatenated with the output of the decoding path of the same block number. Then, a convolution operation is performed on this output, with the same parameters as

previous convolution layers. At the end of the encoding path, the output is an image which is of the same form [128,128,4], as the segmentation map (ground truth).

The model is compiled with the Adam optimiser which has good out of the box performance without need for much parameterisation since it sets an adaptive learning rate for each feature [46]. In other work in the literature Stochastic Gradient Descent is used, and there is the possibility that SGD can lead to greater performance than Adam but it requires more fine tuning of its parameters (learning rate, momentum, decay) in order to work optimally. The loss function used by the model during training is a weighted Dice loss function, described in detail in the next section. The model is then trained using the Keras Model.fit function and the hyper-parameters described in Table 4. Early stopping is also applied with a patience of 4, meaning if there has been no reduction in the loss function in 4 epochs then training will stop. This is done in an effort to reduce overfitting. The optimal learning rate of 5×10^{-4} was discovered through experimenting, trying different orders of magnitude from 1 to 10^{-7} as well as comparing to the existing literature, most of which use learning rates between 10^{-4} and 10^{-3} [32] [30] [38] [3] [39] [34].

Since the dataset, despite being relatively small, is too large to fit in memory, a custom Data Generator has been deployed, which during training will load batches of the dataset to feed to the model. The batch size for training and has been set to 1 since this has led through experimentation to optimal performance. Intuitively this makes sense since a batch size of 1 loads one patient record with the four image modalities and likely allows the network to gain more contextual information than if viewed in larger batches. This has been substantiated by the work of the researchers in the Context Survey, most of which used batch sizes of 1 or 2 [32] [38] [3] [39] [34]. After training, the model is saved as a .h5 file for use by the evaluation later. Also saved are plots of the loss function and dice function over the training epochs which also serve as part of the evaluation.

3.1.3. Dice Coefficient

The Dice similarity coefficient, also known as the Sørensen–Dice index is used to define the loss function as well as to evaluate the model’s performance during training and evaluation [47]. The Dice score measures the similarity between the model’s outputted segmentation and the ground truth. The equation for the Dice coefficient is $2 * |X \cap Y| / (|X| + |Y|)$ where X and Y are two sets and \cap is used to represent their intersection [47]. The base loss function, in line with common practice, is calculated as 1 minus the Dice coefficient. As an extension, the Dice loss is then passed through a weighting function which modifies the loss function to be multiplied the class weights defined in pre-processing. The weighting function creates a tensor with the respective weight for each element in the prediction tensor, which is summed to give the multiplier for the original Dice loss. This increases the loss for tumorous tissue classes [1,2,3], which helps to address the imbalance in target classes found in the dataset.

In the evaluation the Dice coefficient is used to create a set of Dice scores, measuring the accuracy of the prediction. In accordance with the guidelines set out for BraTS by [8], the model is evaluated by its ability to segment the Whole Tumour

(classes 1, 2 and 3), Tumour Core (classes 1 and 3) and Enhancing Tumour (class 3). This means calculating the Dice coefficients for these class combinations.

3.1.4. Evaluation Metrics

To evaluate the Model, it is first loaded from memory then used to predict a set of y outputs for the validation set using the Keras Model.predict function. This set of predictions is then flattened, passed to the Dice evaluation function described in the previous section as well as SkLearn's classification report, which is able to output the Precision, Recall and F1 scores for each class. In evaluation, examining the precision and recall scores allows the isolation of problem classes, where the model has particular trouble in extracting a feature set which can make accurate predictions. Finally in evaluation, a set of overlays are created where a number of the MRI images are overlaid with their corresponding prediction by the Model. The same is then done for the ground truth prediction. Comparing these then allows for greater exploration of the model behaviour.

3.2. Individual Work

The Individual architecture is available on GitHub via https://github.com/DSmith537/Brain_Tumour_Segmentation_UNet_Indi. Making the project open source in this way allows other researchers to utilise the work in this project.

3.2.1. Base Architecture

The objective of the individual work was to explore the use and performance of different loss functions using a baseline architecture. The Group Model serves as this baseline, with modifications to the architecture described below which allow the use of different loss functions. There have been no changes made to the hyperparameters outlined previously except for the loss function. This allows the experimentation of loss functions to be more scientifically rigorous, since no other changes to the model were made except the variable of the loss function used.

The group architecture was modified by adding an additional class LossFunctions, which holds the various loss functions used in the experimentation as well as a picker class used by the training function. The picker function returns the chosen loss function which the user selects by passing a variable from the Model class. Another modification was to save the model individually with its loss function configuration, allowing a number of models using each loss function to be trained, saved, and evaluated separately.

3.2.2. Loss Function Selection

As per the literature, commonly used loss functions for medical image segmentation problems are Mean Squared Error, Dice Loss, Categorical Cross Entropy, Combo Loss and Focal Loss [32] [30] [38] [3] [39] [34]. All of these will be experimented with and in each case except for Focal loss, will also be tested with the weighting modification as used by the Group Model, described in 3.1.3. Focal loss has not be run with weighting since it is already well adapted to handle class imbalance.

i. Mean Squared Error

Mean Squared Error (MSE) is computed as the mean of squares of errors between the ground truth labels and predictions.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad [48]$$

n = number of data points

Y_i = Observed Values

\hat{Y}_i = Predicted Values

MSE works well in many use cases and is often used as a starting point when selecting a loss function. This motivates the selection of MSE for experimentation, since it will serve as a good baseline when comparing the performance of the loss functions overall.

ii. Dice Loss

The Dice coefficient and its use in a loss function has already been defined in 3.1.3.

$$\text{Dice Loss} = 1 - \left(2 * \frac{|X \cap Y|}{(|X| + |Y|)} \right) \quad [47]$$

Dice Loss is widely used elsewhere in image processing tasks to compare the similarity of two images. It has been shown to perform well at segmentation tasks and has been used extensively by the literature [26, 39, 35] due to its adroitness at dealing with class imbalance. The group model architecture modified the Dice loss to be weighted as described in 3.1.3., which further enabled the loss function to deal with class imbalance.

iii. Categorical Cross Entropy

Categorical Cross Entropy (CCE) loss is a modification of Cross Entropy which can handle categorical data as is used in this project. Cross Entropy is a measure from the field of information theory, generally calculating the difference between two probability distributions. Since the prediction and outputs are both probability distributions, representing the probability of each class for each pixel in the image, these can be compared using Cross Entropy.

$$CCE = -\sum_{i=1}^n y_i \log \hat{y}_i \quad [49]$$

$$\begin{aligned} y_i &= \text{Observed Value} \\ \hat{y}_i &= \text{Predicted Value} \\ n &= \text{Output Size} \end{aligned}$$

The result will be a positive number measured in bits and will be equal to the entropy of the distribution if the two probability distributions are identical. Cross Entropy was used by [32], in the original U-Net architecture in [33] and has shown to have smoother gradients during training, motivating its inclusion in the experimentation.

iv. Combo Loss

An option outlined by [40] and used by [3] was to combine the Dice and Cross Entropy losses to a single loss function. This paper presents a modified Combo loss, a weighted summation of the Dice and Cross Entropy Losses.

$$ComboLoss = (\alpha * Dice) + (\beta * CCE)$$

$$\begin{aligned} Dice &= \text{Dice Loss} \\ CCE &= \text{Categorical Cross Entropy Loss} \\ \alpha &= 0.3 \\ \beta &= 0.7 \end{aligned}$$

In the formula α is a number between 0 and 1, β is then defined as $1 - \alpha$. The reason that Combo Loss has been modified compared to [40] and [3] is that the use of the α and β allows for the weighting of the Dice and Cross Entropy losses. Their values have been heuristically derived through experimentation. The benefit of Combo loss is that it utilises the curve smoothing of CCE and the class imbalance adroitness found with Dice.

v. Focal Loss

Focal loss can be seen as a variation to Cross Entropy, which down-weights the contribution of easier examples and enables the model to focus on more difficult features to learn.

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad [50]$$

The formula for Focal loss, modifies Cross Entropy with a modulating factor $(-\alpha_t(1 - p_t)^\gamma)$. In this project $\alpha = 0.25$ and $\gamma = 2$ in accordance with [50]. When an example is misclassified and p_t is small, the modulating factor is near 1 and the loss is unaffected. The focusing parameter γ smoothly adjusts the rate at which easy examples are down-weighted. Intuitively, the modulating factor reduces the loss contribution from easy examples and extends the range in which an example receives the low loss. This means it works well on highly imbalanced datasets which this project deals with, making it a good final loss function to examine for this experimentation.

4. Results

This section outlines the results for the group and individual models. This includes plots of the Loss function and Model Accuracy during training. Also included from the evaluation on the test set, the Dice Coefficient scores for the classes outlined in 3.1.3. and Precision, Recall and F1 scores for target classes [1 (Necrotic Core) , 2 (Edema) ,3 (Enhancing Tumour)]. The Precision scores measure the specificity of the model and its ability to make accurate predictions. Recall measures the sensitivity of the model and its ability to find all occurrences of a class in its predictions. F1, the harmonic mean of Precision and Recall, will be closer to 1 when Precision and Recall are. As noted in the Methodology section, the results will focus on the Models performance at multi-class segmentation. Finally, the segmentation overlays outlined in 3.1.4. for the prediction and ground truth are included. For all models the same slice from the test set is taken, meaning each model prediction is for the same ground truth.

4.2. Group Model

4.2.1. Loss and Accuracy During Training

The Accuracy plot (Figure 14) for the group model shows smooth growth over epochs, converging on a maximum around epoch 40. The accuracies for the training and validation sets are closely aligned, indicating that the model will generalise well to the test set. The loss function (Figure 15) similarly shows a smooth learning path over training, converging at a minimum around epoch 40 with good alignment between data sets.

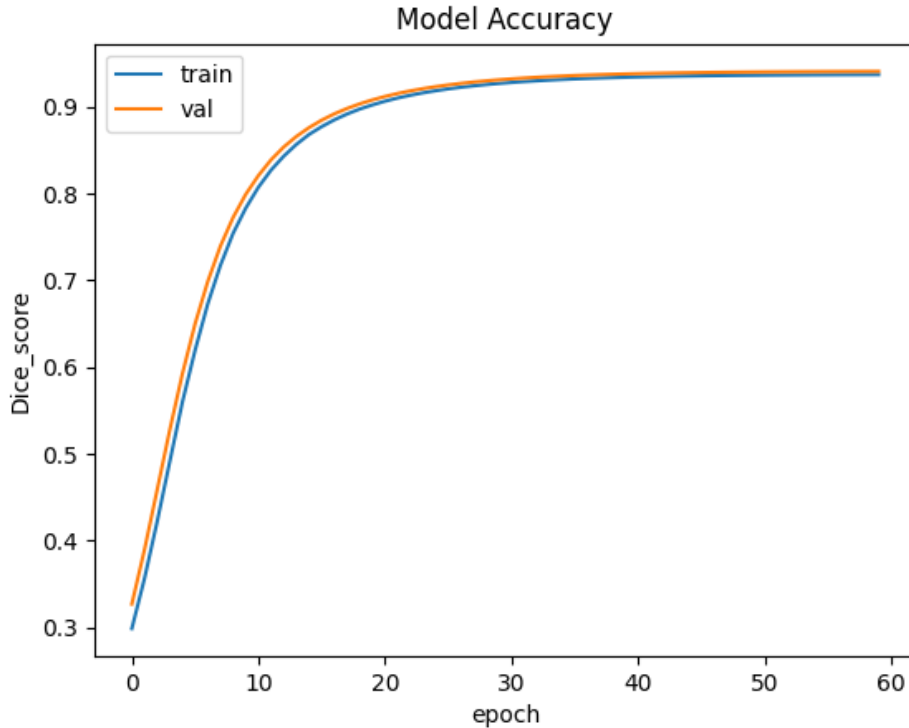


Figure 14 - Group Model Accuracy over Training Epochs for Training and Validation sets.

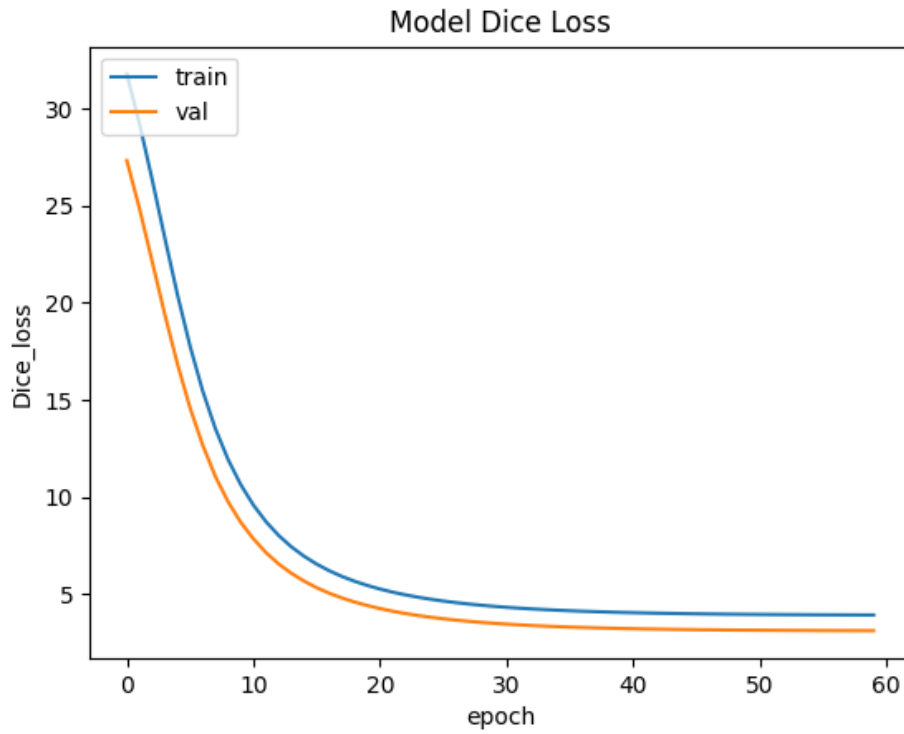


Figure 15 - Group Model Loss Function over Training epochs for Training and Validation sets.

4.2.2. Dice, Precision, Recall and F1 Scores

Whole Tumour: 86.2%

Tumour Core: 79.2%

Enhancing Tumour: 74.5%

	Precision	Recall	F1
Necrotic Core (class 1)	0.83	0.34	0.48
Edema (class 2)	0.64	0.8	0.71
Enhancing Tumour (class 3)	0.64	0.9	0.74

Table 4 - Group Model Precision, Recall and F1 scores for target classes when evaluated on test set.

When evaluated on the test set (Table 4), the Group model achieved good Dice scores, with all three scores comparable to those found in the Context Survey. Examining the Precision, Recall and F1 scores, the Necrotic core has a significantly lower F1 score than other classes, owing to its very low recall score. It has a relatively high precision so it would appear that when it does make predictions of class 1, they are accurate, but the Model is not good at finding class 1's instances.

4.2.3. Example Overlays

Comparing the prediction and ground truth for an example segmentation (Figure 16), the model appears to be predicting both the edema and enhancing tumour with good accuracy, though there are erratic predictions for the necrotic core which shows the greatest disparity to the ground truth segmentation, this aligns with what was shown by the precision, recall and f1 scores.

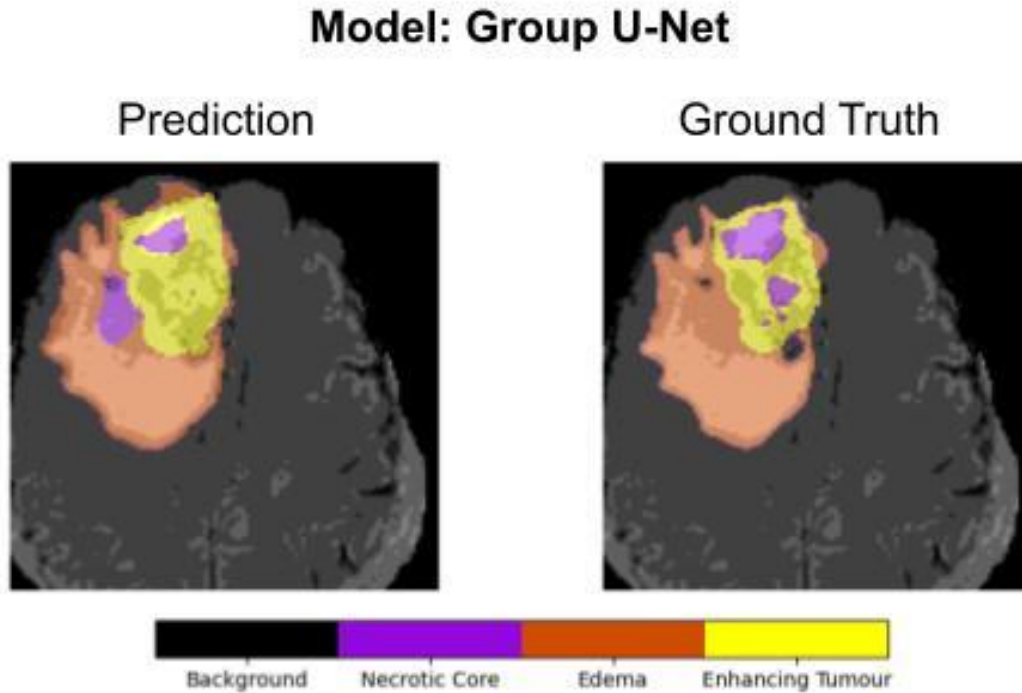


Figure 16 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation.

4.2. Individual Work

4.2.1. Summary of Results

The plot below outlines the Dice scores achieved by the Model when trained with each of the 9 Loss functions. Also included in this section is a plot of the F1 scores for each of the Loss Functions. F1 has been used since it is the harmonic mean of Precision and Recall. A Table of Dice scores and a table of Precision, Recall and F1 Scores is available in A.I.

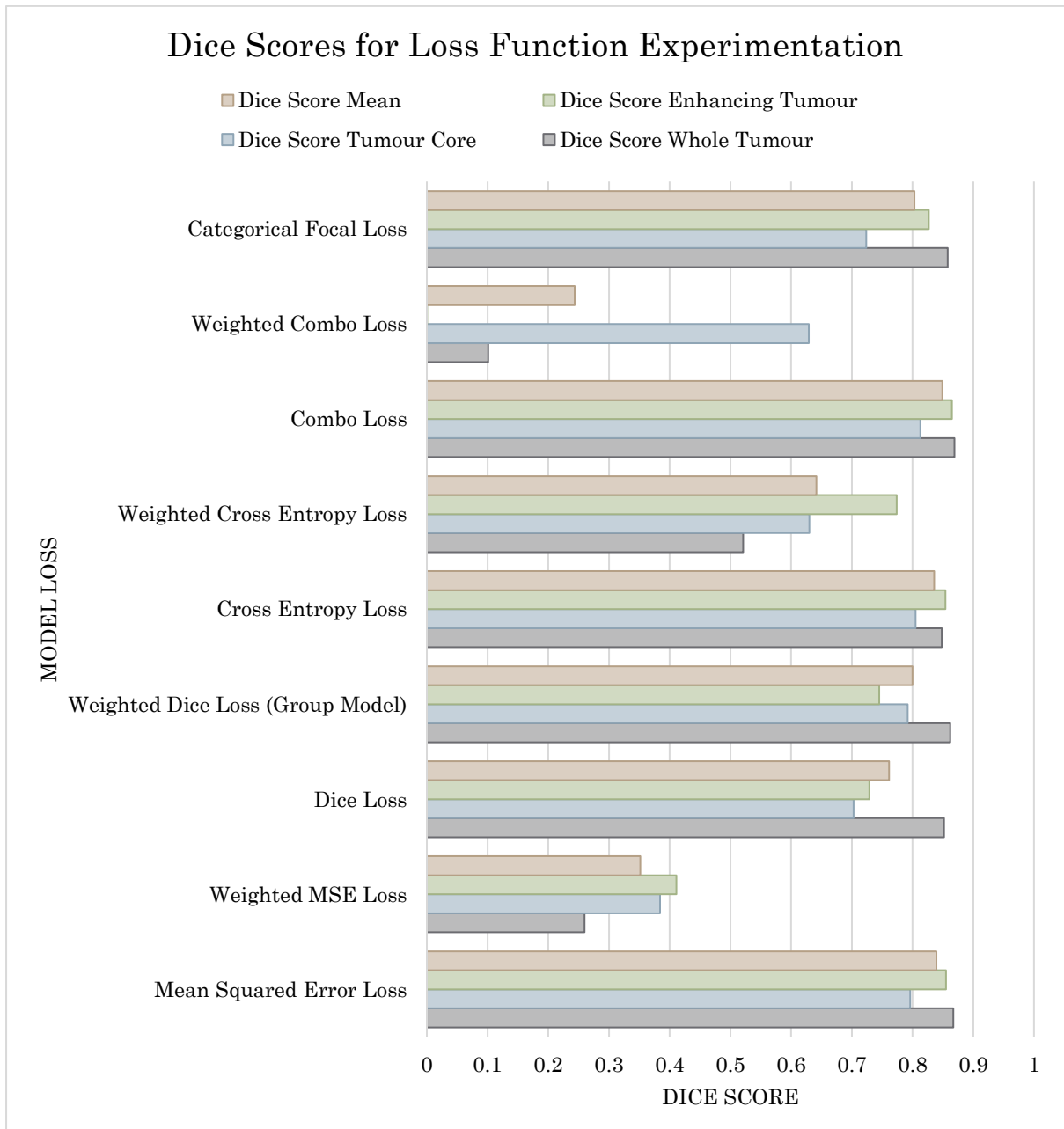


Figure 17 – Dice Scores when evaluating test set using Model with different Loss Functions. Includes Tumour Core, Enhancing Tumour, Whole Tumour, and the Mean of these three target classes.

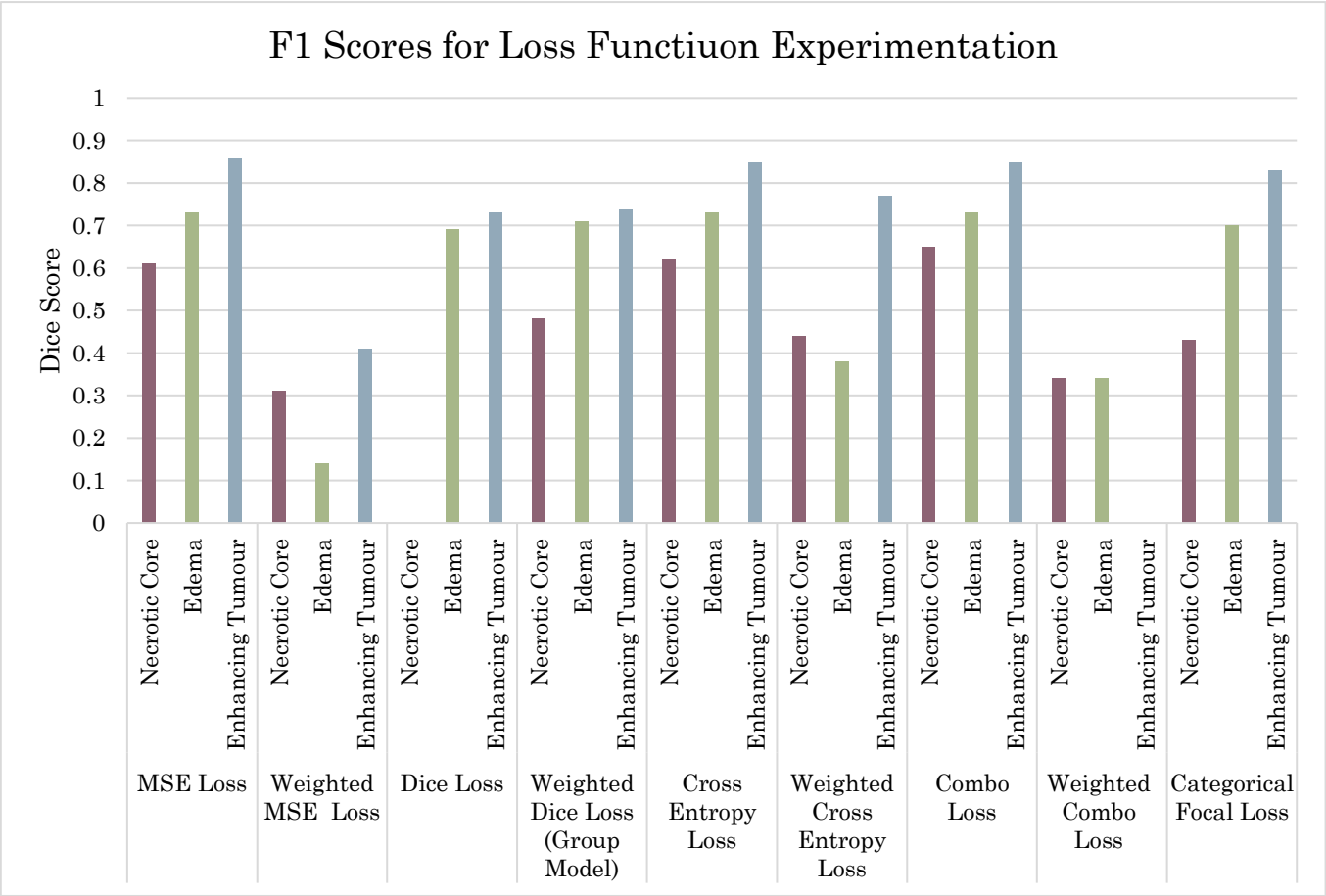


Figure 18 – F1 Scores when evaluating test set using Model with different Loss Functions. Includes Necrotic Core, Edema and Enhancing Tumour.

4.2.2. Mean Squared Error

I. Unweighted

The accuracy plot for MSE (Figure 19) shows a reasonably smooth training curve and there is good alignment between the training and validation sets accuracies. The loss function (Figure 20) also has the curve one would expect but it is less smooth than was observed with the Group model. Around epoch 20 the loss function for the training and validation sets begin to diverge somewhat and at epoch 40 the loss function has levelled off, causing early stopping. However, the gradient for the accuracy when stopping at epoch 30 is greater than 0, indicating there is more learning that can be done but the loss function is an inhibiting factor. The Dice scores for Mean Squared Error are generally very strong, it has the best score for the Whole Tumour and very good performance for the Tumour Core, both slightly improved compared to the Group Model. The Enhancing Tumour score is significantly improved compared to the group model as are the Precision, Recall and F1 scores, in particular the recall for the Necrotic Core, which has increased to 0.5. Examining the example segmentations (Figure 21), there is much better alignment in the Necrotic Cores and the model as generally predicted the tumour shape well.

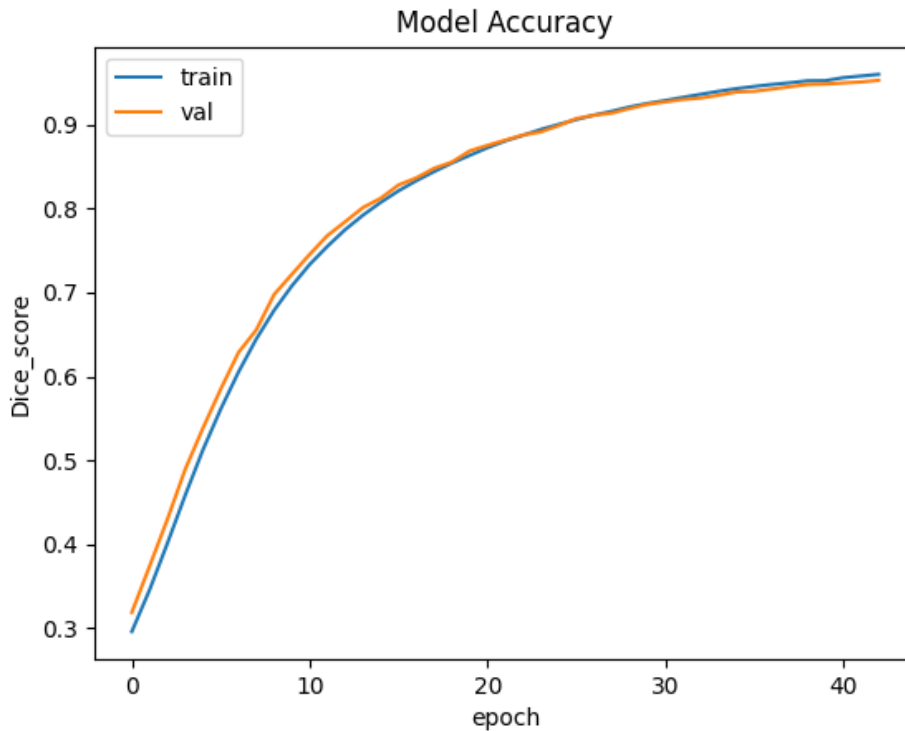


Figure 19 - MSE Loss Model Accuracy over Training Epochs for Training and Validation sets.

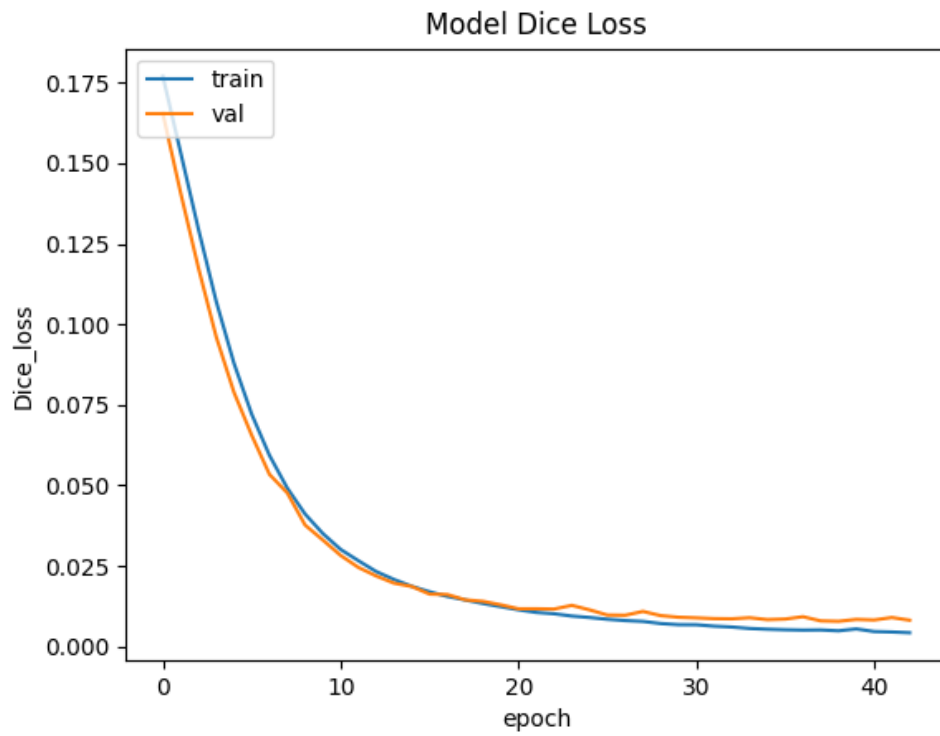


Figure 20 - MSE Loss Function over Training epochs for Training and Validation sets.

Model: Individual U-Net - Mean Squared Error Loss

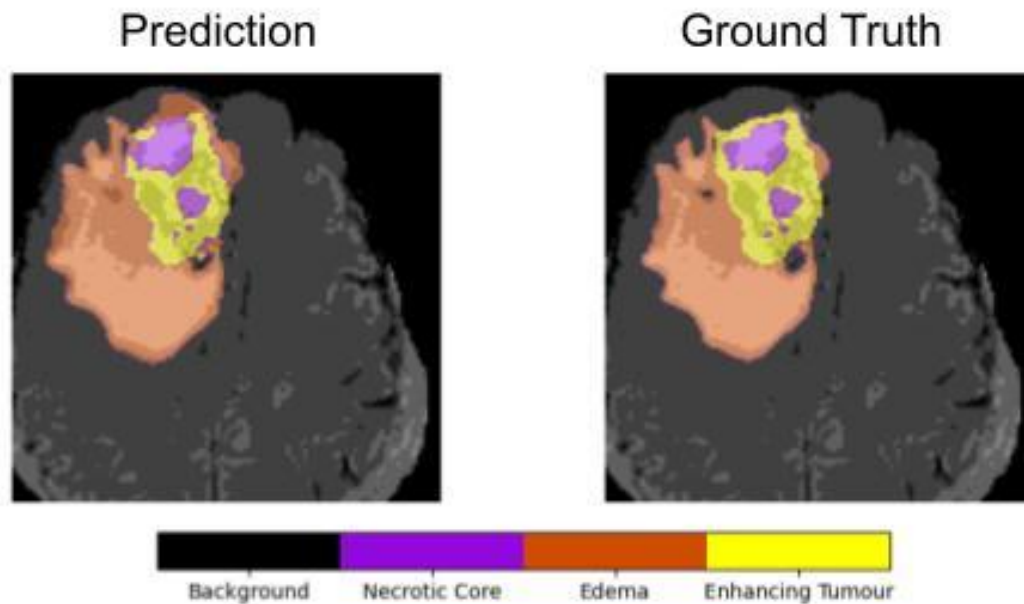


Figure 21- Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

II. Weighted

The accuracy and loss curves (Figures 22 and 23) for the Model trained with the weighted version of MSE loss are much more erratic. The training stops early after just 10 epochs with no more reduction in the loss function. The training increases reasonably smoothly for the training set but very erratically for the validation set and is not trained long enough to converge. Most metrics (Dice, Precision, F1), degraded severely though the recall scores remain relatively high. The overlay images (Figure 24) show the system making abundant predictions of tumorous tissue, which include the correct segmentation, explaining the relatively high recall scores.

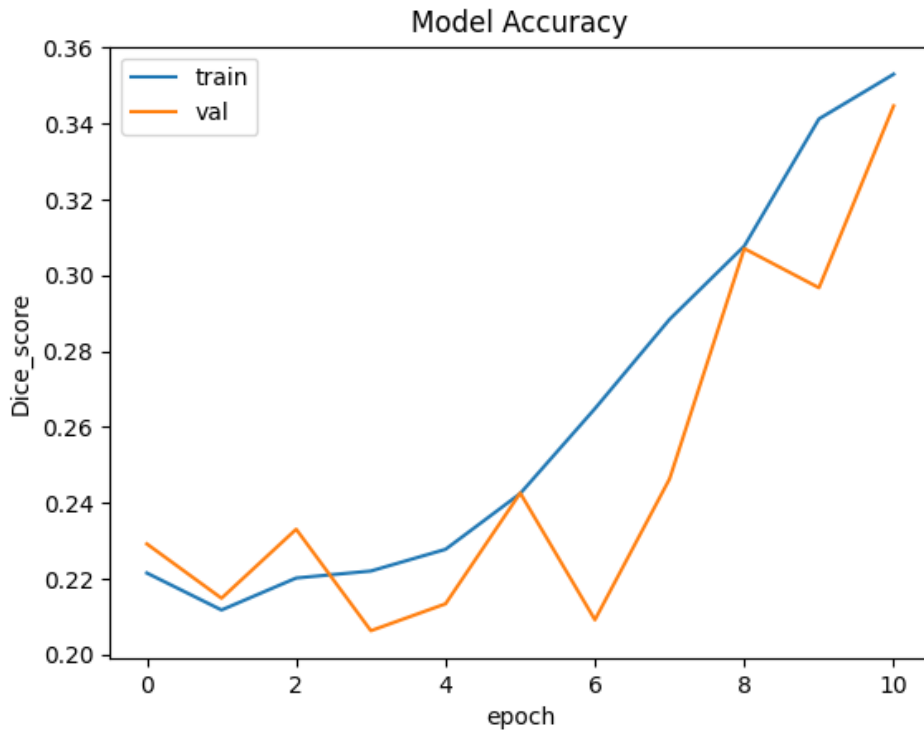


Figure 22- Weighted MSE Loss Model Accuracy over Training Epochs for Training and Validation sets.

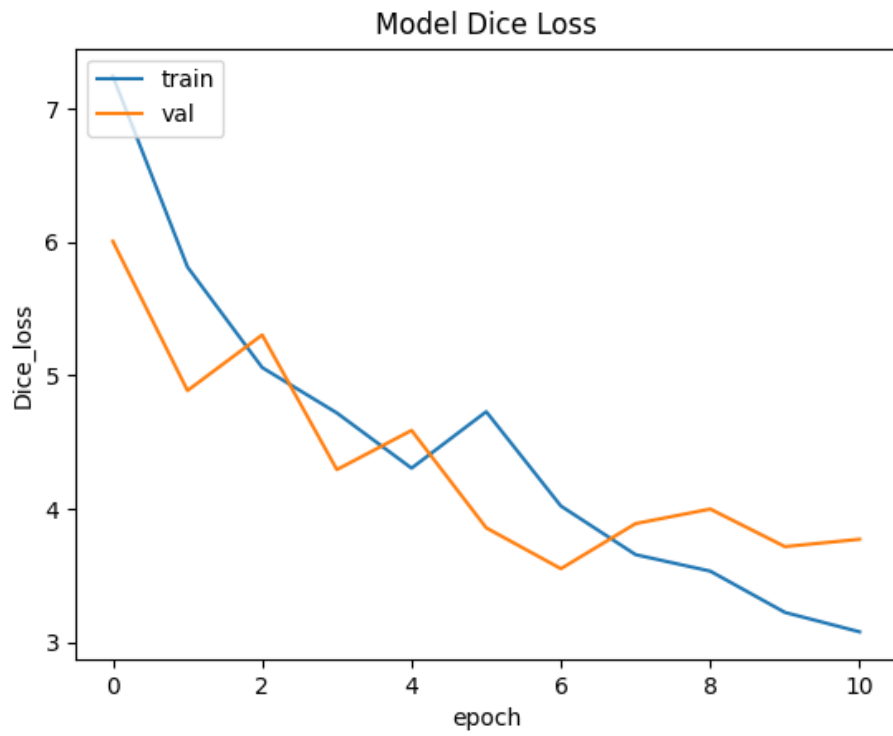


Figure 23 - Weighted MSE Loss Function over Training epochs for Training and Validation sets.

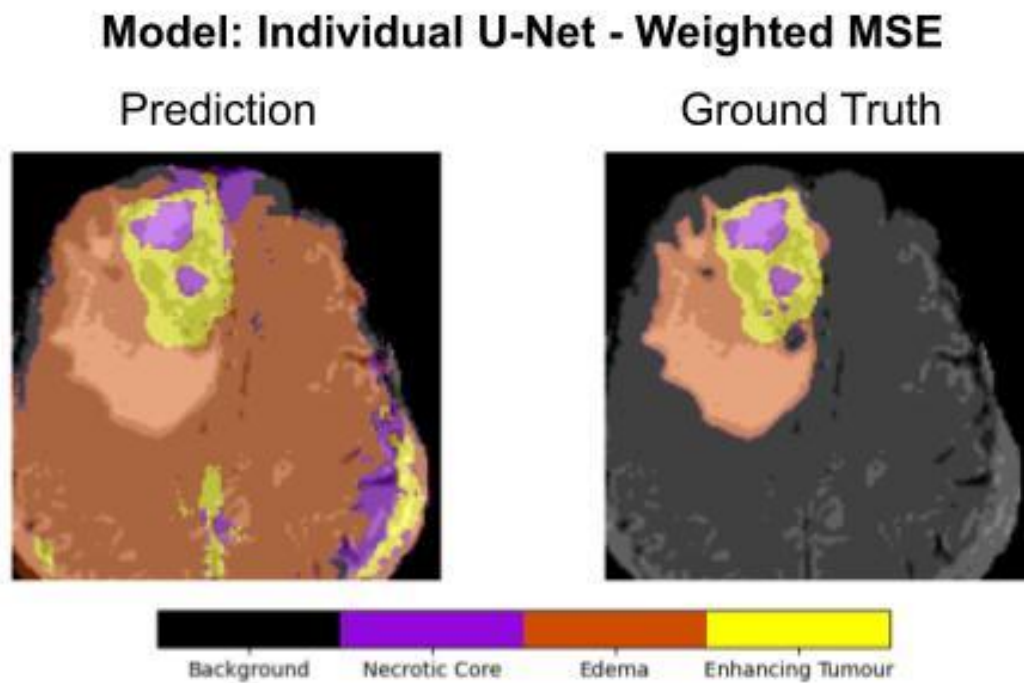


Figure 24 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

4.2.3. Dice Loss

Using an unweighted Dice function to train the model shows a very similar accuracy and loss curve (Figures 25 and 26) to the weighted benchmark version. For both plots, the training and accuracy are well aligned, indicating good generalisability. The Dice scores are worse than the weighted loss benchmark in all cases. Examining the Precision, Recall and F1 scores, there are scores of 0 for the Necrotic Core and reasonable scores for the Edema and Enhancing Tumour. The segmentation overlays illustrate this (Figure 27), the model does not make any predictions of Necrotic Core in the segmentation.

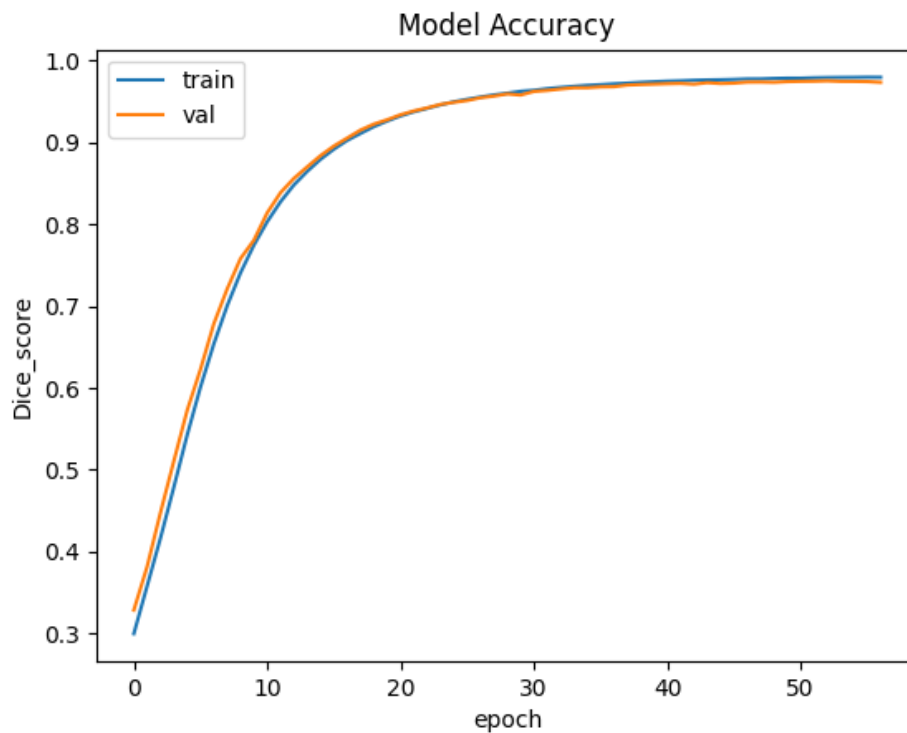


Figure 25 - Dice Loss Model Accuracy over Training Epochs for Training and Validation sets.

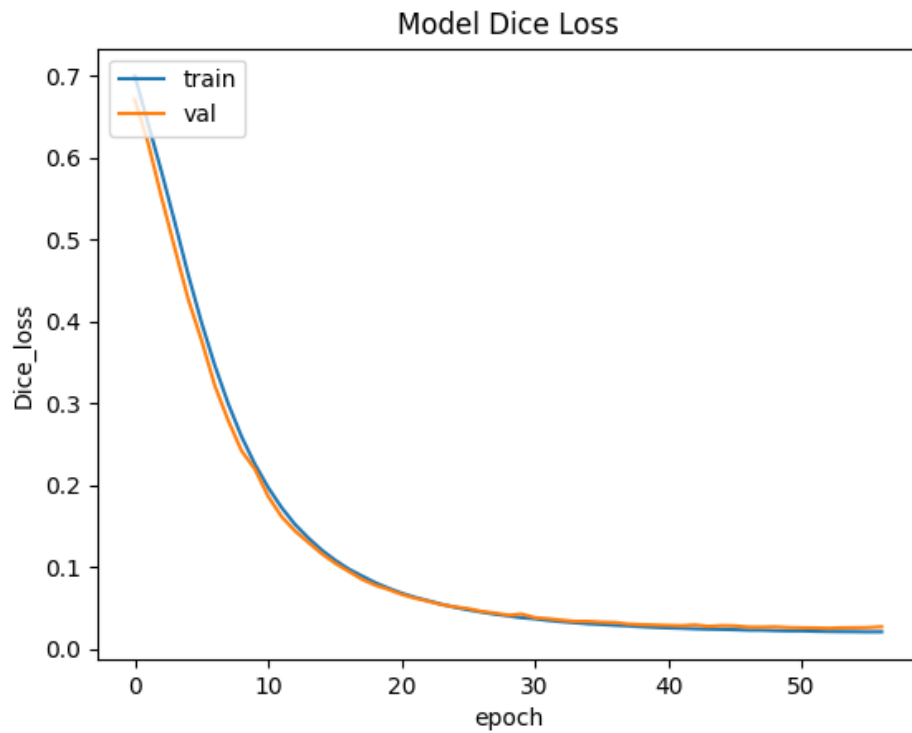


Figure 26 - Dice Loss Function over Training epochs for Training and Validation sets.

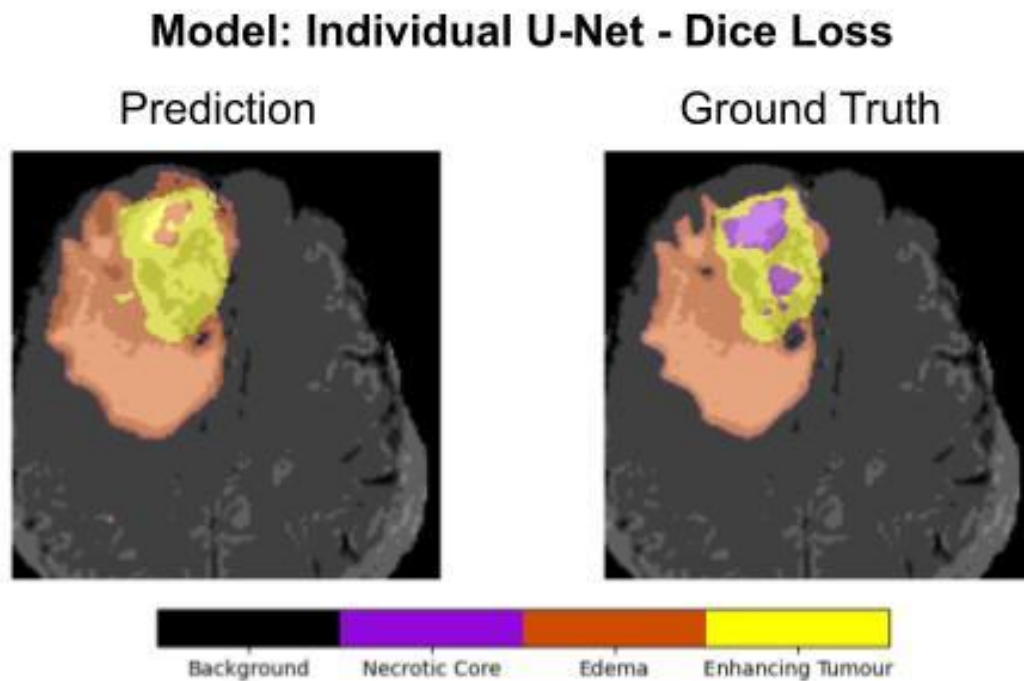


Figure 27 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

4.2.4. Cross Entropy Loss

I. *Unweighted*

When Cross Entropy loss was used to train the model, it showed a smooth training curve (Figure 28) with accuracy in training and validation sets well aligned and convergence before early stopping at epoch 40. The loss function (Figure 29) for the validation set diverges from the training set around epoch 20. The Dice scores for Cross Entropy were lower than the Group Model for the Whole Tumour but slightly higher for the Tumour core and much higher for the Enhancing Tumour. The F1 scores for all classes are improved compared to the Group Model. Precision for Edema and Enhancing Tumour regions is slightly degraded, but Recall has improved, with the opposite being true for the Necrotic Core. Examining the example segmentations (Figure 30), both the prediction and ground truth are very similar. The necrotic core is slightly larger than the ground truth and the model seems unable to identify smaller Necrotic Core regions.

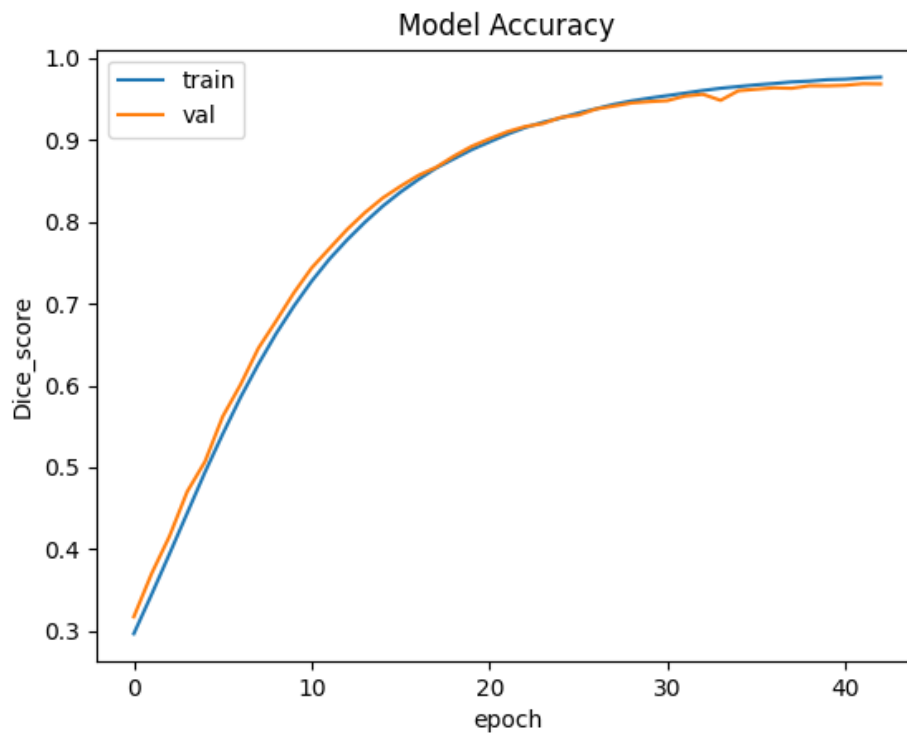


Figure 28 - Cross Entropy Loss Model Accuracy over Training Epochs for Training and Validation sets.

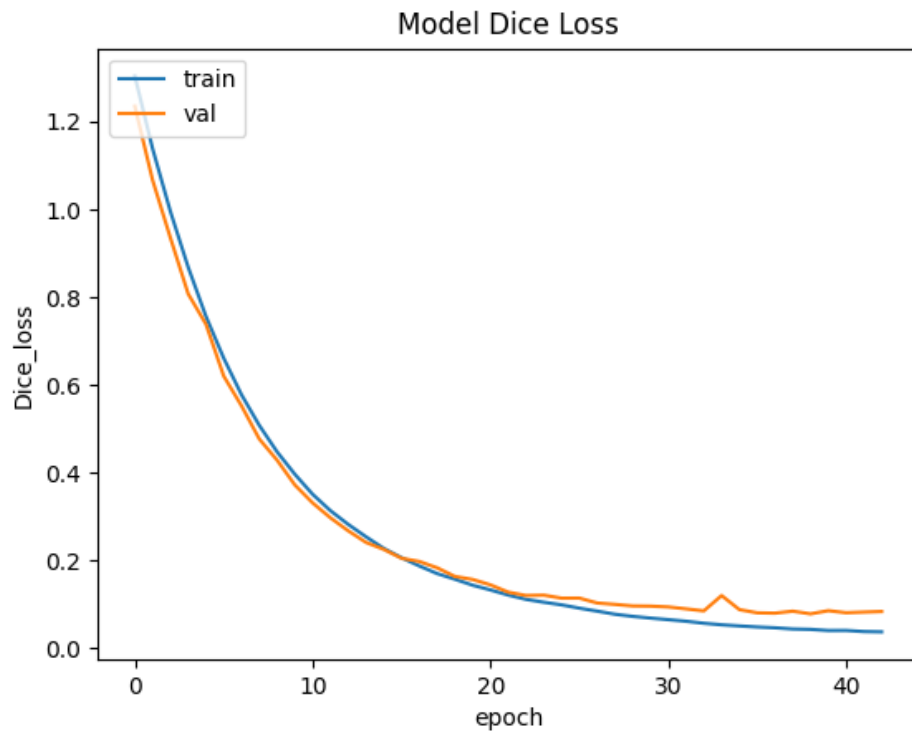


Figure 29 - Cross Entropy Loss Function over Training epochs for Training and Validation sets.

Model: Individual U-Net - Cross Entropy Loss

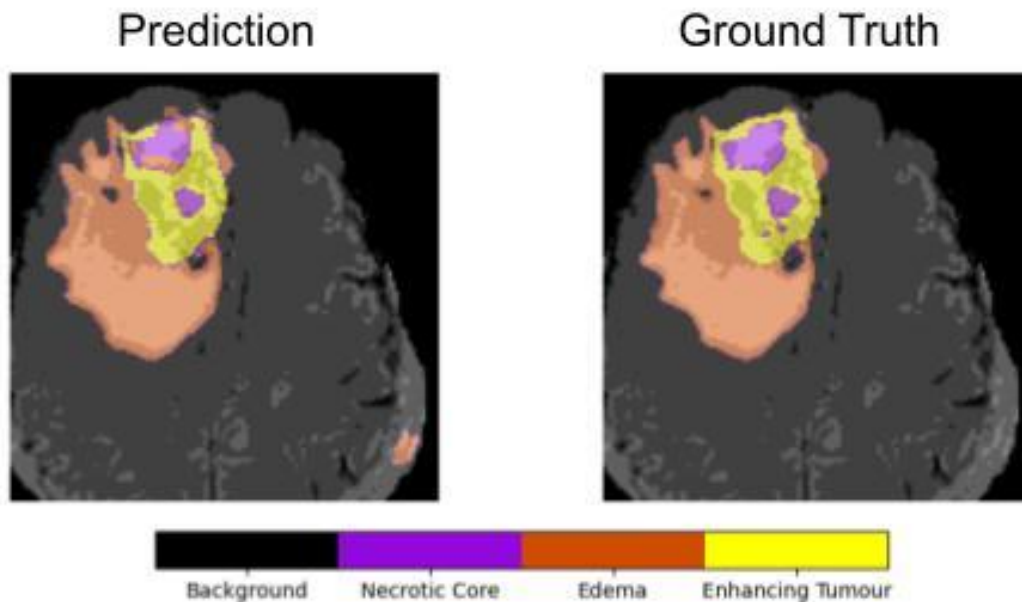


Figure 30 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

II. *Weighted*

The weighted version of Cross Entropy shows very similar performance to weighted MSE. The training plots (Figures 31 and 32) are erratic, indicate early stopping and show variance in their smoothness between the training and validation sets. The accuracy scores are also degraded and there is similar behaviour displayed in the segmentation overlays (Figure 33), with the model making far too many predictions of tumorous regions.

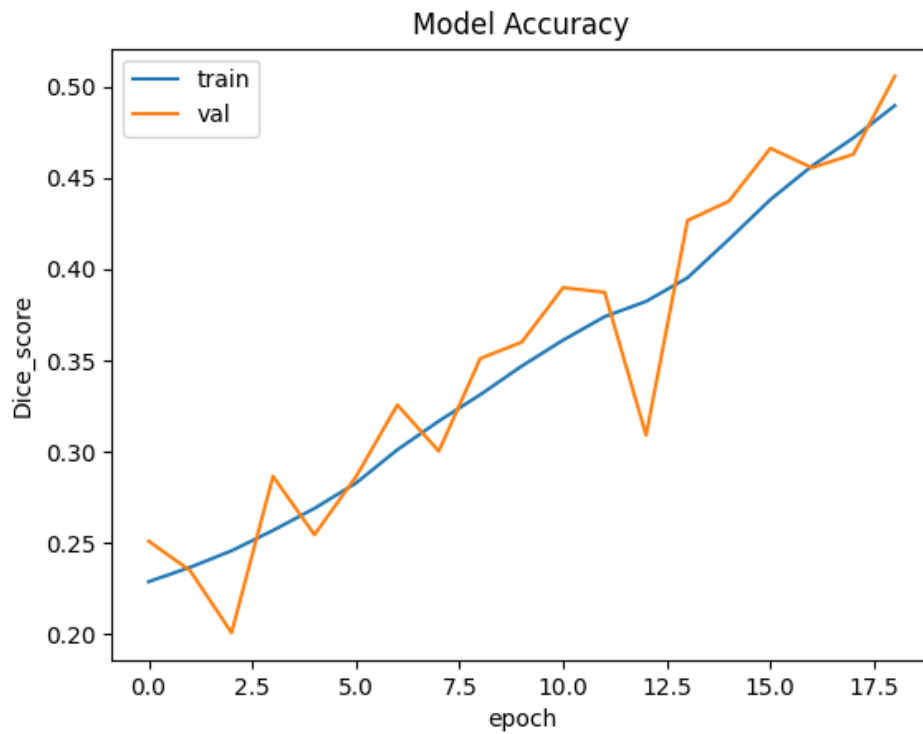


Figure 31 - Weighted Cross Entropy Loss Model Accuracy over Training Epochs for Training and Validation sets.

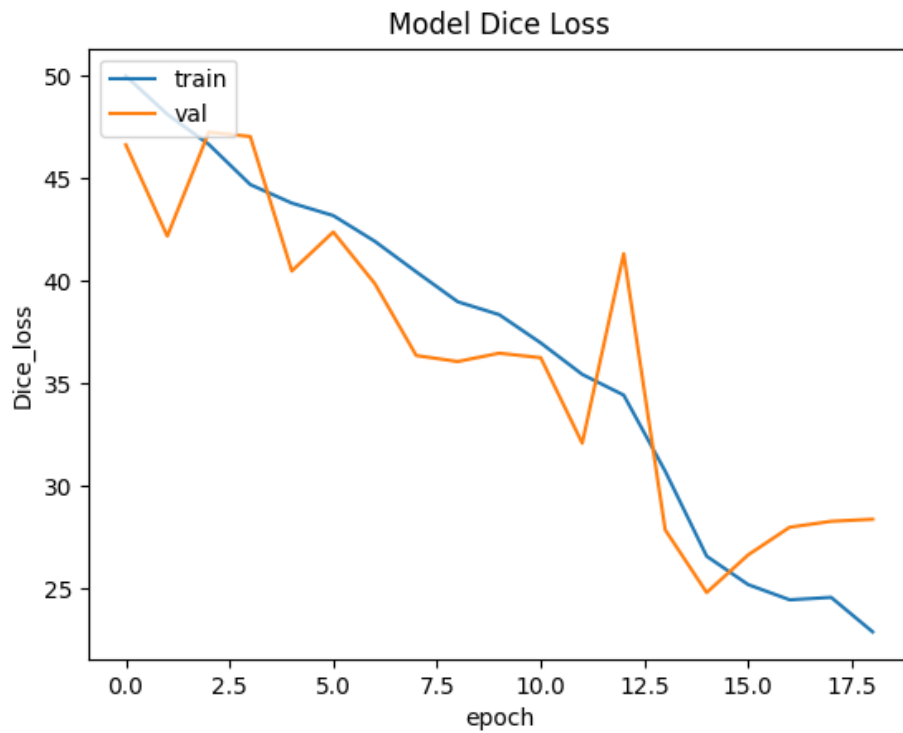


Figure 32 - Weighted Cross Entropy Loss Function over Training epochs for Training and Validation sets.

Model: Individual U-Net - Weighted Cross Entropy

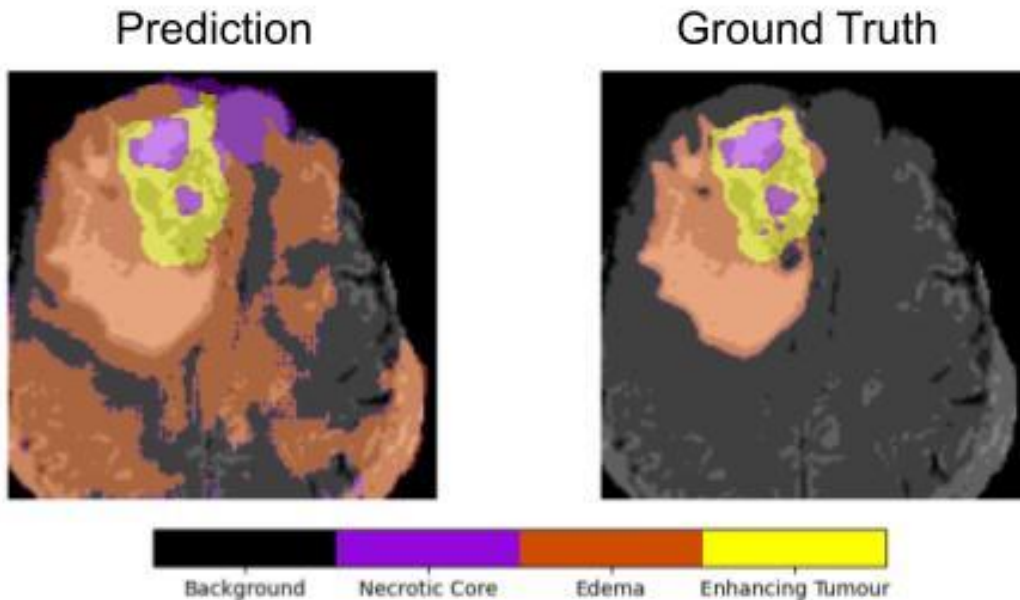


Figure 33 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

4.2.5. Combo Loss

I. *Unweighted*

Combo loss, the combined loss function of Dice and Cross Entropy, shows similar behaviour to said functions used individually (Figure 34 and 35). The curves for training and validation are smooth and well aligned, converging before early stopping at epoch 40. The loss functions, similar to with Cross Entropy, begin to diverge around epoch 20. The Dice scores are improved in all cases compared to the Group Model, particularly the Enhancing Tumour. Precision for the Necrotic Core is slightly lower, but recall is much higher resulting in a higher overall F1. Precision, Recall and F1 are similar though slightly improved for the Edema. Much improved are the Precision and F1 for the Enhancing Tumour. The segmentation overlays (Figure 36) show great similarity in the cores of the tumour, this is the first model which makes predictions for the small Necrotic Core regions and the Enhancing Tumours are very similar.

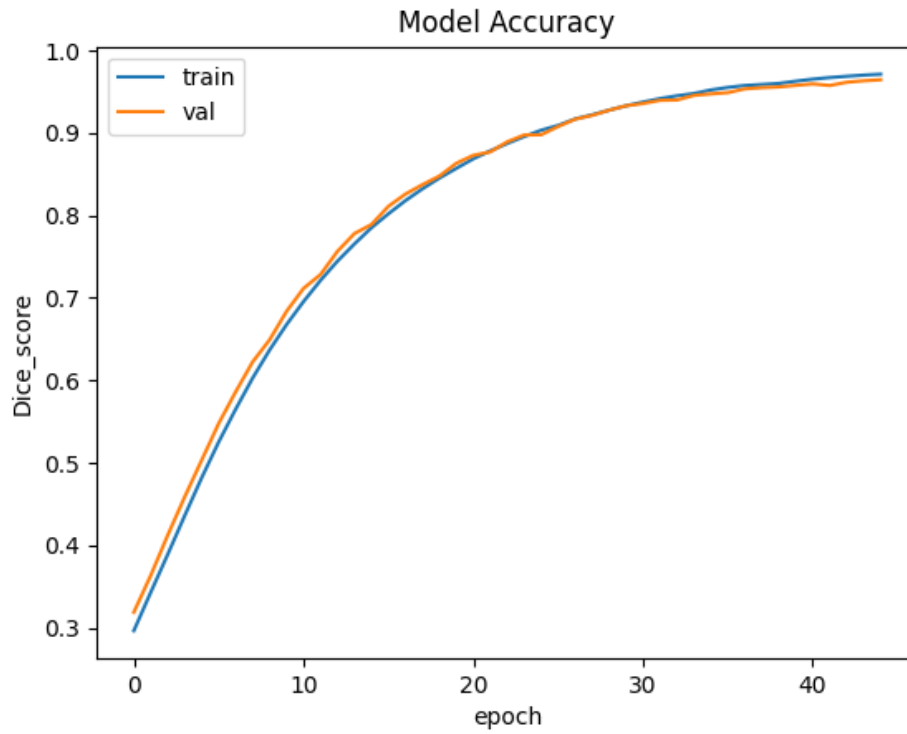


Figure 34 - Combo Loss Model Accuracy over Training Epochs for Training and Validation sets.

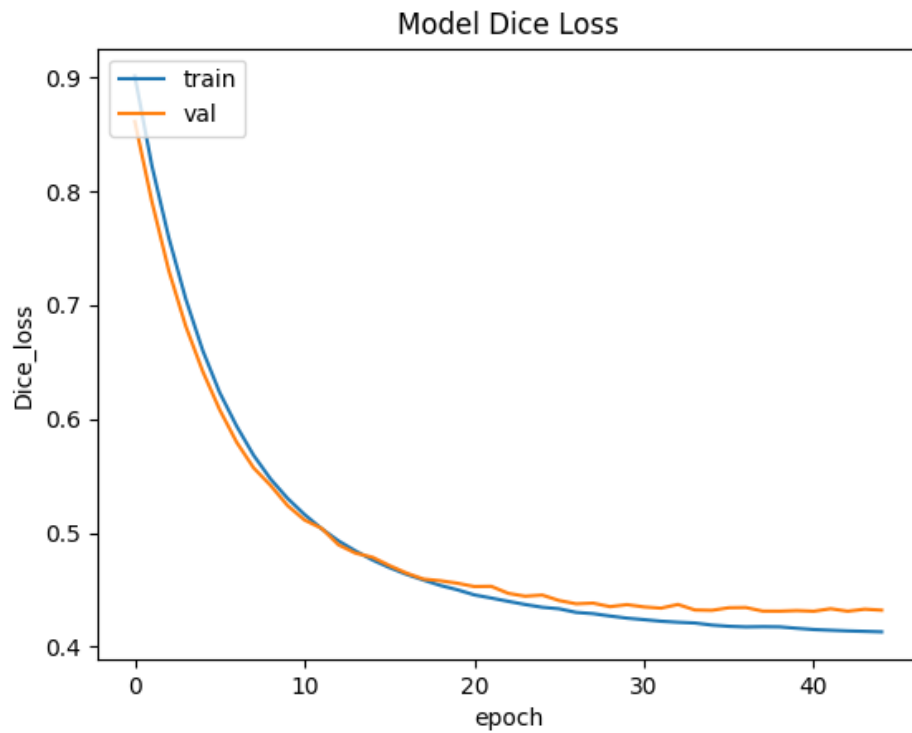


Figure 35 - Combo Loss Function over Training epochs for Training and Validation sets.

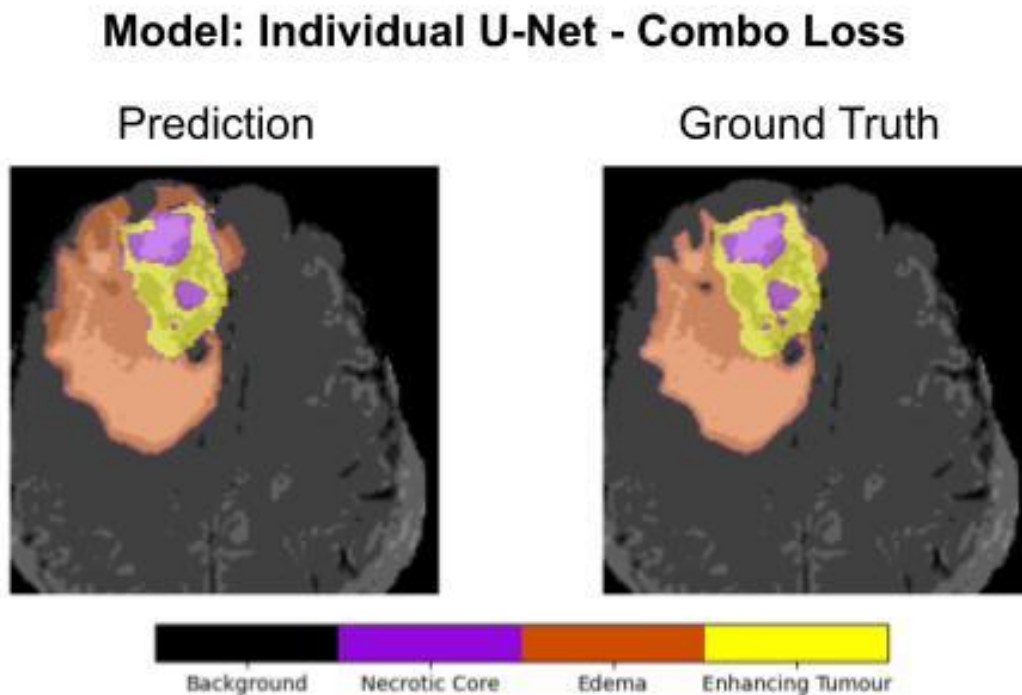


Figure 36 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

II. *Weighted*

Training the Model using the weighted version of Combo Loss showed behaviour to weighted MSE and Cross Entropy (Figures 37 and 38), the learning is much more unstable, especially on the validation set. Dice, Precision, Recall and F1 scores are degraded in all cases compared to the Group Model. The segmentation overlays (Figure 39) display the model's inability to properly segment any tumour type effectively.

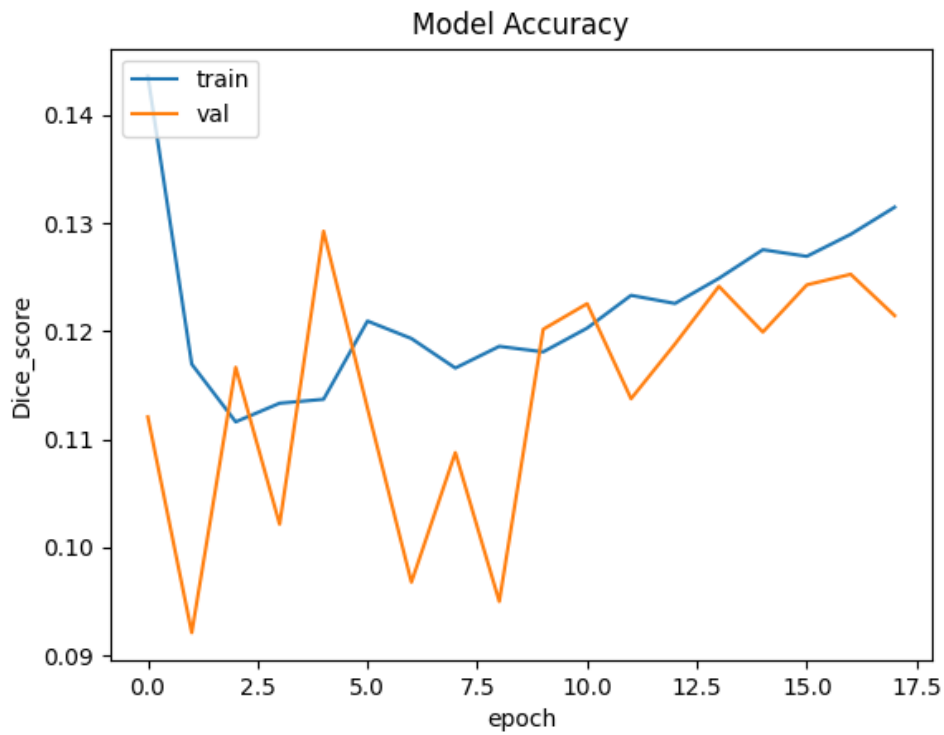


Figure 37 - Weighted Combo Loss Model Accuracy over Training Epochs for Training and Validation sets.

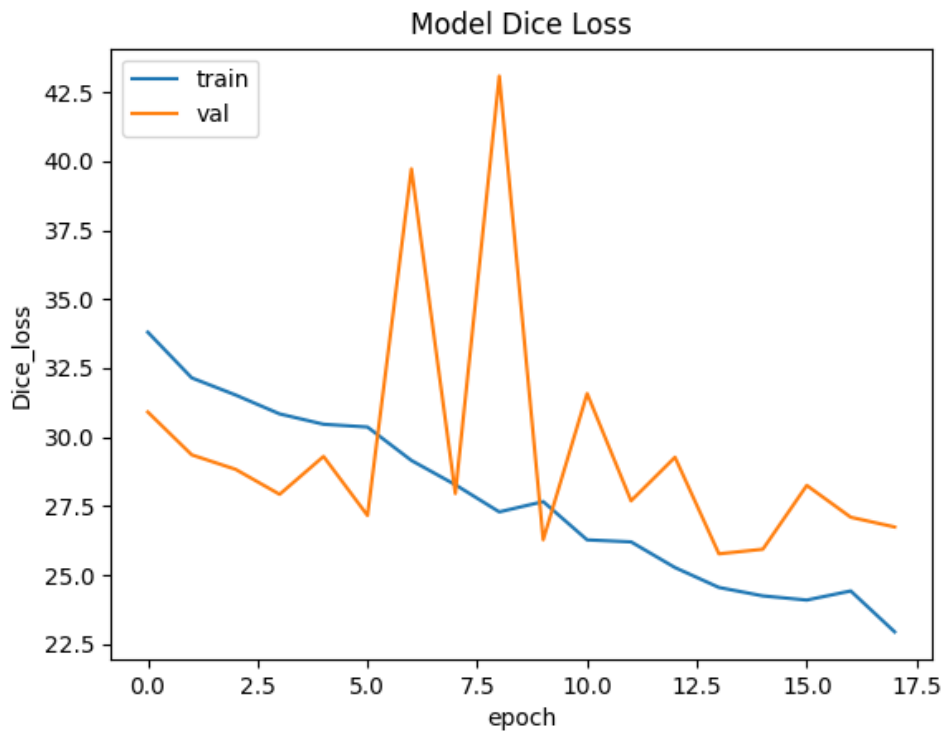


Figure 38 - Weighted Combo Loss Function over Training epochs for Training and Validation sets.

Model: Individual U-Net - Weighted Combo Loss

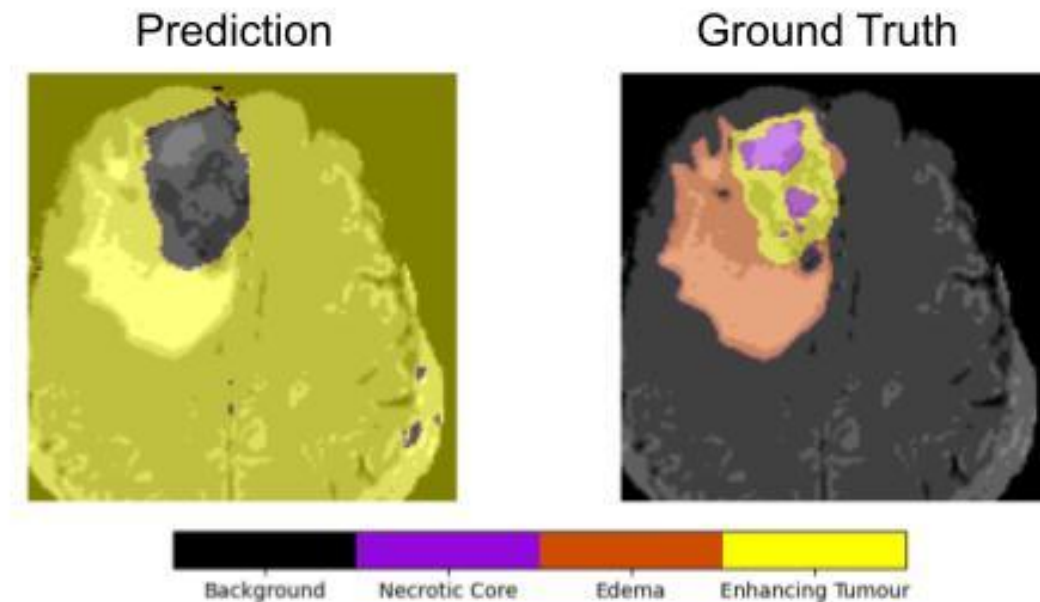


Figure 39 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

4.2.6. Focal Loss

The training curve (Figure 40) when using Focal Loss is smooth and well aligned for the training and validation sets. The Model stopped training particularly early compared to other losses, stopping early around epoch 25. The loss function (Figure 41) seems to have reached a minimum, with a divergence between the training and validation sets beginning, as with other loss functions. However, the accuracy curve has not converged by the point of early stopping. The Dice scores are lower in all cases except the Enhancing Tumour, compared to the Group Model. Using Focal loss, the model once again has trouble identifying the Necrotic Core, indicated by the lower Precision, Recall and F1 scores and the segmentation overlay (Figure 42). The overlays in general show a greater disparity between the prediction and ground truth than other models.

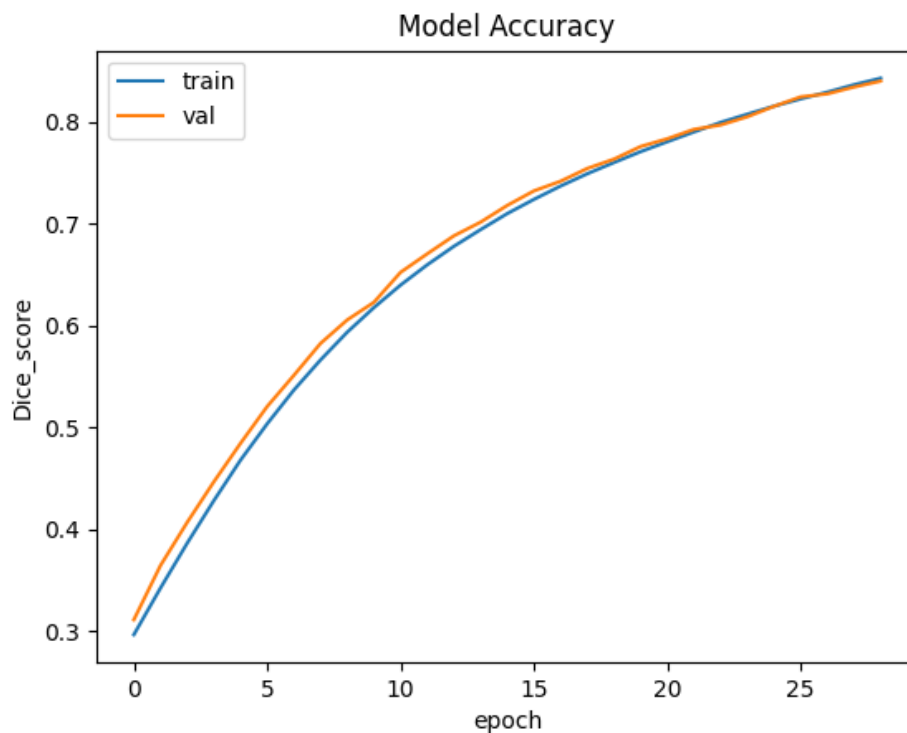


Figure 40 - Focal Loss Model Accuracy over Training Epochs for Training and Validation sets.

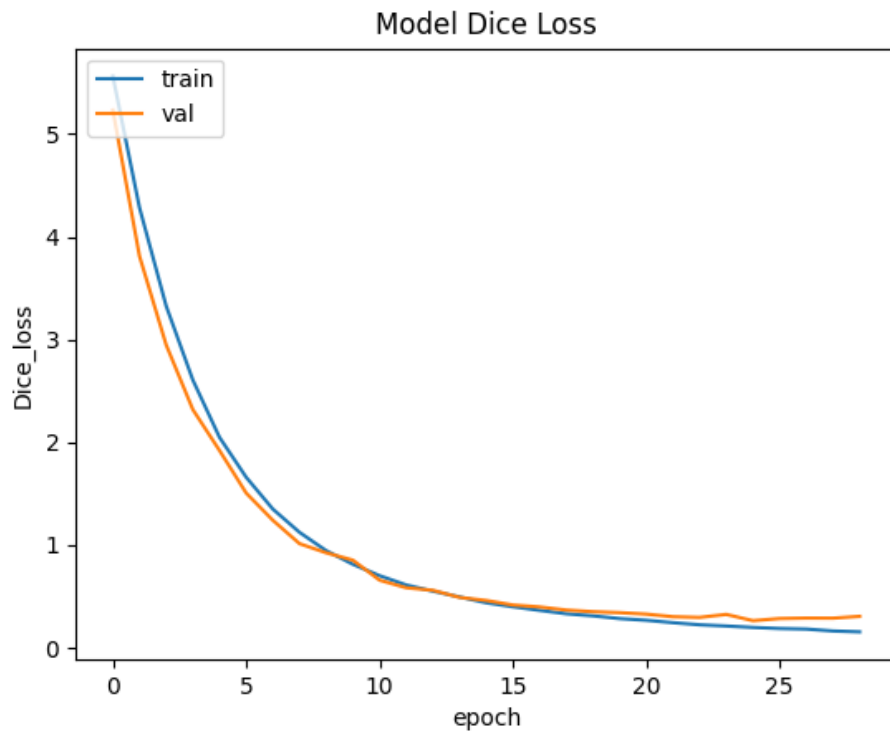


Figure 41 - Focal Loss Function over Training epochs for Training and Validation sets.

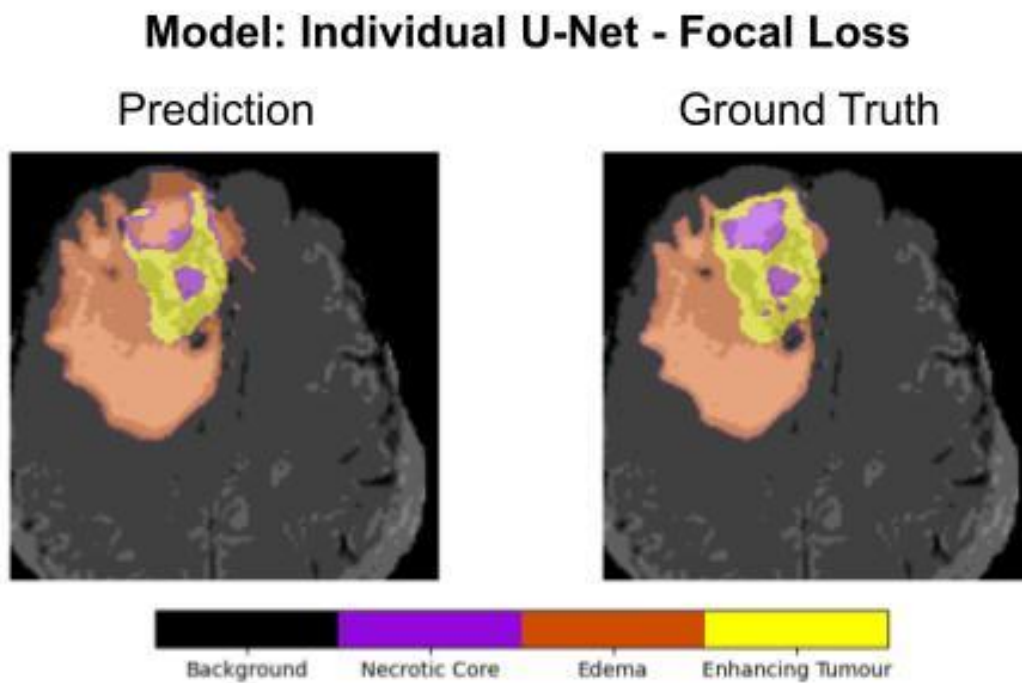


Figure 42 - Prediction and Ground Truth segmentation maps for an example from the test set. Coloured regions indicate tumorous tissue. Key indicates the alignment of tissue types to region colours in the segmentation

5. Discussion and Evaluation

5.1. Discussion of Results

The plot below compares the Dice metrics for the Group Model and the best performing Individual Model (Combo Loss) with those found in Table 1 in 2.5. A table of these results is included in A.I. The Group model has performed well, tied with [30] based on their mean scores. It performed particularly well segmenting the Whole and Enhancing Tumour regions. The Individual Model performed even better, with the 2nd highest mean Dice score of those surveyed, it has shown to be particularly adroit at segmenting the Enhancing Tumour with the highest Dice score of any. These scores demonstrate that both the Group and Individual Models have the potential to contribute to the improved performance of brain tumour segmentation in the wider field. There would certainly be scope for further development of this current architecture with the use of data augmentation, proper parameterisation using grid search and allocation of more computational resources to the problem. These will be discussed further in the evaluation.

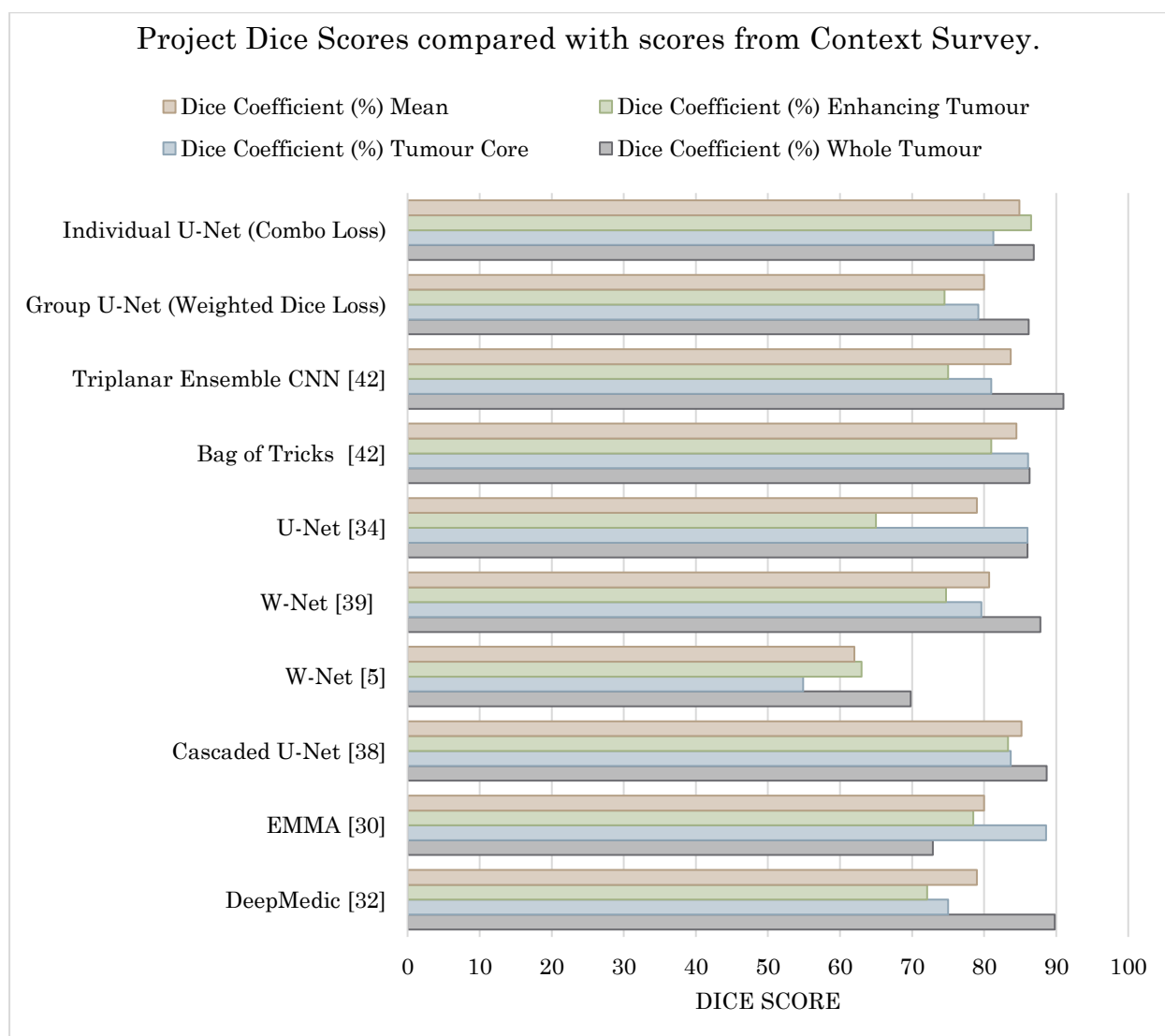


Figure 43 - Dice Scores of Group and Individual Models compared to results in Context Survey (Table 8). Includes scores for the Whole Tumour, Tumour Core, Enhancing Tumour and Mean of the three.

Examining the performance of different loss functions for this particular problem led to improved performance compared to the Group Model and many of the current state of the art applications surveyed in the Context Survey. This work at loss function selection can be beneficial for other researchers when selecting loss functions for tumour segmentation problems. Since 2015, as noted in 2.4.1., the BraTS dataset has undergone a number of changes, including the removal of one target class and the change from a dual problem for High-Grade and Low-Grade Gliomas to an integrated one. This means comparing the results of models using different versions of BraTS should be done with caution.

5.2. Evaluation

5.2.1. Value Alignment

The project’s success will first be observed with the desirable features set out in 2.5.

I. Dice scores and other performance metrics.

The performance of both the Group and Individual Models has been very good, achieving performance comparable to that found in the Context Survey, and in some instances outperforming them. The performance could have been improved by greater parameterisation, using methods such as grid search [51] to algorithmically find the optimal hyperparameters such as batch size, learning rate or optimiser. This would be a more rigorous method of finding the optimal hyper-parameters for the Model. The Combo Loss function could also benefit from a grid search approach in deciding the optimal alpha and beta values when weighting the two loss functions it utilises. There was some finding though experimentation that Stochastic Gradient Descent had the potential to outperform Adam as an optimiser. However, SGD requires more fine tuning of its parameters than Adam, using SGD in combination with grid search to find the optimal learning rate, momentum and decay could increase the performance of both the Group and Individual Model. Another area for improvement would be the allocation of more computational resources. As mentioned in 3.1.2. the Model’s filter size has had to be reduced significantly in order to accommodate VRAM limitations. With more computational resources or optimisation in the code’s VRAM efficiency, larger filters could be used, allowing for the Model to assemble more sophisticated feature maps, improving performance as was shown by [52] and [53].

II. Computational Complexity

Limitations in hardware necessitated a relatively low computational complexity for the system. As outlined in 3.1.1. the development hardware used was relatively modest, which the Group and Individual Architecture has been developed to accommodate. One limitation in the project architecture is the amount of hard drive storage necessary, due to the approach of saving a formatted version of the dataset in stored memory. An advantage of using Docker to run the architecture via the GPU is that the system can be run on machines with modest processors.

III. Addressing class imbalance of training set

The Group Model uses a weighted loss function which allows it to handle the class imbalance found in the dataset [54]. The Individual Model also uses loss functions which are adept at learning with class imbalance present. Both models then, have worked well at addressing the issue of imbalance in the training set. More work could be done though, by using more intelligent patching or data augmentation. The use of a Data Generator means data augmentation can be more easily implemented [55].

IV. Time Costs

Using 50 epochs and each epoch running for around 60 seconds, in experimentation the Model took 53 minutes to train without any early stopping. Using a 10% split for the test set resulted in 40 records to predict for and evaluate and 20 overlay images to compile and save, this ran in 5 minutes. These are one-time, anecdotal measurements of the time elapsed between running the command to execute the python file and the point at which the outputs have fully displayed and/or saved. Using Python's time package which allows for run time to be calculated would be a more accurate measurement and is an area of further work.

V. Use of sufficiently large and representative datasets

The BraTS dataset is among the largest of its kind but only has 369 records so would be considered a small dataset for Machine Learning Algorithms. U-Net was selected due to its adroitness at handling smaller datasets [35]. The dataset has been designed to be as representative of the general population of patients with brain tumours [8, 9, 10]. However, due to its limited size, one cannot place certainty on its generalisability, even with the Model's good performance comparing the training, validation, and test set accuracies. A solution here could be the use of data augmentation as was used by [33] and [38].

VI. Generalisability

The splitting of the dataset into testing, training and validation sets, and the lack of any changes after running the test evaluation, ensures there is no data leakage. Both Models showed consistent performance between the training, validation, and test sets, indicating good generalisability when presented with unseen data. However as was discussed in the context of using large and representative datasets, The BraTS dataset though large for its kind, is not large enough to guarantee generalisability.

VII. Accessibility

In seeking to make the project's code base as accessible, the use of high-level packages and commenting wherever possible ensures that the code is easy to follow. Using Docker for execution had the benefit of packaging up the Python image and any packages required to run the architecture. Finally making the code for the project open source on GitHub allows researchers to access the project's code and utilise it.

VIII. Interpretability

Overall, the project has been successful in producing outputs which are interpretable to the user. For long processes during execution, tqdm bars are used. Print functions give the user visibility into the models processing as it executes. The classification reports and overlays are labelled with meaningful names (Necrotic Core, Enhancing Tumour etc.) rather than class labels (0,1,2,3) and the use of overlay images allows the models behaviour to be interpreted.

IX. End to End architecture.

The project does well at achieving an end-to-end solution. The independently run architecture aids in efficiency, where data loading and training need not be re-run when running the evaluation stage. The use of segmentation images illustrates the concept where the model could output overlaid images for use by stakeholders if the approach was adopted in a clinical environment. Currently the outputs are based on the already saved and defined dataset, rather than using data inputted by the user. This could be extended by adding a function which allows the user to input a set of MRI images (in their raw .nii.gz format). Then, a set of predicted segmentation maps, evaluation metrics for said predictions and overlaid images similar to those seen in the Results section. This would demonstrate the use case in which clinicians upload a set of MRI data and receive a corresponding set of predicted segmentations.

5.2.2. Summary and Future Work

Based on the values set out in 2.5. the model is a good example of a Deep Learning architecture for brain tumour segmentation. It has shown to perform well with Dice scores comparable or better than the current state of the art applications. The Model has been developed with a modular architecture using a commonly used language and packages. It is computationally reasonably inexpensive due to the limited computational resources, has shown to generalise well on unseen data, runs and executes in a reasonable time and presents an end-to-end solution. The use of a Data Generator leaves scope to easily add data augmentation which could address the issue of class imbalance and the dataset's size.

Much of the future work has been covered in the previous section. Expanding on this, more data from other datasets could be combined with that of BraTS. The Model is already well suited to this as the use of Data Generators allows a dataset of any size since the system need not load the entire dataset into memory. There is also great scope to develop a more sophisticated model, using approaches such as Ensemble and Adversarial networks have shown promise as in [30] and the Individual work demonstrated that simply by changing the loss function one can configure the model to be better suited for different classes. The existing model's performance could likely be improved with the use of grid search to algorithmically find the optimal values for certain hyperparameters such as learning rate, or the alpha and beta values in Combo loss.

6. Conclusion

To conclude, this project has presented a 2D U-Net based architecture, capable of performing binary and multi-class segmentation of brain tumours using MRI data. The Group Model is an adaption of previous work by [6, 33, 42], using a U-Net based architecture with a symmetrical encoding and decoding path, capable of preserving context within an image. When evaluated using the Dice metric, the Group Model achieved strong performance, achieving Dice scores of 0.862, 0.792, 0.865 for the Whole Tumour, Tumour Core and Enhancing Tumour respectively. This U-Net model was then used as a baseline for experimentation using 9 different loss functions including Dice loss, Cross Entropy, Combo Loss and Focal Loss. The loss function which contributed to the most favourable model performance was the individually developed and crafted Combo Loss, a weighted summation of Dice and Cross Entropy, with 30% weighting for Dice and 70% weighting for Cross Entropy. This model achieved Dice scores of 0.869, 0.813 and 0.865 for the Whole Tumour, Tumour Core and Enhancing Tumour respectively.

As outlined in 5.1. both models have achieved performance comparable to or greater than the current state-of-the-art applications reviewed in the Context Survey, with the mean Dice score for the Group Model 6th highest and the Combo Loss Model 2nd highest of the 8 papers reviewed in Table 8. The only model which achieved better performance was [38]. The use of the commonly used packages, TensorFlow and Keras as well as uploading the Project's code to an open-source repository (GitHub) means that researchers can easily utilise the methodologies of this project which contributed to its success.

The Individual development, analysing the use of different loss functions for the problem of brain tumour segmentation, has not been examined by the literature reviewed in the Context Survey. This work can aid future researchers in selecting a loss function which will optimise their model's performance. This project also good development potential in future work, as outlined in 5.2.1. lack of computational resources necessitated a reduction in the model's sophistication. Lower filter and batch sizes were used to accommodate for the lack of VRAM. A sensible development, therefore, would be to increase the filter size and test the model on a system with greater computational resources. There is also scope for greater parameterisation in the model, using methodologies such as grid search to algorithmically find the optimal hyperparameter values.

Fully automated segmentation may never be used in practice, due to the high stakes regarding the accuracy of segmentation. However, semi-automated approaches are already seeing deployment in the field. As AI models for segmentation become more sophisticated, the time clinicians need to spend correcting segmentations when using a semi-automated approach will reduce, further increasing the benefit of time saving and could lead to an increase in the overall accuracy of segmentation. This project has contributed to improving the fields understanding of tumour segmentation with DL by presenting an architecture with several novel developments including experimentation with loss functions, both models' favourable performance and the sharing of this project on open platforms.

7. References

- [1] C. R. UK, "Brain, other CNS and intracranial tumours statistics," [Online]. Available: <https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/brain-other-cns-and-intracranial-tumours>. [Accessed 01 06 2021].
- [2] M. P. C. B. R. Manuela Quaresma, "40-year trends in an index of survival for all cancers combined and survival adjusted for age and sex for each cancer in England and Wales, 1971–2011: a population-based study," *Lancet*. <http://dx.doi.org/10.1016/>, 2014.
- [3] R. R. D. H.-B. Andreas Stefani, "Autofocus Net: Auto-focused 3D CNN for Brain Tumour Segmentation," University of St Andrews, The Institute of Cancer Research UK, 2019.
- [4] Esteva et. al., "A guide to deep learning in healthcare," *Nature Medicine*, Vols. <https://doi.org/10.1038/s41591-018-0316-z>, 2019.
- [5] J. K. a. M. A. Abhishta Bhandari, "Convolutional Neural Networks for Brain Tumour Segmentation," *Insights Into Imaging*, no. <https://doi.org/10.1186/s13244-020-00869-4>, 2020.
- [6] Díaz-Pernas, "A Deep Learning Approach for Brain Tumour Classification and Segmentation Using a Multiscale Convolutional Neural Network," *Healthcare*, vol. <https://doi.org/10.3390/>, 2021.
- [7] M. N. R. K. G. D. K. T. Rikiya Yamashita, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. <https://doi.org/10.1007/s13244-018-0639-9>, pp. 611-629, 2108.
- [8] Menze et. al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," IEEE, 2015.
- [9] Bakas et. al., "Advancing The Cancer Genome Atlas Glioma MRI Collections with Expert Segmentation Labels and Radiomic features.," *Nature*, no. DOI: 10.1038/sdata.2017.117, 2017.
- [10] Bakas et. al., "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BraTS Challenge," 2019.
- [11] D. S. Tkachova, "Brain Tumour Segmentation (in MRI) using Deep Learning Project," University of St Andrews, 2020.
- [12] Iida et. al., "Metastasectomy as optimal treatment for late relapsing solitary brain metastasis from testicular germ cell tumor: A case report," *BMC Research Notes*, no. DOI:10.1186/1756-0500-7-865, 2014.
- [13] D. Louis et. al., "The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary".
- [14] M. L. J. W. F. W. T. L. a. Y. P. Jin Liu, "A Survey of MRI-Based Brain Tumor Segmentation Methods," *TSINGHUA SCIENCE AND TECHNOLOGY*, vol. 19, pp. 578-595, 2014.
- [15] R. B. Wenya Linda Bi, "Beating the odds: extreme long-term survival with glioblastoma," *Neuro-Oncology*, vol. doi: 10.1093/neuonc/nou166, pp. 1168-1195, 2014.
- [16] C. R. UK, "Glioma," 10 2019. [Online]. Available: <https://www.cancerresearchuk.org/about-cancer/brain-tumours/types/glioma-adults>. [Accessed 13 6 2021].

- [17] C. P. Davis, "Brain Cancer Facts," MedicineNet, 24 08 2020. [Online]. Available: https://www.medicinenet.com/brain_cancer/article.htm. [Accessed 13 06 2021].
- [18] Tandel et. al., "Multiclass magnetic resonance imaging brain tumor classification using artificial intelligence paradigm," *Computers in Biology and Medicine*.
- [19] N. H. S. (NHS), "NHS UK," 04 2020. [Online]. Available: <https://www.nhs.uk/conditions/benign-brain-tumour/diagnosis/>. [Accessed 13 06 2021].
- [20] C. W. R. University, "Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics," 2016. [Online]. Available: <https://case.edu/med/neurology/NR/MRI%20Basics.htm>. [Accessed 13 06 2021].
- [21] J. K. M. A. Abhishta Bhandari, "Convolutional neural networks for brain tumour segmentation," *Insights into Imaging*, vol. 11, no. <https://doi.org/10.1186/s13244-020-00869-4>, 2020.
- [22] Havaei et. al., "Brain tumor segmentation with Deep Neural Networks," *Medical Image Analysis*, vol. 35, no. <http://dx.doi.org/10.1016/j.media.2016.05.004>, pp. 18-31, 2016.
- [23] Akkus et. al., "Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions," *Journal on Digital Imaging*, Vols. DOI 10.1007/s10278-017-9983-4, 2017.
- [24] J. S. D. V. D. Jaber Juntu, "Bias Field Correction for MRI Images," *Computer Recognition Systems*, pp. 543-551.
- [25] IBM Cloud Education, "Neural Networks," IBM, 17 08 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/neural-networks>. [Accessed 14 6 2021].
- [26] B. K. Xide Xia, "W-Net: A Deep Model for Fully Unsupervised Image Segmentation," Boston University, 2017.
- [27] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks," towards data science, 15 12 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed 18 06 2021].
- [28] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks," towards data science, 15 12 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed 04 07 2021].
- [29] A. I. A. , A. A. M. K. a. H. F. A. H. Mahmoud Khaled Abd-Ellah, "Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks," *EURASIP Journal on Image and Video Processing*, vol. 97, no. <https://doi.org/10.1186/s13640-018-0332-4>, 2018.
- [30] Kamnitsas et. al., "Ensembles of Multiple Models and Architectures for Robust Brain Tumour Segmentation," Imperial College London, Springer, 2018.
- [31] M. e. al., "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge," 2019.
- [32] Kamnitsas et. al., "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Medical Image Analysis*, vol. 36, no. <https://doi.org/10.1016/j.media.2016.10.004>, pp. 61-78, 2016.
- [33] P. F. a. T. B. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Cornell University, 2015.

- [34] Dong et. al., "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks," *MIUA*, 2019.
- [35] M. H. C. Y. T. M. T. V. K. A. Md Zahangir Alom, "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation," *IEEE*, 2018.
- [36] N. N. S.-A. A. Fausto Milletari, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," Cornell University, 2016.
- [37] E. S. T. D. Jonathan Long, "Fully Convolutional Networks for Semantic Segmentation," University of Berkley, 2015.
- [38] C. D. M. L. a. D. T. Zeyu Jiang, "Two-Stage Cascaded U-Net: 1st Place Solution to BraTS Challenge 2019 Segmentation Task," South China University of Technology, 2020.
- [39] W. L. S. O. T. V. Guotai Wang, "Automatic Brain Tumor Segmentation using Convolutional Neural Networks with Test-Time Augmentation," King's College London, 2018.
- [40] S. Jadon, "A survey of loss functions for semantic segmentation," *IEEE*, 2020.
- [41] Google, "Normalization," Google, 03 06 2021. [Online]. Available: <https://developers.google.com/machine-learning/data-prep/transform/normalization>. [Accessed 06 07 2021].
- [42] Carinanorre, "carinanorre/Brain-Tumour-Segmentation-Dissertation," Github, 14 08 2020. [Online]. Available: <https://github.com/carinanorre/Brain-Tumour-Segmentation-Dissertation#readme>. [Accessed 02 06 2021].
- [43] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition," O'Riley, 2019, pp. 338 - 341.
- [44] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition," O'Riley, 2019, pp. 456 - 461.
- [45] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition," O'Reilly Media, Inc., 2019, pp. 492-495.
- [46] V. Bushaev, "Adam — latest trends in deep learning optimization.," Towards Data Science, 22 10 2018. [Online]. Available: <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>. [Accessed 01 08 2021].
- [47] D. C. M. M. Dr Daniel J Bell, "Dice similarity coefficient," Radiopaedia, [Online]. Available: <https://radiopaedia.org/articles/dice-similarity-coefficient?lang=gb>. [Accessed 29 07 2021].
- [48] Wikipedia, "Mean squared error," 31 07 2021. [Online]. Available: https://en.wikipedia.org/wiki/Mean_squared_error. [Accessed 28 07 2021].
- [49] J. Brownlee, "A Gentle Introduction to Cross-Entropy for Machine Learning," Machine Learning Mastery, 22 21 2020. [Online]. Available: [https://machinelearningmastery.com/cross-entropy-for-machine-learning/#:~:text=Cross%2Dentropy%20can%20be%20calculated,%20log\(Q\(x\)\).](https://machinelearningmastery.com/cross-entropy-for-machine-learning/#:~:text=Cross%2Dentropy%20can%20be%20calculated,%20log(Q(x)).) [Accessed 31 07 2021].
- [50] S. Trivedi, "Understanding Focal Loss—A Quick Read," Medium, 02 05 2020. [Online]. Available: <https://medium.com/visionwizard/understanding-focal-loss-a-quick-read-b914422913e7>. [Accessed 01 08 2021].
- [51] V. M. Sebastian Raschka, "Python Machine Learning - Third Edition," Packt, 2019, pp. 207-209.
- [52] Zebin et. al., "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, no. DOI: 10.1109/ACCESS.2019.DOI, 2019.

- [53] J. L. James Mou, "Effects of Number of Filters of Convolutional Layers on Speech Recognition Model Accuracy," UnitedHealth Group, 2021.
- [54] V. M. Sebastian Raschka, "Python Machine Learning - Third Edition," Packt, 2019, pp. 220-221.
- [55] J. Brownlee, "How to Configure Image Data Augmentation in Keras," Machine Learning Mastery, 05 07 2019. [Online]. Available: <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>. [Accessed 30 07 2021].
- [56] Y.-M. Z. C.-L. L. uan-Xing Zhao, "Bag of Tricks for 3D MRI Brain Tumor Segmentation," *International MICCAI Brainlesion Workshop*, vol. 11992, pp. pp 210-220, 2020.
- [57] M. R. R. M. R. W. Richard McKinley, "Triplanar Ensemble of 3D-to-2D CNNs with Label-Uncertainty for Brain Tumor Segmentation," *International MICCAI Brainlesion Workshop*, vol. 11992, pp. pp 379-387, 2020.
- [58] A. A. A. A. A. B. Sindhu Devunooru, "Deep learning neural networks for medical image segmentation of brain tumours for diagnosis: a recent review and taxonomy," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. <https://doi.org/10.1007/s12652-020-01998-w>, pp. 455-483, 2021.
- [59] G. L. P. M. Angulakshmi, "Automated Brain Tumour Segmentation Techniques—A Review," Wiley, 2017.

A. Appendix

A.I. – Supplementary Tables

Table A.1. – Division of Group Work

Action Item	Owner
Read images, pre-process them and store them as Numpy files.	AG360
Implementation of Data Generator to load dataset in batches during training.	AG360
Modify Model architecture include Data Generator batching.	AG360
Fit the model with the batching data	AG360
Build Model Architecture	DS291
Calculate weights of each class	DS291
Build custom weighted loss function	DS291
Model Optimization	DS291
Calculate Dice scores for Evaluation and Training	DS291
Loss and Accuracy plots	DS291
Overlay images of predicted brain tumour	JH384
Classification Report, used by Evaluation	JH384
Test, Train, Validation split, used by Evaluation and Model	JH384

Table A.1. - Division of work within group.

Table A.2. – Loss Function Dice Scores

Loss Fn	Dice Score		
	Whole Tumour	Tumour Core	Enhancing Tumour
Mean Squared Error	86.7%	79.6%	85.5%
Weighted MSE	26%	38.4%	41.1%
Dice	85.2%	70%	72.9%
Weighted Dice (Group Model)	86.2%	79.2%	74.5%
Cross Entropy	84.8%	80.5%	85.4%
Weighted Cross Entropy	52.1%	63%	77.4%
Combo Loss	86.9%	81.3%	86.5%
Class Weighted Combo Loss	10.1%	62.9%	0.01%
Categorical Focal Loss	85.8%	72.4%	82.7%

Table A.2. – Summary of Dice scores from evaluation on the test set. Examining the use of different loss functions when used to train a benchmark model. Highlighted are the group model benchmark (weighted dice) and the best performing loss function (combo loss).

Table A.3. – Loss Function Dice Scores

Loss Function	Class	Precision	Recall	F1
Dice Loss	Necrotic Core (class 1)	0	0	0
	Edema (class 2)	0.59	0.83	0.69
	Enhancing Tumour (class 3)	0.61	0.91	0.73
Weighted Dice Loss (Group Model)	Necrotic Core (class 1)	0.83	0.34	0.48
	Edema (class 2)	0.64	0.8	0.71
	Enhancing Tumour (class 3)	0.64	0.9	0.74
Mean Squared Error	Necrotic Core (class 1)	0.79	0.5	0.61
	Edema (class 2)	0.66	0.82	0.73
	Enhancing Tumour (class 3)	0.88	0.83	0.85
Weighted Mean Squared Error	Necrotic Core (class 1)	0.19	0.77	0.31
	Edema (class 2)	0.08	0.68	0.14
	Enhancing Tumour (class 3)	0.26	0.97	0.41
Cross Entropy	Necrotic Core (class 1)	0.8	0.51	0.62
	Edema (class 2)	0.8	0.67	0.73
	Enhancing Tumour (class 3)	0.84	0.87	0.85
Weighted Cross Entropy	Necrotic Core (class 1)	0.35	0.61	0.44
	Edema (class 2)	0.24	0.91	0.38
	Enhancing Tumour (class 3)	0.66	0.94	0.77
Combo Loss	Necrotic Core (class 1)	0.78	0.55	0.65
	Edema (class 2)	0.69	0.78	0.73
	Enhancing Tumour (class 3)	0.88	0.84	0.85
Weighted Combo Loss	Necrotic Core (class 1)	0.24	0.58	0.34
	Edema (class 2)	0.03	0.89	0.34
	Enhancing Tumour (class 3)	0.13	0	0
Focal Loss	Necrotic Core (class 1)	0.64	0.33	0.43
	Edema (class 2)	0.6	0.83	0.7
	Enhancing Tumour (class 3)	0.89	0.78	0.83

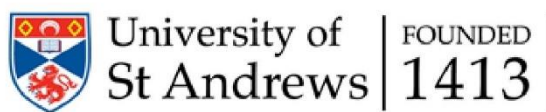
Table A.3. – Summary of Precision, Recall and F1 scores for target classes from evaluation on the test set. Examining the use of different loss functions when used to train a benchmark model. Highlighted are the group model benchmark (weighted dice) and the best performing loss function (combo loss)

Table A.4. – Project Dice Scores with Context Survey

Model	Dataset Used	Dice Coefficient (%)			Mean
		Whole Tumour	Tumour Core	Enhancing Tumour	
DeepMedic [32]	BraTS 2015	89.8	75.0	72.1	79
EMMA [30]	BraTS 2017	72.9	88.6	78.5	80
Cascaded U-Net [38]	BraTS 2019	88.7	83.7	83.3	85.2
W-Net [3]	BraTS 2017	69.8	54.9	63.0	62
W-Net [39]	BraTS 2018	87.8	79.6	74.7	80.7
U-Net [34]	BraTS 2015	86.0	86.0	65.0	79
Bag of Tricks [56]	BraTS 2019	86.3	86.1	81.0	84.5
Triplanar Ensemble CNN [57]	BraTS 2019	91.0	81.0	75.0	83.7
Group U-Net (Weighted Dice Loss)	BraTS 2020	86.2	79.2	74.5	80
Individual U-Net (Combo Loss)	BraTS 2020	86.9	81.3	86.5	84.9

Table A.4. - Dice Scores for Context Survey Research which use versions of the BraTS dataset, extended with Group and Individual Model scores.

A.II. – Ethical Approval Letter



School of Computer Science Ethics Committee

15 July 2021

Dear David, Jamel, Aditi and Ziyu,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS15652	Approved on:	15.07.2021	Approval Expiry:	15.07.2026
Project Title:	Deep Learning to Segment Brain Tumours using MRI data				
Researcher(s):	David Smith, Jamel Houd, Aditi Goswami, Ziyu Song				
Supervisor(s):	Dr David Harris-Birtill				

The following supporting documents are also acknowledged and approved:

1. Application Form

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
 - the details provided in your ethical application
 - the University's [Principles of Good Research Conduct](#)
 - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the ['additional documents' webpage](#) for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Wendy Boyter

SEC Administrator

School of Computer Science Ethics Committee
Dr Juan Ye/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX
Telephone: 01334 463252 Email: ethics-cs@st-andrews.ac.uk
The University of St Andrews is a charity registered in Scotland: No SC013532

User Guide - Brain Tumour Segmentation CNN

1. Source Code Directory Listing

Source Directory Item	Description
> bin_overlays	Directory, containing segmentation overlays for Group Model, binary segmentation.
> indi_overlays	Directory, containing segmentation overlays for Individual Model, segmentation.
> overlays	Directory, containing segmentation overlays for Group Model, multiclass segmentation.
> dataset	Directory, containing original dataset.
> dataset_conv > X > Y	Directory, containing dataset after pre-processing. X directory stores converted MR Images, Y directory stores ground truth labels.
>docker_image Dockerfile requirements.txt	Contains docker file and requirements, used to create Docker image for system execution.
> models	Stores .h5 files of all models.
ClassWeights.py	Calculates and saves weights for each target class in binary segmentation.
ClassWeights_bin.py	As above for binary segmentation.
DataLoad.py	Loads dataset and performs pre-processing.
DataLoad_bin.py	As above for binary segmentation.
DiceScore.py	Calculates Dice loss and dice scores, used by Model during training and by Evaluation classes.
GroupUnetEval.py	Evaluates Group UNet, loading model and using it to predict for either the test or validation set, then evaluating the results.

GroupUnetEval_bin.py	As above for binary segmentation.
GroupUnetModel.py	Contains Group U-Net Model architecture, compiles model, trains it on training set, then saves model as an h.5 file.
GroupUnetModel_bin.py	As above for binary segmentation.
IndiUnetEval.py	Evaluates Individual UNet, loading model and using it to predict for either the test or validation set, then evaluating the results.
IndiUnetModel.py	Contains Individual U-Net Model architecture, compiles model, trains it on training set, then saves model as an h.5 file. Can be configured for use of 9 different loss functions.
LossFunctions.py	Contains loss functions for use by model and evaluation as well as a selector function.

2. System Setup

The system is executed using a Docker image, which is able to containerise the Python and external libraries used by the architecture. A high level description of this process is described below:

- Set up a Docker container with TensorFlow

```
docker pull nvcr.io/nvidia/tensorflow:21.05-tf2-py3
```
- Create a custom Docker file and requirements file:

```
mkdir ~/tf2-custom
touch ~/tf2-custom/Dockerfile
touch ~/tf2-custom/requirements.txt
cd ~/tf2-custom
```
- Append the requirements.txt file with the following:

```
pytest
nilearn
matplotlib
sklearn
keras
seaborn
SimpleITK
```
- Build the custom image.

```
docker build --build-arg local_uid=$(id -u) --build-arg local_user=$USER -t tf2-custom .
```

3. Execution

- The Python files in the directory can now be executed on the GPU using the following:

```
docker run -v $CodeDir:$CodeDir -w $CodeDir --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 --runtime=nvidia --user $(id -u):$(id -g) --rm $DockerDir $pyFile
```

```
$CodeDir = path to source directory  
$DockerDir = path to Docker image  
$pyFile = Class to be run
```

- To execute the **Group Model** run the following pipeline:
 1. DataLoad.py
 2. ClassWeights.py
 3. GroupUNetModel.py
 4. GroupUNetEval.py
- To execute the **Group Model for Binary Classification**:
 1. DataLoad_bin.py
 2. ClassWeights_bin.py
 3. GroupUNetModel_bin.py
 4. GroupUNetEval_bin.py
- To execute the **Individual Model** run the following pipeline:
 1. DataLoad.py
 2. ClassWeights.py
 3. IndiUNetModel.py
 4. IndiUNetEval.py
- To select a different loss function for the Individual Model, in the Model class, comment out the existing loss function and uncomment the new loss function to be used.

```
#Loss function selection  
  
#chosen_loss = 'MSE'  
  
#chosen_loss = 'MSE_weighted'  
  
#chosen_loss = 'Dice'  
  
#chosen_loss = 'Dice_weighted'  
  
#chosen_loss = 'Cross_Entropy'  
  
#chosen_loss = 'Cross_Entropy_weighted'  
  
chosen_loss = 'Combo'  
  
#chosen_loss = 'Combo_weighted'  
  
#chosen_loss = 'Focal'
```