Brain Tumor Segmentation on the Basis of a U-Net Architecture

Adrian Buchwald[2], Nastassia Heumann[1], Stephen Tobin[2], Armin Puran Youssef[2], and Till Zemann[2]

[1]Hasso Plattner Institut firstname.lastname@student.hpi.uni-potsdam.de
[2]Institut für Informatik und Computational Science, Universität Potsdam
firstname.[middlename.]lastname@uni-potsdam.de

Abstract

We propose a selection of relatively small-scale, supervised variants of the U-Net Convolutional Neural Network (CNN) architecture (Ronneberger, Fischer, and Brox 2015) with partitioned inputs for the BraTS challenge 4D (3D + channel) brain tumor segmentation dataset. The motivation behind the downscaling is to broaden accessibility of student, researcher and clinician contributions to and use of Deep Neural Network (DNN) models, which often require considerable computational resources. We compare a 2D-slice approach with two 3D architectures, on the basis that 3D models performed better than 2D models when implemented with self-supervised training. In both of the 3D architectures, inputs are partitioned to reduce the minimum required resources. One of them includes surrounding context while the other does not. Surprisingly, the 2D model performed better than the 3D models. However, none of the three models had especially good performance. We include results from a (yet smaller) 3D model, which served as the basis for our subsequent developments, which performed better than our larger models. We consider the factors that may have yielded this performance, with a view to correcting/improving them in future. All code is publicly available on our GitHub repository: brainTumourProject

1 Introduction

A major social benefit of the growing capacity of deep neural networks (DNNs) to solve complex problems is that of human health. DNNs holds ever increasing promise for a range of medical purposes, such as prediction of a patient's risk of diabetes or the discovery of novel drugs for the treatment of disease. In the present executive report we build a DNN for the segmentation of various types of tumor tissue within multi-parametric, volumetric scans of human brains, in line with the goals of the 2017 BraTS Challenge (Bakas, Akbari, et al. 2017a, Bakas, Akbari, et al. 2017c, Bakas, Akbari, et al. 2017b, Bakas, Reyes, et al. 2018, Menze et al. 2015).

In the following text, we review a selection of relevant surveys and articles and identify a basic architecture to use as a basis for our model. Next we provide a summary of the publicly available data set with which we train and test our model. Then we proceed to illustrate the structure of (n variants of) our model implementation, including motivation for our chosen loss function and evaluation metrics. Finally, we discuss the performance of our model(s) (with reference to previously published models), and address ways in which the model could be further modified or restructured to yield better or more efficient performance.

1.1 Related Work

The development of U-Net (Ronneberger, Fischer, and Brox 2015) followed a period of expansion in the application of convolutional neural networks (CNNs) for large-scale image classification, resulting in AlexNet (Krizhevsky, Sutskever, and Hinton 2012), a model with dramatically higher accuracy than existing models (Deng et al. 2009). Notwithstanding the success of AlexNet, the goals of biomedical image processing often go beyond classification of whole images to classification of the constituent pixels/voxels of the image (i.e., semantic segmentation). Additionally, AlexNet and related network architectures depend on the availability of larger data sets than have typically been available in the biomedical context. While Ronneberger and colleagues' approach is not the first to attempt to adapt the successes of CNNs for image processing to the biomedical domain, it is offers substantial gains in both performance and efficiency over earlier approaches, thanks to

1. an encoder-decoder architecture with an abundance of feature channels in the decoder,
2. image augmentation by means of elastic deformation,
3. weighted loss to achieve high accuracy at segment boundaries.

The first of these points is perhaps the most critical difference between U-Net and a plain Fully Convolutional Network (FCN, Long, Shelhamer, and Darrell 2015) upon which it is based. While FCN segmentation predictions are typically based on the output of the two (or more) final max-pooling layers after multiple convolutional (and intermediate max pooling) layers (only a contracting path), U-Net subsequently also applies multiple deconvolutional layers to the previously convolved image (expanding path), allowing pixels/voxels and their surrounding context to be represented in higher-resolution before the segmentation map prediction is made.

With regard to the second point, data augmentation has been shown to improve performance of NNs when only relatively small data sets are available. Thirdly, applying higher weights to the loss function at segment boundaries ensures that the network learns these more challenging cases, even if they constitute just a small portion of the total image. We will return to this issue in our discussion of loss functions below.

Finally, the U-Net architecture includes skip connections, such that images from the contracting path are cropped and concatenated with images in the expanding path, allowing information both from the highly encoded representation of an image and from its less processed, more input-faithful form to contribute to learning, thus enabling greater spatial precision in segmentation.

It should be noted that Ronneberger used a data set consisting of stacks of 2D, electron microscope images of neural structures. Milletari and colleagues (Milletari, Navab, and Ahmadi 2016) modified Ronneberger's approach, primarily so that their V-Net model would accept 3D inputs rather than stacks of 2D images. While these two approaches are directly comparable in terms of their architecture and number of classes (both involve binary segmentation), unfortunately, the type of image (neural vs. prostate structures) and the non-overlapping metrics applied in the evaluation of the models do not lend themselves to easy comparison. However, this motivated us to compare these two approaches on a single data set with comparable evaluation metrics. Further, we were encouraged by findings of Taleb and colleagues (Taleb, Loetzsch, et al. 2020; Taleb, Lippert, et al. 2021) that processing of 3D images could lead to better model performance on segmentation tasks.

Beyond these very directly related set of papers, a number of recent review papers summarizing the variety of approaches to brain tumor segmentation are available. We found that of Liu (Liu et al. 2020) to offer a particularly thorough account of deep-learning based approaches, capturing critical aspects and motivations of models as well as their crucial differences. Particularly important for us among the numerous abstract and practical factors that they consider is that of efficient computation. In light of supercomputers of apparently ever growing computational power, students and smaller-scale research groups face a high barrier to entry into deep learning research. Given the relatively modest resources at our disposal for the current project, we decided to take an approach that would enable maximal use of modest resources, in particular by partitioning the data into smaller volumes to ensure that memory resources would not be exhausted during learning.

Finally, we considered that our approach to data partitioning (with fixed partition volumes) might yield poor predictions at partition boundaries, given that training within one partition would proceed largely independently of training within another partition. Thus, the concatenation of partition predictions might display discontinuities of a kind not attested in the original data set. As such we decided to introduce a variant 3D model in which additional surrounding image volume (or zero-padding at image edges) would be added to the target partition, in the expectation that the shared context regions would yield greater consistency among partitions when subsequently concatenated at prediction time.

1.2 Dataset

The dataset consists of 750 quadruples of 3D brain MR images split into training and test data. Since no corresponding labels are publicly available for the test data, only the 484 images that constitute the training data can be used for our purposes. Each image is structured along the three standard dimensions of a Cartesian volume, the fourth dimension (henceforth 'channel') corresponding to images produced by one of four different imaging techniques (capturing different tissue properties) of the same brain volume. The dimensions of the images within each channel are (240, 240, 155) voxels. The label dataset consist of just one channel and is of the same dimensions as the training/validation images. They contain three classes and one background class encoded via the numbers 0 to 3 with 0 being the background class. The three remaining classes are "edema", "non-enhancing tumor" and "enhancing tumour". Of these classes the background class makes up 98.2% of the data. This information is summarized in Table 1.

1.3 Summary

To summarize, we propose to build on a U-Net architecture to compare 2D and 3D approaches to semantic segmentation of types of brain tumor tissue. Among the two 3D approaches, one will involve learning contiguous image partitions without context, while the other will involve learning them with context. We plan to implement our models in such a way as to enable training with relatively modest computational resources.

Table 1: Class distribution

Dataset	Class	Voxel count	Proportion
Train	Background	$2.2 * 10^9$	0.982
	Edema	$2.6 * 10^7$	0.011
	Non-enhancing tumor	$7.2 * 10^6$	0.003
	Enhancing tumour	$8.0 * 10^6$	0.004
Test	Background	$4.7 * 10^8$	0.982
	Edema	$5.3 * 10^6$	0.011
	Non-enhancing tumor	$1.6 * 10^6$	0.003
	Enhancing tumour	$1.7 * 10^6$	0.004

2 Methods

2.1 Data preprocessing

The BraTS challenge dataset is provided in a format and structure that is easy to comprehend, navigate, and immediately use without any further preprocessing. We considered the possibility that the MR images may be noisy and might need to be cleaned of characteristic rician noise (Prucnal and Teich 1979). However, in order to implement such cleaning we would need more detailed information about scan sessions than is made available in the BraTS dataset. Additionally, the data was collected at numerous different sites and has been carefully anonymized. As such, denoising the images in this way was not practicable. Further, visual inspection of the images did not reveal particularly noisy signals. We also considered augmenting the dataset by flipping the images along the saggital plane (such that the hemispheres would be switched), modifying image sharpness, blurring, as well as elastic deformation (see e.g., Achanta and Hastie 2015). However, we opted not to apply augmentation, since our training was already quite resource-intensive with the original dataset alone.

(a) Before cropping

(b) After cropping

Figure 1: Axial slice of overlayed MR images before and after the background was cropped

As such, the preprocessing we applied did not need to be particularly extensive. However, in our exploration of the data, we noticed that the images have a substantial amount of background surrounding the brain image itself. We decided to crop much of this background out in the axial plane to reduce the size of the images and therefore training time. To determine a cropping region that would not excise any important content, we created a composite image of the entire training dataset and identified the regions of the axial plane filled with zero-valued pixels across all slices (see Figure 1).

2.2 Architecture and Training

2.2.1 Preliminary Exploration

To begin our exploration of relevant models, we developed a very simple, one-layer CNN, with two convolutions and two deconvolutions. We report some sample predictions from this model, which, while extremely shallow, appeared to have begun learning the outlines of the tumour segments over the course of 100 epochs.

2.2.2 2D U-Net Model

The 2D U-Net that we have developed is heavily inspired by Buda, Saha, and Mazurowski 2019. Since it is mainly used as a state-of-the-art model with which to compare our own 3D model, most of the architecture has been kept unchanged. The only adaptation that has been made for our data set was setting the number of input and output layers to four. The overall architecture consists of four U-Net blocks consisting of two (2D) 3x3-kernels and one 2x2 max pooling layer. Additionally, there is a bottleneck-block consisting of one 3x3 kernel. The channel dimension is multiplied by two with every block starting at a dimension of 32 after the first convolutional block. Outputs of the U-Net are concatenated and upconvoluted in U-Net fashion. The input of the model consists of a stack of 2D horizontal slices of the original MRI image per input channel. The outputs are four stacks of 2D segmentation channels, one for each class.

Since the input of this model consists of 2D slices of the original 3D dataset, no further context is provided for training/learning beyond each single slice of the data. As such we decided to develop parallel 3D models, that could incorporate additional context during training.

2.2.3 3D U-Net Models

In order to accommodate resource constraints, in particular, limited GPU memory, in our 3D approach we began by partitioning the images. The previously cropped cuboids (henceforth cubes) are first split into a batch of 8 minicubes (see Figure 2). A subset of minicubes is then passed through the U-Net to obtain a prediction and loss. The Dice loss is an intersection-over-union measure of difference and is calculated for each predicted minicube and the corresponding labeled minicube.

The size of a minicube batch is determined by the maximum number of minicubes that fit into the GPU memory. In our case two minicubes are used each training step. In order to efficiently iterate through the training data, we loop through training four times while maintaining the same minicube batch (1 image = 4 batches @ 2 minicubes). Thus a new image is sampled once every four learning steps, reducing data loading overhead by a factor of four.

In our initial explorations of training the 3D model, we found anomalies in the composite prediction produced by concatenating the individual minicube predictions, such that predicted segmentations were inconsistent with one another at minicube boundaries. To counteract the loss of information at these internal surfaces of the cube, we decided to supplement our initial 3D model by adding context, more specifically, by extending the minicubes by 20 pixels along each axis, transforming the input shape from the original (80, 96, 96) to (100, 116, 116) (see Figures 2 & 3).

The additional context yielded greater consistency and more alignment among predictions at the borders of neighboring minicubes. The most dramatic border discontinuities disappeared. The architecture was structured so that combined prediction would match the cropped input size of (160, 192, 192).

A consequence of adding context is the consumption of more memory. The extended minicube batches barely fit into our GPU's memory and we experienced regular crashes due to reaching the memory limit, which made development and experimentation more difficult. Given these memory constraints, we had to be precise and quite sparing about what data to keep in the GPU's memory. Ultimately, we found that the model with added context required an HPC GPU. However, this may be attributable to unnecessary storage of temprary tensors in the implementation. Further adaptation of the implementation may allow for training on smaller scale computers, as originally planned.

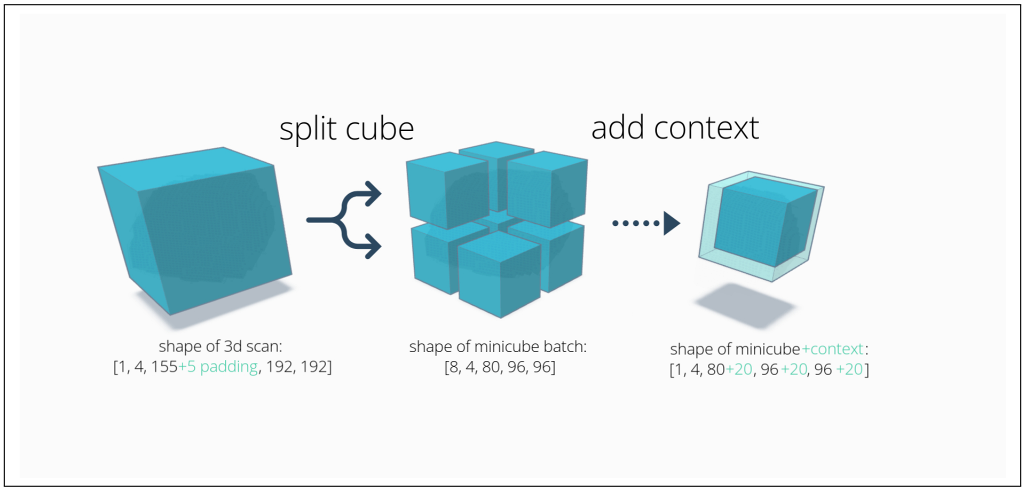


Figure 2: Preparation of 3D brain MRI scans with shape [batchsize, modalities, z, y, x]

There are several motivations behind our decision to develop a 3D segmentation model that is possible to run on a low end GPU.

First, it would allow researchers and students who do not have access to large-scale computing resources, and are thus constrained by their hardware, to train competetive models in spite of these constraints. Training on a GPU instead of a CPU substantially reduces training time, thus enabling the testing of various hyperparameters and model configurations, which might otherwise take a prohibitively long period of time. Thus, our approach lowers barriers to entry with regard to equipment and resources necessary for initiating investigations that may yield meaningful contributions to scientific inquiry.

Second, 3D models can help improve the accuracy of brain tumor segmentations over 2D models. This has been reported by Taleb, Loetzsch, et al. 2020, Taleb, Lippert, et al. 2021 and colleagues, who used an indirectly related training task and observed better performance in wholly 3D implementations compared to implementations drawing on 2D slices of 3D data. Though the models of Taleb and colleagues were self-supervised, in contrast to our supervised models, a compelling explanation for this difference is the greater availability of greater context for each voxel in a 3D sample compared to the relative sparsity (lesser dimensionality) of context for each pixel in a 2D sample.

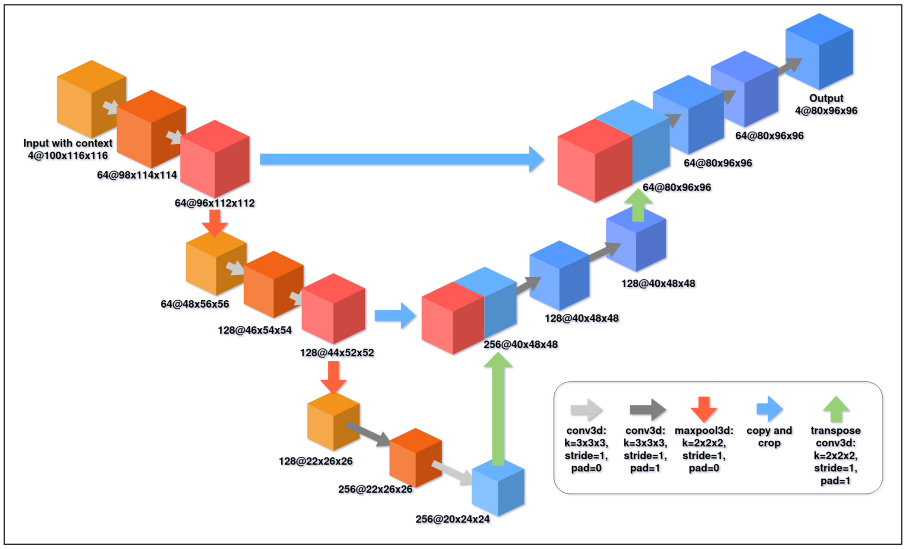


Figure 3: 3D U-Net with context architecture

This motivated us to continue to pursue the 3D model in spite of the challenges we encountered due to our hardware constraints.

Finally, our approach would allow predictions to be made on novel data using an office computer with a built-in low-end NVIDIA GPU of the kind in use in hospitals, clinics and doctors' offices, thus allowing a wider range of medical professionals to obtain estimated tumour segmentations.

2.3 Selection of Loss Function

Perhaps the most widely used loss function for classification problems, such as the present semantic segmentation problem, is the cross-entropy or (multinomial) logistic loss function (Zhang et al. 2021). However, in data sets in which categories are substantially imbalanced, learning of minority categories may be prevented because of their relative scarcity in the data set. In the present data set, the background class (no tumour) is by far the dominant class. Application of cross-entropy loss function could yield relatively low loss values by predicting the background category for every image voxel.

Dice loss is based on a measure that substantially reduces this problem. The goodness of a prediction is determined as (twice) the cardinality of the intersection of prediction and corresponding ground-truth label, normalized by the cardinality of their union. It has been widely employed in training and assessment of neural networks for biomedical image data. Dice loss (Dice) is based on the Sørensen-Dice coefficient $SDC \in [0; 1]$ (Sorensen 1948, Dice 1945), which is essentially the well known F1-score, and (in the binary case) is calculated as shown in equation 1 below.

$$\text{Dice} = 1 - \frac{2|\hat{y} \cap y| + 1}{|\hat{y}| + |y| + 1} \tag{1}$$

Thus assuming a tumour is present in an input instance, an across-the-board prediction of background labels would yield an SDC with a large denominator and a small numerator, resulting in a loss value that represents not just overall accuracy but also precision. Dice Loss can be generalized to the multiclass case by taking the average of the individual category losses, yet the gradient of this function can become unstable with very small categories Yeung et al. 2022. While we expect that activation functions such as ReLU would reduce this instability, we also considered an additional loss functions that might better match the distribution of categories and nature of the ground truth segmentation, namely Focal-Tversky Loss (Abraham and Khan 2019, Jadon 2020, N.B. the latter paper contains a few inaccuracies with respect to its sources).

Focal-Tversky loss is a combination of two loss functions, as the name implies. First, focal loss (FL; Lin et al. 2017) is a weighted variant of cross-entropy (CE) loss (Zhang et al. 2021), provided below with notation adjusted for convenience of the following explanations:

$$\text{CE}(p_t) = -log(p_t) \tag{2}$$

with

$$p_t = \begin{cases} p & \text{if y=1} \\ (1-p) & \text{otherwise} \end{cases}$$

and

$$\text{CE}(p, y) = \begin{cases} -log(p) & \text{if y=1} \\ -log(1-p) & \text{otherwise} \end{cases}$$

Well-classified voxels (those whose probability of matching the ground truth label is high) are assigned lower weights, while poorly classified voxels (with a low probability of matching the ground truth) are assigned higher weights. This is implemented with a modulating factor dependent on p_t (defined above), which contains a non-negative focusing parameter γ (see equation 3). Thus, focal loss emphasizes the difficult cases during training.

$$\text{FL} = -(1-p_t)^{\gamma} * log(p_t) \tag{3}$$

Tversky loss (TL; Salehi, Erdogmus, and Gholipour 2017) is based on the Tversky index (TI; 4) which, like Dice, is an intersection-over-union measure. In contrast to focal loss, which adjusts weights on the basis of classification difficulty, Tversky loss adjusts the weights of false positive and false negative predictions separately, allowing learning to focus on more specific aspects of the error than focal loss. This is implemented by dividing the cardinality of true positive pixels/voxels by the sum of the cardinality of true positives, false positives and false negatives (see equation 5).

$$\text{TI}(P, Y; \alpha, \beta) = \frac{|PG|}{|PG| + \alpha|P \backslash G| + \beta|G \backslash P|} \tag{4}$$

with

$$\begin{aligned} P &:= \text{set of predicted labels} \\ G &:= \text{set of ground-truth labels} \\ \alpha &:= \text{false positive penalty} \\ \beta &:= \text{false negative penalty} \end{aligned}$$

$$\text{TL} = 1 - TI \tag{5}$$

Thus, Focal-Tversky loss (FTL; equation 6) allows learning not only to focus on overall difficult cases, but also to adjust weights depending on the type of prediction error that has been made. Generalization from the binary to the multiclass case can be implemented by summing losses over all categories i:

$$\text{FTL} = \sum_i (TL_i)^{\frac{1}{\gamma}} \tag{6}$$

$$\text{with } \gamma \in [1; 3] \tag{7}$$

When γ is 1, FTL is equivalent to TL. With increasing values of γ, easier cases (with smaller losses) are downweighted and harder cases (with larger losses) play a proportionally larger role during learning.

In summary, we plan to make use of (i) Dice Loss because of its ubiquity in deep learning for medical imaging and (ii) Focal Tversky Loss because it allows for flexible adaptation to different subtypes of error.

2.4 Evaluation

We report Precision, Recall and F1 (see Table 2) for our 3D architectures, along with a confusion matrix (see Figure 4). These measures are widely used in the literature and easy to find formulas for. Precision corresponds to the proportion of all positive predictions that were correct, while Recall corresponds to the proportion of true positive cases that were correctly predictions. F1 is a harmonic mean of the two, whereby each contributes equally.

With regard to the 3D architectures, the relatively high scores for the background class is due to the fact that this is the most prevalent class and the models predicted this class across the board (hence the zero values for precision for the remaining classes). The NaN values for the remaining values for Recall and F1 are due to the presence of zero / NaN denominators.

We also report sample losses over training of a subset of our models (see Figure 5). Notably, the smallest, experimental architecture (SegNet) produced the most promising loss pattern over epochs. Given that we did not observe notable overfitting of the training data compared to the test data (see Figure 5), it was not necessary to incorporate regularization.

Needless to say, these results are disappointing. However, in the following section we will consider aspects of the model and training that may have led to this performance.

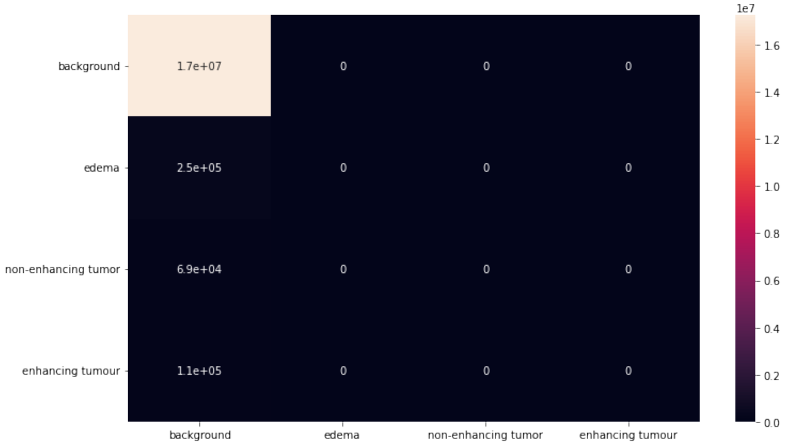


Figure 4: Confusion Matrix for 3D U-Nets (identical for 3d models with and without context)

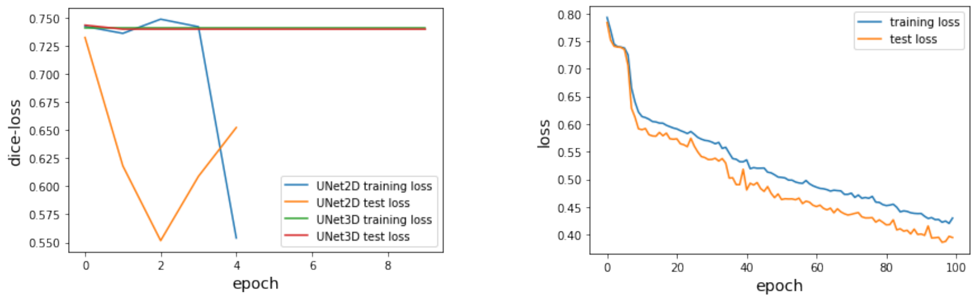
3 Discussion

We proposed to develop a U-Net based architecture for segmentation of brain tumour from the BraTS 2017 challenge dataset. Additionally, we considered one 2D and two 3D approaches (one with and one without reference to surrounding image context), expecting that the 3D approaches would outperform the 2D approach and that the approach with context would outperform the approach without context. We adapted our design to be amenable to relatively modest computational resources, in order make these methods more accessible to students, researchers and clinicians who might not have access to high-performance computing (HPC) facilities.

Given that we invested most time into developing the 3D models, we focus on evaluation of these two models, leaving the 2D model aside. The results did not support our predictions with respect to

Table 2: Evaluation Metrics

Architecture	**Class**	**Precision**	**Recall**	**F1**
2D	Background	unavailable	unavailable	unavailable
	Edema	unavailable	unavailable	unavailable
	Non-enhancing tumor	unavailable	unavailable	unavailable
	Enhancing tumour	unavailable	unavailable	unavailable
3D	Background	1.0	0.975	0.988
	Edema	0	NaN	NaN
	Non-enhancing tumor	0	NaN	NaN
	Enhancing tumour	0	NaN	NaN
3D with context	Background	1.0	0.975	0.988
	Edema	0	NaN	NaN
	Non-enhancing tumor	0	NaN	NaN
	Enhancing tumour	0	NaN	NaN



(a) (2D and 3D) U-Net: 10 training epochs

(b) SegNet: 100 training epochs

Figure 5: Sample Dice-Loss during training for different architectures

model performance. The two 3D models both performed equally poorly. So we now address factors that might have led to this unexpected performance.

First, we note that the 3D model with context did not make efficient use of memory and that any further training would require additional inspection of the code to improve memory usage. Second, although we aimed to base our architecture on U-Net, in order for it to be possible to train the models at all using our immediately available resources, we had to design relatively shallow models. Given that additional depth can provide major improvements in model performance, we suspect that this lack of depth likely had a detrimental effect.

A further explanation for the model performance relates to Focal-Tversky loss, which we experimented with initially but ultimately abandoned in favour of Dice (and Cross-Entropy) loss, which appeared to yield better performance. The hyperparameters (α, β, γ) are all tunable, not learned, and given that the efforts we had already made to train on a small scale did not leave much memory/time, we were not able to perform a grid search (or other approach to hyperparameter tuning) to identify the best set of hyperparameters for the dataset. That said, setting α (coefficient of false-positives penalty) higher and β (coefficient of false negative penalty) lower may have led to better performance. Additionally, an error in selection of the γ parameter, which was not correctly provided in some of our reference literature, may have substantially hampered the training of our models.

Although the performance of these models was far from what we were hoping for, we fully expect that a revised 3D model would still perform well, given prior observations from the literature on 3D models (in particular with the comparison of 2D models).

Given that our more heavily developed models did not perform well, we felt it appropriate to present some of the predictions from the initial, exploratory model that we developed. This shallow CNN began to identify the edges of the brain tumour categories after 100 epochs of training (see Figure 6). This was the promising basis for our subsequent development.

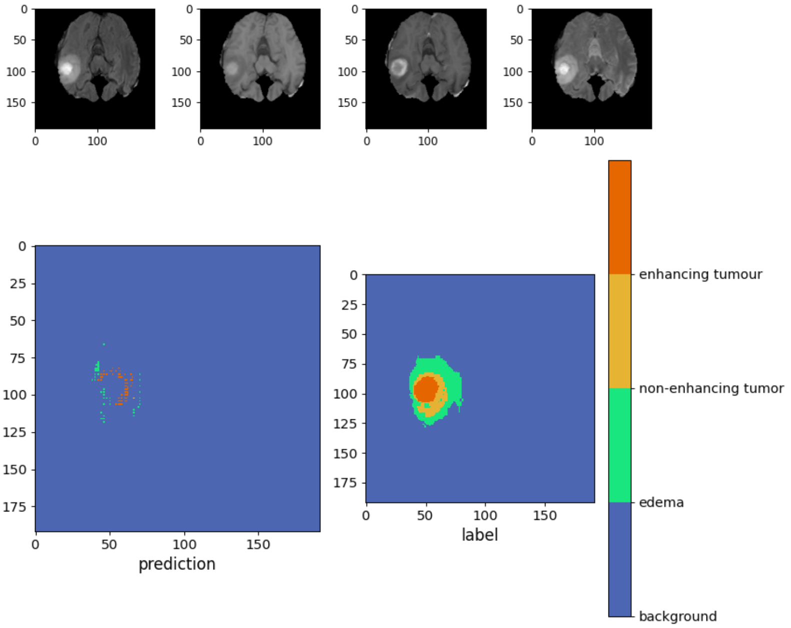


Figure 6: Sample prediction using the SegNet

With regard to our proposal to use limited resources for training and predicting from these models, while it is a strong advantage to have access to HPC resources, with further optimization of our implementations, we believe it would still be possible to train and predict from such models using smaller scale computational resources.

Acknowledgments

We would like to thank Prof. Dr. Chistoph Lippert, Noel Danz, Tahir, Miriyev, Eshant English and Wei-Cheng Lai for sharing their experience and expertise with use over the course of the semester. While the performance of the reported models may have been less than we expected, we feel much better prepared to design, build and test complex neural networks with greater insight and rigour than we would otherwise be without having had this opportunity.

References

Abraham, Nabila and Naimul Mefraz Khan (2019). “A novel focal tversky loss function with improved attention u-net for lesion segmentation”. In: *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE, pp. 683–687.

Achanta, Rakesh and Trevor Hastie (2015). “Telugu OCR framework using deep learning”. In: *arXiv preprint arXiv:1509.05962*.

Bakas, S, H Akbari, et al. (2017a). “Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features”. In: *Scientific data* 4.1, pp. 1–13.

– (2017b). *Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection [Data Set]. The Cancer Imaging Archive*.

– (2017c). *Segmentation labels for the pre-operative scans of the TCGA-GBM collection. The Cancer Imaging Archive*.

Bakas, S, M Reyes, et al. (2018). “Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge”. In: *arXiv preprint arXiv:1811.02629*.

Buda, Mateusz, Ashirbani Saha, and Maciej A Mazurowski (2019). “Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm”. In: *Computers in Biology and Medicine* 109.

Deng, Jia et al. (2009). “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Dice, Lee R (1945). “Measures of the amount of ecologic association between species”. In: *Ecology* 26.3, pp. 297–302.

Jadon, Shruti (2020). “A survey of loss functions for semantic segmentation”. In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, pp. 1–7.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25.

Lin, Tsung-Yi et al. (2017). “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.

Liu, Zhihua et al. (2020). “Deep learning based brain tumor segmentation: a survey”. In: *arXiv preprint arXiv:2007.09479*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Magadza, Tirivangani and Serestina Viriri (2021). “Deep learning for brain tumor segmentation: a survey of state-of-the-art”. In: *Journal of Imaging* 7.2, p. 19.

Menze, B et al. (2015). “The multimodal brain tumor image segmentation benchmark (BRATS)”. In: *IEEE transactions on medical imaging* 34.10, pp. 1993–2024.

Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi (2016). “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 fourth international conference on 3D vision (3DV)*. IEEE, pp. 565–571.

Prucnal, P and M Teich (1979). “Single-Threshold Detection of a Random Signal in Noise with Multiple Independent Observations, Part 2: Continuous Case”. In: *IEEE Transactions on Information Theory* 25.2, pp. 213–218.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Salehi, Seyed Sadegh Mohseni, Deniz Erdogmus, and Ali Gholipour (2017). “Tversky loss function for image segmentation using 3D fully convolutional deep networks”. In: *International workshop on machine learning in medical imaging*. Springer, pp. 379–387.

Sorensen, T (1948). “A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons”. In: *Kongelige Danske Videnskabernes Selskab* 5.4, pp. 1–34.

Taleb, Aiham, Christoph Lippert, et al. (2021). “Multimodal self-supervised learning for medical image analysis”. In: *International Conference on Information Processing in Medical Imaging*. Springer, pp. 661–673.

Taleb, Aiham, Winfried Loetzsch, et al. (2020). “3d self-supervised methods for medical imaging”. In: *Advances in Neural Information Processing Systems* 33, pp. 18158–18172.

Yeung, Michael et al. (2022). “Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation”. In: *Computerized Medical Imaging and Graphics* 95, p. 102026.

Zhang, Aston et al. (2021). “Dive into deep learning”. In: *arXiv preprint arXiv:2106.11342*.