CS 7643: Semantic Segmentation of Multimodal Brain MRIs for Tumor Segmentation

Adriel Bustamante
abustamante31@gatech.edu

Gabriel O'Hara
gohara7@gatech.edu

Jonathan Pang
jpang9@gatech.edu

Brooks Roney
broney6@gatech.edu

Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

This paper seeks to implement and analyze the performance of a semantic segmentation deep neural network on the BraTS2020 challenge dataset of multimodal brain MRIs. We explore two different architectures, U-Net and TransUNet, and different loss functions with the goal of achieving successful pixel-level tumor segmentations on slices of these brain MRIs, measuring our success with the Dice coefficient. We found that TransUNet outperforms U-Net, with TransUNet reaching a Dice coefficient of 0.87480 and U-Net reaching a Dice coefficient of 0.84240.

1. Introduction/Background/Motivation

In this project, our team set out to investigate the task of semantic segmentation in the context of brain tumor analysis. Specifically, we aimed to implement a standard U-Net [10] model and evaluate its performance on the BraTS2020 dataset, which contains multimodal MRI scans with pixel-level tumor annotations. We then extended this baseline by incorporating transformer-based components into the architecture, following the design of TransUNet [2], to assess whether these modifications yielded measurable improvements in segmentation quality. Our overarching goal was to understand the relative strengths and weaknesses of convolutional and hybrid convolution-transformer architectures in this domain.

Traditionally, segmentation of brain tumors in clinical settings is performed manually by radiologists, which is both time-consuming and subject to inter-observer variability. In research and clinical machine learning applications, convolutional neural networks (CNN), particularly encoder-decoder architectures like U-Net, have become the de-facto standard due to their ability to localize features while maintaining spatial accuracy. However, conventional CNNs struggle to capture global contextual information, especially in high-resolution medical images. Recent work has proposed hybrid architectures that integrate transformer modules to address this limitation by modeling long-range dependencies. Despite their promise, it remains unclear whether these more complex models offer significant gains in medical imaging tasks, especially when training data is limited or domain-specific.

Accurate and reliable automated tumor segmentation can substantially reduce the workload of clinical experts, enhance diagnostic precision, and enable faster treatment planning. From a research perspective, understanding the trade-offs between classical CNN-based models and more recent transformer-augmented architectures could inform future design choices for medical vision systems. If transformer-based architectures consistently outperform convolutional baselines in this setting, it may justify the additional computational cost and complexity in real-world deployments.

We used the BraTS2020 dataset, a well-known challenge dataset in the field of medical image segmentation. It comprises of pre-operative MRI scans from patients with glioblastoma and lower-grade glioma, annotated with voxel-wise labels for tumor subregions including the enhancing tumor, tumor core, and surrounding edema. Each case includes four MRI modalities: T1, T1c, T2, and FLAIR, which together offer complementary views of the underlying pathology. The dataset is curated with input from multiple institutions and includes standardized preprocessing steps such as skull stripping and co-registration, enhancing its consistency and utility for model training and evaluation. Its size, label quality, and diversity make it particularly well-suited for benchmarking segmentation models.

In its native form, the dataset is made up of 3D MRI

scans in NIfTI file format. To simplify the task, we use a sliced version of the dataset, where each horizontal slice of the MRI is used as an individual 2D image.

Because this is a challenge dataset, the canonical test dataset is gated away from the public and not accessible to anyone except the BraTS challenge administrators. To combat this, we split the public training dataset into smaller training, validation, and test datasets.

2. Approach

We tested 2 models on their ability to produce pixel-level segmentations of tumors on pre-operative MRI scans, using the BraTS2020 dataset. The models tested were a U-Net model, a convolutional network originally proposed for biomedical segmentation, and a derivative hybrid U-Net-transformer architecture model, TransUNet. These 2 models were trained, tested, and compared to understand their efficacy on tumor identification. These models were chosen due to their success on other similar datasets, as shown by Ronnenberger et al. [10] and Chen et al. [2].

Prior to implementing the two networks, two problems became immediately obvious. The first: there would be a severe class imbalance across the dataset, due to the tumor regions being small relative to the rest of the image. Such a class imbalance could naturally result in a model creating high-Dice coefficient segmentation masks that only return the background class. The second issue is that U-Nets are known to output segmentation masks that are smaller than the input image, an issue pointed out in the original paper. Following the suggestion in the original paper, we padded the input image with a reflection padding strategy such that the output segmentation mask was the correct height and width of the target segmentation mask.

On top of the initial, easily-foreseen issues, there were several snags and unexpected problems that arose throughout the process. In implementing the original U-Net, there was uncertainty in the form that the segmentation target should take. In the original implementation by Ronneberger et. al. [10], the U-Net returns an N-channel image, where N is the number of classes being segmented (including the background class). However, consistent with any other multi-class problem, the output could be reduced to a mono-channel image, where every pixel value is the predicted class. Depending on this choice, the segmentation target and loss functions would need to take a different form. To simplify later experiments with different losses, we chose to reduce the output to a mono-channel image of labels.

3. Experiments and Results

Throughout our experimentation, we measured our model's performance using a multi-class adapted Dice similarity coefficient, $D = \frac{1}{C}\sum_{c=0}^{C}\frac{2|A_c \cap B_c|}{|A_c|+|B_c|}$, where A_c and B_c represent the target and predicted segmentation masks for a given class $c \in C$ respectively.

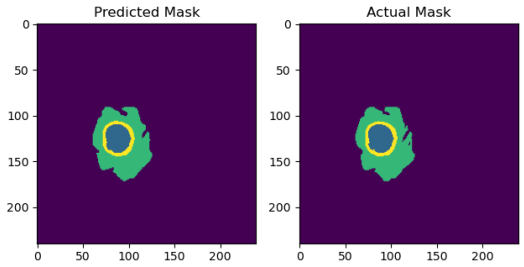

Figure 1. Example image segmentation with baseline U-Net model

All experiments were performed using PyTorch for deep learning, and all model and loss implementations were from scratch. All models were trained for 10 epochs. We use the Adam optimizer[7] to make the hyperparameter search less demanding, and to improve convergence time when training.

As a pre-processing step before training, we pruned our dataset to remove images with full background mask training images that took up less than 20% of the total image area. This was done in an attempt to address some of the class imbalance issues, as well as to accelerate training time. As significant parts of all target segmentations are still overwhelmingly composed of background labels, we determined that the loss of these images would not negatively impact the final results.

As a baseline, we use the standard U-Net implementation with unweighted cross-entropy as the loss function, and a learning rate of 1e-4. In keeping with the original paper, the baseline uses a batch size of 1.

This baseline achieves a Dice coefficient of **0.84240** on the test dataset, with visually impressive results, as shown in Figure 1. However, the fact that U-Net can perform admirably is not, in itself, news; U-Net has become a standard approach for semantic segmentation tasks because of its relatively straightforward architecture and reliability, especially given that it was originally proposed for biomedical segmentations like these. Can we improve on these results?

3.1. U-Net Model

To answer this question, we first explored using different loss configurations. Namely, we sought to test unweighted cross-entropy, weighted cross-entropy, unweighted focal loss[8], and weighted focal loss. In the case of weighted losses, we use a set of class-balanced weights $\alpha = \frac{1-\beta}{1-\beta^{n_i}}$, where β is a hyperparameter and n_i is the number of samples of a given class i, as proposed by Cui et. al. [3].

There were also some experiments with Dice loss [9], a relaxed and differentiable adaptation of Dice coefficient, due to surveys of semantic segmentation losses [1][5] that noted its inherent robustness to imbalanced datasets. However, several iterations of the Dice loss implementation continually resulted in models that showed no signs of learning, even after tuning hyperparameters or changing model architectures. Instead, losses remained near-constant for these models, and Dice coefficients calculated with validation data remained static, neither improving or worsening. Consequently, we discarded Dice loss as a viable option for this experiment.

Given that focal loss first emerged as a proposed answer to class imbalances in training data, and that class-balanced focal loss was intended as an extension to better address class imbalances, the natural expectation was that focal loss would outperform cross-entropy across the board. By this logic, even weighted cross-entropy should outperform unweighted cross-entropy, when given a set of class-balanced weights α. However, this could not be further from the truth.

Instead of the expected benefits from addressing the class imbalance via the loss function, vanilla cross-entropy handily outperformed the other losses. Contrary to expectations, the more weighting applied to lesser-represented classes, the worse the performance, as seen in Table 1.

One possible explanation of this may come from the original paper [3] that proposes α. Cui et. al. specifically call out class-balanced loss as an answer to long-tailed datasets. While our dataset is imbalanced, it is possible that by simple dataset pruning we were able to address the imbalance enough that the dataset is no longer truly long-tailed. Consequently, by inversely weighting lesser-represented classes, we cause the model to be more sensitive and trigger false-positive detections. This is partially borne out by visually inspecting some of the predicted segmentations, such as in Figure 3. Despite only one small region corresponding to a non-background class in the target, the predicted segmentation contains several regions of a completely unrelated class not seen in the target.

An additional explanation may come from the original dataset. While class-balanced losses address data imbalance, they have no additional sensitivity for classes that are highly correlated with each other, spatially or otherwise, such as in this dataset. The BraTS2020 dataset (and all other BraTS datasets) have labels corresponding to different regions of the same tumor. So any non-background classes in this dataset are more likely to be clustered together, instead of interspersed through the image. With this in mind, class-balanced losses may actually be a sledgehammer that treats these classes too separately, instead of factoring in their spatial proximity and inter-class relationships.

While all of these losses show improvement over time (Figure 4), the baseline unweighted cross-entropy exhibits the strongest performance throughout all of training, starting with the second-lowest loss and the highest Dice and carrying those titles through to the end. Unweighted focal loss has the closest performance to unweighted cross-entropy across train, validation, and test, and even manages to have the lowest training and validation loss. This does expose a flaw in our testing methodology: all focal losses were tested with the same hyperparameter $\gamma = 1$, which is the tunable parameter in focal loss. Lin et. al. [8] mention their experimental results showed $\gamma = 2$ to work best for them, but that still involved several rounds of testing and evaluation, which our focal loss testing omitted. Because focal loss is intended to address hard-to-classify samples, it is possible that tuning focusing parameter γ to be any higher may still have had an overall-negative impact, but the fact that unweighted focal loss exhibits the lowest loss during training suggests that this experimental space still has some room to explore.

We only explored two learning rates, as when choosing to use the Adam optimizer[7], we had an expectation that learning rate would have a less drastic impact than with vanilla stochastic gradient descent. This is partially borne out by the results presented in Table 1, though there are still some observations of note. Namely, reducing the learning rate did not improve the performance of the model on the test dataset, save for $\beta = 0.5$ CE, and unweighted focal loss. Without additional training runs to confirm that these results are statistically significant, it is impossible to draw solid conclusions from this, and we cannot suggest that a lower learning rate will improve the performance of this model when using either of those two losses.

We can, however, note that at a reduced learning rate of 1e-5, the test performance becomes worse faster as the β parameter changes. This may be a result of the smaller learning rate causing the optimizer to not find local optima quickly enough, and getting stuck at less-optimal positions in the loss landscape at the end of training, where models with a higher learning rate either reached closer-to-optimal positions in the loss landscape faster, or skipped over certain local optima altogether.

Taking a look at the learning curves (Figure 2) for both the unweighted versions of the tested losses, at both learning rates, we can see that this former point may be the case: for both learning rates, the learning curves for each of the two losses appear to be converging to nearby points. If the models with the lower learning rates were allowed to train for 1 to 2 more epochs over the higher learning rate models, it is reasonable to expect that they would reach and possibly surpass the performance of these higher LR models.

Loss	Test Dice @ LR=1e-4	Test Dice @ LR=1e-5
Unweighted CE	**0.84240**	**0.83930**
CE, $\beta = 0.5$	0.73105	0.78333
CE, $\beta = 0.75$	0.72120	0.70263
CE, $\beta = 0.99$	0.64190	0.62797
Unweighted Focal	0.77056	0.78319
Focal, $\beta = 0.5$	0.74594	0.68743
Focal, $\beta = 0.75$	0.65048	0.62573
Focal, $\beta = 0.99$	0.51534	0.53782

Table 1. U-Net Test Dice Across Different Losses

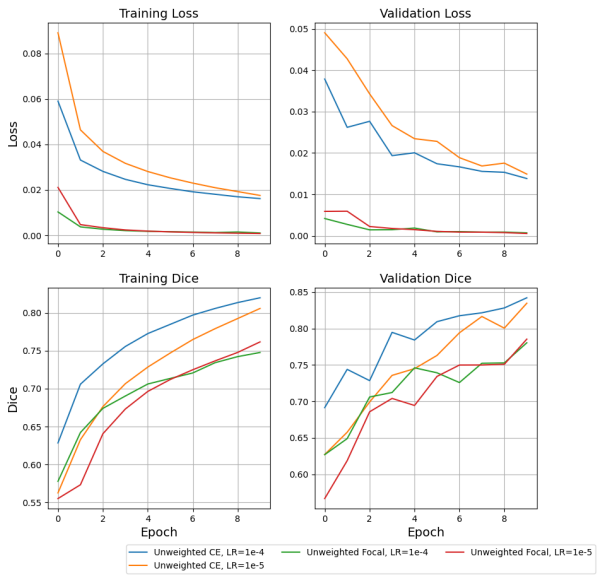

Figure 2. U-Net Training Curves across Learning Ratesβ=0.99

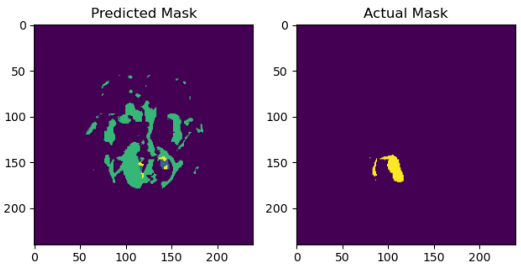

Figure 3. Predicted vs. True Segmentation with focal loss, β=0.99

3.2. TransUNet Model

Similar to the U-Net model, we explored the testing accuracy of different hyperparameter combinations for the TransUNet model, first introduced by Chen et. al. [2] as an

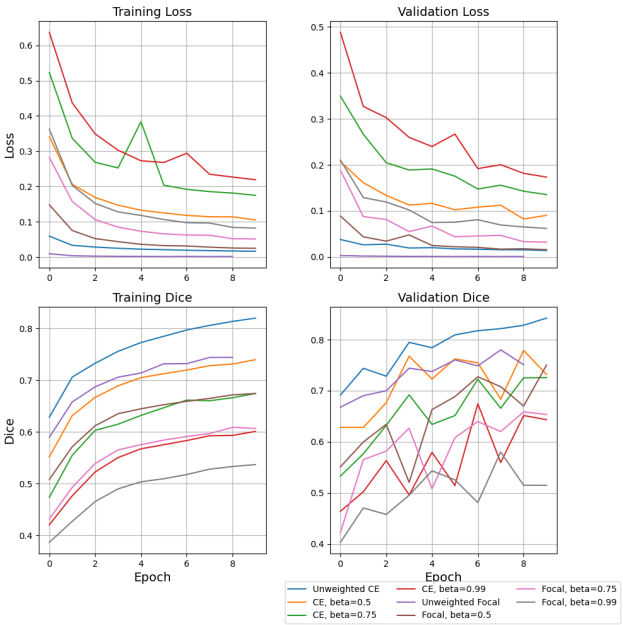

Figure 4. Baseline U-Net with Different Losses

adaptation of the classic U-Net. Where U-Net uses convolutional layers for both its encoder and decoder components, resulting in the eponymous "U"-shaped architecture, TransUNet replaces the encoder path with Transformer blocks. The input image is first passed through convolutional layers to extract hidden features, which are then linearly projected and passed through Transformer blocks. The motivation of this adaptation is to use the well-documented ability of a Transformer to model and encode global information using self-attention to reduce the loss in feature resolution that is inherent to models with max pooling operations (such as U-Net).

As a baseline, we again use unweighted cross-entropy as the loss function, a learning rate of 1e-4, and a batch size of 1. This baseline achieves a Dice coefficient of **0.80036** on the test dataset, slightly lower than the Dice coefficient of **0.84240** found for the baseline U-Net.

Just as we performed for the U-Net model, we first explored different loss configurations, unweighted cross-entropy, weighted cross-entropy, unweighted focal loss, and weighted focal loss.

The results mirrored the results from the UNet loss type tuning experiment. Once again, vanilla cross-entropy outperformed the other losses. As before, the greater the weighting applied to lesser-represented classes, the worse the model performed. The results are shown in Table 2 and training and validation curves are shown in Figure 5. Possible reasons for the unexpected results are discussed in Section 3.1. Worth noting, however, is that as with U-Net, all focal losses were tested with the same hyperparameter $\gamma = 1$. The fact that unweighted focal loss exhibits the low-

Loss	Test Dice @ LR=1e-4	Test Dice @ LR=1e-5
Unweighted CE	**0.80036**	**0.87480**
CE, $\beta = 0.5$	0.68717	0.83365
CE, $\beta = 0.75$	0.68035	0.78808
CE, $\beta = 0.99$	0.57009	0.73566
Unweighted Focal	0.75144	0.84543
Focal, $\beta = 0.5$	0.65378	0.74749
Focal, $\beta = 0.75$	0.55289	0.68219
Focal, $\beta = 0.99$	0.41114	0.55809

Table 2. TransUNet Test Dice Across Different Losses

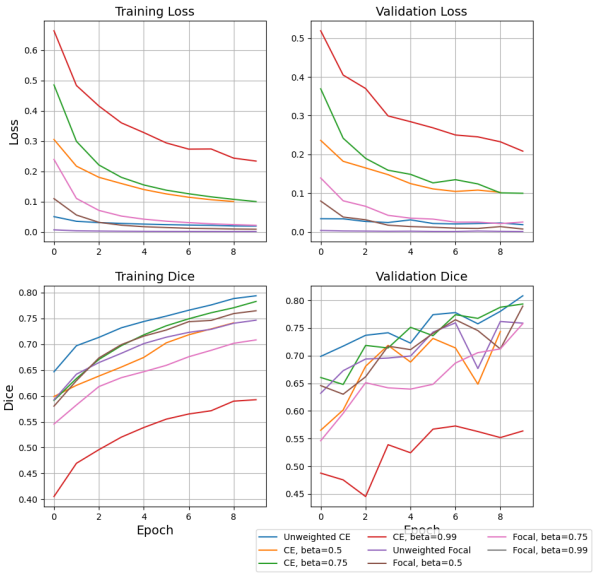

Figure 5. TransUNet with Different Losses

est loss among focal loss tests during training suggests that tuning γ could result in positive results.

Next, we explored the effect of different learning rates and batch sizes on test Dice. We, again, only explored two learning rates using the Adam optimizer, 1e-4 and 1e-5. Contrary to the small changes in test Dice using different learning rates, we found significant changes in test Dice for TransUNet. Where a lower learning rate caused our standard U-Net to produce worse predicted segmentations, a lower learning rate actually helped TransUNet significantly, leading it to outperform the baseline U-Net. This is the only model to have outperformed the baseline U-Net. The model may be benefiting from the smaller learning rate as a result of self-attention mechanisms that may be more sensitive to changes in global information than the basic encoder in a U-Net.

Similarly, for batch size, we explored the effects of batch sizes of 1 and 16 on test Dice. For a batch size of 1, decreasing the learning rate demonstrated significant improvements

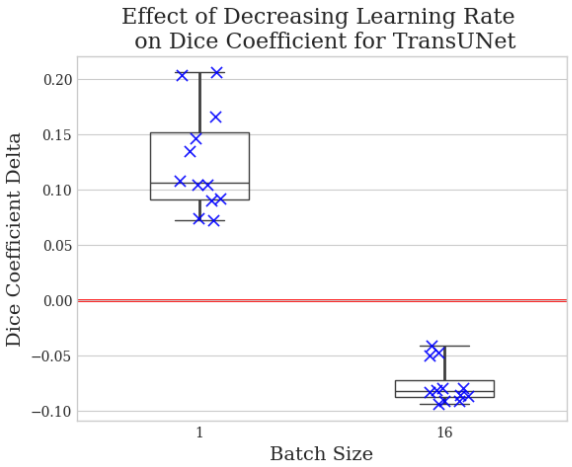

Figure 6. Learning Rate on TransUNet Performance Compared to Base Case Across Batch Sizes (Dice Coefficient)

in model performance with all other hyperparameters fixed. However, for a batch size of 16, performance degraded as the learning rate was decreased from 1e-4 to 1e-5. It has been demonstrated that models trained with a small batch size tend to generalize better due to increased gradient noise [6]. Decreasing the learning rate further decreases the gradient noise, which deteriorated the model's ability to optimize. Other works have demonstrated that learning rate should scale proportionally to batch size in order to preserve gradient noise [4]. With this in mind, increasing the learning rate when the batch size is 16 may have yielded results comparable to runs with a batch size of 1.

The results of the learning rate experiment are shown in Figure 6. The optimal learning rate and batch size found was 1e-5 and 1, respectively. The increase in test Dice for the unweighted cross-entropy and unweighted focal loss models at a batch size of 1 are shown in Table 2.

4. Conclusions

Ultimately, we were able to achieve better segmentations with TransUNet than with U-Net, though not by a significant margin. However, the experimentation that led to these results had a few issues. Among these gaps in the methodology, the team did not tune focusing parameter γ for any of the focal loss experiments, instead opting to keep it at $\gamma = 1$. Our learning rate exploration was limited to only two options, though they were separated by an order of magnitude. Consequently, it remains unknown what the effect of a higher learning rate would be on any of these models. In the name of expediency, we also truncated all of our experiments at 10 epochs, though after reviewing the learning curves, it seems entirely possible that longer training could have resulted in even better-performing models, so we may have hamstrung all of these models.

Despite these gaps, the fact that a TransUNet with a lower learning rate was able to outperform the baseline U-Net is noteworthy. This jump in performance, despite the lower learning rate, may be a result of Transformer blocks maintaining a higher level of feature resolution. Instead of decimating information via max pooling layers, the Transformer self-attention mechanisms may be better at incorporating global information from the images, requiring smaller gradient updates to find good local optima. Conversely, a higher learning rate for TransUNet may actually be causing it to overshoot these local optima, possibly even creating a smoothing effect during the gradient updates that inhibit model generalization.

Given that biomedical image segmentation is a field with high potential utility for healthcare providers around the globe, further survey studies should be conducted comparing the different options available for segmentation, as well as the effects of different loss functions. While certain studies already focus on different loss functions specifically for segmentation, there appears to be insufficient focus on the different architectures available, and specifically how they perform on key biomedical benchmark datasets.

As for our own analysis, any continuation of this work should begin with a wider hyperparameter exploration, with an emphasis on loss-specific hyperparameters like γ, and an introduction of more possible loss functions, and longer training times, as 10 epochs may be insufficient to properly analyze the final performance of these models.

5. Work Division

All members of the team contributed to the completion of the project, including contributing to completing the project objectives and writing the report.

Adriel Bustamante is responsible for creating a wrapper around the BraTS dataset to facilitate use of the Kaggle dataset. He also wrote the U-Net implementation, and implemented the cross-entropy and focal loss functions. Adriel also provided analysis relating to the different hyperparameters in the U-Net and their effects on model performance.

Gabriel O'Hara is responsible for the idea and implementation of pruning the images to reduce the severity of class imbalance. This also made training faster, resulting in productivity improvements for all team members. Gabriel also integrated Tensorboard into the training scripts, facilitating easier analysis of the models. He also contributed to the analysis of effects of different hyperparamters on TransUNet.

Jonathan Pang was responsible for the hyperparameter tuning scheme, including exploring the hyperparameter space for both U-Net and TransUNet models. He also provided analysis of the different hyperparameters and their effects on the model performance.

Brooks Roney implemented the TransUNet model and performed initial training and evaluation of the model. He also wrote the background and approach sections of this report.

Work division is summarized in Table 3.

5.1. Code

All code was worked on collaboratively and stored via Georgia Tech's internal GitHub. Link to repo here: https://github.gatech.edu/abustamante31/cs7643-final-project

References

[1] Reza Azad, Moein Heidary, Kadir Yilmaz, Michael Hüttemann, Sanaz Karimijafarbigloo, Yuli Wu, Anke Schmeink, and Dorit Merhof. Loss functions in the era of semantic segmentation: A survey and outlook, 2023. 3

[2] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation, 2021. 1, 2, 4

[3] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples, 2019. 2, 3

[4] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017. 5

[5] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, page 1–7. IEEE, Oct. 2020. 3

[6] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. 5

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 2, 3

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 2, 3

[9] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation, 2016. 3

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1, 2

Student Name	Contributed Aspects	Details
Adriel Bustamante	Implementation and Analysis	Implemented original U-Net, training loop, dataset wrapper around BraTS data, loss functions, provided analysis for different loss functions
Gabriel O'Hara	Data Cleaning and Implementation	Implemented data cleaning functionality to improve training time, training performance, as well as Tensorboard integrations
Jonathan Pang	Implementation, Training, and Analysis	Implemented hyperparameter exploration scheme, trained U-Nets and TransUNets, provided analysis for different hyperparameters
Brooks Roney	Implementation and Writeup	Implemented TransUNet, trained sample TransUNets for initial evaluation, wrote initial sections of report discussing background and approach

Table 3. Contributions of team members.