A Comparison Between Dice Loss and Tversky Loss on Training a 3D U-Net Model to Detect Tumor Tissue in Brain MRI Scans

Dov Winkleman

Hunter College MA in Statistics and Applied Mathematics

Fall 2021

Advisor: Dr. Iordan Slavov

INTRODUCTION

The detection of tumor tissue in imaging scans is an important task in the medical health field. Early detection is especially important, as it may help prevent any tumor metastasis, save lives, and at the very least reduce treatment time or intensity for the patient. However, Radiologists sometimes differ in their interpretations of what is or is not tumor tissue in imaging scans. And in certain situations, an experienced, specialized Radiologist might not be available to interpret the scans. In these and other situations, a well-trained machine learning model can be a valuable resource. In this paper, a 3D U-Net CNN model was trained with two different loss functions to determine which one performs better at detecting tumor tissue, necrotic tissue, or peritumoral edema in 3D brain MRI scans.

There are several notable difficulties that are specific to training a machine learning model on 3D medical imaging data. One regards the loss functions used for the model's gradient descent. Since most of the volume in an MRI brain scan is not tumor tissue, the ground truth categories would be highly skewed towards the background category, and using a typical binary or categorical crossentropy would not be efficient. A model could achieve very high accuracy on the training and validation sets by easily and correctly predicting which tissues are not tumor. Since we are more interested in predicting which tissues are tumor, this method would be futile.

Another difficulty is in acquiring enough data to sufficiently train a model. Machine learning models require large amounts of data for training, and obtaining brain MRI images with ground truths properly labeled by a professional Radiologist can be difficult and expensive. A third difficulty lies in the size of 3D MRI image data. The file sizes can be quite large and keeping it all in computer memory at once to train the model might not even be possible due to memory size. For this reason, a generator function is used, which generates data in batches.

DATA

The brain MRI data used for this study was taken from the BraTS2020 brain tumor segmentation challenge. The files were in niftii format, taking up approximately 33 GB of data, and contained MRI images from 368 patients, with each patient receiving T1, T2, T1-CE (contrast enhanced with gadolinium) and T2-FLAIR (Fluid Attenuated Inversion Recovery) imaging modalities (CBICA, 2020). These images were reviewed and given ground truth labels by professional board-certified Neuroradiologists, who labeled the necessary tissues as either tumor, necrotic, or edema. The images were acquired across 19 different institutions, using various scanners and clinical protocols. The images then went through a degree of initial pre-processing where they were co-registered to the same anatomical template, interpolated to the same resolution (1 mm^3) and skull-stripped (all non-brain tissue, like the skull, removed).

Scan Types

T1-weighted MRI sequences are defined by their short signal relaxation times (TR and TE) (Murphy, 2020). For brain scans, this means that any fluid (i.e., CSF) has a low signal

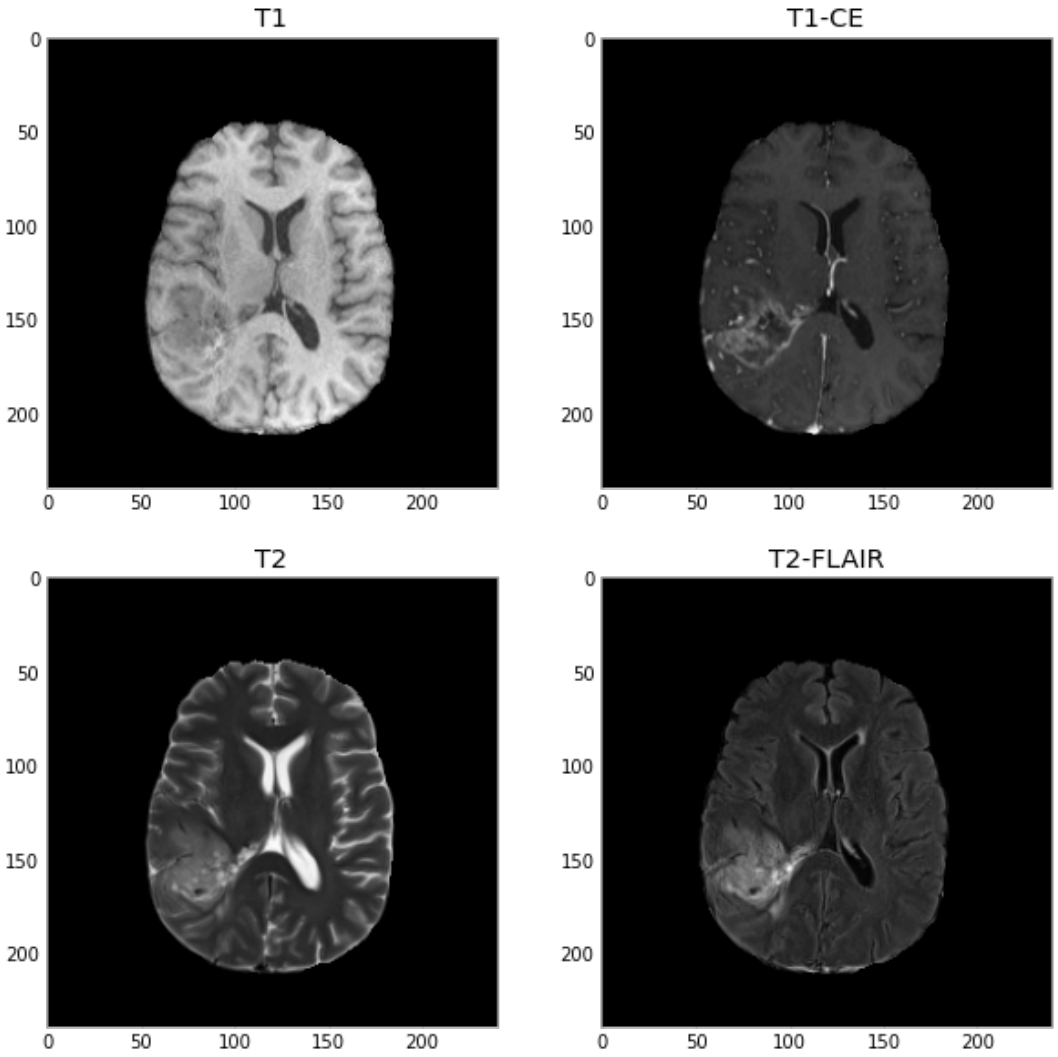


Figure 1: The 4 types of MRI brain scans acquired for a single patient.

intensity, grey matter has an intermediate signal intensity, and white matter's intensity is a little higher than grey matter.

Contrast enhanced T1-CE scans differ from regular T1 scans by the addition of gadolinium enhancement to the blood stream. This causes tumors and areas of inflammation to display a higher signal intensity.

T2 scans are characterized by longer signal relaxation times than T1 scans. Fluids have a higher signal intensity, white matter appears darker, while grey matter appears slightly lighter than white matter. Pathological tissues, like tumor tissues, which have more liquid than regular brain tissue, appear brighter. (Center for Functional MRI, n.d.).

T2-FLAIR scans attenuate any signal from brain fluids in T2 scans. This helps detect any edema that may be surrounding the tumor.

Tissue Classifications

There are a maximum of three types of tumor or tumor related tissues that were classified by expert Radiologists for each MRI scan. These include tumor tissue, the edema (swelling) that might surround the tumor, and necrotic (dead) tissue, usually found inside a tumor. The ground truths for the above patient can be seen in Figure 2.

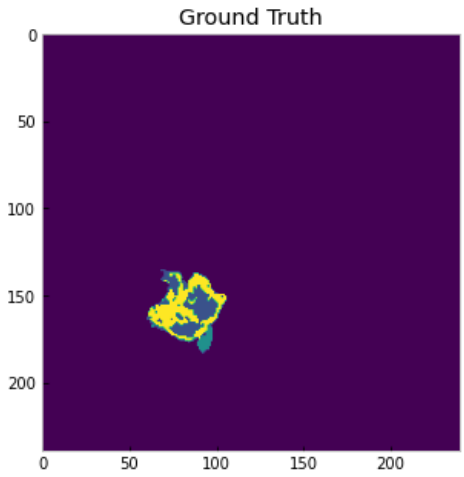


Figure 2: Ground Truth labels from above patient. Green = edema, yellow = tumor, blue = necrosis.

Training, Validation, and Test Sets

Of the 368 total patients, 20% (74) were put aside to use as a test dataset. Of the remaining 294, 20% of those (59) were put aside as a validation dataset, leaving 235 patients for the training set. After creating 5 patches per patient of the brain volumes with the percentage of non-tumor data below a certain threshold (see Methods, below), there remained 190 patients in the training set, 47 in the validation set, and 64 in the test set.

METHODS

Preprocessing

After the niftii files were imported into Python, they were first corrected for bias field fluctuations in a process called Bias Field Correction (BFC). This step is required for machine learning analysis of MRI images. Intensities in the tissues in MRI images may differ due to magnetic field fluctuations, rather than tissue variations. An example of this would be if some grey matter in the frontal cortex were displayed with the same pixel intensity as white matter in a different portion of the brain, when in fact the intensities are different. This random fluctuation in intensities would affect model training and cause accuracy to suffer. BFC corrects these imperfections. (BrainSuite, 2016). The BFC algorithm used N4 normalization, and utilized the Otsu threshold filter, since no mask region was specified.

Following Bias Field Correction, the images were converted to the H5py format. This format allows large data files, in this case each subject's 4 imaging modalities and the ground truth file, to be stored in one single file and accessed like a NumPy array. This was crucial for the data generator function to perform efficiently. During this conversion process, the first 5 and last 15 slices of each scan were removed, as they contained no pertinent information.

For a model to extract information in 3 dimensions, it must be fed whole 3D volumes of data at a time, rather than just 2 dimensional slices. Since these file sizes are very large and an entire volume would need to fit into memory all at once for proper training, smaller sub-volumes were extracted to facilitate this. Five different 3D patches, each containing less than 96% non-tumor/edema/necrosis area, were extracted for each patient. This patch extraction function worked by randomly choosing a voxel near a corner, choosing a 3D sub-volume based on this

corner value with the dimensions 180x160x24 (the third value being the number of slices), and if the non-tumor area was less than the threshold level, save this as a sub-volume. This process was looped 1000 times until 5 sub-volumes with adequate brain tissue were created for each subject, all with different initial corner voxels. Sixty-seven patients did not have enough tumor/edema/necrosis area large enough to meet the cutoff, and these patients were not included in the study. Standardization of the data, where the mean pixel value was set to 0 and the standard deviation of the pixel intensity was set to 1, was also conducted during this process.

Loss Functions

Two different loss functions were used for training. 'Soft' Dice Loss was first used. Dice Loss compares the intersection between the ground truth (p) and the model's predicted value (p-hat) with the total of those values (Jadon, 2020). If the model makes a good prediction, then the intersection will be large, and 2 times the intersection will be close to the ground truth plus the prediction. Subtracting this from 1 will cause the loss to move closer to 0 as the predictions become better, which is necessary for gradient descent. 'Soft' Dice Loss differs from 'regular' Dice Loss by the fact that the values in the denominator are squared, allowing the function to be continuous. Without squaring the denominator values, the function contains asymptotes approaching positive and negative infinity.

$$SoftDiceLoss = 1 - \frac{2p\hat{p} + (laplace\ smoothing\ factor)}{p^2 + \hat{p}^2 + (laplace\ smoothing\ factor)}$$

$$TverskyLoss = 1 - \frac{TP + (laplace\ smoothing\ factor)}{TP + \alpha FN + (1-\alpha)FP + (laplace\ smoothing\ factor)}$$

The second loss function used was Tversky Loss. This looks at the relationship between true positives (TP) in the numerator, and true positives plus false negatives (FN) plus false positives (FP) in the denominator. The weights of the FNs and FPs are attenuated by a constant, α. The closer α is to 1, the more weight is given to false negatives, and the closer it is to zero, the more weight is given to false positives. For brain imaging data, a false positive is an instance where the model predicts tumor tissue where there is none, and a false negative is where the model fails to predict tumor tissue where in fact it exists. Ideally, when attempting to predict tumor tissue in MRIs, one would want to give more weight to false negatives. It would seem to be better to wrongly assume that some tissue is a tumor, than to falsely claim that tumorous tissue is healthy. But for training, the optimal α value was found to be 0.05, which would indicate that drastically reducing the weight given to the false negatives increases model accuracy. This value was determined by training models on only 10 epochs of data and assessing their score on the test set. While evaluating the results based on the test set score would likely introduce bias, this method was used due to the wild fluctuations of the validation set scores. An α value of 0.01 scored very well on the test set but did not appear to be trainable, so it was not used. Its training and validation loss per epoch were close to linear.

Additionally, several metrics were used to evaluate the model for each loss function, including the Dice coefficient, Dice coefficient specific to each tumor tissue classification, precision, and Tversky Loss.

α:	0.01	0.03	**0.05**	0.1	0.3	0.7	0.9	0.95
Test set Loss:	0.3	0.42	**0.41**	0.48	0.63	0.58	0.55	0.76

Table 1: Test runs (10 epochs each) to determine optimal Tversky α value. As noted earlier, alpha=0.01 was not used due to the model's lack of trainability with that value.

MODEL

A 3D CNN 'U-Net' model was used to train the data. Where a typical CNN might be interested in image classification, a U-Net is also interested in localizing structures in images and distinguishing borders (Zhang, 2019). The model is symmetric and has a contraction path and an expansion path, with two concatenation paths connecting various parts across each of these paths. This model is illustrated in Figure 3 and Appendix A. The contraction path consists of a group of two 3D Convolution layers, each followed by a Batch Normalization layer. This first grouping feeds into a Max Pooling layer for down-sampling, and also feeds into a concatenation path that leads to the final output grouping. The Max Pooling layer feeds into two more 3D Convolution layers that are also followed by Batch Norm layers. This group feeds into the second and final Max Pooling layer, while also feeding into a concatenation path that leads to the first up-sampling layer. This final Max Pooling layer feeds into two consecutive 3D Convolution layers (both followed by Batch Norm), which feed into the first Up-Sampling layer. This layer feeds into two 3D Conv layers (each followed by Batch Norm layers), into another Up-Sampling layer, into two more 3D Conv layers (each also followed by Batch Norm layers), and finally leads to the output of the model. The Adam optimizer was used, with a learning rate of 0.001.

The model was trained with each loss function for 40 epochs using a GPU in Google Colab Pro Plus. Training took 10 hours for Soft Dice Loss and 11 hours for Tversky Loss.

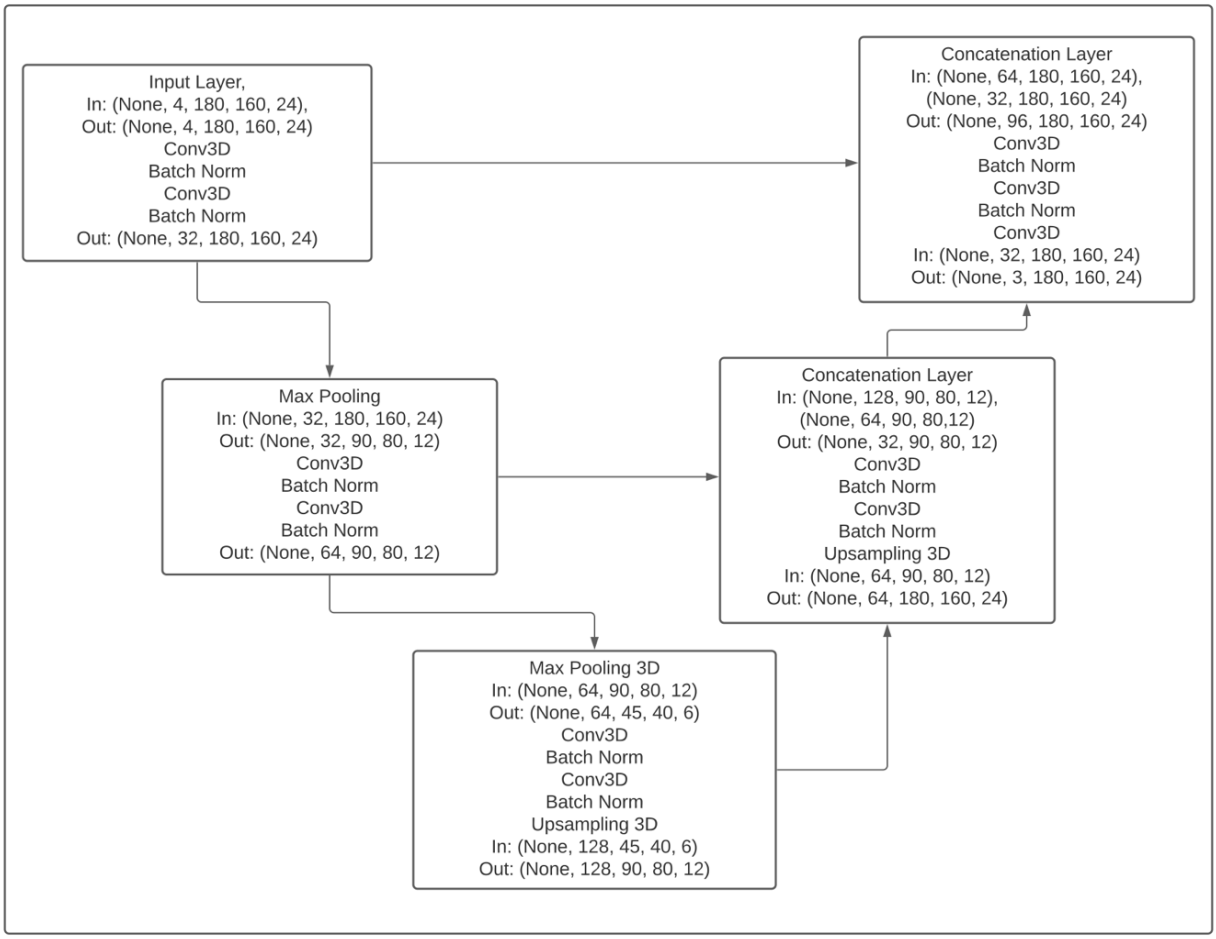


Figure 3: 3D U-Net model flowchart

RESULTS

The results from training are summarized in Table 2 and Figure 4:

Loss Function:	Epochs	Training time	Test set Loss	Dice Coefficient	Dice edema	Dice tumor	Dice necrotic	Precision	Tversky Loss ($\alpha = 0.05$)
Soft Dice Loss	40	10 hrs	**0.34**	**0.56**	**0.58**	**0.48**	**0.38**	**0.71**	0.43
Tversky Loss ($\alpha = 0.05$)	40	11 hrs	0.40	0.47	0.49	0.46	0.26	0.38	**0.40**

Table 2: Results from both models on the test set. Note that lower values represent better performance for loss functions, but higher values represent better performance for all other metrics.

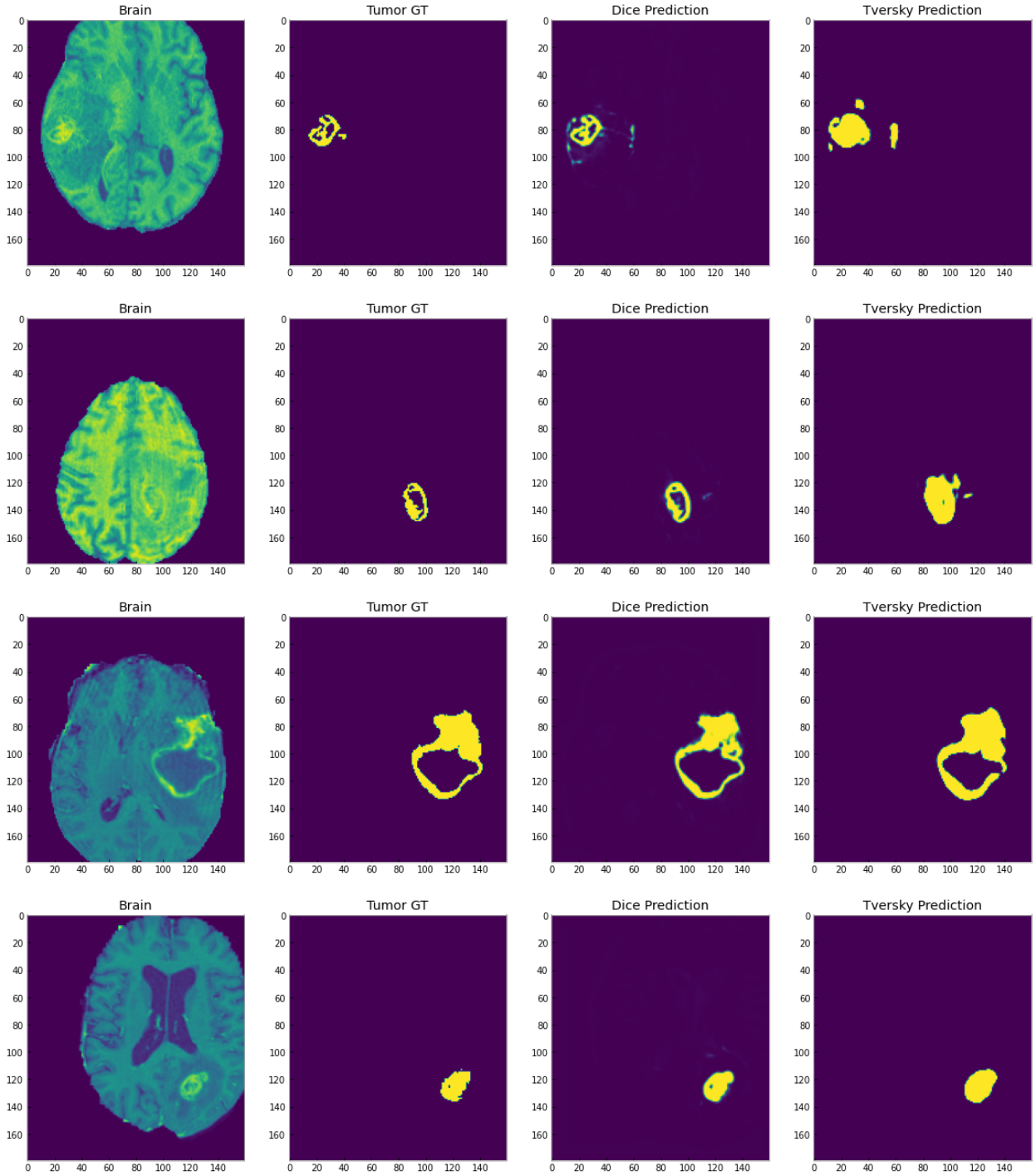
Figure 4: Validation/Training loss per epoch for each model.

As can be seen in Table 2, Soft Dice Loss performed better than Tversky Loss on the test dataset and on all metrics, except Tversky Loss. Most notably, test set loss was 0.34 for Soft Dice Loss, while it was 0.4 for Tversky Loss. Precision was almost twice as good for Dice as it was for Tversky.

Figure 4 shows that Tversky Loss plateaued quicker than Dice Loss, near the 25th epoch mark, with a loss of around 0.4 on the Validation set. Soft Dice training loss was still reducing at 40 epochs, but the Validation loss appeared to plateau around a loss value of 0.35.

CONCLUSION

It is clear that Soft Dice Loss performed better on this 3D U-Net model than Tversky Loss did. Upon analysis of specific tissue predictions by this model for each loss function (figures 5-8), it would appear that Tversky is over-estimating the tumor area and underestimating necrosis and edema. This is consistent with the data in Table 2, where the Dice Coefficient value for tumor tissue between the two loss functions is close, but Dice Loss performs better for edema and necrosis. This is also consistent with the alpha value, which gave much higher weight to false positives. It also explains the difference in precision (TP / (TP + FP)) between the two loss functions. By increasing the false positives, the denominator of the precision equation also increases, which decreases the precision value.



Figures 5-8: Examples of Predicted tumor tissue

One possible reason for Tversky Loss's worse performance is that the Tversky Loss function is not continuous. A literature search did not yield any results for a continuous version of the function. An attempt was made to code a continuous version, but time limits prevented its completion. A continuous function with a global minimum is an ideal function for gradient descent in order to prevent exploding or vanishing gradients. Since the Tversky function is a

reciprocal function, it is noncontinuous, and has asymptotes at y = +/- infinity and x = +/- infinity.

There are several improvements that can be considered for this study. The first would be an improved method of customizing the alpha value for Tversky loss. The false positive rate was clearly too high to make accurate enough tumor tissue predictions. More time spent customizing other model parameters, the acquisition of more data, experimentation with transfer learning datasets, additional loss functions to use for comparison, and more experimentation with current loss functions are some other improvement options that should be explored. For instance, it might be interesting to code the loss functions to have gradient descent be calculated only on the tumor data, and not the edema or necrosis data.

REFERENCES

Anshik (2021). *AI for Healthcare with Keras and Tensorflow 2.0*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7086-8

BrainSuite. (2016, October 21). *Bias Field Correction.* Retrieved December 22, 2021, from http://brainsuite.org/processing/surfaceextraction/bfc/

CBICA, Perelman School of Medicine at the University of Pennsylvania (2020). *Multimodal Brain Tumor Segmentation Challenge 2020: Data*. Retrieved December 22, 2021, from https://www.med.upenn.edu/cbica/brats2020/data.html

Center for Functional MRI - UC San Diego (n.d.). *Structural MRI Imaging*. Retrieved December 22, 2021, from http://fmri.ucsd.edu/Howto/3T/structure.html

Jadon, S. (2020, October*). A survey of loss functions for semantic segmentation*. arXiv, https://arxiv.org/abs/2006.14822

Murphy, A. (2020, April 2). *MRI sequences (Overview)*: Radiology reference article. Radiopaedia Blog RSS. Retrieved December 22, 2021, from https://radiopaedia.org/articles/mri-sequences-overview?lang=us

RNA (pseydonym). (2021, July 18). *Loss function library - keras & pytorch*. Kaggle. Retrieved December 23, 2021, from https://www.kaggle.com/bigironsphere/loss-function-library-keras-pytorch

Zhang, J. (2019, October 18). *UNet line by line explanation*. Medium. Retrieved December 22, 2021, from https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5

BraTS2020 Dataset Citations

Bakas, S. Akbari, H. Sotiras, A. Bilello, M. Rozycki, M. Kirby, J.S. et al. (2017). *Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features*. Nature Scientific Data, 4:170117 (2017) DOI: 10.1038/sdata.2017.117

Bakas, S. Akbari, H. Sotiras, A. Bilello, M. Rozycki, M. Kirby J., et al. (2017). *Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection*, The Cancer Imaging Archive, 2017. DOI: 10.7937/K9/TCIA.2017.KLXWJJ1Q

Bakas, S. Akbari, H. Sotiras, A. Bilello, M. Rozycki, M. Kirby J., et al. (2017). *Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-LGG collection*, The Cancer Imaging Archive, 2017. DOI: 10.7937/K9/TCIA.2017.GJQ7R0EF

Bakas, S. Reyes, M. Jakab, A. Bauer, S. Rempfler, M. Crimi, A. et al. (2018). *Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge*. arXiv, https://arxiv.org/abs/1811.02629

Menze, B. H., Jakab , A. Bauer, S. Kalpathy-Cramer, J. Farahani, K. Kirby, J. et al. (2015). *The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS).* IEEE Transactions on Medical Imaging 34(10), 1993-2024 (2015) DOI: 10.1109/TMI.2014.2377694

Appendix A: U-Net Model architecture