# Brain Tumor Segmentation

Bora Erden, Noah Gamboa, and Sam Wood

## BACKGROUND

We set out to build a convolutional neural network to classify tumors and tumor subsections in MRI brain images. Medical image analysis is a very important field, and we believe that computer algorithms have the potential to reproduce or even improve upon the accuracy of human experts. Using algorithms to automate medical image analysis could save time and money for hospitals and patients, and improved accuracy would be a great benefit to cancer patients.

## PROBLEM STATEMENT

Given an input of an MRI brain volume, our neural network outputs a semantic segmentation of the volume that separates the tumor from the rest of the brain. We produce this segmentation using a deep down-sampling-up-sampling network of 3D convolutions. We evaluate the segmentation produced by our network using the DICE coefficient between the predicted segmentation and the ground-truth label. The Dice coefficient between two sets *X* and *Y* is calculated as:

$$Dice(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

## DATASET

Our dataset consists of around 500 brain volumes, each consisting of 155 two-dimensional slices. Every brain volume in the dataset contains a glioma, a common type of brain tumor. The brain volumes are clinically-acquired MRI scans, labelled manually by neuro-radiologists. In addition to demarcating the tumor, the labels divide the tumor into four subsections: edema, non-enhancing core, necrotic core, and enhancing core. The dataset was provided by the 2017 BraTS Multimodal Brain Tumor Segmentation Challenge. An example brain slice and associated label is shown below.

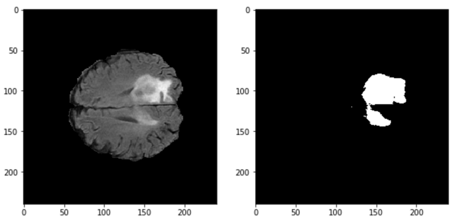

Figure 1: An example slice from a brain volume, at left, with its ground-truth tumor segmentation.

## METHODS

We use a fully convolutional neural network. After running the input brain volume through a series of down-sampling max-pooling convolutional layers, we feed the data through a series of up-sampling transpose convolutional layers (See Fig. 2). The final output of the network is the same shape as the input, but each "pixel" of the output, rather than containing visual information, contains the unscaled probability that the corresponding pixel in the input belongs to the tumor.

We experimented with both two-dimensional and three-dimensional convolutional architectures. Since the inputs to our network are two-dimensional brain slices that make up connected three-dimensional volumes, we assumed that three-dimensional convolutions should be able to achieve better results. However, the memory demands of three-dimensional convolutions forced us to train on smaller batches and sacrifice some width and depth from our network.

While we assumed that using the Dice coefficient directly as our objective would give the best results, we also experimented with optimizing against cross-entropy loss and against sensitivity-specificity loss (which weights sensitivity, the percentage of correctly-identified tumor pixels, much more highly than specificity, the percentage of correctly-identified non-tumor pixels, in an attempt to compensate for the scarcity of tumor pixels).

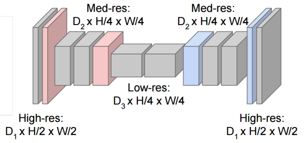

Figure 2: While this does not represent the exact architecture of our model, it demonstrates the principles of a down-sampling-up-sampling convolutional network.

## LOSS FUNCTIONS

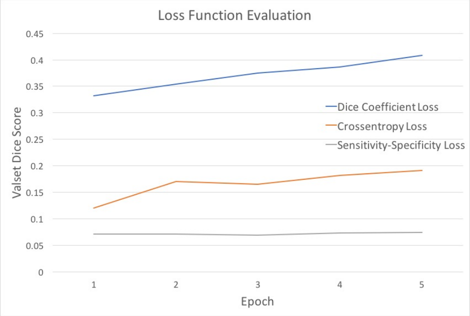

Figure 3: As expected, using the Dice coefficient as the objective yielded the best results.

## HYPERPARAMETERS

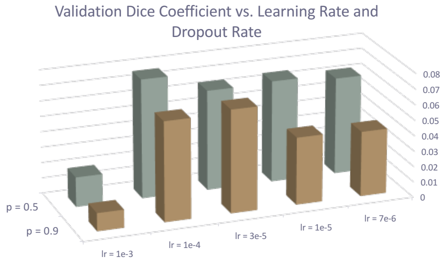

Figure 4: A dropout rate of 0.5 and high learning rate of 1e-4 led to the best validation set dice coefficient after one epoch of training

## MODEL ARCHITECTURE

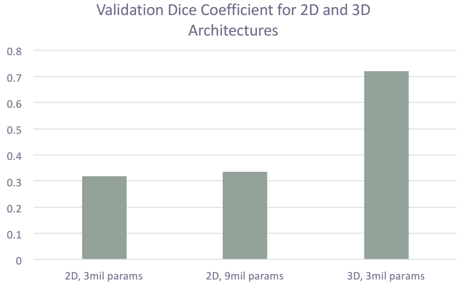

Figure 5: We compared a 3D net to various 2D architectures. The 3D net significantly outperformed the 2D nets, even though it had access to far fewer trainable parameters, due to memory constraints.

## CONCLUSIONS

As predicted, optimizing against the Dice coefficient objective best allows us to maximize the Dice coefficients for our model's predictions.

We found that a 3D-convolutional architecture significantly outperformed our 2D-architectures, even when the 2D nets were significantly wider and deeper.

Moving forward, we plan to investigate res-net style architectural choices that would allow us to achieve increased accuracy using our 3D model, despite the width limitations we face as a result of the memory demands of 3D convolutions. For example, some papers have suggested concatenating the outputs of earlier down-sampling convolutions to the outputs of later up-sampling convolutions. We will investigate the effect of this strategy on the success of our model.

We also plan to take advantage of the subsection segmentation data, perhaps using a vote among an ensemble of models for multiclass segmentation.