Deep Learning for Brain Tumor Segmentation

Neel Rao and Charles Tsao
Department of Computer Science
Stanford University
(neelvrao and chtsao)@stanford.edu

Abstract
In this project, we wanted to build a tool to segment MRI brain scans into four parts: non-tumor, whole tumor, tumor core, and enhancing tumor core. We started by taking a pre-existing model, which attempted to solve the problem by chaining together three cascaded neural networks, each doing one step of the segmentation. We ran two experiments on this model. First, we added an extra layer to each neural network. Secondly, we experimented with the model performance by using a weighted cross-entropy loss function instead of the dice loss function that was originally used. Neither experiment resulted in higher performance.

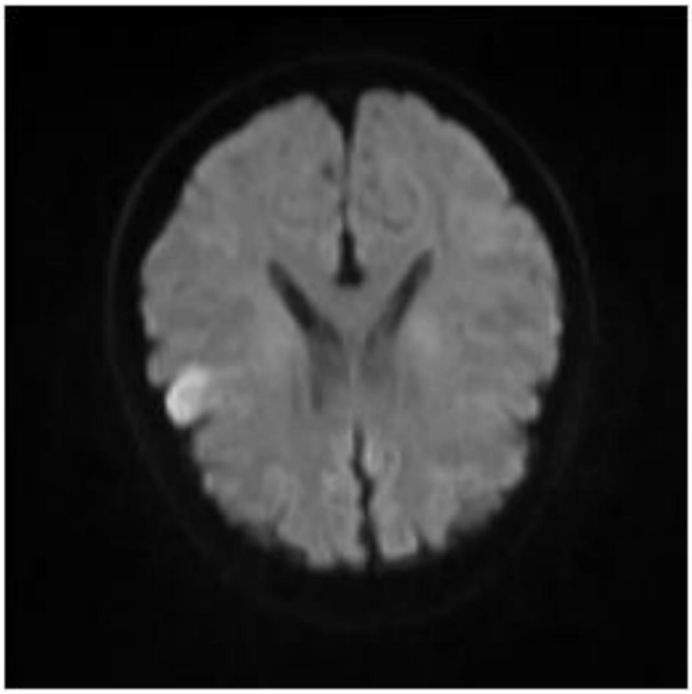
1. Introduction
Brain tumor segmentation is an important task in the modern healthcare world. Tumor segmentation provides doctors with the level of specificity they need to properly treat patients. Trained radiologists usually do this task. In our project, we wanted to build a tool that could do the same work with a high degree of accuracy, to hopefully provide a blueprint for a radiologist to then double check, expediting the process and making the task more efficient. We started with a pre-trained model built for this task, and then augmented it to try to improve performance. The model consists of three cascaded neural networks. The first one takes as input multi-modal 3D volumes of a patient's brain, and uses a series of Convolutional Neural Networks (CNNs) to output a mask of the whole tumor. The second network inputs a cropped region around the tumor and uses CNNs to output a mask of the tumor core. The third network inputs a cropped region around the tumor core, and uses CNNs to output a mask of the segmented enhancing tumor core. In training, the cropped regions are taken from the ground truth data, but in testing the cropped regions are set based on the predictions of the previous network.

2. Related Work
The literature is extensive on brain tumor segmentation, due in part to the popularity of artificial intelligence for healthcare as a research topic, as well as the prevalence of large-scale data science competitions with these topics. There are a wide variety of ways people have attempted to build models. The majority of current approaches are either generative or discriminative. The generative approaches show the probabilistic distributions of tumor and healthy tissue appearances, and are good at generalizing to unseen images. They rely on previous knowledge of what a healthy brain looks like and apply that to tumorous brains to spot the differences. This type of approach is outlined in "A Generative Model for Brain Tumor Segmentation in Multi-Modal Images" (Menze et. al). Discriminative approaches, on the other hand, try to learn the relationship between image

intensities and tissue classes. These models directly learn differences between lesions and other tissues, rather than relying on spatial priors. Discriminative models have done this through the extraction of low-level image features, which includes raw pixel values (Havaei et al.), Gabor filterbanks (Subbanna et al.) or alignment-based features (N.Tustison and Avants, 2013). However, these models generally require large amounts of data. Recently, discriminative models have become the state-of-the-art standards. One of the most well-known models in this space is the U-Net, which uses convolutions similar to the model we chose (Abdulkadir, et al.). However, U-Net inputs 3D images, whereas the model we used inputted 2D images, and trained each view (sagittal, axial, coronal), separately, before putting them all together in the end. This strategy allows for a lot of memory to be saved, as well as speeding up the testing.

3. Dataset and Features

We used a dataset from the 2017 Tumor Segmentation (BraTS) Challenges, held at the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI). The training dataset consisted of 285 3D multimodal brain scans that contained either low-grade gliomas (LGG) or high-grade gliomas (HGG). Out of the 285 brains, 210 contained HGGs and 75 contained LGGSs. We then tested the model on a validation set of 45 brains, also provided by the BraTS challenge. Each brain scan consisted of five slices each from the coronal, sagittal, and axial views. The 2D receptive fields for the first, second, and third networks are 217×217, 217×217, and 113×113, respectively.

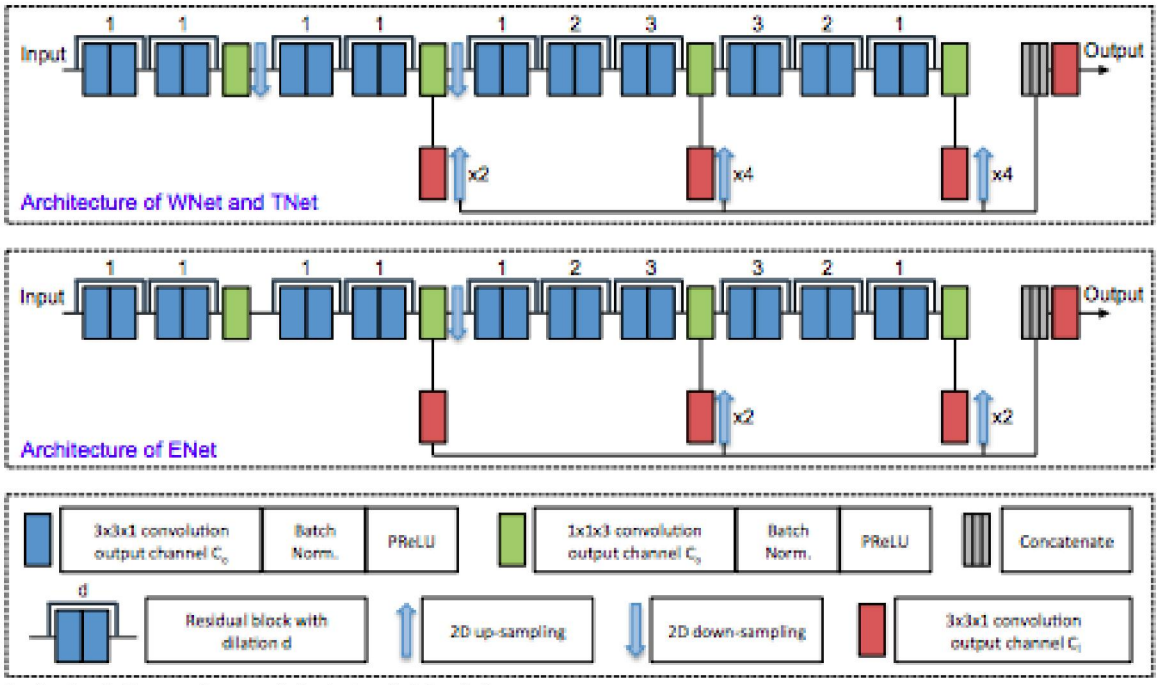


Example of a slice from axial view

4. Methods

The first and second neural networks used four sequential layers, each with two blocks made up of two 3x3x1 convolutional layers, a Batch Norm layer, and a PReLU layer, followed by a 1x1x3 convolution output channel and a 2D downsampling layer. After the first chunk, each convolution output layer was followed by a 3x3x1 convolution output channel with 2D upsampling. The third network uses the same architecture, but with less upsampling, because of the smaller input size(just a tumor core, not an entire brain). This all can be seen in the picture below. This model makes use of residual connections, an architecture that makes the image more manageable to learn from by lowering the resolution of an image as it is passed through the layer, and then raising it back up, rather than crunching the image to a lower size.

As background, convolutional neural networks use convolutions to extract features from images. Convolutions are grids that weight pixels differently based on their placement. CNNs pass the convolution over the input, multiplying the convolution by the input step by step, outputting a grid with a size dependent on the size and stride of the convolution. They usually identify lines at first, and as the image progresses through the network, can identify more complex features. Batch Norm normalizes each input, using trained parameters to determine how much normalization is needed. PReLu is an output function, similar to Leaky ReLu, except that the coefficient of leakage is itself a trained parameter.



Labeled Overview of Original Model

In our added layer experiment, we changed every networks size to be five identical blocks, not four, as stated before. In our second experiment, we changed our loss

function from a dice loss function:

$$\mathrm{DL}_2 = 1 - \frac{\sum_{n=1}^{N} p_n r_n + \epsilon}{\sum_{n=1}^{N} p_n + r_n + \epsilon} - \frac{\sum_{n=1}^{N} (1 - p_n)(1 - r_n) + \epsilon}{\sum_{n=1}^{N} 2 - p_n - r_n + \epsilon}$$

to a Weighted Cross Entropy loss function.

$$\mathrm{WCE} = -\frac{1}{N} \sum_{n=1}^{N} w r_n \log(p_n) + (1 - r_n) \log(1 - p_n),$$

5. Experiments/Results/Discussion

Our first experiment was adding an extra layer to each of the neural networks. We expected this to help with the later stages of the segmentation, the tumor core and enhancing tumor core segments, because the extra layer would theoretically be able to pick up more complex features in the images it was inputted. However, this was not the case. The new model actually did much worse on these two tasks, relative to the original model 9as seen in appendix). We believe this is because the extra layer ended up forcing the model to overfit to the training set. The new model had a higher training accuracy, but a lower test accuracy.

Next, we ran an experiment to see whether a change in loss function would help the model. When we trained with dice coefficient loss, the model had a tough time converging, so we decided to try to train with weighted cross entropy loss, which we know is easier to backpropagate. We also lowered the learning rate and raised the learning rate decay, to help the backpropagation converge. While we were right, and the gradient converged much faster, training with the new loss function ended up leading to worse results (as seen in appendix). We are not precisely sure what the reason was for the change in performance. We know that dice loss performs better on class imbalanced datasets, but ours is not class imbalanced, because all of the brains in our training set have tumors. Therefore, we believe that the results may have happened by chance, and not specifically motivated by a reason we can diagnose. Indeed, the qualitative results look roughly similar to the naked eye.

6. Conclusion/Future Work

We took a pre-existing deep learning model for brain tumor segmentation, and ran some experiments on it to see if we could improve its performance. We added an extra layer to each of the neural network, and we changed the loss function used in training, as well as the learning rate and learning decay rate. Neither experiment resulted in better performance. For the added layer model, we believe this is because the extra layer caused the model to overfit to the training set. If we had more time, we would try to attach a fourth neural network to the output of the model, and use it to try to predict patient survival rates from tumor scans and some other key variables, like patient age. When we tried it here, we were unable to get results better than random. We think that as the BraTS competition progresses, and more data is released, we might be able to be more successful at this last task.

Contributions

Both teammates contributed to the planning, design, and research components of the project, thinking through what exactly we wanted to do and how to execute it. Charles handled the coding, training and testing of the project, whereas Neel handled the poster creation and wrote the final report, as well as most of the project milestone and project proposal.

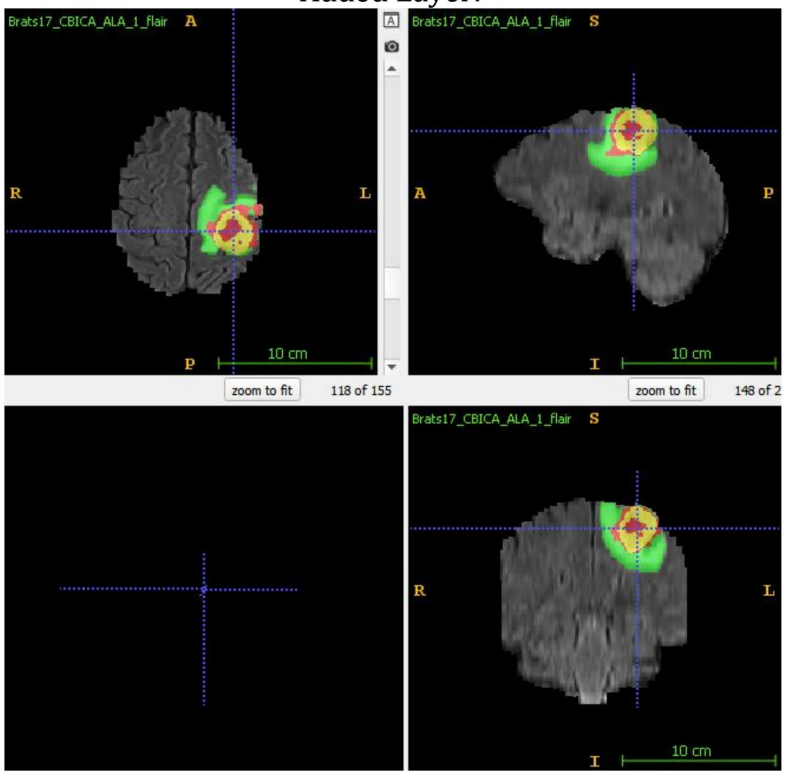
Appendix/Figures

Mean Accuracy Benchmarks for Each Model

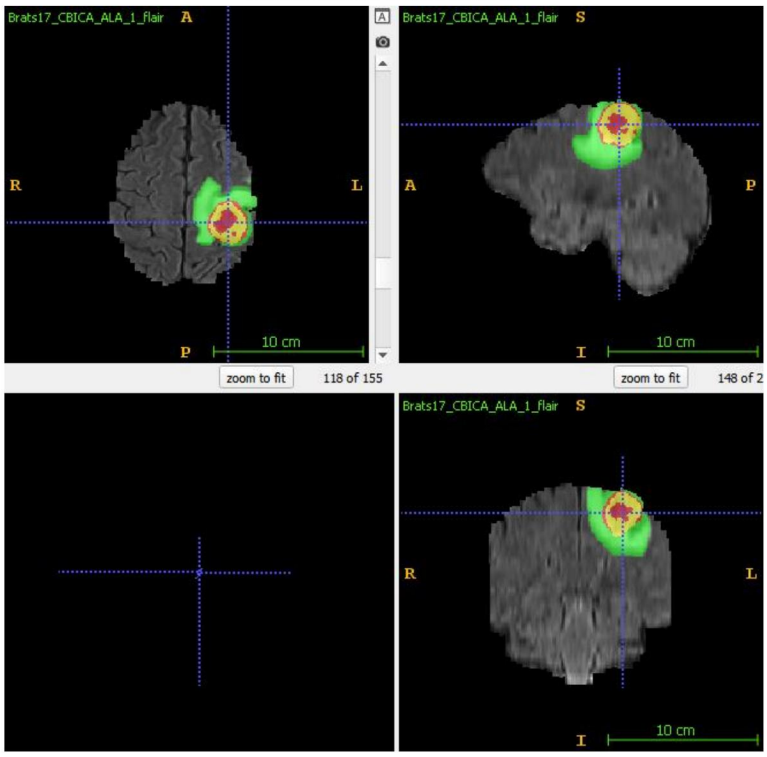
Model	Dice_ET	Dice_WT	Dice_TC	Sensitivity_ET	Sensitivity_WT	Sensitivity_TC
Original	0.75761	0.89899	0.83502	0.78272	0.9221	0.82967
Added Layer	0.70386	0.86193	0.70802	0.80848	0.89627	0.86119
New Loss Function	0.73527	0.88759	0.73865	0.71132	0.89701	0.76053
	Specificity _ET	Specificity_ WT	Specificity _TC	Hausdorff95 _ET	Hausdorff95_ WT	Hausdorff95_TC
Original	0.99823	0.99377	0.99789	3.78315	5.72536	7.25928
Added Layer	0.99798	0.99266	0.99192	8.03155	22.60405	12.03764
New Loss Function	0.99887	0.99344	0.99659	5.05962	5.93903	10.41551

Qualitative Data (Brats17_CBICA_ABT_1)

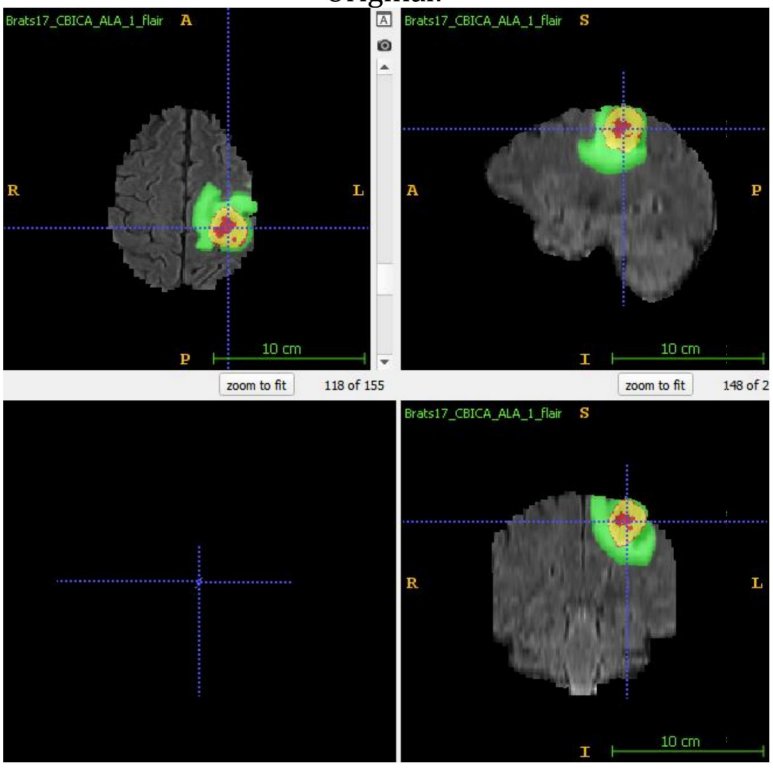
Added Layer:



New Loss Function:



Original:



Works Cited

Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net : Learning dense volumetric segmentation from sparse annotation. In: MICCAI. pp. 424–432 (2016)

Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I. Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso^, Tom Vercauteren^. "NiftyNet: a deep-learning platform for medical imaging." Computer Methods and Programs in Biomedicine, 158 (2018): 113-122. https://arxiv.org/pdf/1709.03485

Guotai Wang, Wenqi Li, Sebastien Ourselin, Tom Vercauteren. "Automatic Brain Tumor Segmentation using Cascaded Anisotropic Convolutional Neural Networks." In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Pages 179-190. Springer, 2018. https://arxiv.org/abs/1709.00382

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Medical Image Analysis 35, 18–31 (2016)

Menze, B.H., Van Leemput, K., Lashkari, D., Weber, M.A., Ayache, N., Golland, P.: A generative model for brain tumor segmentation in multi-modal images. In: MICCAI. pp. 151–159 (2010)

Subbanna, Precup, Collins, Arbel, 2013

N. Subbanna, D. Precup, L. Collins, T. Arbel **Hierarchical probabilistic gabor and mrf segmentation of brain tumours in mri volumes.**
in Proceedings of MICCAI, Vol. 8149 (2013), pp. 751-758

Tustison, Wintermark, Avants, 2013

N. Tustison, M.C.D. Wintermark, B. Avants **Ants and árboles**
in Proceedings of BRATS Challenge - MICCAI (2013)