Computer Vision Radiology

William Bakst
Stanford University
wbakst@stanford.edu

Linus Meyer-Teruel
Stanford University
lmeyerte@stanford.edu

Cameron Andrews
Stanford University
cba@stanford.edu

Abstract

This paper proposes a pipeline for the preprocessing and pixel-wise segmentation of primary and secondary glioblastomas and low-grade gliomas using deep learning. We incorporated inception layers into a three dimensional UNet architecture, yielding improved model performance over established benchmarks on the BRATS dataset by a statistically significant margin. Our models performed comparably to the state-of-the-art on the segmentation of the enhanced tumor tissues, the most difficult diagnostic task associated with glioblastoma segmentation.

1 Introduction

Cancer was responsible for 600,000 deaths in the United States in 2016, and is one of the leading causes of death in the world today. Early detection is the most effective tool we have to stymie this horrible disease, and with the rise and refinement of deep learning methods coupled with the dramatic increase in the power and accessibility of computational resources, machines seem perfectly positioned to alleviate some of the glaring inefficiencies and shortcomings of the diagnostic status quo. If new technology fails to empower our diagnosticians and equip them to do more with less, our generation will be the first since the advent of modern medicine to see a decline in the quality of care afforded to them. For the reasons stated, we are excited to apply convolutional neural networks to medical image analysis, specifically geared towards the segmentation and characterization of glioblastoma from MRI scans.

2 Challenges

There exist a number of acute technical challenges that we are weary of. First and foremost is the manipulation and interpretation of our image data. MRI data is recorded in three dimensions, and requires a network architecture that can properly process it as such. In addition, 3D imaging modalities are more prone to blurring and image quality degeneration as the result of patients moving around on the stage during the course of their 30-40 minute scan. Accounting for this in a scalable fashion will most likely be difficult, and we plan to use many forms of data augmentation to address this issue.

Another crucial challenge is bounding and image pre-processing. Some of the DICOM images can be as large as 1GB, and handling them can be computationally expensive, to say nothing of training models. Bounding these images to a point that models can be reasonably trained on available hardware will be critically important to speeding up the development process, allowing us to iterate quickly and ultimately produce a good model.

3 Dataset and Features

The Brain Tumor Segmentation Challenge has become a popular competition for those working on deep learning in medicine. Contestants develop algorithms (or, increasingly, neural networks) in

order to characterize acute glioblastomas, including both primary and secondary tumors [4, 1]. The BRATS dataset includes multiple imaging modalities (T1/T2 MRI tissue contrasts, T2 FLAIR, and T1 contrast-enhanced MRI), a variety of preenting phenotypes (primary/secondary tumors, solid and infiltrative growing tumor profiles), a range of patient states (some images are acquired prior to treatment, some post-treatment, and as such a number of images display common abnormalities caused by radio-therapy and cavities from resection). In total, the data consists of 274 training MR images, of which 220 are high grade gliomas and 54 are low grade gliomas. Included with the training images are ground-truth pixel labels for each scan, with label values consisting of five classes: necrosis, edema, non-enhancing tumor, enhancing tumor, and healthy tissues.

The dataset is broken up into two segments: the data itself (skull-stripped brain MRIs) and the ground-truth segmentation of the data, hand-annotated by radiologists. Each case has between 2-4 images associated with it, often comprised of all four imaging modalities listed above. All images are in stored in an MHA filetype.

3.1 Preprocessing

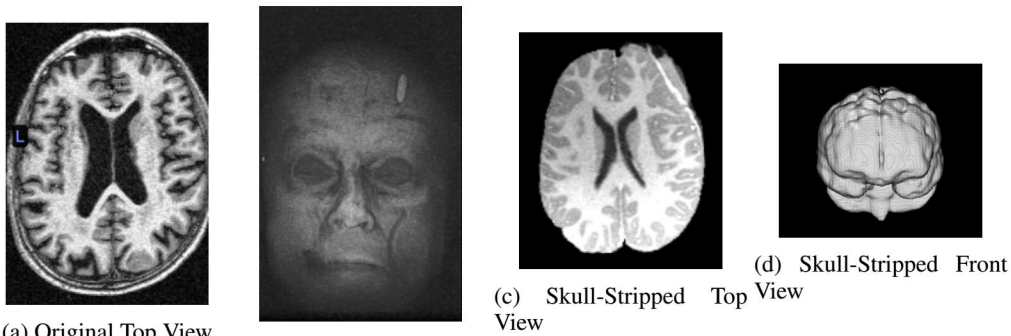
The standard pipeline for the classification and segmentation of brain MRIs is as follows:

3.1.1 Read in image

We read in the 4 images per patient (corresponding to 4 different MR modalities) using the skimage library, which is ubiquitous amongst projects handling this kind of data. The images are generally between 10-15 MB per case (500kb for ground truth, scans are 2-4 MB), and are 240x240 voxels.

3.1.2 Skull-stripping

Operating on this dataset simplifies this pipeline, as the images have already been skull-stripped. That said, we developed our own skull-stripping pipeline based on a 2016 paper that used deep-learning to identify and remove all non-brain features. The purpose of this is to allow us to begin operating on our proprietary dataset we are acquiring from one of the largest outpatient imaging companies in the United States, but do not yet have access to. However, since the BRATS dataset is already skull-stripped, we were able to shift our focus to the development of the network itself.



(a) Original Top View (b) Original Front View (c) Skull-Stripped Top View (d) Skull-Stripped Front View

Figure 1: Skull stripping pipeline.

3.1.3 Resizing the images

Another standard step in any deep learning pipeline, image resizing, was performed by the skimage library. We experimented with resizing to different input dimensions, and eventually decided to resize the images to 64x64x64 pixels for the baseline and 128x128x128 for our UNet implementation. To resize the input images, we simply inputted the brain MRIs and ground truth segmentations into skimage and used the built-in resize functions to generate data of the appropriate dimensions.

3.1.4 Normalize image

Signal intensity for MRI images varies significantly based on the tumor profile, physiological variance between patients, the hardware used to capture the image, the amount of Gadolinium (or another comparable contrast agent) used for image capture, and the machine settings used to capture the image. As such, normalizing signal intensity of the input images is critical. We used a standard score-based method to normalize. This involves subtracting the mean intensity from the sample voxel, divided by the standard deviation in voxel intensity:

$$x' = (x - \mu)/\sigma$$

Standard deviation and mean were calculated over the whole training set. We used these same values for normalizing the test set to be consistent with the measure that the neural net was trained on. We believe this normalization method to be superior to alternatives (such as feature scaling) because it keeps parameters from getting assigned artificially large or zero values during training, and because it is significantly less sensitive to outliers (MRI images can have very bright spots in the image that are not inherently cancerous).

3.2 Augmentation

We augmented our dataset by rotating the 3D images across all 3 axis, resulting in a six fold increase in the size of our training set. This had a significant positive impact on our model performance: our F1 training score increased from —- to —- and —- to —- on our models with and without inception layers respectively. Our scores on the validation set similarly increased from —- to —– for our model without inception layers, and from —- to —- for our model with them. While there are more complex augmentation methods described in the literature, we believe that our model would over-fit our data if we did not expand our dataset before augmenting further.

4 Architecture

Our general model concepts are loosely based on the idea of an all convolutional network, meaning that we do not use any fully connected layers [5].

4.1 Baseline Model

Our baseline model is an simple 3 layer convolutional net. Each layer is CONV–MAXPOOL–ReLU where we use same padding for both the convolutional and the max pooling operations.

4.2 UNet-3D

Our UNet-3D model is built based on the original UNet-3D paper [2]. We use the same number of standard convolution layers, maxpools, and upconvolutions (15, 4, 4 respectively) for a total of 23 layers and 19 million parameters. Before each max-pooling, we have a double covolutional layer of the form CONV-ReLU-BATCHNORM-CONV-ReLU-BATCHNORM.

4.3 UNet-3D with Inception Layers

For this model we have altered the original UNet-3D architecture by replacing the double convolutional layers with single inception layers [6]. This increases the total convolutions to 42 while keeping the number of maxpools and upconvolutions constant but uses only approximately 2 million parameters.

4.4 ResNet50-3D

For this model we implemented the ResNet50 that we saw in class [3] using 3D convolutions instead of 2D convolutions. The original ResNet50 that we saw in class performed a simple binary classification for its output, but since we are classifying each pixel in a 3D image, we modified the network to incorporate upsampling (which we saw in lecture) in order to preserve the image size

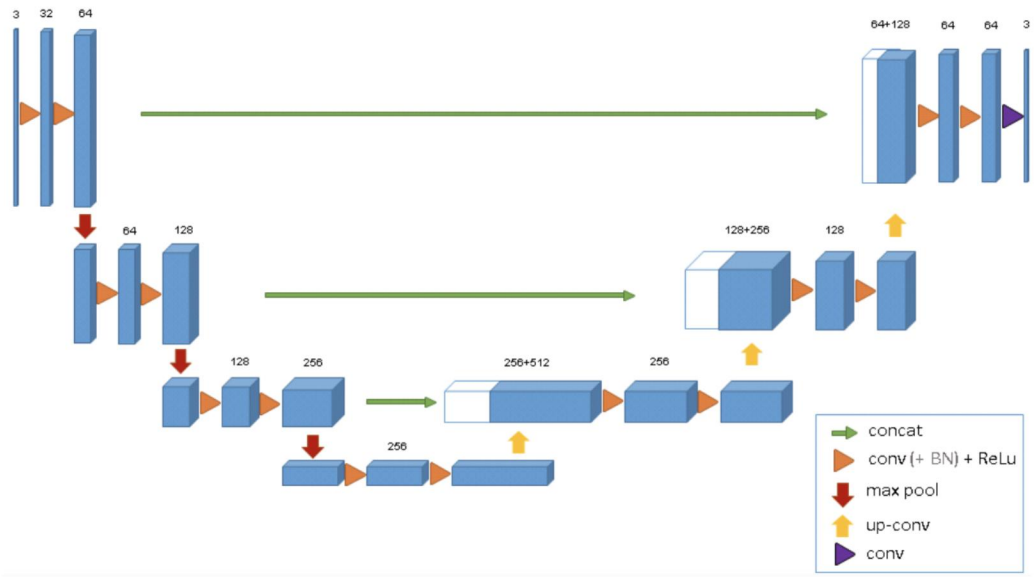
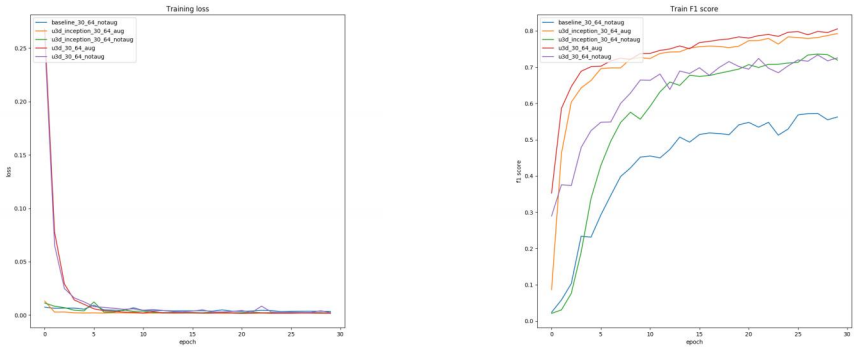


Figure 2: UNet Model Graph

when making our final predictions. This model is currently still in development and has not been fully evaluated yet.

5 Evaluation

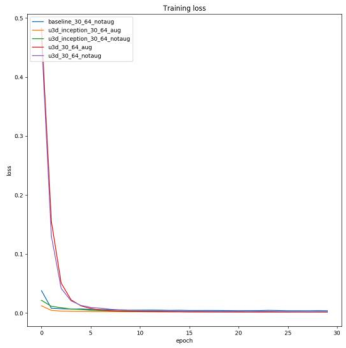
Our evaluation metric will be F1 score on the BRATS dataset, where we will attempt to properly segment an MRI scan to identify the regions with enhancing cancerous tumors. We compare our results for this task the the BRATS 2015 winners, which had a highest mean F1 score of 75. For our loss we will be using binary cross entropy loss. The data is downsized to 155x64x64 to allow for the model to fit into the GPU's memory.



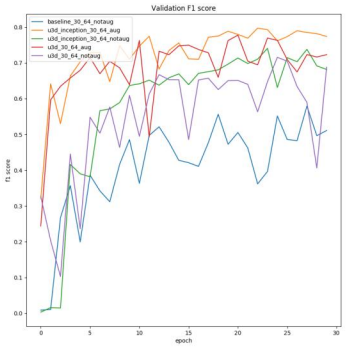
(a) Train Loss

(b) Train F1

Figure 3: Training Loss



(a) Validation loss



(b) Validation F1 score

Figure 4: Validation Results

Model	Train Loss	Dev Loss	Train F1	Dev F1	Params
Baseline	0.037	0.0033	0.572	0.579	69,729
U3D	0.0021	0.0022	0.734	0.716	19,078,337
U3D Aug	0.0013	0.0016	0.805	0.778	19,078,337
U3D Inception	0.0019	0.0019	0.736	0.741	2,614,949
U3D Inception Aug	0.0014	0.0017	0.793	0.797	2,614,949

5.1 Evaluation Results

Our Unet3D model with Inception layers managed to outperform the standard U3D model, with a validation F1 score of 0.797 compared to 0.778. This occurred both with data augmentation and without, demonstrating an overall improvement over the original model with a ten fold decrease in parameters. This was likely due to the models ability to view the image at multiple scales at each convolutional layer. These results also demonstrate that our models might beat the original models submitted to the BRATS 2015 competition. However, as the competition ended, we were unable to evaluate our model on the official test set.

6 Future Work

A handful of 3D, fully convolutional architectures have done well on the diagnosis and segmentation of everything from brain tumors to multiple sclerosis. We decided to implement to widely used UNet-3D and ResNet-3D models (still in development), but there are many more models that we would still like to build and evaluate in order to compare to our current models and results. For future work, we would do some follow-up research on a handful of architectures such as SegNet, FCN, Nabla-Net, DeepMedic, and others. We would also perform a more robust hyperparameter search, as this takes quite a long time (a bit too long for the scope of this project).

We also plan to evaluate our models on different, similar datasets such as the new BRATS 2017 dataset, which was hand-labeled by a committee of doctors, as well as a proprietary dataset that we have recently been given access to. We believe that our models will be able to perform quite well on these datasets given the performance we have seen on our current dataset.

7 Contributions

William Bakst: General project setup (git repo, basic files, etc.), Preprocessing, Model evaluation program, Baseline implementation, ResNet50-3D, Final Writeup.

Cameron Andrews: Data acquisition, Skull-stripping, ResNet50-3D, Final Writeup.

Linus Meyer-Teruel: Preprocessing, UNet-3D (with/without inception), Final Writeup.

References

[1] The Virtual Skeleton Database: An Open Access Repository for Biomedical Research and Collaboration. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3841349/.

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. URL http://arxiv.org/abs/1606.06650.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[4] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. R. Gerstner, M. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. T. Ryan, D. Sarikaya, L. H. Schwartz, H. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Székely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. B. Ünal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. V. Leemput. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging*, 34(10):1993–2024, 2015. doi: 10.1109/TMI.2014.2377694. URL https://doi.org/10.1109/TMI.2014.2377694.

[5] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014. URL http://arxiv.org/abs/1412.6806.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594. URL https://doi.org/10.1109/CVPR.2015.7298594.

Code: https://github.com/wbakst/CS-230-Project