Brain Tumor Segmentation from MRI Scans

Jacky Huang
Stanford University
jackyh@stanford.edu

Abstract

Magnetic Resonance Imaging (MRI) scans are frequently used by physicians to diagnose and plan treatments for brain tumors. One component of this workflow involves the segmentation of the tumor from the scan. Due to the time consuming nature of this task, automated segmentation algorithms are of interest to the medical research community. This paper explores how deep learning can be applied to segment out tumors from MRI scans, focusing on a patch-based training approach utilizing the U-Net architecture. Training and evaluation was done on the Multimodal Brain Tumor Segmentation Challenge (BraTS) 2018 dataset, achieving a Dice score of 0.54676 and a 95th percentile Hausdorff distance of 6.30415 for the enhancing tumor (ET) segmentation on the validation dataset.

1 Introduction

Magnetic Resonance Imaging (MRI) scans are a common medical imaging tool used by medical professionals in the diagnosis of brain tumors. By using a magnetic field and radio waves, the scanner is able to map out a detailed volumetric (3D) image of the patient's brain. Radiologists are then able to analyze these scans to determine exactly where the tumor is located. However, manual segmentation of the tumor by radiologists takes a lot of time and effort, and is prone to error. Due to this bottleneck, there is a large interest in researching automated algorithms for tumor segmentation. Having such a tool can significantly cut down their workload as they will only need to correct the mistakes made by the algorithm instead of having to classify every voxel manually. This project investigates how deep learning can be used to achieve this task. Given an input 3D MRI scan from the BraTS 2018 dataset of a patient with a glioma, we apply image segmentation techniques to obtain an output segmentation of the tumor in the same format.

2 Related work

Image segmentation is a widely studied area of computer vision which differs from standard classification tasks in that it requires a classification on a pixel level. To provide this localization, Ciresan et al. [10] used a sliding window approach to classify each individual pixel based on its surroundings with a convolutional neural network based architecture. As this required running the classification network on each individual pixel, it was quite slow. Ronneberger et al. [18] came up with the U-Net architecture based on fully convolutional networks, which is able to do this task much more quickly and accurately.

The U-Net architecture by Ronneberger et al. was a 2D convolutional neural network based architecture. As such, it is unable to accurately capture the volumetric properties of the MRI scans as it can only process one slice at a time, thus ignoring the relationships between adjacent slices. A 3D variant of the U-Net architecture was quickly proposed by Cicek et al. [20] to handle this issue. This new network is essentially an extension of the 2D variant, namely by replacing all 2D operations with their 3D versions. Instead of taking in a slice of the image, it can generalize to take in a 3D subvolume, which is ideal for medical imaging segmentation tasks.

CS230: Deep Learning, Spring 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

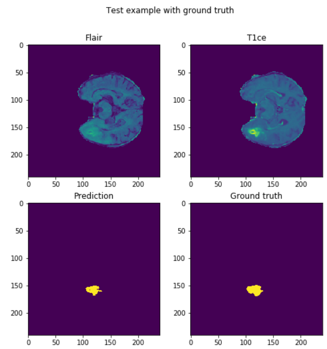

Figure 1: The top two images represent a specific slice for the Flair and T1ce modalities. Bottom left is our baseline prediction for the enhancing tumor, and bottom right is the ground truth.

In the 2018 BraTS challenge, Isensee et al. [12] built upon the 3D U-Net architecture to solve the task of tumor segmentation. Having placed second overall, they showed that a generic 3D U-Net with minor changes was able to score very well on the dataset. By optimizing the training aspect rather than the architecture design, such as through intelligently picking hyperparameters like patch size and using various data augmentation techniques, they were able to achieve a Dice score of 0.7788 and a Hausdorff distance of 2.90 for the enhancing tumor on the test set. On the other hand, the first place team overall in the BraTS 2018 challenge follows a CNN based encoder-decoder approach with an additional variational auto-encoder to reconstruct the original image for regularization purposes during training [16]. This approach achieved a Dice score of 0.7664 and a Hausdorff distance of 3.7731 for the enhancing tumor on the test set.

3 Dataset

The dataset we have chosen for training is the MICCAI BraTS 2018 dataset [6, 15, 4, 5, 2, 3]. The dataset, which is 2.8GB in size, is split into two parts for training and validation, consisting of 285 samples and 66 samples respectively. Evaluation of the overall model was done using the validation set. During the model training process, the training set itself was randomly split into a 80/20 training/validation split.

The training subset is further divided into low grade glioma (LGG) and high grade glioma (HGG) classes. Although each sample consists of four NifTI files representing different modalities (T1, T1ce, T2, Flair), we are only interested in using the T1ce and Flair modalities as these are most commonly available in clinical settings. Each file is a 3D image of size (240, 240, 155) where each voxel represents the intensity at that location. The orientation of the image is from the axial view. Although some preprocessing such as skull stripping has already been applied to the image, the voxel intensities are not standardized as they are collected from different institutions. The training set also has an additional file of the same dimensions which represents the ground truth for the segmentation as evaluated by experts. This contains annotations for the whole tumor (WT), tumor core (TC), enhancing tumor (ET) and the background classes. For this project, we are only interested in segmenting the enhancing tumor. Figure 1 gives an example of what the data looks like as well as an example segmentation.

One issue is that the 3D nature of the images means that running the data through a deep learning model will be computationally expensive and take up a lot of memory. It is not feasible to pass in the entire image all at once to the network. We avoid this issue by training with patches. Additionally, as this is a segmentation problem, we can expect that the background class will dominate the tumor classes. This is indeed the case; we find that non-tumorous voxels account for 98.88% of all labels in the training set, while the other regions account for 0.2% (ET), 0.28% (TC), and 0.64% (WT) of the labels. We experimented with a statistical based subsampling algorithm to generate the patches in a way to avoid training with patches that have a low presence of enhancing tumors.

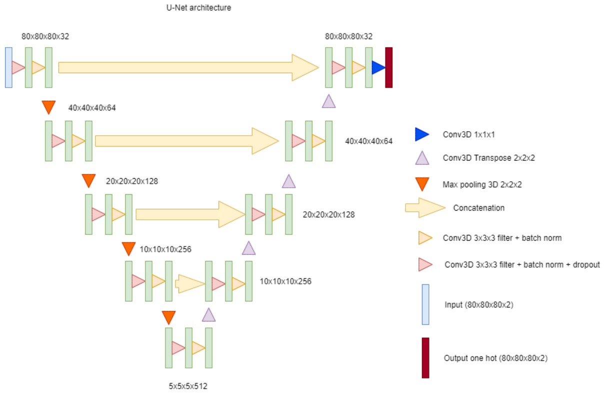

Figure 2: The architecture for our modified U-Net network.

Due to the large size of the data, we use a Keras [9] data generator to generate patches of size (80, 80, 80) to train with on the fly using the subsampling algorithms. Preprocessing is done to each generated patch, including standardization of the patch to have 0 mean and unit variance, as well as standard data augmentation techniques for images such as translations, flipping and rotations.

4 Methods

As mentioned above, U-Net models are generally considered a good choice for image segmentation problems. We decided to base our architecture on U-Nets as we sympathize with the sentiment of Isensee [12] in keeping the architecture simple and focusing on the training component as we felt this would help deal with the disadvantage of using only 50% of the data. These types of architectures consist of a “contracting" path followed by a “expanding" path. In the contracting path, the dimensions of the input are halved in every convolutional block via the use of max pooling layers, while the number of feature channels are doubled. This part of the network is similar to a typical convolutional network in that it aims to extract the image features from the input. On the other hand, the upsampling path is the inverse to the downsampling path; the dimensions of the input are doubled in every convolutional block through up convolutions, while the number of feature channels are halved. The goal of this path is to reconstruct the segmentation mask by using the image features from the contracting path, and thus a concatenation is also used to glue these layers together. Finally, a 1x1x1 convolution is used to create the one-hot segmentation mask as the output. Due to issues with overfitting, we decided to use a slightly modified U-Net architecture as in Figure 2 for our experiments by adding batch norm and dropout (with probability 0.5) layers. ReLU activation was used except for the final activation, where we decided to go with softmax.

We decided to use soft dice loss [13]

$$\mathcal{L}(y, \hat{y}) := 1 - \text{Dice}(y, \hat{y}) = 1 - \frac{2 \sum_{x,y,z} (y * \hat{y}) + \epsilon}{\sum_{x,y,z} (y * y) + \sum_{x,y,z} (\hat{y} * \hat{y}) + \epsilon} \tag{1}$$

as our loss function because the main evaluation metric is the Dice score. Here, y is a tensor of the true target labels, $\hat{y}$ is a tensor of our predicted probabilities, $\epsilon = 1$ is a smoothing constant and $*$ denotes elementwise matrix multiplication. An additional bonus of using such a loss function is that it is able to deal with imbalanced data.

4.1 Baseline

For our baseline model, we decided to go with a generic 2D U-Net architecture with a depth of 4 with 64 filters initially, as 3D models are hard to train. We used the implementation from the keras-unet package [19]. We process each slice in the z dimension separately and combine the results together. While this allowed us to speed up computation time compared to the 3D U-Net, it ignores the fact that tumors are three dimensional. Nevertheless, we found that this was a good baseline to work from.

In addition to the preprocessing steps mentioned above, we filtered out the z slices with no enhancing tumor pixels present, and only use those slices with at least one pixel containing the enhancing tumor. This was designed to combat the large class imbalance, as we would expect that the majority of the image would be non-tumorous. Applying this additional preprocessing step helped speed up training and improved the accuracy of the predictions.

4.2 3D U-Net

In order to fully take advantage of the 3D nature of the input, we implemented a 3D generalization of the U-Net architecture [20] as seen in Figure 2. We chose to keep the depth of the U-Net model as 4, as we found that it was a reasonable compromise between complexity and efficiency. However, due to the much larger amount of parameters in a 3D model, we needed to drop the initial filter size to 32 as well as do patch based training in order to fit in GPU memory. For this project, we explored different subsampling algorithms to generate the patches.

The subsampling algorithm present in the baseline model can easily be extended to a 3D patch. Instead of taking a 2D slice of size (240, 240, 1) as the input, we can take a 3D subvolume of size (240, 240, n_z). Here, we picked $n_z = 16$. In an analogous manner to the preprocessing steps of the baseline model, we pick only the patches which have at least t many voxels containing the enhancing tumor. We experimented with $t = 0, 100, 1000, 1000$ and found that 100 worked well for our purposes.

Another approach we explored was a statistical sampling approach to generate the patches. We tried out different patch sizes but decided to go with (80, 80, 80). Instead of generating patches with uniform probability as in Isensee et al. [12], we designed a simple statistical algorithm that optimizes for picking patches with more tumorous voxels, thus ensuring higher data quality. We use the observation that tumors are generally condensed together in one region of the brain, rather than spread out in a uniform fashion. Thus, we compute the center of mass (c_x, c_y, c_z) of the enhancing tumor and add i.i.d Gaussian noise with 0 mean and σ variance to generate the midpoint of a patch. This is generated on the fly during training, and preprocessing is applied to each of these patches. We picked $\sigma = 20.408164$, which roughly leads to a 95% probability of the patch containing the center of mass. The idea behind this is that it would help the model to fine tune its classification abilities on a more granular level by avoiding the background as much possible, and to be able to speed up convergence by training with patches that (on average) contains lots of enhancing tumor voxels. Due to issues with the smoothness of the predictions, inference was done by predicting on overlapping patches (with stride 20 in each direction) and then taking the average to get the final segmentation rather than splitting the image up into disjoint 80x80x80 patches and doing the predictions separately for each patch.

We also played around with an additional hyperparameter $\alpha \in [0, 1]$ such that we generate the patch using the Gaussian method with probability α, and the uniform random method with probability $(1 - \alpha)$. The goal of this weighted average was to find a balance in order to ensure that the algorithm is not too aggressive in classifying voxels as tumorous (large α), while also not being too passive (small α). We observed that playing around α did not influence the results in a significant manner to justify the additional hyperparameter.

5 Experiments/Results/Discussion

We experimented with different optimizers and learning rates but we found that Adam optimizer with a learning rate of 0.0001 provided the most stable training. Our batch size was 2; this was the biggest power of 2 such that the network would fit in GPU memory. We trained for a total of 35 epochs with 3000 patches per epoch on an NVIDIA RTX 2080 Ti. We decided to stop at epoch 35 as it seemed like the training loss was plateauing. The metrics we used were the same ones used for BraTS evaluation; Dice score (as seen in (1) but with predictions rather than probabilities), Hausdorff distance d_H [14], sensitivity and specificity. We have

$$d_H(y_{\text{et}}, \hat{y}_{\text{et}}) = \max\{\sup_{u \in y_{\text{et}}} \inf_{v \in \hat{y}_{\text{et}}} d_2(u, v), \sup_{u \in \hat{y}_{\text{et}}} \inf_{v \in y_{\text{et}}} d_2(u, v)\}, \tag{2}$$

where d_2 denotes the ℓ^2 metric, and the et subscript indicates the subset of the image containing the enhancing tumor class. Intuitively, Hausdorff distance gives a measure of whether or not the segmentations of the tumor are close to where it should be.

Inference on the validation set was done and submitted to the CBICA Image Processing Portal [17] for evaluation, as seen in Table 1.

	Dice Score	Hausdorff Distance	Sensitivity	Specificity
Baseline	0.45001/0.54291	58.94707/62.19492	0.44159/0.48235	0.92951/0.99707
3D Baseline	0.53241/0.68811	61.25853/53.61282	0.60653/0.74911	0.96609/0.99804
Uniform samp.	0.53003/0.69635	9.53483/**3.74166**	**0.60815/0.86128**	0.99680/0.99920
Gaussian samp.	**0.54676/0.74373**	**6.30415**/4.00000	0.52668/0.65533	**0.99863/0.99972**

Table 1: Mean/Median metric scores after submission of predictions.

From Table 1, we found that for all our models, the mean scores were significantly worse than the median score for all metrics. Looking further into the results, we saw that there were a few outliers in the validation dataset for which all of our models did not find any enhancing tumor voxels, which had a significant impact on the mean scores. Taking out these outliers seems to bring the mean much closer to the median, as seen in Table 2. One possible explanation for these outliers are that they are LGG examples, where there may be no or very little enhancing tumor. Looking into the training dataset seems to support this theory, as we found multiple examples which had no enhancing tumors in the ground truth. In our research, we found that other papers also had problems with these outliers [1, 7, 8, 11]. Further investigation needs to be done into why these outliers exist and how to address these.

From a quantitative standpoint and disregarding the outliers, our model does a decent job at segmenting the enhancing tumor. The low Hausdorff distance means that it has successfully learnt to identify the precise location of the tumor in the brain. While it does really well on the specificity metric, the relatively low sensitivity values suggests that the model is too conservative in predicting voxels as tumorous, possibly as a result of the major class imbalance. As expected, the baseline does the poorest out of all our models - this is expected as it does not use the 3D nature of the input. It is also expected that the 3D extension with 16 slices would do worse compared to the sampling based approaches; the number of training patches of size 240x240x16 is much smaller than the number of patches of size 80x80x80, so overfitting becomes an issue. On the other hand, it is interesting to see that the two different sampling approaches perform at a similar level. One explanation for this is that the training was done for a reasonably long time (around 10 hours each) until both approaches plateaued so in the end there was no big difference as the uniform sampling method was able to adequately explore the patch space. Indeed, we observe in Figure 3 that the Gaussian approach had consistently lower training loss compared to the uniform approach, suggesting that the Gaussian method leads to slightly faster convergence. Finally, the low training loss compared to our Dice score on the validation dataset indicates that the model is overfitting. We applied preprocessing and data augmentation techniques as well as dropout and batch norm layers as described above to mitigate this issue. Figure 4 shows an example of a segmentation.

	Dice Score	Hausdorff Distance	Sensitivity	Specificity
Baseline	0.57916	50.46157	0.56495	0.98964
3D extension of baseline	0.66297	36.84994	0.70951	0.99628
Uniform sampling	**0.72753**	5.89898	**0.80491**	**0.99886**
Gaussian sampling	0.72317	**5.06761**	0.69598	0.99795

Table 2: Mean metric scores after removing outliers.

6 Conclusion/Future Work

In this project we explored the application of deep learning to the area of medical imaging. We used the U-Net architecture to segment out brain tumors from 3D MRI scans, and investigated different methods for generating patches in patch based training. Our methods were fairly accurate in the segmentation of the enhancing tumor, although it did not do well on certain outliers in the validation dataset. The uniform and Gaussian sampling methods performed at a similar level in our primary metrics, although it seems like the Gaussian method would have slightly faster convergence. Future work in this area could involve investigating the differences between these sampling methods on more customized architectures rather than a generic U-Net architecture.

7 Contributions and Acknowledgement

The author was the sole contributor to the whole end to end project described in this paper, including coding, experimentation and analysis. This project was conducted under the supervision of Assistant Professor Haruka Itakura from the Stanford University School of Medicine. The author would like to thank Prof. Itakura for her invaluable advice and mentorship over the course of this project.

References

[1] Alberto Albiol, Antonio Albiol, and Francisco Albiol. Extending 2d deep learning architectures to 3d image segmentation problems. pages 73–82, 2019.

[2] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J. Kirby et al. 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`.

[3] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J. Kirby et al. 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`.

[4] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, and J.S. Kirby et al. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Nature Scientific Data 4:170117*, 2017. Publisher: Springer Nature, DOI: `10.1038/sdata.2017.117`.

[5] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, and Alessandro Crimi et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge, 2018. arXiv preprint, arXiv:1811.02629, URL: `https://arxiv.org/abs/1811.02629`.

[6] Spyridon (Spyros) Bakas, Bjoern Menze, Christos Davatzikos, Jayashree Kalpathy-Cramer, and Keyvan Farahani et al. Multimodal brain tumor segmentation challenge 2018. URL: `https://www.med.upenn.edu/sbia/brats2018.html`.

[7] Eze Benson, Michael P. Pound, Andrew P. French, Aaron S. Jackson, and Tony P. Pridmore. Deep hourglass for brain tumor segmentation. pages 419–428, 2019.

[8] Mariano Cabezas, Sergi Valverde, Sandra González-Villà, Albert Clérigues, Mostafa Salem, Kaisar Kushibar, Jose Bernal, Arnau Oliver, and Xavier Lladó. Survival prediction using ensemble tumor segmentation and transfer learning, 2018.

[9] François Chollet et al. Keras. `https://keras.io`, 2015.

[10] D.C. Ciresan, L.M. Gambardella, A. Giusti, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *NIPS*, page 2852–2860, 2012.

[11] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks, 2017.

[12] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H. Maier-Hein. No new-net. pages 234–244, 2019.

[13] Jeremy Jordan. An overview of semantic image segmentation. URL: `https://www.jeremyjordan.me/semantic-segmentation/`.

[14] User: kjytay. What is hausdorff distance? URL: `https://statisticaloddsandends.wordpress.com/2019/10/02/what-is-hausdorff-distance/`.

[15] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, and J. Kirby et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015. Publisher: Institute of Electrical and Electronics Engineers, DOI: `10.1109/TMI.2014.2377694`.

[16] Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 311–320, 2019. Publisher: Springer International Publishing, DOI: https://doi.org/10.1007/978-3-030-11726-9_28.

[17] CBICA Image Processing Portal. A web accessible platform for imaging analytics. URL: https://ipp.cbica.upenn.edu/.

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

[19] Karol Zak. Keras unet. URL: https://github.com/karolzak/keras-unet.

[20] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation, 2016.

8 Appendix

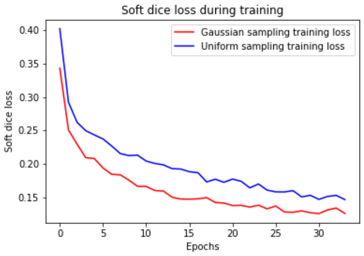

Figure 3: Training loss over epochs of the Uniform and Gaussian sampling algorithms. Soft dice loss is used.

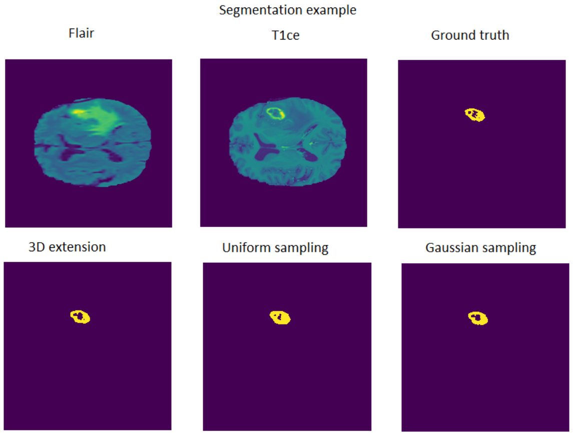

Figure 4: An example of a segmentation on a particular slice of a testing sample. The top row consists of the Flair and T1ce modalities along with the ground truth, and the bottom row consists of our extension, uniform sampling and Gaussian sampling segmentations respectively.