CAP 5516 Medical Image Computing Spring 2022 Assignment 2: BraTS MRI Segmentation Challenge

Kyle Beggs
Department of Mechanical and Aerospace Engineering
University of Central Florida
kbeggs07@knights.ucf.edu

1. Problem Definition

Segmentation of medical images is perhaps the most prevalent imaging task, so it is of great interest to continuously build motivation to push the boundaries of performance. Given a dataset of 3D MRI brain scans, we aim to segment the tumors present.

2. Methods

Since we are training on 3D images, and the assignment is to use a UNet type architecture, there are 2 main routes to go regarding architectures. You can use a 2D UNet and segment slice by slice or a true 3D UNet which uses 3D convolutions. We choose the latter because it is shown to perform better as you contain information in all dimensions for each operation which allows for better localization, although at the cost of extra computation. Here, we use the 3D UNet from [1].

The dataset contains .nii.gz files which all the images are of the same size (240, 240, 155). Preprocessing of the data includes normalizing the intensity and augmenting by random flipping across all 3 axes, small random intensity shifts and scales, and a small random cropping of the image to a final size of (224, 224, 144).

The model was trained using 2 NVIDIA V100 16GB GPUs on the same node with a Cosine Annealing learning rate scheduler initialized at 0.001. The batch size was set to 8. K-Fold cross validation was used and the model was trained from random weight initialization for each fold. 5 folds were used. Some examples of the data are shown in figures 2 and 3.

3. Results

The validation set for each fold is evaluated and shown in Table 1. The training loss for each fold is shown in figure 4. Notice the pink line (fold 3) converges much slower than the other 4 folds, but does catch up by the end of the training. Interestingly enough, it ends up being the 2nd best performing fold.

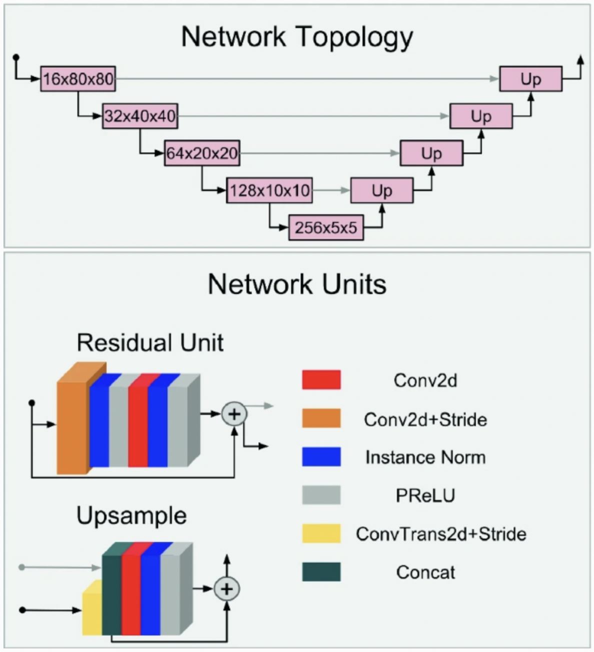


Figure 1: 3D UNet architecture with residual units.

Fold	Dice	Hausdorff
1	0.750	9.665
2	0.759	11.049
3	0.760	9.468
4	0.768	9.783
5	0.732	12.397
Avg.	0.754	10.472

Table 1: Results of each cross fold. The Dice and Hausdorff values shown here are the averages of the 3 segmentations.

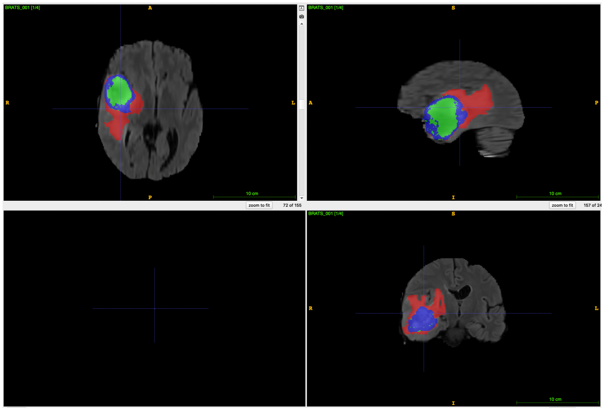


Figure 2: Random visualization of the training image and label.

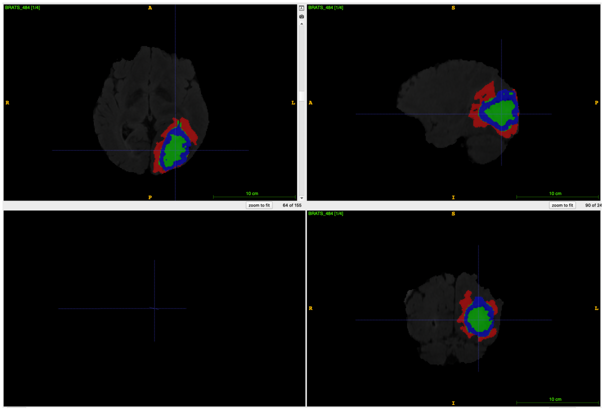


Figure 3: Random visualization of the training image and label.

Visualization of the models segmentation results using the weights found in fold 1 is seen in figures 5b and 6b.

4. Discussion

I have demonstrated the ability to successfully segment brain tumors using the BraTS dataset. K-fold cross validation is a good strategy to ensure you split the dataset into fair training/validation splits because some folds clearly performed better than others. Another big takeway is the large compute time; this training took 22 hours for all 5 folds on 2 V100 GPUs only going to 100 epochs each fold.

The code can be found on GitHub at kylebeggs/3D-UNet-Segmentation-BraTS.

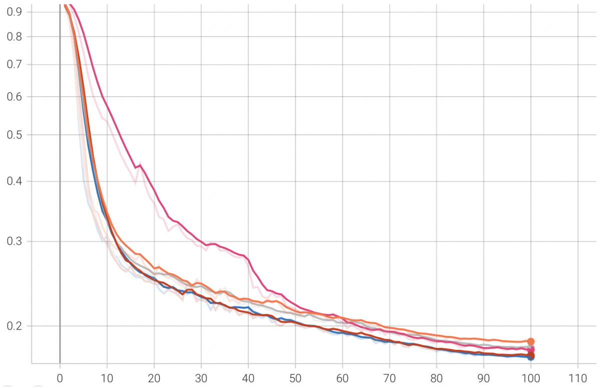
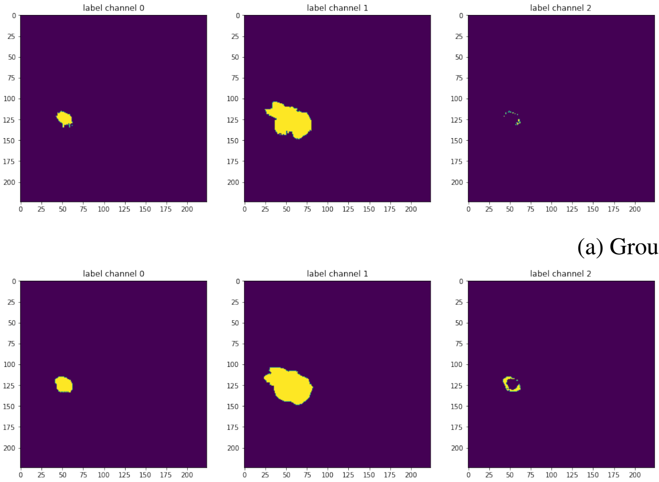


Figure 4: Training loss curves for training set. fold (color): fold1 (orange), fold2 (red), fold3 (pink), fold4 (grey), fold5 (blue).



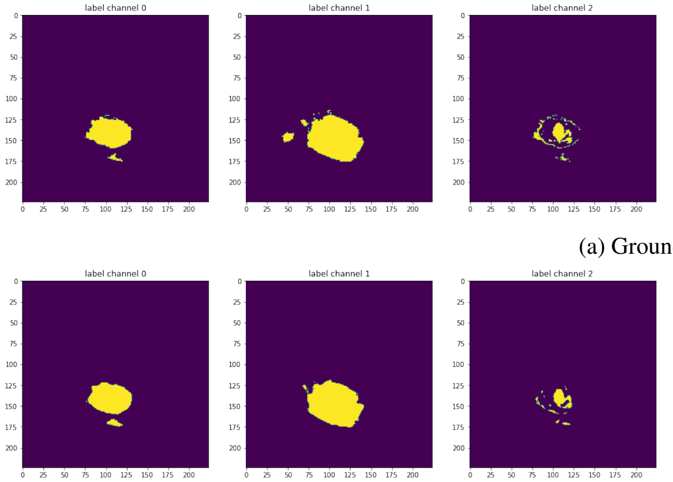
(a) Ground truth.

(b) Fold 1 segmentation.

Figure 5: Comparing ground truth to fold 1 segmentation.

References

[1] E. Kerfoot, J. Clough, I. Oksuz, J. Lee, A. P. King, and J. A. Schnabel, “Left-Ventricle Quantification Using Residual U-Net,” in *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges*, ser. Lecture Notes in Computer Science, M. Pop, M. Sermesant, J. Zhao, S. Li, K. McLeod, A. Young, K. Rhode, and T. Mansi, Eds. Cham: Springer International Publishing, 2019, pp. 371–380. 1



(a) Ground truth.

(b) Fold 1 segmentation.

Figure 6: Comparing ground truth to fold 1 segmentation.