



UNIVERSITY OF CENTRAL FLORIDA

Assignment II:

Deep Learning-based Brain Tumor Segmentation Using MRI

This report presents details of the implementation of the Assignment II where UNET was used for Deep Learning based Brain Tumor Segmentation task.

Submitted By:

Sanjeda Sara Jennifer

1. Introduction and Visualization using ITK-SNAP Software

The objective of this project is to perform brain tumor segmentation using a deep learning model, UNet, on MRI data. The network is evaluated using 5-fold cross-validation to ensure robustness and generalizability. The dataset used for training and evaluation consists of NIfTI format MRI images of brain tumor volumes, where the task is to classify and segment three types of tumors: Enhancing Tumor (ET), Whole Tumor (WT), and Tumor Core (TC).

The Task01_BrainTumour.tar dataset was downloaded from the link that was provided from: <https://drive.google.com/drive/folders/1HqEgzS8BV2c7xYNrZdEAnrHk7osJJ--2>.

After deleting the imagesTs directory, the final dataset contained the following shown in figure 1

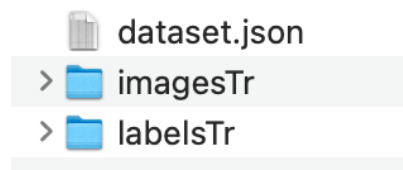
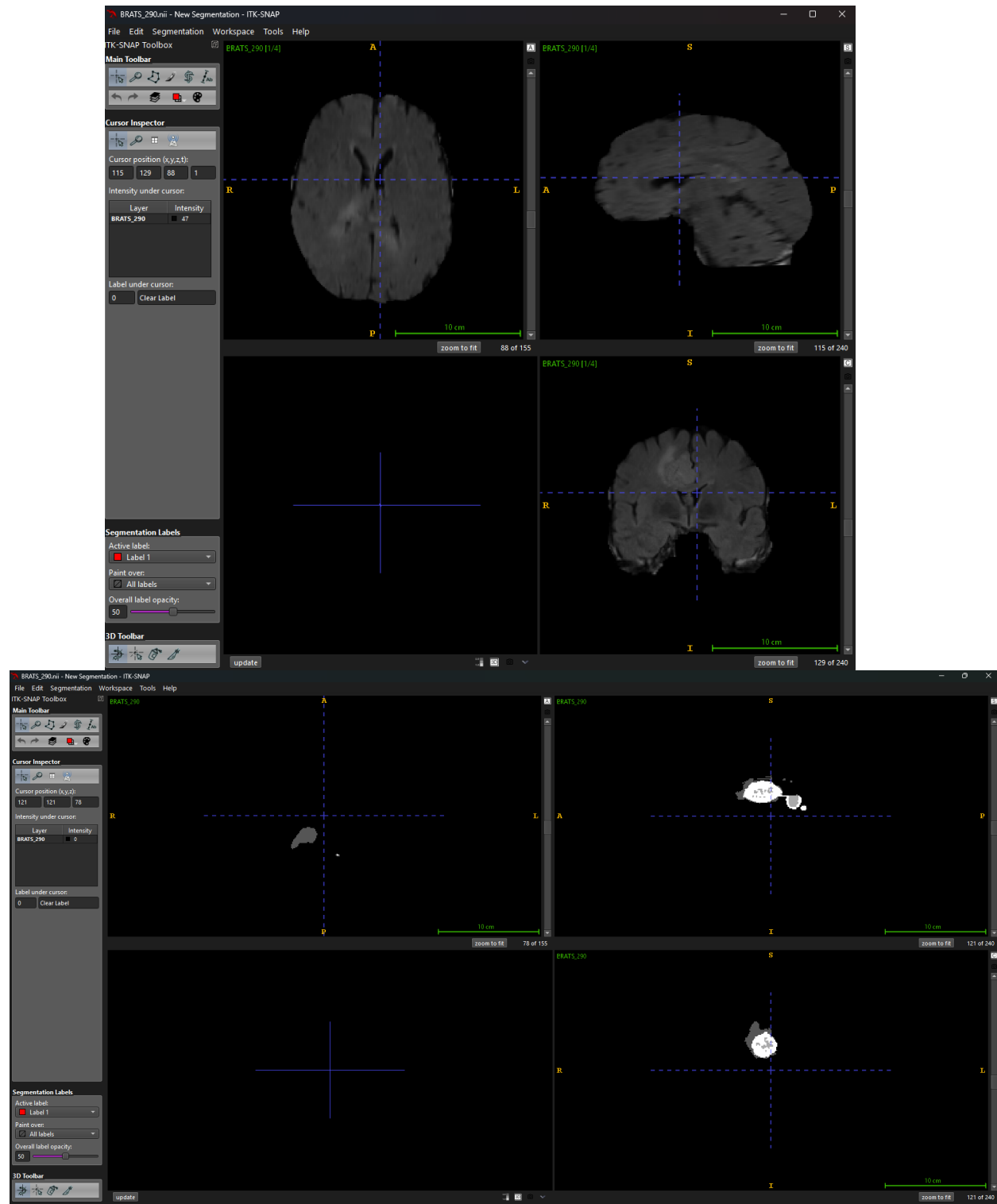


Figure 1: Final Dataset Directory

The ITK-SNAP Software was used to visualize the following images:



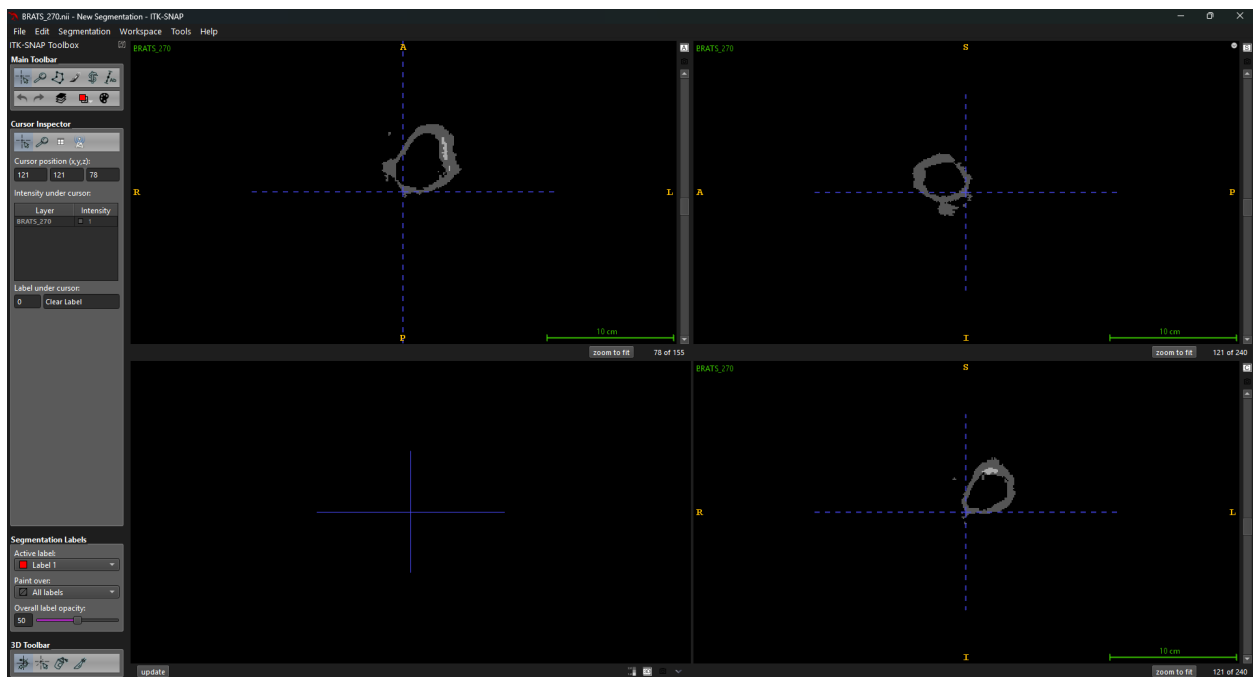
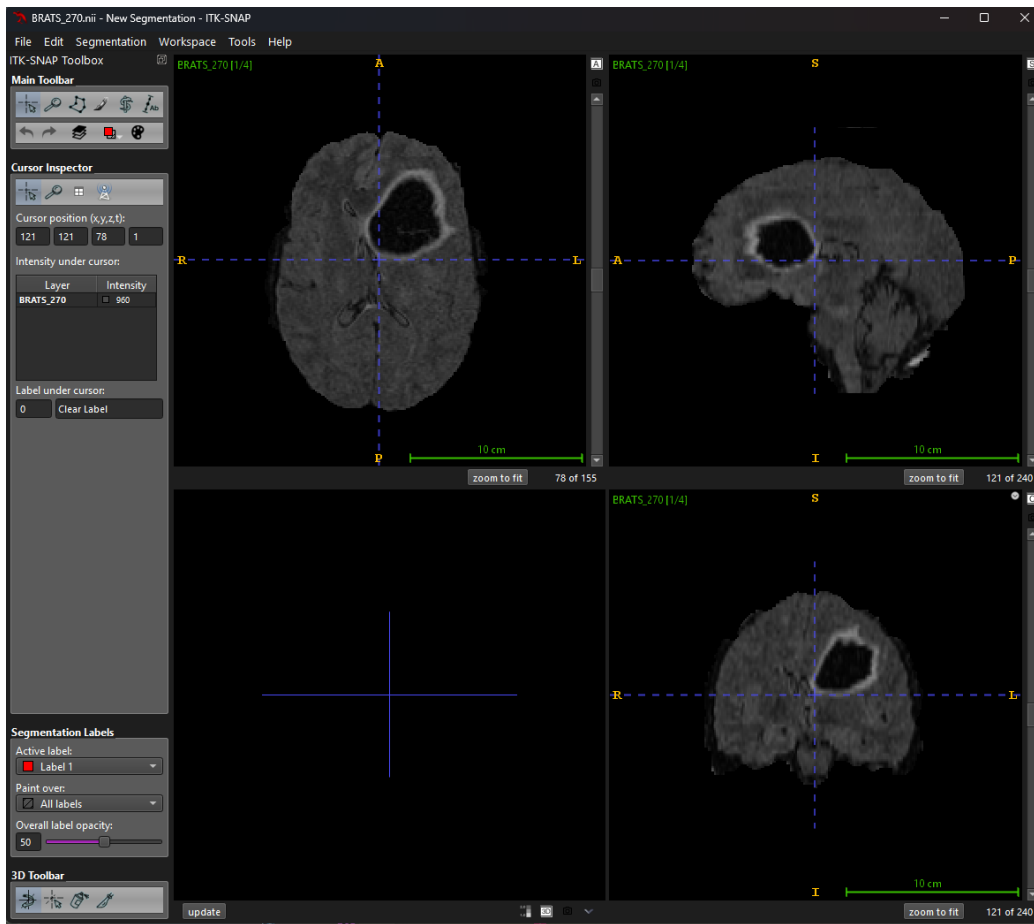


Figure 02: Sample images using ITK-SNAP

2. Implementation details of the network.

UNet model is a convolutional neural network (CNN) designed for semantic segmentation tasks, particularly in medical image analysis. The architecture follows an encoder-decoder structure, where the encoder learns to extract important features from the input, and the decoder reconstructs the segmentation mask from those features.

The encoder consists of multiple convolutional blocks, where each block contains the following:

- Two convolution layers with a kernel size of 3×3 , which capture local spatial information from the input image.
- ReLU activation function after each convolution operation to introduce non-linearity.
- The feature maps are downsampled using max-pooling with a kernel size of 2×2 after each convolution block.

In this network, Conv Block 1 processes the input image and extracts feature at a low resolution and Conv Block 2 extracts higher-level features as the resolution of the image decreases. The bottleneck layer consists of another convolution block, which learns to combine high-level features from the encoder.

The decoder reconstructs the image by progressively increasing the spatial resolution of the feature maps. The upsampling layers help to increase the spatial dimensions. After each upsampling layer, the feature maps are concatenated with the corresponding feature maps from the encoder (skip connections), which help preserve fine-grained details from earlier layers. Upconv2 and Upconv1 upsample the feature maps to the original size of the input image. The final layer is a 1×1 convolution that reduces the output to the desired number of channels, corresponding to the segmentation masks for the three tumor classes.

Dataset and Preprocessing:

The dataset consists of brain tumor images in the NIfTI format. Each volume is a 3D image where each voxel has a 4-channel modality, representing different MRI modalities, and a corresponding mask that labels the tumor areas. The nibabel library was used to load the MRI images and their corresponding segmentation masks. The 3D volumes were processed, and the middle slice was selected for segmentation, reducing the dimensionality to a 2D image. The images were then normalized to have pixel values between 0 and 1, making the training process more stable. Both the input images and masks are converted into PyTorch tensors for processing in the model. The target masks contain different tumor regions, labeled as 1 for ET, 2 for WT, and 3 for TC. Figure 02 shows a sample MRI image with the segmentation mask.

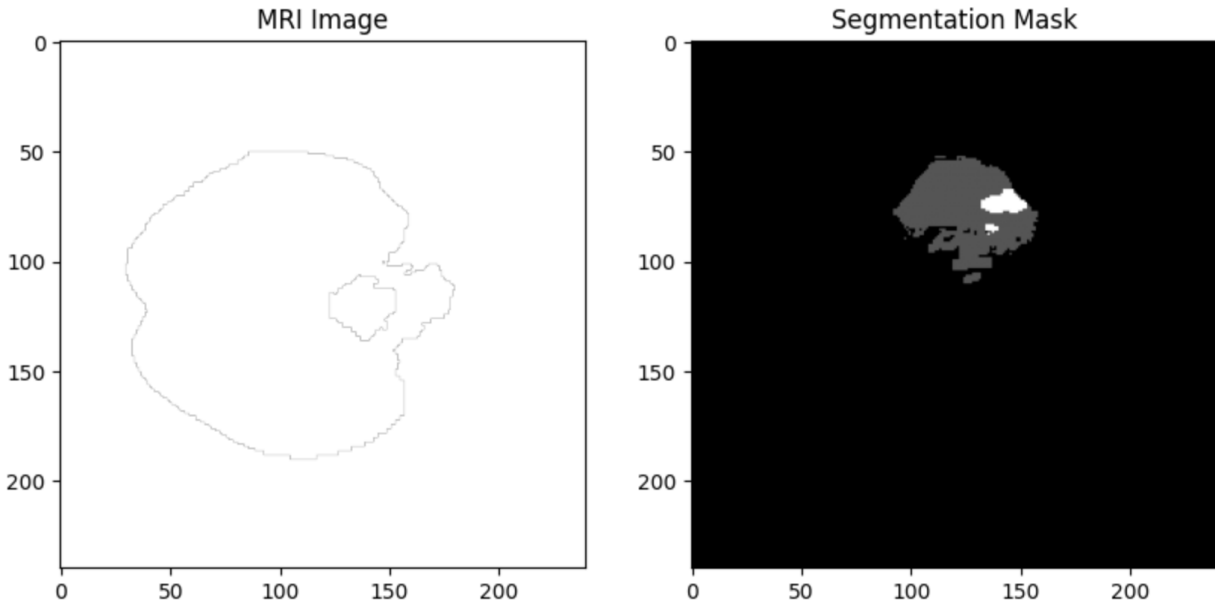


Figure 03: MRI image with the segmentation mask

Model Training with Cross-Validation:

As required by the assignment, 5-fold cross-validation was done to evaluate the model's performance on unseen data and assess its generalizability. The dataset is split into 5 subsets, and the model is trained on 4 subsets and validated on the remaining 1 subset for each fold. This process ensures that every image in the dataset is used for both training and validation, providing a more reliable evaluation. The Adam optimizer was chosen for its adaptive learning rate capabilities and Cross-Entropy Loss function was used, as it is a standard loss function for multi-class segmentation tasks that compute the difference between the predicted and actual class labels at each pixel. The number of epochs that the model was trained for is where the main challenge arises. To experiment with it, it was trained with 25 epochs per fold, then 20, then 10 and 5. Due to the limited computational power, the first three could not complete full execution for the five-folds.

After training, the model was evaluated on the validation set and save the model with the highest Dice Score for the Enhancing Tumor (ET) class. Dice score is a metric for evaluating the overlap between the predicted and true tumor areas, ranging from 0 (no overlap) to 1 (perfect overlap).

Evaluation Metrics:

For evaluation, Dice Score and Hausdorff Distance was used. Dice Score metric evaluates the intersection over union (IoU) of the predicted segmentation mask and the ground truth mask for each tumor class. Hausdorff Distance metric measures the maximum distance between the boundary points of the predicted tumor mask and the true tumor mask. It helps in understanding

how well the model captures the tumor's boundaries. The Hausdorff distance is calculated separately for each class (ET, WT, and TC).

Results and Observations:

After training the model across 5 folds, the evaluation of the Dice score and Hausdorff distance for each tumor class (ET, WT, TC) is conducted.

The Dice score is reported for each class, providing a measure of how well the model identifies each tumor type. The scores for Enhancing Tumor (ET), Whole Tumor (WT), and Tumor Core (TC) are calculated as averages over the validation folds.

The Hausdorff distance is calculated for each tumor class to evaluate how well the model captures the boundaries of the tumors. Lower values indicate better boundary preservation.

Improvement:

This experiment demonstrates the effectiveness of the UNet model for brain tumor segmentation. The cross-validation approach ensures the model's robustness and generalizability, which are critical in medical applications where accurate and reliable segmentation is necessary for diagnosis and treatment planning. The results could be further improved a lot by using a better hardware and having GPU support, which was not feasible during this experiment. It took 8 hours to just run 15 epochs for 1-fold. Having a better device will ensure that the experiments is running efficiently and is not missing out on anything.

3. Segmentation Result:



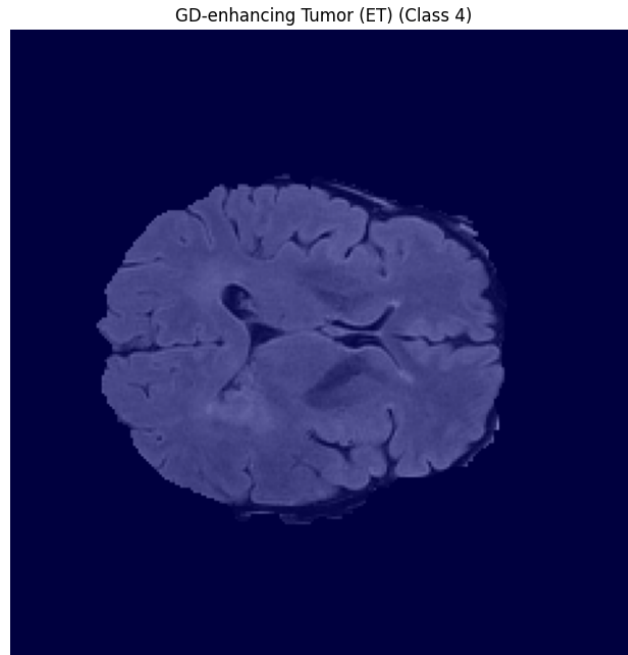


Figure 04: Segmentation Outcome Sample

Fold 1, Epoch 1 – Avg Loss: 0.6424	Fold 2, Epoch 1 – Avg Loss: 0.0780
Fold 1, Epoch 2 – Avg Loss: 0.1096	Fold 2, Epoch 2 – Avg Loss: 0.0780
Fold 1, Epoch 3 – Avg Loss: 0.1023	Fold 2, Epoch 3 – Avg Loss: 0.0763
Fold 1, Epoch 4 – Avg Loss: 0.0993	Fold 2, Epoch 4 – Avg Loss: 0.0768
Fold 1, Epoch 5 – Avg Loss: 0.0996	Fold 2, Epoch 5 – Avg Loss: 0.0742
Fold 1, Epoch 6 – Avg Loss: 0.0952	Fold 2, Epoch 6 – Avg Loss: 0.0746
Fold 1, Epoch 7 – Avg Loss: 0.0945	Fold 2, Epoch 7 – Avg Loss: 0.0725
Fold 1, Epoch 8 – Avg Loss: 0.0960	Fold 2, Epoch 8 – Avg Loss: 0.0711
Fold 1, Epoch 9 – Avg Loss: 0.0944	Fold 2, Epoch 9 – Avg Loss: 0.0719
Fold 1, Epoch 10 – Avg Loss: 0.0947	Fold 2, Epoch 10 – Avg Loss: 0.0705
Fold 1, Epoch 11 – Avg Loss: 0.0925	Fold 2, Epoch 11 – Avg Loss: 0.0734
Fold 1, Epoch 12 – Avg Loss: 0.0916	Fold 2, Epoch 12 – Avg Loss: 0.0688
Fold 1, Epoch 13 – Avg Loss: 0.0889	Fold 2, Epoch 13 – Avg Loss: 0.0676
Fold 1, Epoch 14 – Avg Loss: 0.0897	Fold 2, Epoch 14 – Avg Loss: 0.0694
Fold 1, Epoch 15 – Avg Loss: 0.0870	Fold 2, Epoch 15 – Avg Loss: 0.0670
Fold 1, Epoch 16 – Avg Loss: 0.0839	Fold 2, Epoch 16 – Avg Loss: 0.0688
Fold 1, Epoch 17 – Avg Loss: 0.0839	Fold 2, Epoch 17 – Avg Loss: 0.0654
Fold 1, Epoch 18 – Avg Loss: 0.0840	Fold 2, Epoch 18 – Avg Loss: 0.0666
Fold 1, Epoch 19 – Avg Loss: 0.0819	Fold 2, Epoch 19 – Avg Loss: 0.0671
Fold 1, Epoch 20 – Avg Loss: 0.0779	Fold 2, Epoch 20 – Avg Loss: 0.0660
Fold 1:	Fold 2:
--> Dice Score (ET) : 53.30%	--> Dice Score (ET) : 57.37%
--> Hausdorff Distance (ET): 30.58 mm	--> Hausdorff Distance (ET): 26.54 mm
--> Dice Score (WT) : 7.69%	--> Dice Score (WT) : 16.11%
--> Hausdorff Distance (WT): 46.15 mm	--> Hausdorff Distance (WT): 31.27 mm
--> Dice Score (TC) : 7.69%	--> Dice Score (TC) : 4.19%
--> Hausdorff Distance (TC): 46.15 mm	--> Hausdorff Distance (TC): 36.07 mm

Figure 05: Snippet of the 20 epoch per fold training

Fold	Dice Score (%)			Hausdorff Distance (mm)		
	ET	WT	TC	ET	WT	TC
1	53.3	7.69	7.69	30.58	46.15	46.15
2	57.37	16.11	4.19	26.54	31.27	36.07
3	52.15	15.49	4.07	40.16	46.15	46.15
4	60.73	16.82	7.49	37.32	50.00	50.00
5	72.19	3.02	5.08	35.95	50.00	50.00

Table 1: Fivefold validation with 20 epoch per fold

Fold	Dice_ET	Hausdorff_ET	Dice_WT	Hausdorff_WT	Dice_TC	Hausdorff_TC
1	0.293416	39.111413	7.692308e-02	46.153846	7.692308e-02	46.153846
2	0.312783	38.707197	7.177851e-10	50.000000	1.151813e-09	50.000000
3	0.229068	43.266671	7.692308e-02	46.153846	1.132653e-09	50.000000
4	0.301672	34.685089	7.010233e-10	50.000000	4.464760e-10	50.000000
5	0.412534	30.800910	6.183499e-10	50.000000	5.081168e-10	50.000000

Table 2: Fivefold validation with 5 epoch per fold