Brain Tumor Localization using Deep Neural Networks

Aniket Deo Priya Bharti
50425006
aniketde@buffalo.edu

Atul Arvind Singh
50425045
atularvi@buffalo.edu

. Mohammad Aamir
50424940
mohamma7@buffalo.edu

Prateek Bhuwania
50420033
prateekb@buffalo.edu

Yash Rathi
50366308
yrathi@buffalo.edu

Abstract

Image segmentation is a technique for partitioning a digital image into multiple sets of pixels that have aided in developing a number of clinical applications, such as image processing for evaluating brain changes and identifying diseased areas. In last few decades, different segmentation techniques with diverse degree of accuracy have been designed and documented in the literature. In this project, we investigated the most prominent strategies for brain MRI segmentation, and implemented advanced variants of U-Nets, the capabilities of which have yet to be demonstrated on the **'BraTS'** dataset.

1 Introduction

The advancement in brain magnetic resonance imaging (MRI) has resulted in a large volume of high-quality data. The image segmentation for brain tumor detection is a challenging issue due to their varying structure and appearance in multi-modal MRI. Manual segmentation of brain tumors (Gliomas) requires expert medical knowledge, takes time, and is prone to human mistakes. Manual segmentation could result in inaccurate diagnosis due to human error or lack of expertise. So, building a model that can surpass a trained neurologist in tumor diagnosis would be quite advantageous, allowing for a more accurate, reliable, and consistent approach to disease detection, treatment planning, and monitoring. Improvements in deep learning (DL) and deep neural networks (DNNs) hold immense potential for application in computer-aided brain tumor data interpretation. CNN can learn from examples and achieve innovative segmentation accuracy in 2D and 3D medical picture modalities.

1.1 Image Segmentation

Image (in this case MRI) segmentation is the process of dividing an image into semantically relevant, homogenous, and non-overlapping sections with comparable qualities such as intensity, depth, color, or texture. The segmented picture is either an image of labels identifying each homogenous region or a set of contours describing the region borders.

The segmentation results are then employed in a variety of applications, including anatomical structure analysis, finding pathological regions, surgical planning, and for visualization.

1.2 Problem with image segmentation

The segmentation process is complex since brain tumors vary in size, shape, and location across individuals. This reduces the usability and utility of previous shape and position information, which is commonly employed for robust segmentation of many other anatomical features. The majority of the images derived from a sequence of MRI scans are slices. After segmenting these individual 2D images slice-by-slice we can define data in 3D space, it has been found that segmenting images in 3D space would help to define and segment certain brain diseases with higher accuracy. However, this frequently comes at the expense of increased complexity and longer scanning times for higher resolution image at the expense of patient discomfort. Image acquisition in adult brain MRI investigations is around 20 minutes, but image acquisition time in paediatric MRI studies is limited to 5 to 15 minutes.

2 Dataset

The **'BraTS'** 2020 training dataset is over 33 GB in size and is available as 'NIfTI' files ('.nii.gz'). It contains 369 cases, each with four 3D MRI modalities (native (T1), post-contrast T1-weighted (T1c), T2-weighted (T2), and T2 Fluid Attenuated Inversion Recovery (FLAIR)) that have been rigorously aligned, resampled to '1x1x1' mm isotropic resolution, and skull stripped. The input image dimensions are 240x240x155. There are three tumor subregions annotated: the **enhancing** tumor, the peritumoral **edema**, and the **necrotic** and non-enhancing tumor **core**. The annotations were combined into three nested subregions: whole tumor (WT), tumor core (TC), and enhancing tumor (ET). An example image from the dataset is shown below:

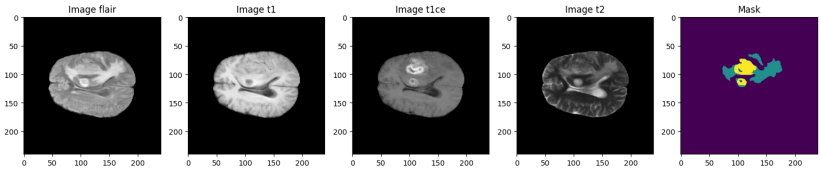

Figure 1: Example image from BraTs dataset

3 Literature Survey

Over the last few decades, there have been numerous studies on automated brain tumor segmentation. Most contemporary techniques employ generative or discriminative approaches. The probabilistic distributions of architecture and appearance of the tumor or healthy tissues are explicitly modeled in generative techniques. They frequently exhibit strong generalization to previously encountered pictures by leveraging domain-specific prior information. Accurate probabilistic distributions of brain tumors, on the other hand, are difficult to model. On the other hand, Discriminative techniques learn the link between image intensities and tissue types directly, and they require a collection of annotated training images.

4 Data Pre-processing

The dataset consists of images in NIfTI (.nii) format. NIfTI is a raster format that requires at least 3-dimensional data in the form of voxels, or pixels having width, height, and depth. The most often used format for multi-dimensional neuroimaging data is NII (or NIfTI) files. We can't read these images like we read JPEG or PNG. Instead, we have to use NiBabel lirary to read the images. Each images consists of 100 slices taken at different time frame. You can visualize this as a GIF file. We compared all the images to find out that the Flair and the T1c images have the most variation, so we chose to work on these images. But since the entire image is in 3-dimensions, we cropped the image to contain just the RoI (region of interest, in this case the brain). Rest of the black portion is eliminated. We then, stack these 2 types of images on each other for every slice. Also, since the dataset has multi-dimensional images, TensorFlow struggles to handle multiple image slices at the

same time, so we created a data loader that stacked the images as described above and adjusted the batch size to 1 so TensorFlow could perform properly. TensorFlow required additional resources for batches larger than one, which were beyond our system resource configuration. We also split the training dataset into Train, Validation and Test set in the ratio 56:20:24 respectively. We also used Min-Max Scaler to normalize the data.

5 Approach

UNet is a relatively new and commonly utilized architecture in the field of neurological research. Our primary goal was to assess the efficacy of various U-Net variations, such as Attention U-Net, Residual U-Net, Dense U-Net, Inception U-Net, RESNET U-Net, and others, in semantic segmenting types of brain tumors. The following is our workflow:

- After preprocessing and developing the data loader, we designed a vanilla U-Net variant to serve as a baseline against which other UNET variants could be compared. We've added a few skip connections, which allows the U-Nets to generate an image in the decoder phase using fine-grained features learnt in the encoder.
- After experimenting with different loss functions such as categorical cross entropy, Tversky loss (specific to Bioimaging Image Segmentation), and others, we discovered categorical focal dice loss (another loss specific to Bioimaging Image Segmentation) to be the Ideal Loss Function after testing multiple model files. We also used IOUScore, MeanIoU, and FScore, as well as accuracy, to assess the efficacy of our model. These parameters play a significant role in the bioimaging area.
- We then took the weights of the pre-trained RESNET, VGG, and InceptionV3, Inception RESNET, and added them to U-Net as backend architecture, then tested them with our loss function and fine-tuned other hyper parameters and trained and evaluated the new trained model.

5.1 Originality

We referred the following research papers as part of originality and worked on implementing the W-Net[4] and U-Net VAE[7] architectures.

5.1.1 WNet[4]

WNet is made up of two UNets: one for encoding and one for decoding. In both the encoder and decoder, the first Unet will optimize Soft Normalized Cut Loss(soft and cut loss), while the second Unet will optimize Reconstruction Loss(rec loss). The structure of the WNet model is as follows: $Encoder \rightarrow Decoder \rightarrow Encoder \rightarrow Decoder$. It has 46 convolutional layers in 18 modules. Two convolutional layers of three are present in each module. The first nine modules comprise the network's dense prediction basis, while the second nine comprise the reconstruction decoder.

5.1.2 UNet with VAE[7]

A simple U-Net can learn conditional distributions over semantic segmentations when used in conjunction with a variational auto-encoder. For ambiguous images, it can simulate complex distributions and provide multiple segmentation hypotheses. The ability to determine the joint probability of all pixels in the segmentation map is one of the architecture's most important features. The network may potentially be able to learn low-probability hypotheses and forecast them at the appropriate frequency. We created a simple U-Net, added VAE with mu and sigma at the end of the encoding stage, reshaped the VAE network output to its prior encoder shape, then decoded the image to get the desired result. However, we were unable to properly implement it. When attempting to train on a merged network, we ran into a tensor mismatch problem and due to time constraints, we were unable to do it properly debug and train the network.

6 Implementation

6.1 Architecture

The UNet is a U-shaped architecture with two symmetric parts. The left side is an encoder with a convolution layer and the right side is a decoder with transpose convolution layers. Our base UNet model takes an input image (size: 128x128x2) and then passes it through the encoder part. In this stage, we have used the ReLU activation function, the padding as 'same', a 3x3 kernel, maxpool, and a dropout for downsampling the input image. This image is then passed into the decoder part that upsamples the image using the Conv2D + upsampling layer with the ReLU activation function. We concatenate the higher resolution feature maps from the encoder part with the up-sampled features in the last layer to learn representations more efficiently in successive convolutions. The activation function in the last layer is softmax for multi-class classification.

We have built six different models:

- **Vanilla UNet:** as mentioned above.
- **U-Net with ResNet:** Traditional neural networks with more layers suffer from the vanishing gradient problem. ResNet has a skip connection that addresses the problem of vanishing gradient.
- **U-Net with VGG:** The VGG19 has smaller kernel that only uses less computation and has higher non linearity. It has a better model fitting ability and ease of optimizing convolution model.
- **U-Net with InceptionV3:** Inception has high performance because it tackles the issue of redundant activation functions in a deep network. So, it provides high performance with little increase in computation requirement.
- **U-Net with Inception and ResNet:** Resnet with Inception enhances single-frame recognition performance in image segmentation for classification problems. It also stabilizes the training of the model through activation scaling.
- **W-Net:** WNET is beneficial in situations where labeled pixelwise supervision is difficult to come by, or when fresh data sets may necessitate extensive re-labeling. Using a depth-wise separable convolution, it examines spatial and cross-channel correlations separately which allows the network to achieve better performance with the same amount of parameters. There are no fully connected layers in the network, hence it can learn images of any size.

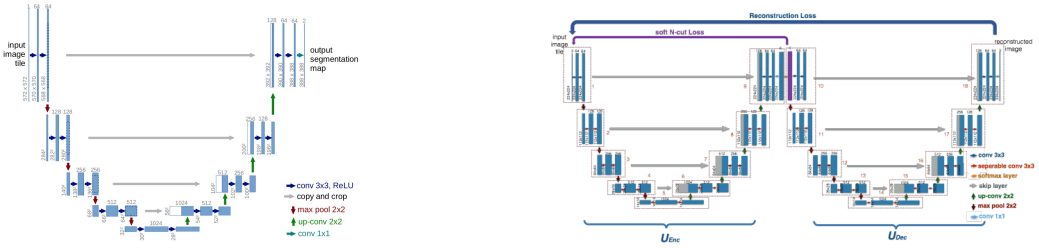

Figure 2: U-net architecture[1] and W-net architecture[4]

6.2 Training the model

After building the UNet model, we trained it for *twenty* epochs using the training set, and evaluated for the validation set. Each epoch took about *45-50 minutes* in the Kaggle notebook, totalling nearly *15 hours* to train a complete model. WNet took even more time to train. After twenty epochs of training, the weights were adjusted, minimizing the loss. We have trained six models on the entire dataset, and it took **over 100 hours** to train and fix errors/issues.

[1] https://medium.com/geekculture/u-net-implementation-from-scratch-using-tensorflow-b4342266e406

6.3 Testing the model

We tested seven models on test set. We measured the performance of several models using four metrics (categorical focal dice loss, Accuracy, MeanIoU, IOUScore, and FScore) and compared them to get insight into their strengths.

6.3.1 Categorical Focal Dice Loss

We use focal dice loss for effective brain tumor segmentation, where there is a substantial class imbalance not only between foreground and background but also across various sub-regions of tumors in brain magnetic resonance (MR) images.

$$loss = \frac{2 * Intersection + Smooth}{sum(Actualvalues) + sum(Predictedvalues) + Smooth}$$

6.3.2 Intersection over Union (IOU Score) and MeanIoU Score

The predictions are gathered into a confusion matrix, which is then weighted by sample weight and used to calculate the measure. It is a method of comparing the similarity and diversity of datasets. The IoU Score compares the similarity of limited sample sets. An IoU of zero means no overlapping. An IoU of one means complete overlap. When there are several classes in an image, we compute the average of all IoU classes in the images using MeanIoU.

$$IoU = \frac{TruePositives}{TruePositives + FalsePositives + FalseNegatives}$$

6.3.3 F-Score

The F-score is a measure of how accurate a test is. It is derived from the test's precision and recall.

$$F - score = \frac{2 * Recall * Precision}{Recall + Precision}, where$$
$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}, and$$
$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

7 Evaluation Results

You can find our code at: **`https://github.com/hipswan/cse676_brain_tumour_localization.git`**

Metric	Simple UNet	UNet with ResNet	UNet with VGG	UNet with InceptionV3	UNet with Inception and ResNet	WNet
Categorical Focal Dice Loss	0.3857	0.3982	0.4783	0.3388	0.4034	0.3367
Accuracy	0.9915	0.9886	0.9856	0.9904	0.9918	0.9912
IoU Score	0.5720	0.5225	0.4430	0.5734	0.5603	0.6061
Mean IoU Score	0.8355	0.4892	0.5227	0.3757	0.8583	0.3757
F-Score	0.6639	0.6109	0.5327	0.6684	0.6628	0.6913

8 Conclusion

We trained our model both on sparse and entire training data to test the full range and capabilities of our models. On sparse training data, Inception-Resnet performed the best in terms of accuracy and IoU score, meanwhile Wnet did not produce satisfactory result. We further extended the training data to entire training data set along with epochs, the Wnet did great recovery and performed on par with Inception-Resnet model with a accuracy of 99.12% and IoU score of 0.60. We believe that using a larger training dataset helped to reduce reconstruction loss and ensure that enough information is kept following the encoding stage.

9 Graph and Plots

9.1 Simple UNet

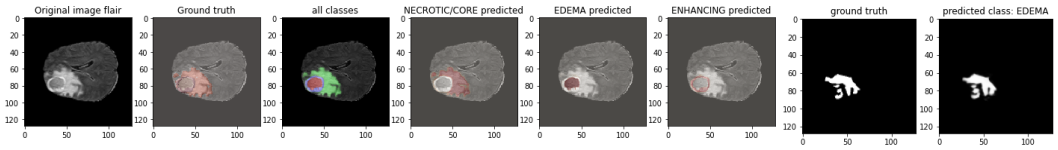

Figure 3: Simple UNet Sample image (First 6 images) and Prediction (Last 2 images)

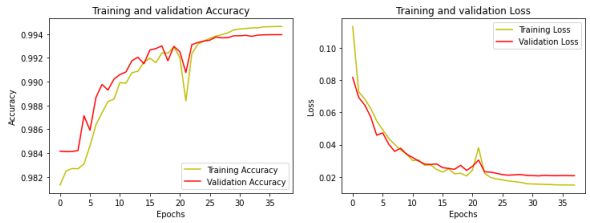

Figure 4: Simple UNet Training and Validation Accuracy (First image) and Loss (Last image)

9.2 UNet with ResNet

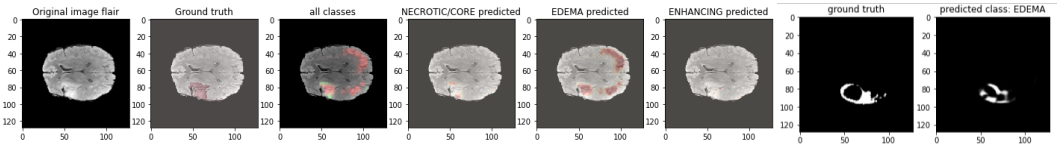

Figure 5: UNet with ResNet Sample image (First 6 images) and Prediction (Last 2 images)

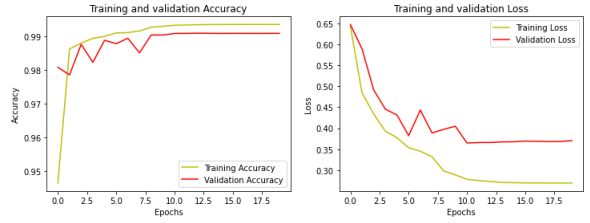

Figure 6: UNet with ResNet Training and Validation Accuracy (First image) and Loss (Last image)

9.3 UNet with VGG

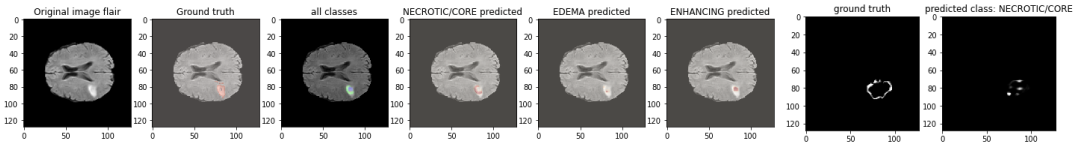

Figure 7: UNet with VGG Sample image (First 6 images) and Prediction (Last 2 images)

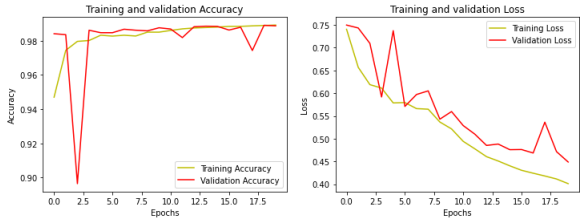

Figure 8: UNet with VGG Training and Validation Accuracy (First image) and Loss (Last image)

9.4 UNet with InceptionV3

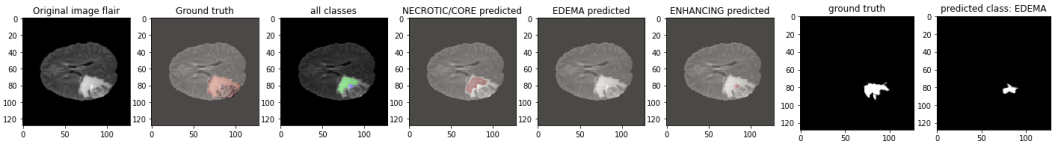

Figure 9: UNet with InceptionV3 Sample image (First 6 images) and Prediction (Last 2 images)

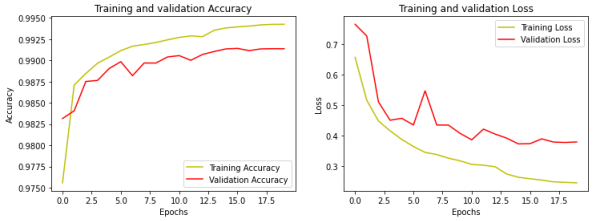

Figure 10: UNet with InceptionV3 Training and Validation Accuracy (First image) and Loss (Last image)

9.5 Unet with Inception and ResNet

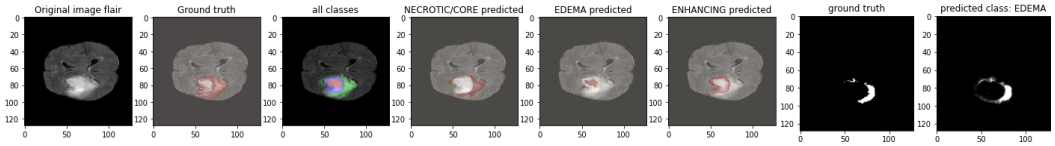

Figure 11: UNet with Inception and ResNet Sample image (First 6 images) and Prediction (Last 2 images)

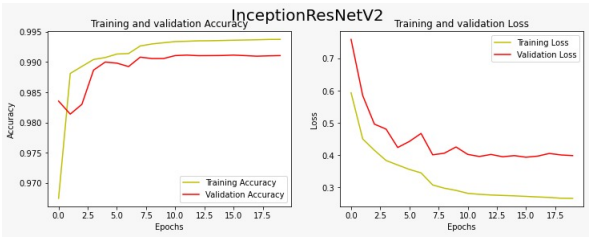

Figure 12: Unet with Inception and ResNet Training and Validation Accuracy (First image) and Loss (Last image)

9.6 WNet

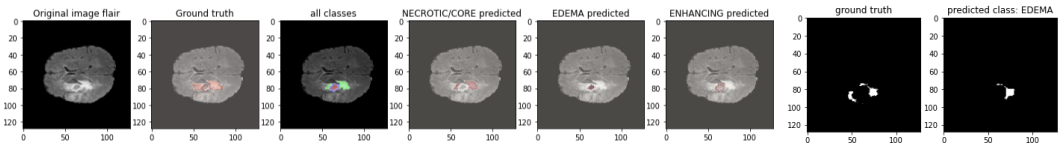

Figure 13: WNet Sample image (First 6 images) and Prediction (Last 2 images)

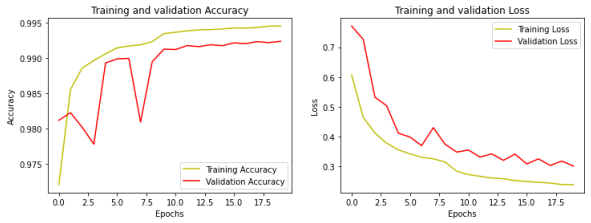

Figure 14: WNet Training and Validation Accuracy (First image) and Loss (Last image)

References

[1] Dana Cobzas et al. "3D Variational Brain Tumor Segmentation using a High Dimensional Feature Set". In: Jan. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4409130.

[2] Raphael Meier et al. "Appearance-and Context-sensitive Features for Brain Tumor Segmentation". In: Sept. 2014. DOI: 10.13140/2.1.3766.7846.

[3] Mohammad Havaei et al. "Brain tumor segmentation with Deep Neural Networks". In: *Medical Image Analysis* 35 (Jan. 2017), pp. 18–31. ISSN: 1361-8415. DOI: 10.1016/j.media.2016.05.004. URL: http://dx.doi.org/10.1016/j.media.2016.05.004.

[4] Xide Xia and Brian Kulis. "W-Net: A Deep Model for Fully Unsupervised Image Segmentation". In: *CoRR* abs/1711.08506 (2017). arXiv: 1711.08506. URL: http://arxiv.org/abs/1711.08506.

[5] Guotai Wang et al. "Automatic Brain Tumor Segmentation Using Cascaded Anisotropic Convolutional Neural Networks". In: *Lecture Notes in Computer Science* (2018), pp. 178–190. ISSN: 1611-3349. DOI: 10.1007/978-3-319-75238-9_16. URL: http://dx.doi.org/10.1007/978-3-319-75238-9_16.

[6] B.N. Sreenu. *(74) 73 - Image Segmentation using U-Net - Part1 (What is U-net?) - YouTube*. https://www.youtube.com/watch?v=azM57JuQpQI&t=5s. Dec. 2019.

[7] Ke Li, Lingwei Kong, and Yifeng Zhang. "3D U-Net Brain Tumor Segmentation Using VAE Skip Connection". In: *2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC)*. 2020, pp. 97–101. DOI: 10.1109/ICIVC50857.2020.9177441.