Brain MRI segmentation with U-Net Concordia University

Manoj Tooprani (40162843) *, Sai Preetham Reddy Veerannagari (40162727) †,
GitHub: https://github.com/Man0j3/Image_processing_project
Email: *email2manojt@gmail.com, †saipreethamveerannagari@gmail.com,

***Abstract*—A recent study found genetic subgroups of lower-grade glioma tumors associated to form factors. In this research, we provide a fully automated method for assessing tumor imaging features using deep learning-based segmentation, and we analyze whether these properties can predict tumor genomic subgroups. Preoperative imaging and genetic data for 110 people with lower-grade gliomas from five different institutes were included in the Cancer Genome Atlas. Using autonomous deep learning segmentations, we identified three features that quantify the two-dimensional and three-dimensional characteristics of the tumors. The genomic data from the patients included previously discovered genomic clusters based on IDH mutation and 1p/19q co-deletion, as well as DNA methylation, gene expression, DNA copy number, and microRNA expression. To investigate the link between imaging features and genetic subtypes, we used the Fisher exact test on ten hypotheses for each pair of imaging features and genetic subtypes. To account for multiple hypothesis tests, we used the Bonferroni correction. P-values less than 0.005 were used to determine statistical significance. The strongest associations were found between RNASeq clusters and the bounding ellipsoid volume ratio (p 0.0002) and RNASeq clusters and margin fluctuation (p 0.005). There were additional connections between bounding ellipsoid volume ratio and all molecular subtypes (p 0.02), as well as angular standard deviation and RNASeq cluster (p 0.02). (p 0.02). Our deep learning approach yielded an average Dice coefficient of 82 percent for automatic tumor segmentation, which was used to obtain quantitative picture features, comparable to human performance. A brain tumor's semantic segmentation is critical for treatment and prevention. Several neural network-based architectures have been introduced to improve the performance of brain tumor segmentation. This paper also proposes BU-Net, a 2D image segmentation method, to contribute to brain tumor research. Residual Extended Skip (RES) and Wide Context (WC) along with customized loss function is used in the baseline U-Net architecture. The suggested BU-Net outperformed current state-of-the-art methods. Researchers in the fields of bioinformatics and medicine can benefit greatly from the high-performance BU-Net. Along with U-Net and BU-Net, we also used ResNet50 for Automatic Polyp Segmentation. Colorectal polyps are frequently overlooked during colonoscopy because of their variety in shape, size, and surrounding tissue similarities. Many colorectal polyps can be easily discovered and eliminated with the help of an automatic, accurate, and quick polyp segmentation approach used during the colonoscopy. We use the U-Net with pre-trained ResNet50 as the encoder for the polyp segmentation.**

***Index Terms*—U-Net; U-net-ResNet50; computational biology; segmentation; brain tumor;Deep learning; brain segmentation; MRI; LGG**

I. Introduction

Imaging can be useful before surgery or in situations where resection is not possible. Recent research in this field has revealed a link between tumor shape features derived from MRI and genetic subtypes. However, manual MRI segmentation was only the initial step in getting tumor features. Deep learning is a relatively new machine-learning concept that has recently revolutionized image analysis. In the field of medical imaging, there are numerous examples of successful deep learning applications, particularly in the segmentation of brain MRIs. In recent years, deep learning for autonomous brain segmentation has progressed to the point that it can match an experienced radiologist's performance. A brain tumor is a growth of cells in the brain that multiplies in an abnormal, uncontrollable way. The histology glioma is the most common malignant brain tumor, and it has three sub-regions: tumor core, augmenting core, and total tumor. Many brain tumor segmentation studies focus on gliomas which are the most common brain tumors in adults. Gliomas are of two types: 1) High-Grade Glioma (HGG) and 2) Low-Grade Glioma (LGG). HGG tumors are malignant because they grow quickly and cause damage to brain tissues. Patients with

this tumor will not survive more than 2 years, so they require surgery. LGG tumor patients can be treated and can increase their life expectancy. Magnetic Resonance Imaging (MRI) is the main tool to monitor and analyze Brain tumors. To visualize the brain, the MRI uses four separate modalities: 1) T1-weighted, 2) T2-weighted, 3) post-contrast T1-weighted, and 4) Flair. Because manually dividing brain tumors is challenging, researchers are working to develop a mechanism for segmenting brain tumor regions automatically. In the medical industry, it's critical to be able to distinguish and interpret malignancies, and a thorough understanding is required. Machine learning has grown more trustworthy and sophisticated than before, thanks to developments in medical image processing. FCN (Fully Convolution Network) and U-Net are the most used deep-learning methods in the field of medical image segmentation. In terms of performance U-Net is the most reliable method. The architecture of the U-Net is a U-symmetrical structure in which left side of architecture performs the encoder task and right side of the architecture performs the decoder task. Another feature of this architecture is that the encoder concatenates the decoder's matching layer. Furthermore, the model's performance is improved by combining features from different levels while retaining location information. The main methods for brain tumor segmentation are 2D segmentation based on slice and 3D segmentation based on MRI. In MRI based 3D segmentation there are only few training data with labels, so it is hard to train the large amounts of data. Havaei et al. [1] proposed a specific multipath convolution neural network (CNN) to segment the brain tumor region on the 2D sliced data of the MRI image. Besides, they used two training steps (phases) to deal with unbalanced classes of input data. Shen et al. developed a boundary-aware FCN to improve the segmentation performance. Later, Kamnitsas et al. [2] developed a 3D network called Deep Medic that extracts multi-scale feature maps and incorporates them locally and globally using a two-path architecture. Based on the limitations in the baseline model of U-Net, we proposed a network named BU-Net. BU-Net introduces two modules which are embedded in U-Net for brain image segmentation. The two modules are Residual Extended Skip (RES) and Wide Context (WC). RES converts low-level features to middle-level features. It increases the valid receptive field which remains the problem in previous techniques. When compared to current state-of-the-art brain tumor segmentation approaches, BU-Net has shown encouraging results.

II. Related Work

FCN, U-Net are the encoder-decoder network used by most of the researchers for semantic segmentation task. U-Net is used for both natural and biomedical image segmentation. The encoder learns and captures crucial semantic information ranging from low-level to high-level using numerous convolutions. These features are then concatenated with the encoder's features using skip connections, and then convolution layers are applied to generate the final output, which is a binary mask. The encoder functions as a feature extractor, and the decoder uses the features extracted from the input to generate the desired segmentation mask. A pre-trained network, such as VGG16. VGG19, and others, can be used to replace the encoder. These pre-trained networks have already been trained on the ImageNet dataset and are capable of extracting features. SegNet and TernausNet are two architectures that use pre-trained VGG16 and VGG11 for segmentation. ResNet50 is one of the most used architectures for any transfer learning task, thanks to the success of the residual network. Two 3 x 3 convolutional layers and an identity mapping are used in the residual network. Each convolution layer is followed by a batch normalization layer and an activation function based on the Rectified Linear Unit (ReLU). The identity mapping is a shortcut connection that connects the convolutional layer's input and output. Identity mapping aids in the construction of a deeper neural network by removing the problem of vanishing and exploding gradients.

III. Dataset

The data for this study came from The Cancer Genome Atlas (TCGA) and The Cancer Imaging Archive (TCIA). We identified 120 patients from the TCGA lower-grade glioma collection who had preoperative imaging data that included at least one fluid-attenuated inversion recovery (FLAIR) sequence1. Ten patients had to be discarded because they lacked genetic cluster information. The final set of 110 patients included patients from Thomas Jefferson University (TCGA-CS, 16 patients), Henry Ford Hospital (TCGA-DU, 45 pa-

tients), UNC (TCGA-EZ, 1 patient), Case Western (TCGA-FG, 14 patients), and Case Western – St. Joseph's (TCGA-HT, 34 patients) from the TCGA LGG collection. The complete list of patients who participated in this trial may be seen in Online Resource 1. The 110 patients were placed into 22 nonoverlapping five-person groups. This was done for the sake of cross-validation. Imaging Data: The National Cancer Institute's Cancer Imaging Archive2 was utilized to collect imaging data for TCGA participants. Except for FLAIR, which was used only if any other modality was unavailable, we used all the modalities when they were all available. There were 101 patients who had all of their sequences, 9 lacking the post-contrast sequence, and 6 missing the pre-contrast sequence. Each patient's available sequences are listed in detail in Online Resource 1. For each subject, the number of slices ranged from 20 to 88. To capture the earliest pattern of tumor growth, only preoperative data was evaluated. Due to the rarity of enhancing tumors in LGG, the tumor shape was evaluated using FLAIR aberrations. To provide training data for the automatic segmentation system, a researcher in our lab who had previously worked in neuroradiology imaging manually labelled FLAIR images by drawing an outline of the FLAIR anomaly on each slice. We used software developed in our lab to accomplish this. A board-certified radiologist double-checked all annotations and changed any that were found to be incorrect. The dataset of registered photos, as well as manual segmentation masks for each case used in our investigation, has been made public. Dataset used in: Mateusz Buda, AshirbaniSaha, Maciej A. Mazurowski "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm." Computers in Biology and Medicine, 2019. and Maciej A. Mazurowski, Kal Clark, Nicholas M. Czarnek, Parisa Shamsesfandabadi, Katherine B. Peters, Ashirbani Saha "Radiogenomics of lower-grade glioma: algorithmically-assessed tumor shape is associated with tumor genomic subtypes and patient outcomes in a multi-institutional study with The Cancer Genome Atlas data." Journal of Neuro-Oncology, 2017. This dataset contains brain MR images together with manual FLAIR abnormality segmentation masks. The images were obtained from The Cancer Imaging Archive (TCIA). They correspond to 110 patients included in The Cancer Genome Atlas (TCGA) lower-grade glioma collection with at least fluid-attenuated inversion recovery (FLAIR) sequence and genomic cluster data available. Tumor genomic clusters and patient data is provided in data.csv file. The dataset is taken from Kaggle [1]. A sample of the dataset used is shown in figure 1.

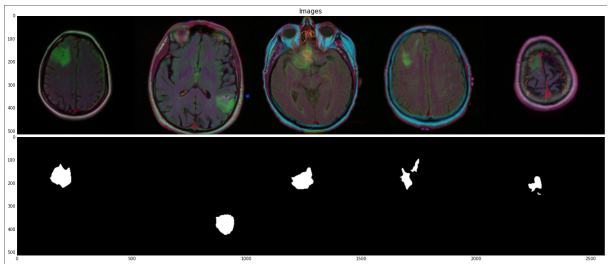

Figure 1. Dataset images sample

IV. Preprocessing

The dataset consist of different categories of patients distributed among different folders. Each patient consist of Brain MRI scan with mask and without mask. The images are segregated into dataframes with mask and without mask. Each images contains its corresponding mask image. A new column diagnosis is created in the dataframe using mask images. Diagnosis value is one if maximum value in numpy array of the mask image is greater than zero. This is shown in the following dataframe figure 2.

	patient	image_path	mask_path	diagnosis
0	TCGA_HT_7616_19940813	/content/drive/MyDrive/kaggle_3m_1/TCGA_HT_761...	/content/drive/MyDrive/kaggle_3m_1/TCGA_HT_761...	0
1	TCGA_HT_7616_19940813	/content/drive/MyDrive/kaggle_3m_1/TCGA_HT_A61...	/content/drive/MyDrive/kaggle_3m_1/TCGA_HT_A61...	0
2	TCGA_HT_7616_19940813	/content/drive/MyDrive/kaggle_3m_1/TCGA_FG_596...	/content/drive/MyDrive/kaggle_3m_1/TCGA_FG_596...	0
3	TCGA_HT_7616_19940813	/content/drive/MyDrive/kaggle_3m_1/TCGA_EZ_726...	/content/drive/MyDrive/kaggle_3m_1/TCGA_EZ_726...	0
4	TCGA_HT_7616_19940813	/content/drive/MyDrive/kaggle_3m_1/TCGA_CS_539...	/content/drive/MyDrive/kaggle_3m_1/TCGA_CS_539...	0

Figure 2. Dataframe columns

The distribution of Brain MRI images based on positive or negative tumours is depicted in figure 3.

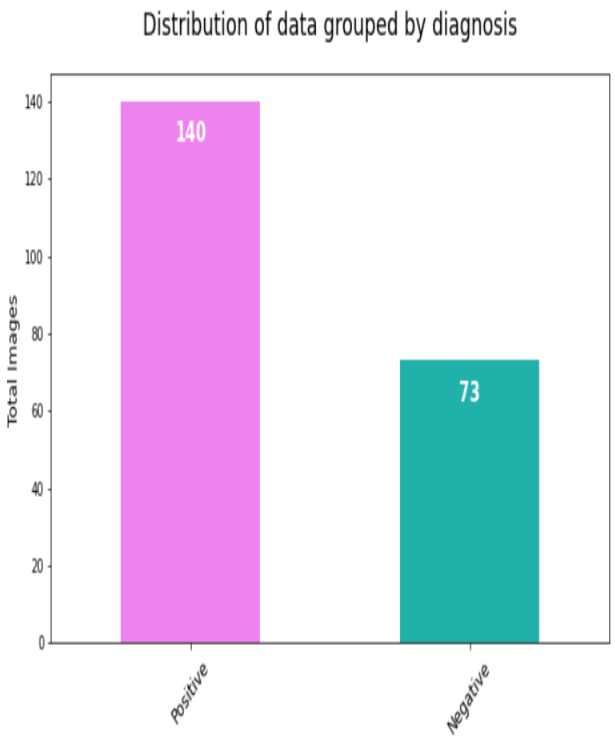

Figure 3. Distribution of data grouped by diagnosis

The distribution of Brain MRI images based on patients is depicted in figure 4.

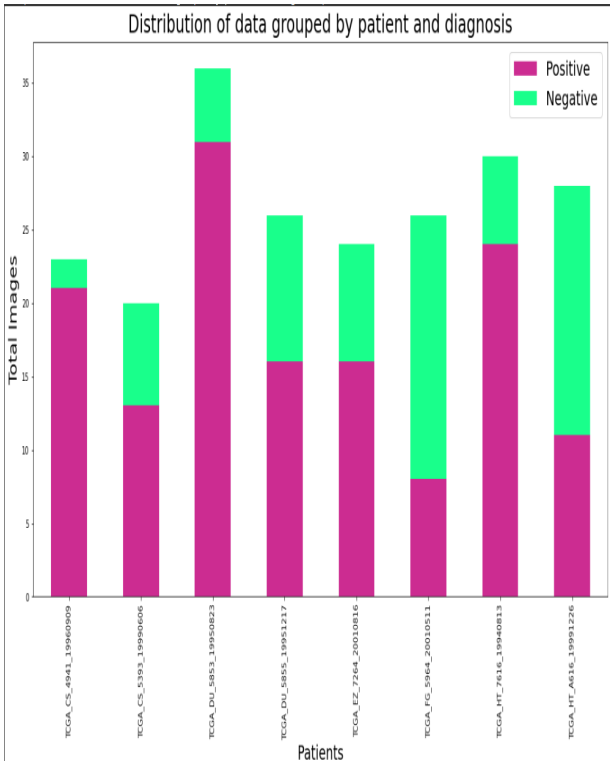

Figure 4. Distribution of data grouped by patient and diagnosis

In the pre-processing step, we use the concept of Albumentation in order to achieve spatial transformations of images. The spatial transforms include HorizontalFlip, VerticalFlip, Transpose, RandomRotate90, ShiftScaleRotate, RandomSizedCrop, Resize and Normalize. Below figure shows all the spatial transformations of the image. few of these operations are shown in figure 5.

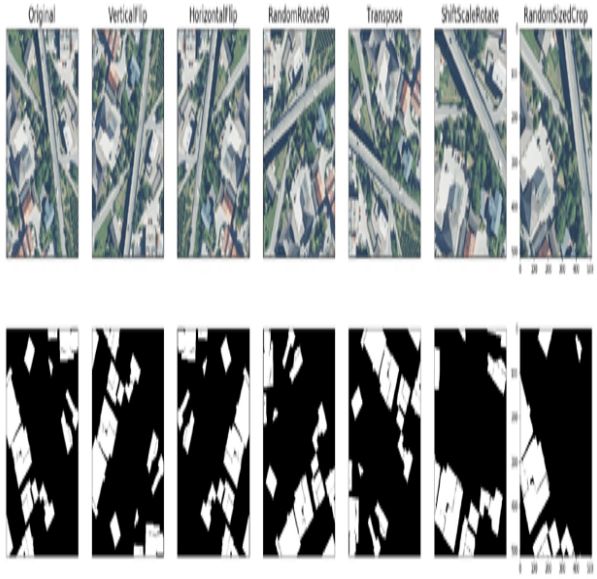

Figure 5. An example of geometry-preserving transforms applied to satellite images (top row) and ground truth binary masks (bottom row) from the Inria Aerial Image Labeling dataset

HorizontalFlip:
In HorizontalFlip transformation, the image is rotated horizontally through the y axis passing through its center.
VerticalFlip:
In VerticalFlip transformation, the image is rotated vertically through the x axis passing through its center.
Transpose:
In this transformation, the rows and columns of the image is swapped.
RandomRotate90:
This transformation helps the image rotate by 90 degrees zero or more times.
RandomSizedCrop:
In this transformation, a random part of the image is cropped and rescale the image to some size.
ShiftScaleRotate:
In this transformation, first we need to translate the image then scale and rotate the image.

Resize:
In this transformation, a particular width and height are provided, and the image will be resized to given width and height.
Normalize:
Normalization of the image can be done by given formula.
img = (img - mean * maxpixelvalue) / (std * maxpixelvalue)
mean, standard deviation (std) and maximum pixel value (maxpixelvalue) are the parameters.
Data split:
The data is split into 90 percent as train data and 10 as validation data. 15 percent of train data is split into 85 percent of train data and 15 percent as test data.

V. Model Architecture

Firstly, in this section, we discuss about image processing, which is useful for input image. Then, we discuss about BU-Net which is used for brain image segmentation along with two modules RES and WC. These two modules are included for better performance. The last architecture used in this was U-Net using ResNet50.

3.1. Image processing

Deep learning models are robust to noise. Hence processing of data is very important before image is given to the network. To make all the images homogeneous, N4ITK algorithm and a bias correction technique is used on all the images. N4ITK algorithm is the most reliable technique for pre-processing of input images. The N4ITK algorithm can correct the bias field of MRI data. All the photos are then standardized to a zero mean with unit variance as a final step.

3.2. U-Net

In previous architecture, no contextual information is shared between the layers. In order to share the information between the layers, a module needs to be introduced., so that all the features in the network can be enhanced. The below figure shows the overall architecture of U-Net which includes RES and WC blocks. The architecture takes the input images of 256 x 256 resolution and outputs the images with the same dimensions. The left part of the architecture acts as an encoder and the right part acts as decoder. In this architecture, convolution layers with padding are used, which allows the same sized output image as given input image. Both the encoder and decoder are divided into number of blocks. Each block of the encoder consists of two convolution layers along with a single max-pooling layer and a dropout layer. Each block on the decoder side starts with Conv2DTranspose layer applied on the output of the previous layer. The output of Conv2DTranspose layer is concatenated with the output generated from the associated RES block. Further, for the transition from the encoder to the decoder, the architecture uses a wide context block. All the convolution layers of BU-Net are followed by batch normalization and ReLU activation function, except for the last convolution layer, which uses a sigmoid activation function. The Keras framework is used to implement the BU-Net. We used hyper-parameter tuning to determine the dropout ratio—a variety of dropout ratios were examined to determine the most ideal dropout ratio; 0.3 was found to be the most optimal dropout ratio for the network. Overall architecture of the proposed U-Net including RES blocks and wide context block is shown in figure 6.

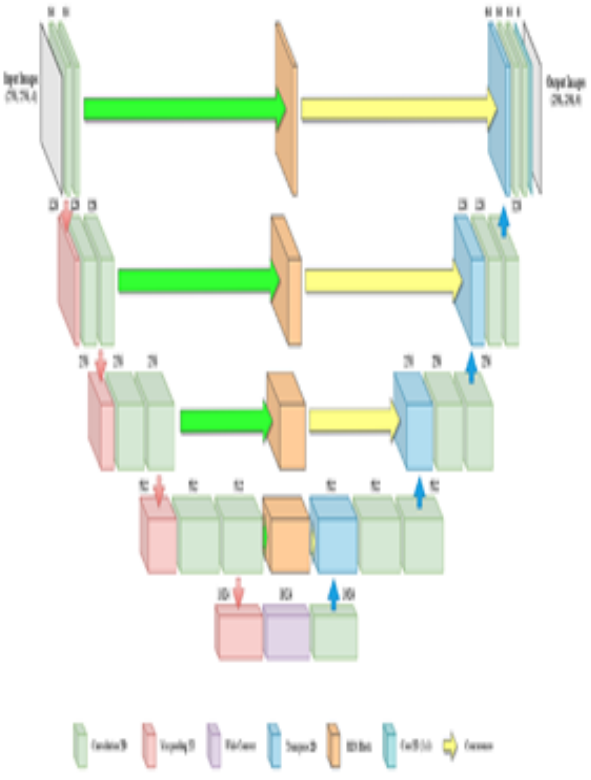

Figure 6. Overall architecture of the proposed U-Net including RES blocks and wide context block

3.2.1 Residual Extended Skip (RES)

The figure 7 shows the architecture of RES block.

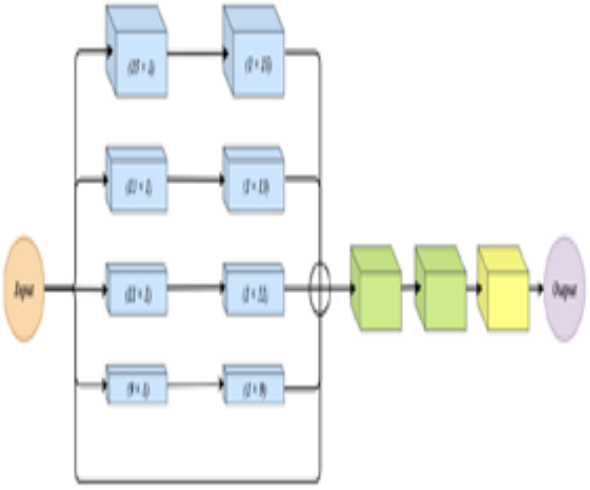

Figure 7. The architecture for Residual Extended Skip (RES) block.

In this architecture, input is given to 5 parallel connections. In the first four connections, two convolution layers are applied. We used N x 1 filter size for first convolution layer and 1 x N filter size for the second convolution layer, in each connection. We used two convolution layers because it generates less number of parameters which benefits the overall architecture. The last connection is a skip connection as it passes the input as it is to the other layers. The outputs of 5 connections are summed up to get a single output with the help of 3 convolution layers placed one after the other. The filter sizes of 3 convolution layers are 3 x 3, 3 x 3, 1 x 1.

3.2.2 Wide Context (WC)

The figure 8 shows the architecture of Wide Context. The input to the architecture is given to 2 parallel connections. Two convolution layers are applied to each of the connection. The convolution layers with filter size N x 1 and 1 x N, respectively are used for first connection. For the second connection the convolution layers with filter sizes 1 x N and N x 1 are used. The outputs of two connections are summed up to produce an output of the WC.

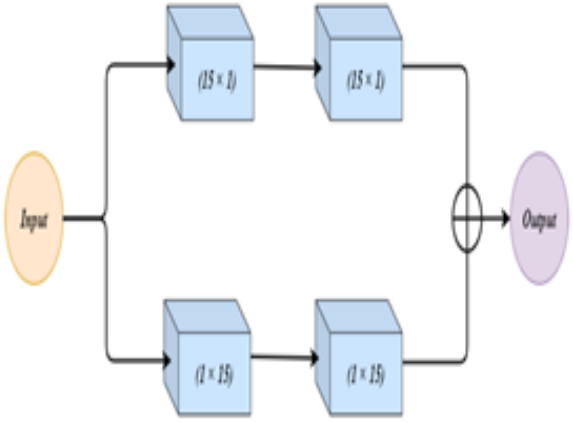

Figure 8. The architecture for the Wide Context (WC) block

3.2.3 Customized loss function

The imbalance class data is one of the challenges with brain tumor segmentation. To address this problem, BU-Net introduced a concept called as Customized loss function. The loss function consists of two objective functions: 1) Weight Cross Entropy (WCE) which oversees classifying tissue cells based on their type. 2) Dice Loss Co-efficient (DLC) which is used to achieve the greatest possible overlap between the ground truth and predicted segmented regions, regardless of class.

The formula for WCE is given in formula 9.

$$WCE = -\sum_{j}^{N} w_j g_j log(p_j)$$

Figure 9. Formula for WCE

The formula for DLC is given in formula 10.

$$DLC = 1 - 2\frac{\sum_j^N w_j g_j p_j}{\sum_j^N w_j (g_j + p_j)|}$$

Figure 10. formula for DLC

where N represents the total number of labels, wj is the assigned weight to the label “j”. pj denotes the predicted binary pixel value of segmented image and gi denotes ground truth binary pixel value of

the segmented image. Customized Loss function is calculated as the sum of WCE and DCL.

L = WCE + DLC.

3.3 U-Net using ResNet50

The figure 11shows the architecture of ResNet50 U-Net. It is also an encoder-decoder based architecture where ResNet50 is used. The input image is passed into the pre-trained ResNet50 encoder, which is made up of a series of residual blocks. The encoder uses these residual blocks to extract the relevant features from the input image, which are then supplied to the decoder. The decoder performs a transpose convolution, which upscales the incoming feature maps to the desired shape. Then, using skip connections, these upscaled feature maps are concatenated with the pre-trained encoder's unique shape feature maps. These skip connections aid the model in obtaining all of the encoder's low-level semantic information, allowing the decoder to generate the desired feature maps. It is then followed by two 3 x 3 convolution layers, each of which is followed by a batch normalization layer and a ReLU non-linearity. The output of the last decoder block is passed to an 1 x 1 convolution layer, which is then passed to a sigmoid activation function, resulting in the desired binary mask.

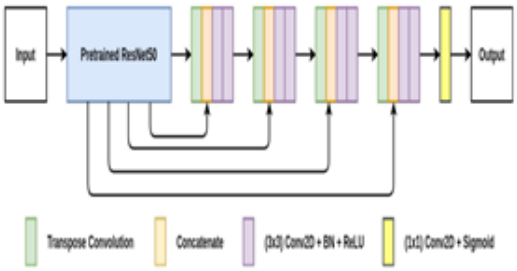

Figure 11. The proposed U-Net-ResNet50 architecture

The architecture followed is shown in the summary by figure 12.

```
UNet(
  (conv_down1): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (conv_down2): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (conv_down3): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (conv_down4): Sequential(
    (0): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (upsample): Upsample(scale_factor=2.0, mode=bilinear)
  (conv_up3): Sequential(
    (0): Conv2d(768, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (conv_up2): Sequential(
    (0): Conv2d(384, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (conv_up1): Sequential(
    (0): Conv2d(192, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
  )
  (last_conv): Conv2d(64, 1, kernel_size=(1, 1), stride=(1, 1))
)
```

Figure 12. U-Net summary

VI. Evaluation and Results

A. System Environment

The model was trained in Google colab software with the following specifications.

- DISC: 128 GB
- RAM: 16GB

B. Hyperparameters

The hyperparameters used in our training model are:

1) Batch size: 40: Batch size is the number of samples used for training in one iteration. In our project, every batch consists of 40 samples which are used for training purposes. Its useful to divide the entire project into batches and keeping the size of one batch as small as possible because it would help the model to run faster.

2) Optimizer: Adam: In our model, optimizer is used to update weights of the layers every time so that the model will make less mistakes while making predictions. Optimizer uses loss function in order to calculate and update weights. Adam optimizer is the common optimizer used in CNN, as it is fast and reliable compared to other optimizers.

3) Loss Function: Cross entropy: Loss function will return the loss value based on the truth value and the model predictions. The loss value which is generated is then taken care of the optimizer to adjust weights in the model. Cross entropy is the loss function which is generally preferred in multi-class classification models. The outputs of cross entropy will range from 0 to 1 for each model class.
4) Number of epochs: 8: Epochs is considered as the number of times the algorithm sees the complete dataset. Each epoch will pass the dataset exactly once to the algorithm. High number of epochs leads to better accuracy of the model because model will be trained a greater number of times.

C. Evaluation Criterion

Dice Score:

Dice score is the measure of how similar the objects are. It is used to increase the performance of image segmentation methods. Dice score is calculated as: Dice score = 2 * number of true values / (2*number of true positives + number of false positives + number of false negatives) Let's consider that we have image A which has values 1 in the pixels where you are trying to find something else zero. Now we have an algorithm to generate image B with same binary values, then the dice score will be as follows: Dice score=(2.|A intersection B|)/(2.|A intersection B|+|B/A|+|A/B|)=2.|A intersection B|/(|A|+|B|)

D. Results:

The graph depicting variation in dice across all the epochs is shown in figure 13.

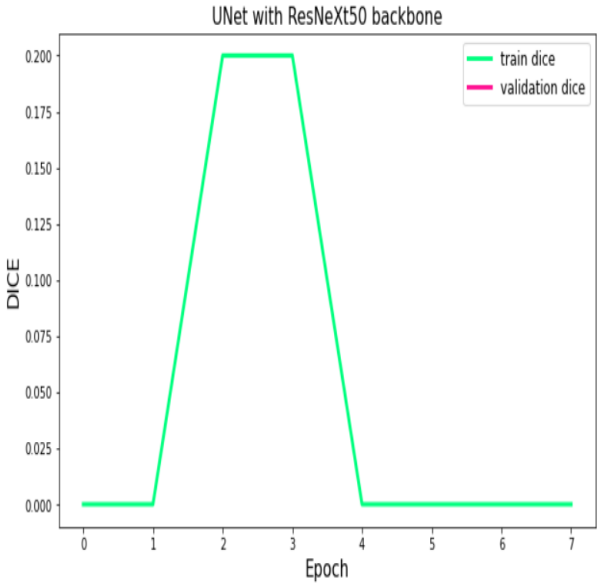

Figure 13. Train Dice of UNet with ResNeXt50 backbone

The prediction of the model on image and its corresponding mask is shown in the figure 14.

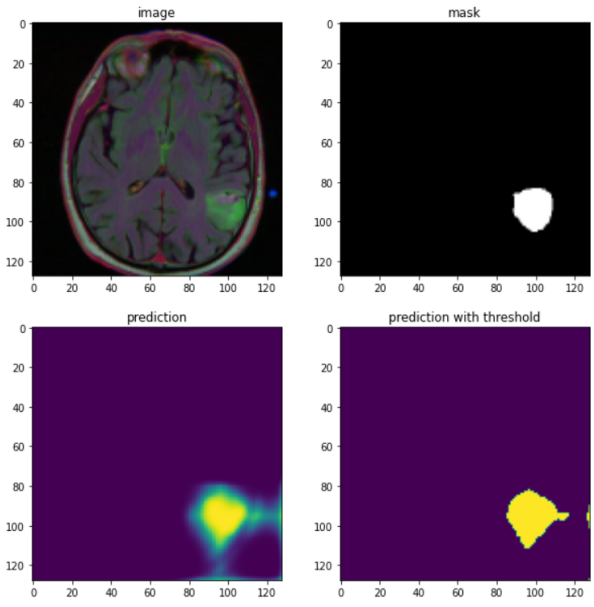

Figure 14. Prediction of UNet with ResNeXt50 backbone

The prediction of UNet with ResNext50 overlaping with ground truth is shown in the figure 15.

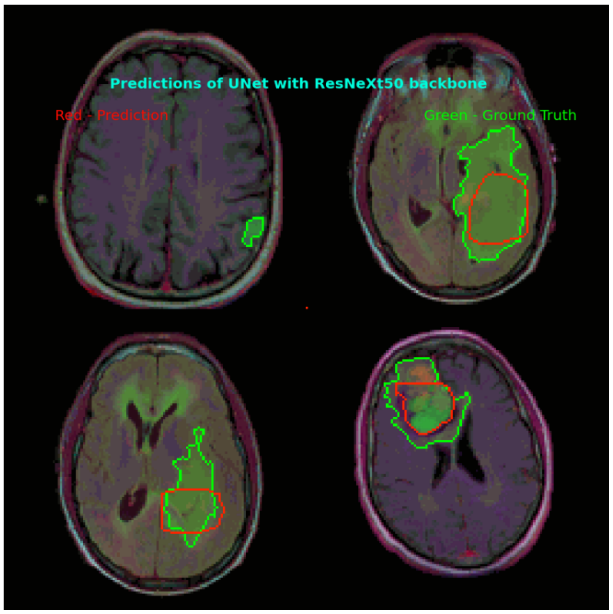

Figure 15. Prediction of UNet with ResNeXt50 backbone

VII. Conclusion

Finally, we showed that MRI features retrieved using deep learning techniques were linked with tumor molecular subtypes of lower-grade gliomas assessed by genomic tests. This indicates promise for non-invasive imaging-based tumor genomics surrogates in brain cancer that are reproducible. We also proposed BU-Net for brain image segmentation. We have introduced two blocks, namely Residual Extended Skip (RES) and Wide Context (WC) to baseline U-Net architecture. An increase in the valid receptive field is achieved using RES block, which improves the overall performance. We also introduced new architecture by name U-Net-ResNet50. We can converge this model in a very short time by replacing the U-Net encoder with pre-trained ResNet50 encoder. Therefore, we can reduce the training time as we are not using encoder from scratch. This is an important step for faster convergence.

References

[1] Kaggle Brain MRI segmentation dataset. [Online] *https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation*.

[2] Brain MRI segmentation reference code. [Online] *https://github.com/RanjanRavi2398/Brain-Tumor-detection*.

[3] Albumentations: Fast and Flexible Image Augmentations. [Online] *https://albumentations.ai/docs/api-reference/full-reference/*.

[4] BU-Net: Brain Tumor Segmentation Using Modified U-Net Architecture. [Online] *https://www.mdpi.com/2079-9292/9/12/2203/pdf*. Rehman, Mobeen Cho, SeungBin Kim, Jee Chong, Kil. (2020). BU-Net: Brain Tumor Segmentation Using Modified U-Net Architecture. Electronics. 9. 2203. 10.3390/electronics9122203.

[5] Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods. [Online] *https://www.sciencedirect.com/science/article/pii/S187705091632587X?via%3*

[6] PyTorch Image Segmentation Tutorial with U-NET: everything (youtube). [Online] *https://www.youtube.com/watch?v=IHq1t7NxS8k*.

[7] Convolutional Networks for Biomedical Image Segmentation. [Online] *https://arxiv.org/abs/1505.04597*. Ronneberger, Olaf Fischer, Philipp Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. LNCS. 9351. 234-241. 10.1007/978-3-319-24574-4_28.

[8] Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. [Online] *https://arxiv.org/pdf/1906.03720.pdf*. Buda, Mateusz Saha, Ashirbani Mazurowski, Maciej. (2019). Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Computers in Biology and Medicine. 109. 10.1016/j.compbiomed.2019.05.002.

[9] Dice score. [Online] *https://stats.stackexchange.com/questions/195006/is-the-dice-coefficient-the-same-as-accuracy*.

[10] Automatic Polyp Segmentation using U-Net-ResNet50. [Online] *https://arxiv.org/pdf/2012.15247.pdf*. Alam, Saruar Tomar, Nikhil Kumar Thakur, Aarati Jha, Debesh Rauniyar, Ashish. (2020). Automatic Polyp Segmentation using U-Net-ResNet50.