Automatic Brain Tumor Segmentation Using U-ResUNet Chain Model Approach

by

Mohd Tanjeem Alam
17101223
Nafisa Nawal
17201075
Nusrat Jahan Nishi
17301070
MD Samiul Sahan
17101504
Mohammad Tanjil Islam
15301110

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Tanjeem

Mohd Tanjeem Alam
17101223

Nafisa Nawal

Nafisa Nawal
17201075

Nishi

Nusrat Jahan Nishi
17301070

Sahan

MD Samiul Sahan
17101504

Tanjil Islam

Mohammad Tanjil Islam
15301110

Approval

The thesis/project titled “Automatic Brain Tumor Segmentation Using U-ResUNet Chain Model Approach” submitted by

1. Mohd Tanjeem Alam (17101223)
2. Nafisa Nawal (17201075)
3. Nusrat Jahan Nishi (17301070)
4. MD Samiul Sahan (17101504)
5. Mohammad Tanjil Islam (15301110)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

Mostafijur
Rahman Akhond

Digitally signed by Mostafijur Rahman Akhond
DN: C=BD, OU=CSE, O=Jashore University of Science and Technology, CN=Mostafijur Rahman Akhond, E=mr.akhond@just.edu.bd
Reason: I am approving this document
Location: your signing location here
Date: 2021.10.05 17:52:12+06'00'
Foxit Reader Version: 10.1.1

Mr. Mostafijur Rahman Akhond
Lecturer
Department of Computer Science and Engineering
Jessore University of Science and Technology

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Identifying brain tumors precisely within the early stage is still a challenging problem for the medical sector consistent with recent research. In a previous research approved by Cancer.Net Editorial Board, it was observed that this year, approximately twenty four thousand five hundred thirty adults will be detected with initial stage cancer tumors of the brain and spinal cord in the United States. So, a developed technology is required to identify this tumor in an early stage to increase the survival rate from this disease. To overcome this problem, many Deep Learning models like CNN (Convolutional Neural Network), LSTM(Long-Short Term Memory) were proposed to detect tumor areas in the primary stage through segmentation and classification in previous research. In our proposed paper, we will attempt to use combination of Res-Unet and Unet model to perform segmentation on brain MRI images. So, basically, our target will be to take brain MRI images as input data and after that, we will try to fit the combination of Unet and Res-Unet model on the dataset to perform segmentation to compare the result with other proposed models to get better result.

Keywords: Brain tumor; Deep Learning; CNN; LSTM; Segmentation; Res-Unet; Unet; Data train-test; Comparison; Result analysis.

Acknowledgement

At first, we would like to proceed with all praise to the Almighty Allah for giving us strength and patience for whom our thesis has been completed successfully without any major interruption.

Secondly, we're grateful to our supervisor Mr. Mostafijur Rahman Akhond sir for his kind support and advice during our work. He helped us whenever we were in need. His support and appreciation have encouraged us a lot to be focused in this work field with patience. Without his proper guidance, we would not be able to continue our research work accordingly. So, his contribution was really the best enlightenment for all of us during this research period.

And finally, we're thankful to our parents without whose support throughout the journey it may not be possible to obtain our goal. So, with their kind support and prayer we are now on the verge of our graduation stage which we hope, is going to be a new journey of our further life.

Table of Contents

Chapter 1

Introduction

The brain is one of the most complicated organs in the human body. It includes about 100 billion brain cells called neurons which are the human body's primary information processing unit, while gila refers to non-neuronal cells that protects and supports the neurons [1]. Uncontrolled development of the human brain cells causes tumors that might be benign or malignant. Benign tumors are considered less harmful as they are non cancerous tumors. They are slow-growing and rarely spread to other regions of the body. Malignant tumors, on the other hand, are extremely dangerous since they are cancerous and rapidly growing tumors. They have a tendency to spread throughout the tumor-affected area's surrounding tissues. Some brain tumors start in the brain called primary tumor whereas others start in other regions of the body and spread throughout the body, including to the brain.They are referred to as secondary tumors. Those who have a family history of cancer are more likely to get a secondary tumor. However, the causes of primary tumors are still unclear [2]. Both types of brain tumors can cause different symptoms depending on which area of the brain is affected. In order to determine the type of tumor and the affected area of the body, medical professionals use a variety of tests and imaging methods.

Different imaging methods, such as MRI and CT scan, are utilized to diagnose brain tumors. However, the use of magnetic resonance imaging (MRI) is very common to aid in diagnosis of brain tumors. MRI helps to distinguish between normal and tumor affected areas of the brain since the scan focuses on those specific areas, making it easier to identify tumor-affected areas. MRI is an image processing technique which was previously known as NMR(Nuclear Magnetic Resonance). A heavy magnetic field and radio waves are used in MRI to generate accurate pictures of the organs and tissues within the body. Basically, it works with strong radiology and magnetic fields which can pass through tissue cells and helps to identify any kind of unwanted cell's growth. In this technique, at first with the help of a radio pulse with proper frequency, we select a particular area of the body and pass the radio frequency and create a magnetic field which can easily find any kind of abnormality inside that selected area. That's why it has become nowadays a revolutionary technique in the medical field to detect health diseases to control it in the primary stage [3].

Brain tumors grow fast in the human body and can spread throughout the body, affecting other organs. Thus, the tumor must be detected as quickly as possible.

In this case, computerized brain tumor segmentation is essential to track tumor progression in the affected area. Brain tumor segmentation is a process of distinguishing and extracting the tumor from normal brain tissues. It is considered one of the most significant and tough processes in many medical-image applications since it generally includes a large quantity of data. The brain tumor segmentation process finds out where active tumorous tissue (vascularized or not), necrotic tissue and edema are located and how far they extend (swelling near the tumor) and this is done by comparing aberrant tissue to normal tissue after finding aberrant spots [4]. Brain tumor segmentation is an important step in the diagnosis of brain tumors since it allows for more precise and faster detection of tumorous cells. In the previous studies various approaches was offered for this purpose. But, deep learning-based frameworks have recently received a lot of attention, as they can reach state-of-the-art outcomes. They have also proven to be highly efficient than other traditional models when it comes to feature extraction, processing and analyzing huge amounts of MRI data [5]. Deep learning based Convolutional neural networks (CNNs), have acquired a lot of popularity in image processing and segmentation. CNN is made up of numerous layers of artificial neurons that reduces complexity of the data leading to less computation with more accuracy [6]. CNN is best suited to situations where the network must deal with a significant amount of data. In brain tumor segmentation a large amount of image-based data is employed to obtain more accurate and superior performance, which is why CNN is the best choice in this case. CNN-based various deep learning approaches like Alexnet, Googlenet, VGG-16,Unet, Resnet were used in many researches for their successful and accurate results in brain tumor segmentation and classification. Among them Unet is primarily introduced in [7] to deal with biomedical images for not only to classify whether there is an infection but also to identify the location of the disease. The Unet used two tracks consisting of encoder and decoder to tackle the problem of class imbalance in brain tumor segmentation [8]. Later on, Residual block of the ResNet network is combined with Unet to pass adequate relevant information through the network and to overcome the gradient loss problem that other convolution models suffer from.The Residual based Unet was first introduced by the author et al. in [9] for the extraction of roads from aerial images. After that, it is used with various networks in various sectors of research. Recently, it has also been used for various medical image segmentation of tumor, heart, liver and skin diseases.

Both Unet and ResUnet not only have acquired success in brain tumor segmentation but also are used in other various sectors of research. In the paper [10], in order to identify various ophthalmological disorders such as diabetes, microaneurysms, arteriosclerosis and hypertension, a Chain Unet with residual Unet is employed for blood vessel segmentation. Moreover, besides brain tumor segmentation, ResUnet and Unet based models achieved tremendous success in segmentation and classification of liver tumors. A dice coefficient of 96 percent is achieved using a ResUnet based model for liver tumor segmentation in the paper [11]. In another research [12], an Unet model with residual attention was tested on MICCAI 2017 data and proved to be very efficient for liver tumor segmentation. According to the author et al., the methodology has significant potential to be applied to other medical fields. Moreover, since tumor segmentation of the thyroid ultrasound images is complicated due to the image's low resolution, a Unet-based framework was used to achieve high accuracy and performance in the segmentation [13]. In recent years Unet and Residual Unet have been employed in a variety of medical fields, along with brain tumor, to achieve more precise and accurate results.

After researching various papers with various models of brain tumor segmentation and classification, we came to an conclusion that the crucial and first step of tumor detection is to distinguish the tumor cells from the brain cell by image segmentation to analyze the tumor growth and to classify the tumor type in order to start appropriate diagnosis according to the segmentation reports. We also understood that for getting results with more accuracy with less amount of data loss, model selection is the most important step. Time is also very important factor in case of identification of tumorous area as some tumor can spread from the brain to all over the body in a very less time. That is why, getting the results in less amount of time is also important. The goal of this research is to accurately segment brain tumor cells by evaluating the proposed Residual Unet model on the BraTS2020 dataset [14] [15] [16] in less amount of time. We've performed a combined Unet and Res-Unet model in our dataset to identify the best accuracy with less data loss to make our model more effective for detecting tumors on the initial stage of it.

Chapter 2

Literature Review

2.1 Literature Review

The paper [17] proposes an automated and modified UNet based deep Convolution neural network to detect the brain tumour more precisely and accurately and also obtained mean dice score of 0.783, 0.868 and 0.805 for enhancing tumor, whole tumor and tumor core respectively after implementation is done on BraTS'2018 training and validation dataset. The model includes a contracting path at the left side and an expanding path at the right side. Other paths are made up of three pre-activated residual blocks, with an extra residual unit serving as a link between them. In all, there are 7 residual models, 25 CNN layers and 15 BN layers in the model. According to the author, the suggested model can produce promising and objective findings that are comparable to manual segmentation results produced by experienced neuro-radiologists.

The network of Separable 3D UNet with separable 3D convolution for automated and more accurate brain tumor segmentation is presented in the paper [18]. In the S3D CNN network, one 2D convolution is used to learn spatial characteristics while one 1D convolution is used to learn temporal features. The R3D UNet network including a contracting path as encoder and expanding path as decoder. The contract path contains five layers and employs downsampling to reduce the amount of the input data whereas the extended path employs upsampling to restore the data's original resolution. On the training data of Brats2018, the proposed model achieved dice scores of 74.93 percent, 89.35 percent and 83.09 percent and on the testing dataset of BRATS2018, it achieved dice scores of 68.94, 83.89 and 78.34 percent for enhancing tumor, whole tumor and tumor core, discretely.

The ContextNet model suggested in the paper [19] is depends on two networks:3D UNet and Residual UNet. To avoid over-parameterization and poor convergence rates, the model employs a residual block based 3D UNet architecture with a Global Planar Convolution module. The model contains residual layers in each layer with additional residual layers after each GPC module, as well as upsampling and downsampling layers and a skip connection, similar to the UNet architecture. On the BraTS 2018 validation and test dataset, the proposed model received dice-scores of 75.2 percent, 89.7 percent and 79.7 percent for Enhancing Tumor, Whole Tumor and Tumor Core, respectively.

The fully convolutional architecture based deep learning approaches have been proposed in the paper [20]. The model is developed with a 3D UNet along with a skip residual connection. The UNet network is made up of four upsampling modules that act as encoders to reduce the size of the input image, resulting in the loss of spatial information which is later recovered by four downsampling modules. The downsampling module acts as a decoder, restoring the input data's original resolution. In order to provide adequate relevant information on the network, residual skip connections are used to integrate feature maps of the corresponding encoder and decoder layers. After performing the experiment, the author concluded that novel training techniques and processes had a greater chance of improving performance than the traditional UNet model and also achieve high mean dice scores of 0.697, 0.828, 0.77 along with Hausdorff95 distances of 25.56, 14.64, 26.69 for enhancing tumor, whole tumor and tumor core, respectively.

A deep residual encoder-decoder CNN included Squeeze-and-Excitation block is introduced in the paper [21] for automatic brain tumour segmentation. The model is basically built based on a 3D ResNet network with encoder decoder, 4 residual blocks and four max pooling layers. After each residual block, SE blocks are added, allowing the picture characteristics to be fully exploited while avoiding the unnecessary information and features. The model is trained and tested on the BraTS2019 dataset and achieved dice scores of 0.70, 0.85 and 0.80 for WT, TC, ET respectively. After adopting 2D,2.5D and 3D CNN models, the research suggested that 3D gives better outcomes and adding SE blocks can enhance the model's performance in identification and segmentation of brain tumors.

The paper [22] proposes a DRRnet network based on the UNet that consists of encoder adaptation blocks, decoder fusion blocks, refine blocks and residual based ResNeXt blocks. The paper has employed the dense based fusion block to increase performance while also reducing computational load. Besides, a 3D form of ResNeXt is utilized in terms of increasing cardinality and making full use of volume images. The author et al. stated that the proposed model is optimized using the BraTs2015 dataset, resulting in dice scores of 0.84, 0.72 and 0.62 for whole tumor, tumor core and enhancing tumor, respectively. However, excessive segmentation is mentioned as a model flaw in the study which will be corrected in future by pre and post processing.

As other traditional convolutional and deep learning networks cause gradient loss which decreases brain tumor segmentation performance and accuracy. In order to solve this issue, a deep residual network based approach is used in the paper [23]. As the Residual network uses shortcut connections and identity connections to the network, gradients can easily flow back which prevents gradients from vanishing. The model is optimized on BraTS2015 dataset and after comparing the performance results of the method with CNN, UNet and UNet-res, the authors determined that the proposed model outperformed them all, with an accuracy rate of 83 percent, 90percent and 85 percentfor the complete, core and enhancing regions, respectively. In terms of computing time, the paper showed that CNN, UNet, UNet-res and the resNet model took 156, 354, 280 and 62 minutes, respectively, implying that the model is three times faster than other networks.

Aboelenein et al. [24] introduced Hybrid two track based UNet model that performs significantly better in brain tumor segmentation than traditional UNet and other networks, which might be very helpful in reducing diagnostic time. The UNet model with two tracks is used to tackle the problem of class imbalance in brain tumor segmentation. Each of the two tracks is made up of encoders and decoders, similar to the traditional UNet which contained convolutional, max-pooling and deconvolutional layers. Moreover, a hybrid loss function consists of the Focal Loss function and the Generalized Dice Loss is introduced to reduce the imbalance rate while also improving the model's accuracy and performance rate. After validating the model on BraTS2018 testing and training dataset and comparing the results with other approaches the paper [24] suggested that the proposed model outperforms other networks with a mean Dice coefficient of 0.745, 0.808, 0.865 and a median Dice coefficient of 0.815, 0.895, 0.883 for the enhancing region, core and whole tumor,respectively.

In the paper [25], the model ResUNet-a,a Residual based UNet, is introduced for semantic segmentation of images which is assessed on the ISPRS 2D Potsdam dataset is evaluated on the ISPRS 2D Potsdam dataset, achieving an average F1 score of 92.9 percent. The traditional Convolutional Layer based model and UNet has the problem of gradient exploding and vanishing. In the model, the traditional UNet architecture is modified by adding a residual block to solve the gradient loss problem. Moreover, a novel variant loss function of dice loss named Tanimoto is introduced in the research paper [25] for getting better result for the speed and performance of the semantic segmentation.

In place of, first order features the paper [26] has used second order modules and residual block based UNet end-to-end model for automatic brain tumor segmentation. The second order features will increase non-linearity and model convergence speed by replacing skip connections of the traditional UNet network. Vanishing gradient is always a crucial problem of convolutional networks. Thus the author et al. used residual blocks in the downsampling part of UNet to prevent the gradient from vanishing and to accelerate network fusion. Moreover, to resolve the issue of class disproportion, a combination of generalized dice loss and weighted cross entropy loss function is utilized in the model. After evaluating the proposed model on both BraTS2018 and BraTS2019 dataset, BraTS2018 achieved a mean dice of 0.873 for whole, 0.808 for core and 0.766 for enhancing tumor while BraTS2019 achieved 0.876 for Whole 0.782 core, 0.717 for enhancing tumor suggesting that the proposed model outperformed the traditional UNet and ResUNet models.

For the volumetric brain tumor image segmentation, the paper [27] proposed a VoxResNet model consisting of VoxRes blocks including 25 volumetric convolutional layers and 4 deconvolutional layers in total. The Voxres block consists of batch normalization, convolutional layers, Relu activation function along with a skip connection from input to output. The batch normalization boosted acceleration and model performance by enclosing additional contextual information for enhancing discrimination capacity. In addition, the convolutional layers lowered input resolution, increasing the field size capacity of the network.After conducting the experiment using MICCAI MRBrainS challenge data, the paper [27] concludes that the proposed model comparatively performed better than other deep learning based models.

A model is introduced to automatically segment the Covid-16 chest CT scan image. The model is an upgraded UNet model that includes aggregated ResNetXT blocks as well as an attention network to enhance the performance and efficiency of the UNet network. The encoder uses convolutional layers whereas the decoder uses non-convolutional layers to do upsampling and downsampling. Furthermore, an attention mechanism is utilized to collect complicated characteristics of the image in order to more precisely differentiate the distinct infection areas of the lung infected by COVID-19. ResNetXt is used instead of a typical ResNet block to minimize the layers number of the model. When compared to U-Net and other models, the suggested technique achieved a 10 percent improvement in multi-class segmentation, according to the author et al. [28].

For colorectal cancer segmentation, the AtResUNet model is proposed in the research paper [29]. The model used UNet as base added with an atrous convolutions and residual connection. UNet is the main architecture of the model and series Atrous Convolutions are included for feature extraction purposes. Furthermore, the Residual network's skip connection facilitates the transmission of sufficient relevant information across the network, ensuring that model performance is improved. The model is assessed based on the DigestPath 2019 Challenge dataset achieving a Dice Coefficient of 0.748. After comparing the results of the model with other deep learning models, the paper suggested that the proposed model has more accuracy compared to other models. The model accrues accuracy near 80 percent which is less than 90 percent and still it is not good enough to segment cancerous areas perfectly and precisely.

In a recent publication [30], it had been identified that they've proposed the LSTM-MA method for brain image segmentation which deals with pixel-wise, superpixel-wise constraint as well. But, after an experimental demonstration, it was found that, in comparison to superpixel-wise constraint, a pixel-wise constraint can give more accuracy during segmentation of a brain image.

Because of limited tissue contrast in tumor sub-regions, automatic segmentation of brain tumors is a difficult task for computer-aided diagnosis. To address this issue, the author et al. [31] suggested a pixel-based technique based on a convolutional 3D to 2D MR patch conversion model which transforms 3D patches into 2D patches after extracting 3D patches from an MRI image and then feeds the 2D patches into 2D kernels for final prediction. Here, both local-inter and global-intra slices are utilized to estimate the class label of the central voxel using a 2D CNN classifier.

The research work [32] proposes a hybrid approach using neutrophils and the Convulsive Neural Network (NS-CNN) to classify tumor regions as benign and malignant. In the proposed method MRI images were segmented using the NS-EMFSE approach. Alexnet then retrieved the features of the segmented pictures from CNN architectures which were subsequently categorized using SVM and KNN classifiers in the classification step. The neutrosophic set – expert maximum fuzzy-sure entropy (NS-EMFSE) approach is used to preprocess the MRI image (T1-GD sequence) and transform the filtered image into a binary before submitting it to the KNN and SVM classifier for segmentation.

After comparing the result of the suggested method with the SVM and KNN classifications, the author et al. [33] determined that the proposed strategy surpassed them all, with an accuracy rate of 99.1 percent in liver tumor classification and 98.6 percent in brain tumor classification. The paper presents a CNN–DWT–LSTM method to classify the computed tomography (CT) images of livers with tumors as benign or malignant and to classify the magnetic resonance (MR) images of brains with tumors as meningioma, glioma and pituitary.In the model, CNN is used to obtain the image's feature vector, then DWT is used to transform the feature vector into a more elaborated, reduced and stronger signal, as well as to detect signal discontinuities and finally, LSTM is used to disassemble the signal and predict a class label.

A customized CNN architecture is used in the paper [34] to classify HRCT lung image patches of ILD patterns. A fully automatic neural-based machine learning framework is designed to extract discriminative features from training data while also performing classification. In the method, image feature extraction is done using convolution and the pooling layer is used for feature dimension reduction. In image classification tasks, one or more 2D matrices are treated as the input to the convolutional layer and multiple 2D matrices are generated as the output. The pooling layer plays an important role in CNN for feature dimension reduction. The publicly available ILD database contains 113 sets of HRCT images are used for evolution to state that the CNN method achieved the best classification performance in comparison to other three approaches (1) SIFT feature with keypoint located at the patch center; (2) rotation-invariant LBP feature with three resolutions and (3) unsupervised feature learning using RBM.

The author et al. [35] proposed a new automated technique for microscopic brain tumor identification and tumor type classification based on 3D CNN and FNN. The architecture of a 3D convolutional neural network (CNN) is created to extract brain tumors before passing the extracted tumors into a pretrained CNN model VGG-19 for feature extraction. The extracted features are then transferred to CbFNN, the method of selection depends on correlation for selecting the best features and finally, these selected features are validated through a feed-forward neural network (FNN) for final classification. The author et al. [35] evaluated the proposed model using three BraTS datasets from 2015, 2017 and 2018 and compared the model's high accuracy and low error rate to other existing approaches, concluding that the suggested model outperforms them.

In the research [36], a Deep learning model based on 3D U-nets was built to perform segmentation of brain tumors using brain-wise normalization and patching procedures, while feature extraction from the estimated tumor label was done to figure out the overall survival days of patients. Normalization was done for each mobility in the pre-processing step and then two patch methods were used to make the GPU capable of dealing with a large image for further processing. Finally, the suggested model is used to complete the segmentation task of brain tumors.Following that, the predicted tumor images would be used to estimate a patient's overall survival days.

In the paper [37], a Convolutional Neural Networks based approach has been proposed for melanoma classification that can help patients and doctors to be able to detect or identify skin cancer classes whether it is benign or malignant.During the preprocessing stage, the images were resized and turned to grayscale to make the process easier for the CPU. After initializing the images and parameters required for the system, the photos are input and stored on the system and then the characteristics of the affected skin cells are retrieved after segmenting the images using the feature extraction approach. Finally, the extracted features are classified using a deep learning-based Convolutional Neural Network (CNN) classifier. After using the publicly available data set, the suggested CNN model obtained an accuracy of 89.5 percent and a training accuracy of 93.7percent.

According to [38], in brain tumor segmentation a magnetic resonance imaging (MRI) based multimodal network, called LSTM multi-modal UNet, consisting of multi-modal UNet and LSTM, performs better than conventional UNet. In order to prove the statement, the author also showed a comparison of outcomes between UNet and the suggested model in terms of dice, Sensitivity and PPV. The proposed model is used to thoroughly take advantage of the interrelationships between modalities and the association between sequences.

In the paper [39], the author et al. have used pre-trained VGG-16 for feature extraction and LSTM for classification. For VGG-16 they have used some common CNN architectures which show the comparisons between layers, input image dimensions, parameters and errors. For LSTM classification they have used RNN neural networks that can model the temporal dependencies. The dataset that was used was named BRATS 2015. A total number of 60 cases was used for this model. Their model has 84 percent accuracy with the VGG-16.

The proposed Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) based modeling is used in the paper [40] to conduct experiments on the available MNIST dataset and IDC Breast Cancer dataset to show that the suggested model achieved better outcomes in image classification in compare to other state-of-the-art classifiers. The images were fed into the LSTM cell after passing through the input layer, Batch Normalization layer and reshape layer. Then, the output is transmitted directly to the convolution layer from the LSTM layer for the extraction of important features. The novel LSTM-CNN based hybrid model has obtained a training accuracy of 99.8 percent and a validation accuracy of 98.2 percent in the MNIST dataset as well as training and validation accuracies of 84.5 percent and 85 percent respectively in the IDC dataset.

2.2 Related Machine Learning Algorithm

i) CNN: Convolutional Neural Network (CNN) is a deep learning-based network that is commonly used for assessing visual data. To compare with other image classification algorithms, CNNs utilize preprocessing very minimally since it learns to improve the filters (or kernels) through automatic learning.

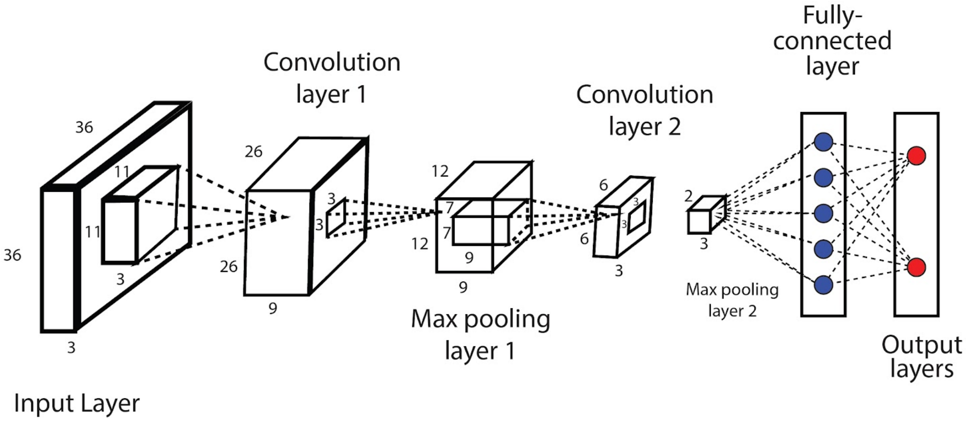


Figure 2.1: CNN Diagram

We can classify convolution neural networks in three steps consisting of an input layer, a convolution layer and lastly an output layer. The first touch point which is an input layer performs several tasks for the convolution layer such as padding, strides. It adds padding around an image so that it doesn't lose the spatial dimension of an image. That means pixels are being added when the image is being processed. Stride is a filter that controls the movement of pixels in an image which shifts pixels in an input matrix. If the number of specified stride is 3 the filter will move the pixel 3 unit. After the padding and strides are applied, convolution is the layer that extracts features from a given input. Convolution then learns from the features that are extracted from the image to detect the object. The kernel will extract features from the input layer which contains pixels of an image height and weights. The kernel then will return the results or outputs which are smaller in dimension [41]. Pooling helps to reduce the image features by downsampling the images. Pooling is done on individual channels by working only with image dimension and width and it does not affect the number of channels. There are several Activation functions which help to decide which information should move forward and which should not. Activation function can be used at the end of the layer or between two convolution layers. Activation function can be marked as linear or nonlinear functions. In terms of performance both linear and non linear gives good performance. We have used sigmoid and RelU activation functions in our model.

Besides the input layer and output layer, a convolutional neural network has a hidden layer that consists of convolution layers, pooling layers, ReLU and fully connected layers. Before transferring the data to the next layer, the convolution layer performs a convolution operation on the input. Afterwards, pooling combines the outputs of multiple clusters of neurons into a single neuron and then each neuron in one layer is coupled to each neuron in the following layer is fully connected layers [42]. The architecture of a standard Convolutional Neural Network might be like this:

Input → Convolution → ReLU → Convolution → ReLU v Pooling → ReLU → Convolution → ReLU → Pooling → Fully Connected.

There are several architectures in the CNN model such as VGG-19, VGG-16, Alexnet, Googlenet, UNet, Resnet.

ii) UNet: UNet is an architecture of Deep Convolutional Neural Network which is specially intended for medical image segmentation. The UNet model is gaining popularity in biological applications since it not only evaluates whether or not there is an infection but also identifies the disease area. It is named UNet due to its U shape which contains two sections: the encoder on the left which is composed of the simple Convolutional process and the decoder on the right, which is composed of transposed 2D convolutional layers [43]. The encoder, also known as the contractive route, functions in the same way as a standard Convolutional model consisting of downsampling steps that double the number of feature channels followed by 3X3 convolution that is applied repeatedly, each followed by ReLU and 2X2 max pooling. On the other hand, the decoder or extensive path consists of upsampling steps followed by a 2X2 convolution that halves the number of feature channels and two 3x3 convolution layers, each followed by a ReLU. Finally, each 64-component feature vector is mapped to the required number of classes using 1x1 convolution layer. The UNet model has 23 convolution layers [7].

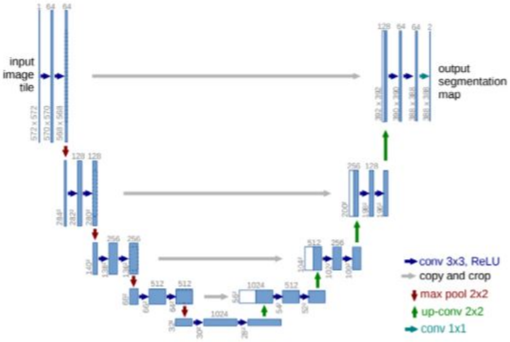


Figure 2.2: UNet Model by Olaf Ronneberger et. el.

Here, each blue box represents a multi-channel feature map whereas white boxes indicate replicated feature maps with the number of channels displayed on top of each box and the different operations indicated by arrows.

iii) ResUNet: ResUNet is the improved version of UNet with additional residual blocks. ResUNet refers to a deep residual Network which is mainly a UNet network with residual blocks. Basically, Zhengxin Zhang et al. [44] developed ResUNet for the road extraction from the Road extraction from aerial images. ResUNet consists of an encoder, a decoder and a bridge connection between the encoder and decoder just like UNet. Only difference is that instead of max pooling operation, ResUNet uses a preactivated. Residual block which helps the network to maintain better flow of data and gradients and also to prevent gradients from vanishing. The encoder has 3 encoder blocks that are made of residual blocks and the output of the encoders act as a skip connection for the corresponding decoder blocks. The bridge is also made from pre-activated residual blocks. The decoder consists of 3 decoder blocks. The decoder takes features from the bridge and skip connections and after concatenating the feature maps with the corresponding skip connection it passes through the residual blocks. Relu activation is used in each encoder and decoder layers whereas the output of the decoder uses sigmoid as activation function and passes through the 1x1 convolutional layer.

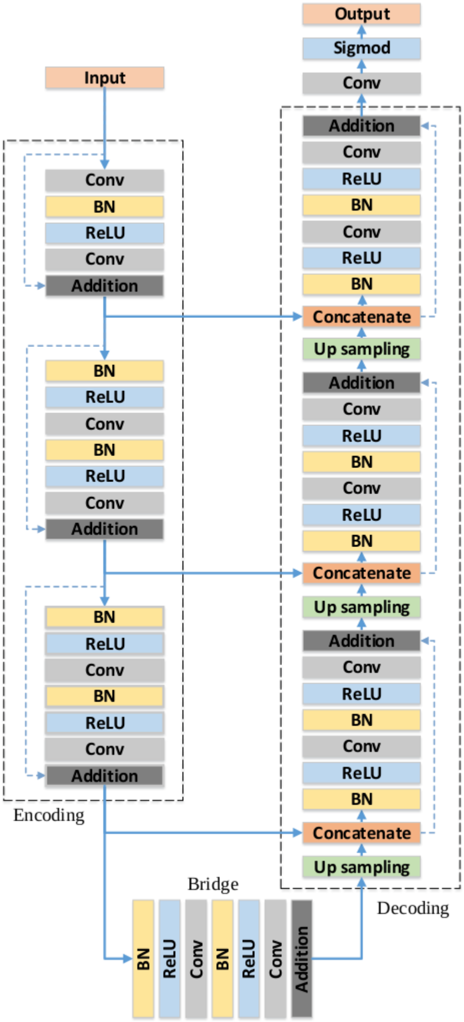


Figure 2.3: ResUNet Model by Zhengxin Zhang et al.

iv) Residual Block: A skip connection based block is referred as a residual block in which the output from one layer is taken and appended to another layer further in the block. The skip connection from input to out works an identity mapping in the architecture. Residual blocks take advantage of residual feature mapping to preserve data and prevent gradient vanishing of image. Multiple convolutional layers can be skipped at once in the residual block, ensuring gradient transmission to deep layers. Batch Normalization is another feature of residual block that prevents gradient disappearing [45].

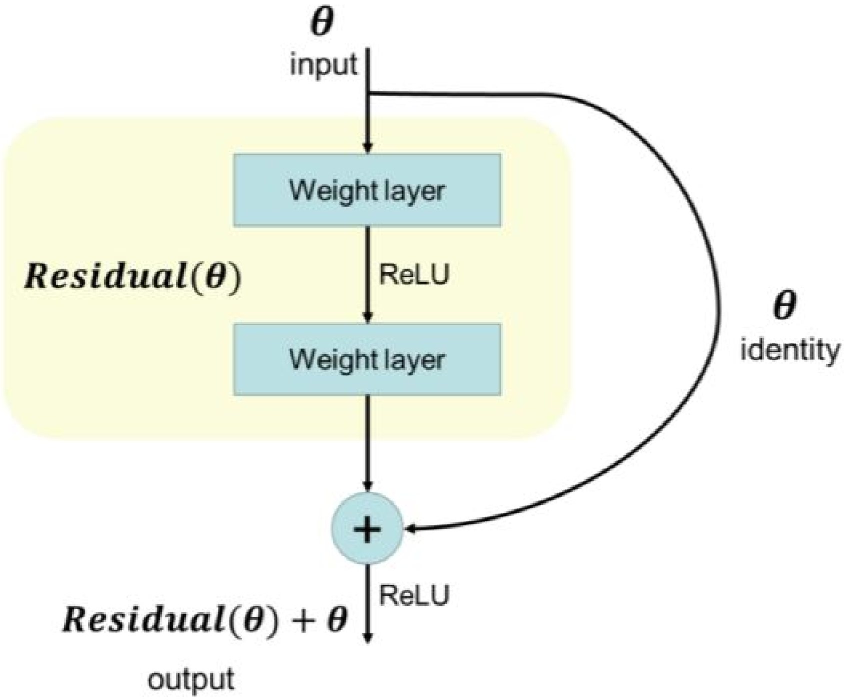


Figure 2.4: Structure of Residual Block

Chapter 3

Working Plan

3.1 Research Plan

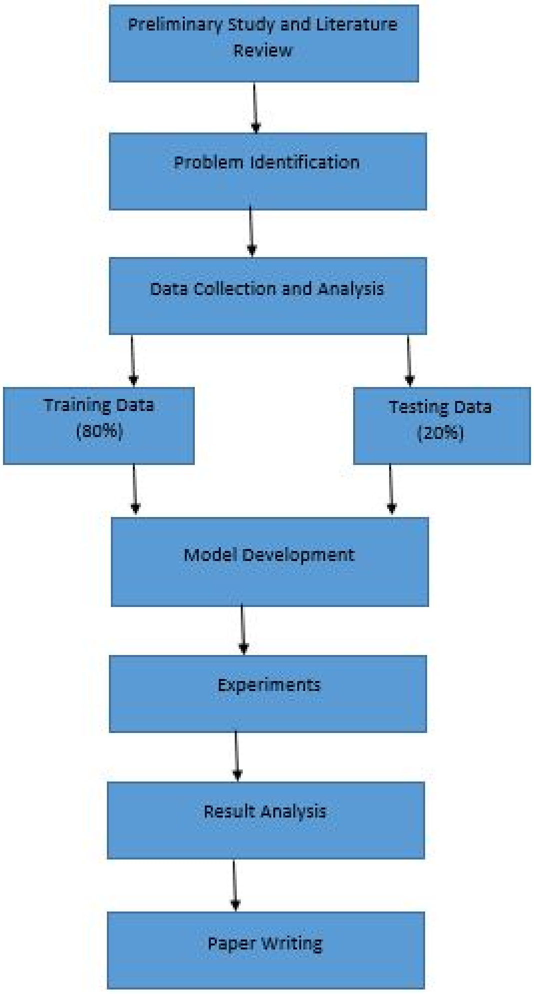


Figure 3.1: FLOWCHART OF THE RESEARCH PLAN

While starting our research work, we decided to follow the above steps to complete our work effectively. At first, we will do some preliminary study and literature review to get some better idea about what was done previously in this field related to our topic. For that, we have to study articles, journals, survey papers, research papers, several book chapters and so on. While doing literature review, we will identify some problems those raised previously during several researches. Some of them can be found solved later and some of them can be still in processing. As, we will do several preliminary study so at that time, we will collect data for our research and after that, will try to analyze those for further processing. Among our collected data, we will divide those into two portion. 80 percent for training data and 20 percent for testing data. Then, we will try to develop our proposed model and after that will start our experiment accordingly. Once our experiment will done, after that we will analyze the result and at last, will start our paper writing finally.

3.2 Methodology

The principal motivation behind our proposed brain tumor identification utilizing the convolution neural network (CNN) model is distinguishing brain tumor at the beginning stage. To do as such, a brain image MRI (Magnetic Resonance Imaging) will be segmented with the help of combination of Unet and Res-Unet model to get the desired output data in our proposed model with high accuracy and less data loss at less number of epoch.

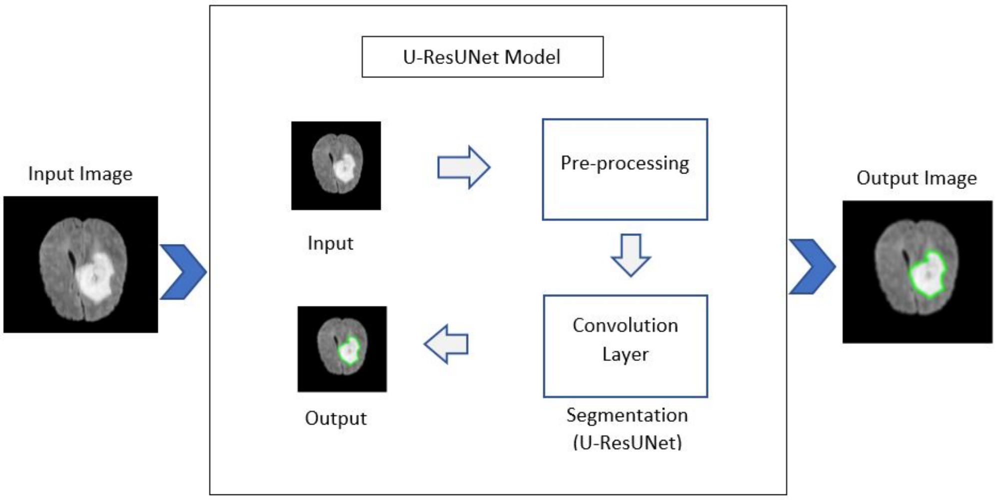


Figure 3.2: BLOCK DIAGRAM OF OUR PROPOSED SYSTEM

The proposed model includes 4 primary stages:

i) Input: The Multi modal Brain Tumor Image Segmentation Benchmark "BRATS Dataset" (2020) [14] [15] [16] is used as input of the model. The BRATS dataset contains 3D volume's, in each volume there are multiple slices of the image. Each images has a dimension. These image slices are in single channel grey-scale. There are 249 set on the training dataset and 45 set of the testing dataset. The dataset also contains survival days for each patients. These MRI 3D images are not in out traditional image format type rather they are in nifty format. We need to convert these nifty format in an array so that we can make a good evaluation process for our model. The dataset has 4 modalities. Then we will load the 3D volumes using nibabel function from keras. Nibabel will read the 3D volumes and convert them into an array. Nibabel converts these 3D slices into 2D slice. Then we can analyze the shapes of each images.

ii) Data Pre-processing: Pre-processing is the technique which helps to make the raw data understandable and readable for the model by reducing noise, duplicate data, handling missing values, moving redundant data and so on, before moving to the further processing. So, this stage is concerned with resizing and formatting the input data in a way that makes it easy for further processing to deal with large size of data set.

```
Path= '../input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData'
p=os.listdir(Path)
Input_Data= []
def Data_Preprocessing(modalities_dir):
    all_modalities = []
    for modality in modalities_dir:
        nifti_file   = nib.load(modality)
        brain_numpy  = np.asarray(nifti_file.dataobj)
        all_modalities.append(brain_numpy)
    brain_affine   = nifti_file.affine
    all_modalities = np.array(all_modalities)
    all_modalities = np.rint(all_modalities).astype(np.int16)
    all_modalities = all_modalities[:, :, :, :]
    all_modalities = np.transpose(all_modalities)
    return all_modalities
for i in p[:20]:
    brain_dir = os.path.normpath(Path+'/'+i)
    flair     = glob.glob(os.path.join(brain_dir, '*_flair*.nii'))
    t1        = glob.glob(os.path.join(brain_dir, '*_t1*.nii'))
    t1ce      = glob.glob(os.path.join(brain_dir, '*_t1ce*.nii'))
    t2        = glob.glob(os.path.join(brain_dir, '*_t2*.nii'))
    gt        = glob.glob( os.path.join(brain_dir, '*_seg*.nii'))
    modalities_dir = [flair[0], t1[0], t1ce[0], t2[0], gt[0]]
    P_Data = Data_Preprocessing(modalities_dir)
    Input_Data.append(P_Data)
```

Figure 3.3: Pre-processing Data

iii) Segmentation: This stage is concerned with segmenting input data using CNN based Unet and Res-Unet model. In this stage, the resized and formatted data fed to the encoding and decoding blocks of the networks for further processing of the input data. At first, the input passes through the encoding decoding blocks of the Unet model. And then the output of the Unet model send to the connected ResUnet for the segmentation. After the input passes through the residual encoder and decoder blocks of the model, we get the final output of the segmented brain tumor image. We has further elaborated the process in Chapter 4 and 5.

iv) Output: After the necessary pre-processing and segmentation of the input data, the segmented processed data is given as output.

Chapter 4

Proposed Model

Our proposed U-ResUnet model uses both the Unet and ResUnet model. First the input goes through the Unet network and the output of the Unet network then passes to the ResUnet network as input. After the upsampling and downsampling of the data in the network, it gives 128x128x4 images as final output. At the very first step, the Unet takes 128x128x2 images as input and after passing all encoding and decoding layers finally gives the output image. The model follows an encode-decoder structure where the encoder path is called the contraction path and the decoder path is called the expansive path. The Unet network consists of 4 encoders, 4 decoders along with a bridge section that connects the encoder and decoder of the network. First, the number of channels changes from 1 to 64 because the convolution process increases the depth of the image. The downward arrow is the max-pooling process that reduces the size of the image in half consisting of stride 2 for downsampling. After that, the process is repeated 3 times more. When we reach the bottom there it contains 2 convolution layers but no max pooling. At the contraction part, we double the number of feature channels. On the other hand, the expansive part which is the decoder will increase the size of the image. Increasing the size of an image is a technique called transposed convolution. After the transposed image is upscaled from 8x8x512 to 16x16x256. Same as before, we will repeat the step 3 times. After we reach the top of the Unet architecture, the output image will then again be sent to the ResUnet as input and the same process will repeat again. But this time the encoding and decoding is done using a pre-trained residual block. The resUnet architecture contains 4 encoders and decoders. First encoder contains the stem block that consists of 2 conv2D layers, 1 batch normalization and 1 ReLU activation function. The existing three encoders are made up of residual blocks. Each decoder has residual blocks and the bridge connection between encoder and decoder is also made up from residual block. Each pre-trained residual block includes two conv2D alongside two batch normalization and two activation function ReLU. Residual blocks contain skip and identity connections that let each gradient flow across the network more efficiently, reducing gradient and data loss. As gradients can easily flow to the deep layers of the network, it speeds up network fusion. After the downsampling and upsampling operation, it gives the 128x128x4 image as the final output of the proposed model.

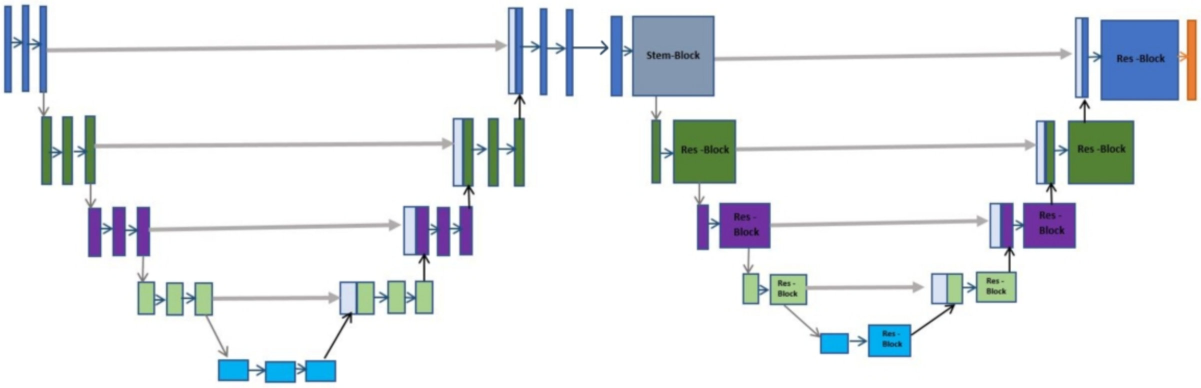


Figure 4.1: U-ResUnet model(Our Proposed Model)

The network architecture of Unet:

The Unet network of our contains 4 encoders and decoders along with a bridge connection. The bridge connection connects both the encoder and decoder of the network. Unet model uses convolution layers and max pooling operation along with ReLu activation function for the segmentation of the image.

Encoder: In the encoder, there are 4 encoder blocks and each block consists of two 3x3 convolution layers along with activation function ReLU. The input data passes through the convolution layers and then the output is transmitted to the next encoder blocks. The output feature maps of each encoder block also transmits into the corresponding decoder blocks using skip connections. A 2x2 max pooling layer is used in each downsampling step to reduce the spatial dimensions and to detect the maximum amount of the features of the features map of the encoder blocks. Each downsampling operation doubles the feature channels while halves the spatial dimensions.

Bridge connection: It connects the encoder and decoder in the network and also passes the feature maps from the encoder to the decoder.

Decoder: Decoder takes feature maps from the corresponding skip connection and also from the bridge connection. 4 decoder blocks consisting of 3x3 convolution layers do the upsampling operation in the model. Every decoder block starts with an upsampling of the feature maps, followed by a 2x2 transpose convolution layer, which reduces the number of feature channels in half. After that these feature maps are concatenated with the skip connection of the appropriate encoder and passes through two 3x3 convolution layers. After completing all upsampling steps, the final output is sent to the connected ResUnet network as an input.

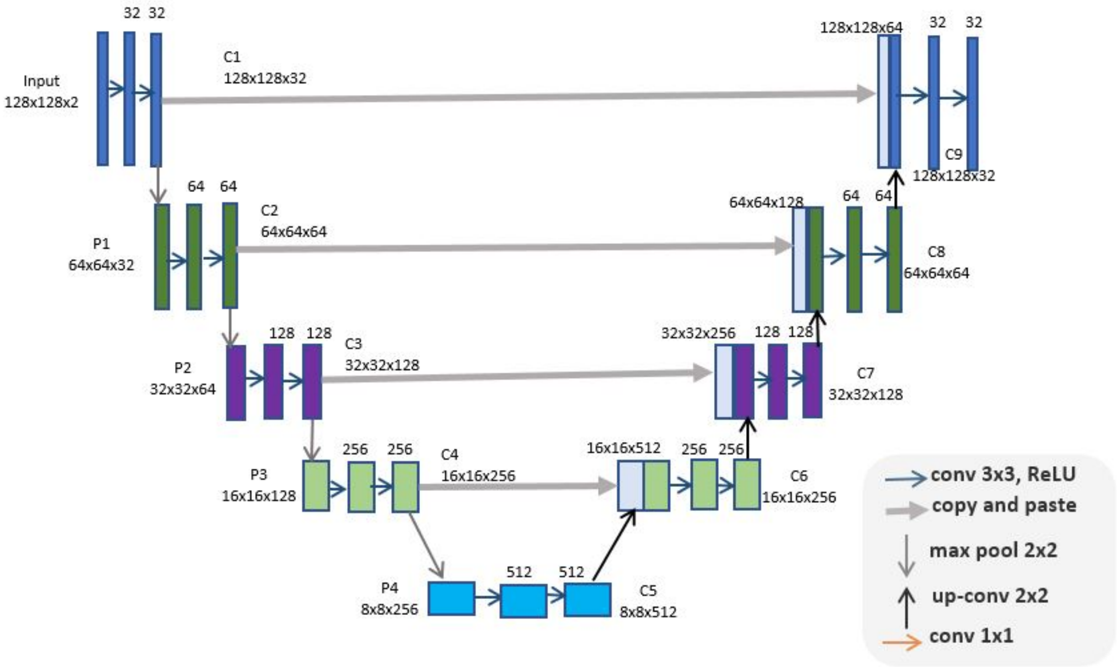


Figure 4.2: Unet Model

The network architecture of ResUnet: A ResUnet network that is connected with the Unet network in our proposed model. We have used the ResUnet architecture in our model to take the advantage of both ResNet and Unet in order to get high performance with fewer parameters. The ResUnet of our model involves 4 encoders and decoders along with a bridge connection that connects the encoder and decoder networks just like the Unet architecture. But the Unet model uses a convolution layer with a ReLU activation function. Whereas in resUnet these layers are replaced by the residual blocks that encompasses convolution layers, batch normalization and ReLU activation function.

Encoder: In the ResUnet, the downsampling operation is done using the encoders. Pre-activated residual blocks are used to construct four encoder blocks of the model. The ResUnet takes the output of the connected Unet network as input and then passes the input through the residual blocks that help the network to learn about the abstract representation of the input data. After passing the input data through each layer of the residual block, an identity function adds the input with the corresponding output before passing it to the next residual encoder block. The final feature maps of each encoder are also transmitted to the associated decoder blocks using skip connections. This skip connection allows data features to be reused while also stabilizing training and convergence. The 3×3 convolution layers in the second, third and forth encoder block utilizes a stride of 2 to decrease the spatial dimensions of the feature maps by half.

Bridge Connection: In the bottom of the ResUnet architecture, a bridge is used to connect both encoder and decoder. This bridge is also made from a residual block that uses a stride of 2.

Decoder: Decoder blocks receive inputs from bridge connections and skip connections from the corresponding encoder blocks to develop a better semantic representation of the feature maps that is used to create a segmentation mask. After each downsampling step, the spatial dimensions of the feature map increases, whereas the number of feature channels decreases. Upsampling is performed by the 4 decoder blocks in the network and each decoder block consists of upsampling and residual blocks. Each decoder block employs 2x2 upsampling, thereby doubling the spatial dimension of the feature maps and after that these feature maps are concatenated with the skip connection of the appropriate encoder. This concatenation step aids the decoder in obtaining the feature mappings learnt by the encoders of the model. After the concatenation is done, the feature map is then transmitted to the residual block of the decoder. Finally, the output of the last decoder passes through a 1x1 convolution layer with Sigmoid activation function to get the pixel wise classification of the model.

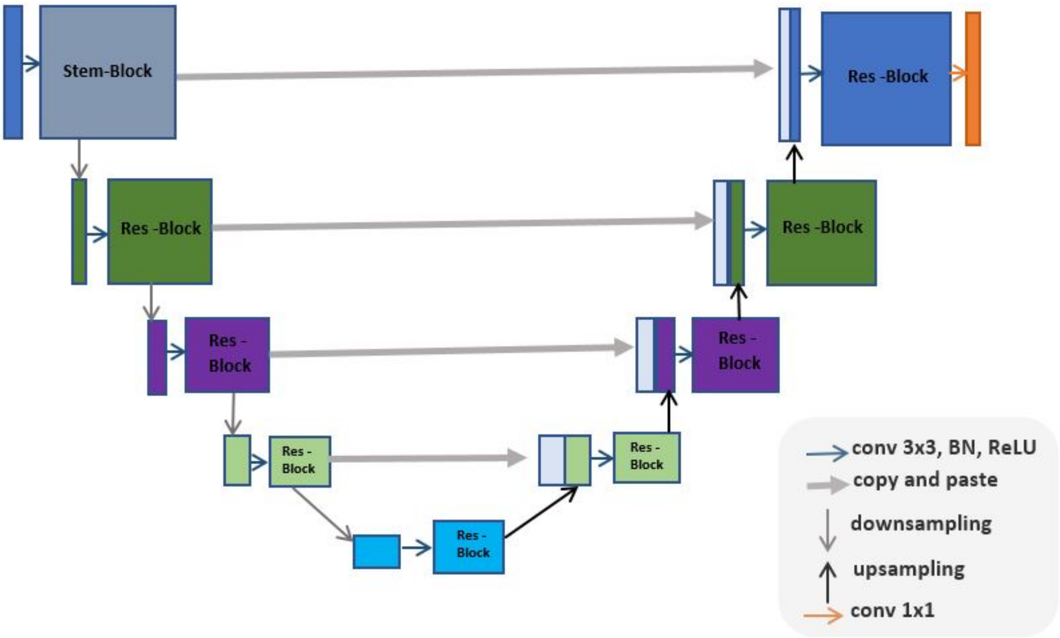


Figure 4.3: ResUnet Model

The network architecture of Residual block: In our model we have used the pre-trained Residual block, a batch normalization and activation function ReLU is followed by a 3x3 convolution layer in the residual block. This is again continued by a batch normalization, ReLU and 3x3 convolution layer. Then an identity connection skips both these layers and adds the input directly with the output of the corresponding residual block before passing the final output to the next residual block of the ResUnet's encoder and decoder. Basically, each layer in the residual blocks feeds into the following layer and also directly into the layers two or three steps away via skip connections. This connection prevents gradients from disappearing by stabilizing gradient updates and preserving the gradient in each layer.

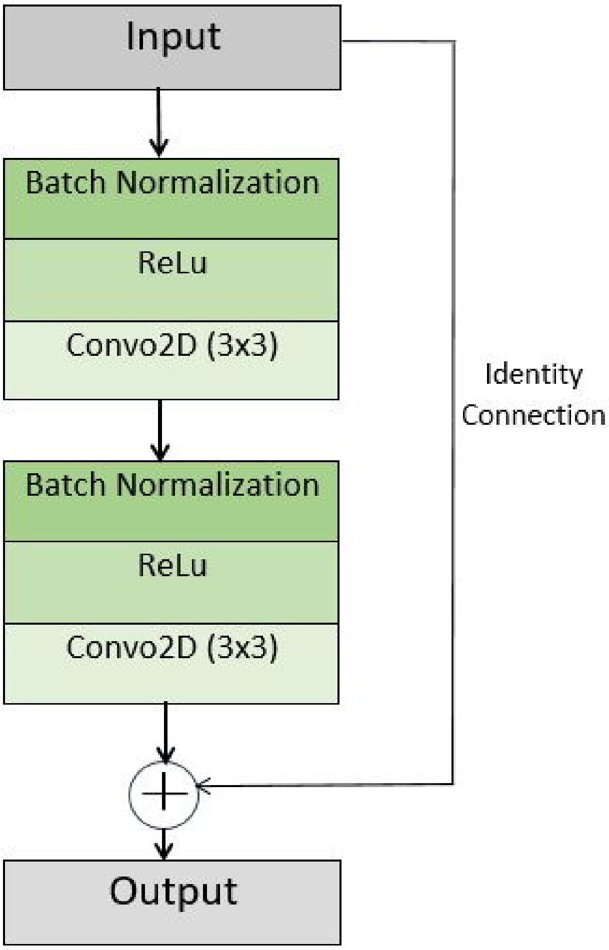


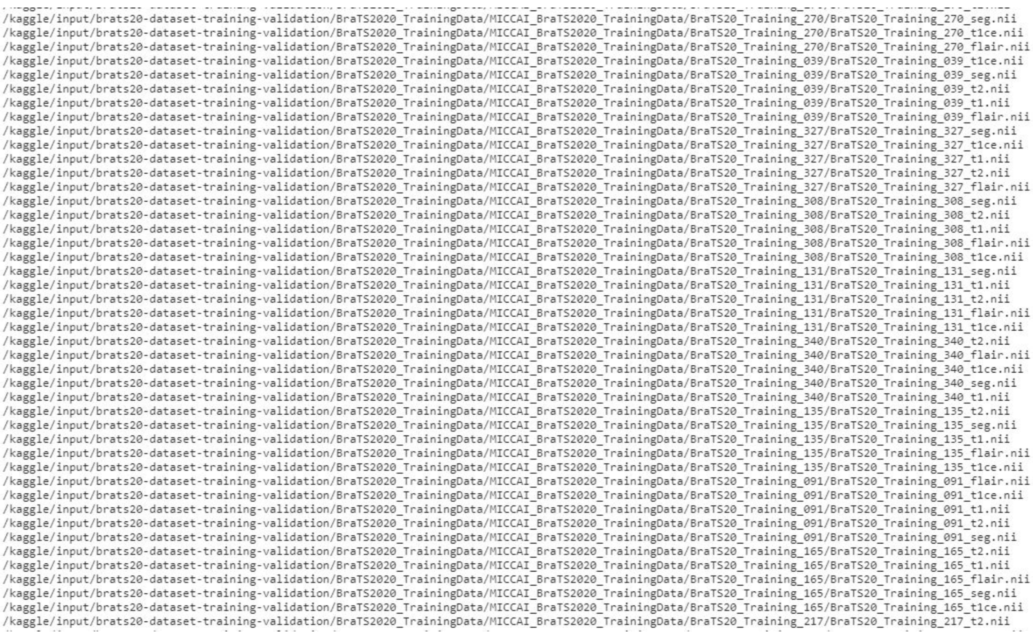
Figure 4.4: Residual Block

Chapter 5

Experimental Setup

5.1 Data Collection

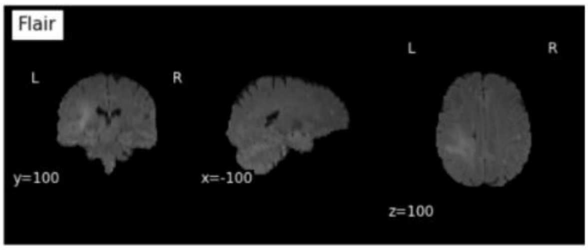
For this experiment, The Multi-modal Brain Tumor Image Segmentation Benchmark "BRATS Dataset"(2020) [14] [15] [16] has been used for model training and testing. This dataset is provided by Medical Image Computing and Computer-Assisted Interventions(MICCAI SOCIETY). The BRATS data set contains 3D brain MRI's and ground truth brain tumor segmentation's collected by physicians which contain 4 MRI modalities per case named T1-weighted(T1), T1ce-contrast (T1ce),T2-weighted (T2) and Fluid-attenuated inversion recovery(FLAIR). These multi-modal scans are in the form of NIFTY files(.nii.gz). This form is commonly used in medical imaging format to store MRI scans and describe different states. This has 3 tumor sub-regions which are the enhancing tumor, edema and the non-enhancing tumor core. These data have been collected from multiple institutions.

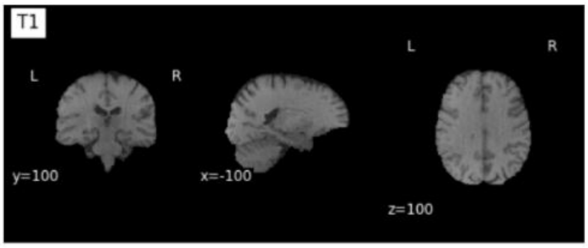


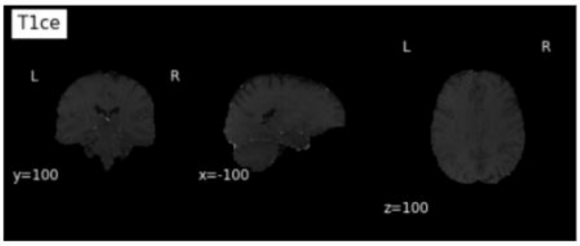
```
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_270/BraTS20_Training_270_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_270/BraTS20_Training_270_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_270/BraTS20_Training_270_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_039/BraTS20_Training_039_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_039/BraTS20_Training_039_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_039/BraTS20_Training_039_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_039/BraTS20_Training_039_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_039/BraTS20_Training_039_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_327/BraTS20_Training_327_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_327/BraTS20_Training_327_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_327/BraTS20_Training_327_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_327/BraTS20_Training_327_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_327/BraTS20_Training_327_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_308/BraTS20_Training_308_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_308/BraTS20_Training_308_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_308/BraTS20_Training_308_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_308/BraTS20_Training_308_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_308/BraTS20_Training_308_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_131/BraTS20_Training_131_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_131/BraTS20_Training_131_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_131/BraTS20_Training_131_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_131/BraTS20_Training_131_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_131/BraTS20_Training_131_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_340/BraTS20_Training_340_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_340/BraTS20_Training_340_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_340/BraTS20_Training_340_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_340/BraTS20_Training_340_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_340/BraTS20_Training_340_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_135/BraTS20_Training_135_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_135/BraTS20_Training_135_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_135/BraTS20_Training_135_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_135/BraTS20_Training_135_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_135/BraTS20_Training_135_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_091/BraTS20_Training_091_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_091/BraTS20_Training_091_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_091/BraTS20_Training_091_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_091/BraTS20_Training_091_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_091/BraTS20_Training_091_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_165/BraTS20_Training_165_t2.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_165/BraTS20_Training_165_t1.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_165/BraTS20_Training_165_flair.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_165/BraTS20_Training_165_seg.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_165/BraTS20_Training_165_t1ce.nii
/kaggle/input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/BraTS20_Training_217/BraTS20_Training_217_t2.nii
```

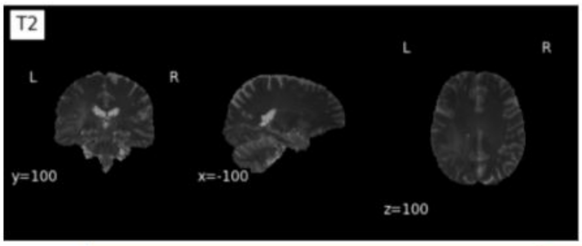
Figure 5.1

The Data set contains training data and validation data. Training data has 369 instances each with 4 MRI modalities per case named T1,T1ce, T2 and FLAIR.The validation data contains 125 instances with 4 MRI modalities. These imaging data set has been segmented manually.









5.2 Data Pre-Processing

Because all information about the brain is transformed into intensity variation, careful consideration of image contrast and size is essential for interpreting or analyzing the images during the tumor identification process. As a result, preprocessing is required to eliminate image noise and undesired labels, as well as to resize and format the data so that it can be used in further processing. In the paper, the author et al. mentioned that pre-processing techniques prepare the image for future processing, improve picture quality and eliminate noise from the image.

In our research work, the dataset we used is saved as a NIFTY format file and the .nii.gz files read into a pandas data frame for preprocessing. To import the .nii.gz format files the nibabel module is used to call all the files at once. To process data from the BraTS dataset NiBabel was used. This function supports the neuroimaging file format. Firstly, To handle the file format and to load the images nib.load() function was called. An image can be loaded by passing the file name and load function.

```
In [5]: TRAIN_DATASET_PATH = '../input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData/'
        VALIDATION_DATASET_PATH = '../input/brats20-dataset-training-validation/BraTS2020_ValidationData/MICCAI_BraTS2020_ValidationData

        test_image_flair=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_flair.nii').get_fdata()
        test_image_t1=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t1.nii').get_fdata()
        test_image_t1ce=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t1ce.nii').get_fdata()
        test_image_t2=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_t2.nii').get_fdata()
        test_mask=nib.load(TRAIN_DATASET_PATH + 'BraTS20_Training_001/BraTS20_Training_001_seg.nii').get_fdata()


        fig, (ax1, ax2, ax3, ax4, ax5) = plt.subplots(1,5, figsize = (20, 10))
        slice_w = 25
        ax1.imshow(test_image_flair[:,:,test_image_flair.shape[0]//2-slice_w], cmap = 'gray')
        ax1.set_title('Image flair')
        ax2.imshow(test_image_t1[:,:,test_image_t1.shape[0]//2-slice_w], cmap = 'gray')
        ax2.set_title('Image t1')
        ax3.imshow(test_image_t1ce[:,:,test_image_t1ce.shape[0]//2-slice_w], cmap = 'gray')
        ax3.set_title('Image t1ce')
        ax4.imshow(test_image_t2[:,:,test_image_t2.shape[0]//2-slice_w], cmap = 'gray')
        ax4.set_title('Image t2')
        ax5.imshow(test_mask[:,:,test_mask.shape[0]//2-slice_w])
        ax5.set_title('Mask')

Out[5]: Text(0.5, 1.0, 'Mask')
```

Figure 5.2: BRATS Dataset import

5.3 Data Segmentation

The process of distinguishing the tumor from normal brain tissues is known as brain tumor segmentation and it is one of the most significant and difficult processes since it generally includes a large quantity of data. According to author et al. in [4], the purpose of brain tumor segmentation is to find out where active tumorous tissue (vascularized or not), necrotic tissue and edema are located and how far they extend (swelling near the tumor) and this is done by comparing aberrant tissue to normal tissue after finding aberrant spots.

Brain tumor segmentation is an important step in the diagnosis of brain tumors since it allows for more precise and faster detection of malignancies. Brain tumor segmentation can be done using CNN-based various deep learning approaches like Alexie, googlenet, VGG-16, Unet. In our paper, after data pre-processing, segmentation is optimized on brain MRI for segmenting tumors from normal brain tissues. In the segmentation CNN based U-ResUnet is applied for better performance and accuracy rate. The both U-shaped model Unet and ResUnet is symmetric and consists of two key parts: the encoder on the left and the decoder on the right [5]. The proposed Unet with ResUnet model helps to achieve good efficiency for biomedical images.

After importing the 3D slices from the BraTS dataset in the pre-processing step, we are showing segments of tumors with nlplt function in figure 5.3 and figure 5.4 .

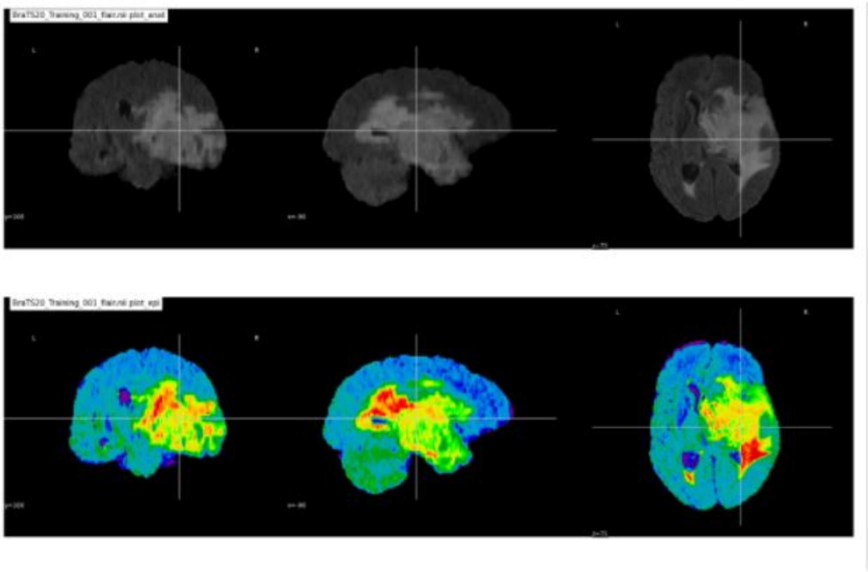


Figure 5.3

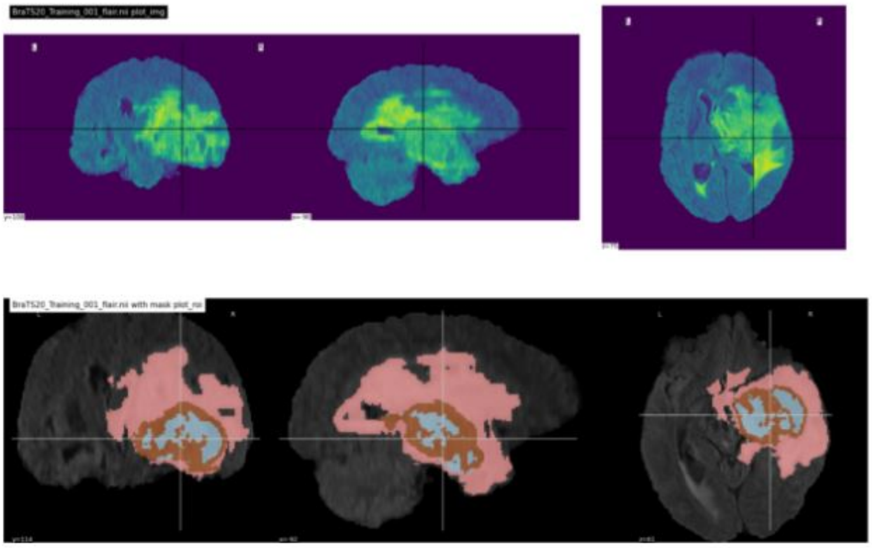


Figure 5.4

5.4 Train-Test Split

For the Independent features "X" and for dependent features "Y" has been added. These are divided into 4 portions for training and testing the brats dataset. The dataset has been divided into 15percent for tests and 80percent for training purposes.
Training Data: The dataset contains 249 patients MRI images. These 3D images were converted then fed into the U-ResUnet model to learn segmentation of the tumor. We kept the training data to 80percent for better segmentation. For the input data, we used (128, 128, 2) image size for training data.

Testing Data: The dataset contains 70 patients MRI images which are labeled as validation data. We converted them into an array and fed them into the Res-Unet model for testing. We kept the test size to 20 percent.

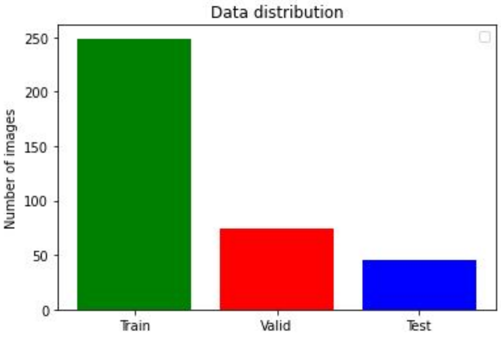


Figure 5.5: Data Distribution Graph

5.5 Model Training

For our experiment, Unet and ResUnet were used. ResUnet refers to Residual Unet. Both Unet and ResUnet is the convolution neural networks that works with encoder and decoder. Unet and ResUnet architecture has performed in training data and predict the machine learning model using BRATS dataset to attain high accuracy and efficiency.

The Unet contains encoder, decoder and bridge. Bridge works as a connector between two networks. First, the input data passes through the encoder blocks during the downsampling step. Then the output is sent to the decoder from the encoder using both skip and bridge connections. Finally, when the upsampling operation in the decoding network is completed, Unet's final output is sent to ResUnet as input.

The Res-Unet contains a total of 3 parts: an Encoding network, Bridge and Decoding network, just like Unet. The bridge connects these 2 networks [46]. The encoding network takes the input as an image and passes it to later encoding blocks. Firstly, the encoders take the input as an image and forward it to later blocks. The encoder has a residual block activated previously. Each encoder output will connect to its corresponding decoder output. In each layer, the spatial dimension of the images decreases by half. To do this strides=2 was applied. Upon applying strides 2 in the first layer the spatial dimension decreases from 256 to 128. The bridge uses a stride value of 1. Later the bridge passes the feature maps to the decoder. The decoder also takes the skip connections from the encoder which then generates a segmentation mask. After each decoder block, the spatial dimension gets doubled which is upsampling.

We have used different activation functions for our input layers and output layers. To activate the input layer ReLU activation function has been applied. We applied the sigmoid function for the output layer. To train our model, MRI images were fed into the neural network, and it shows segmented images. We have experimented with different epochs. We used several epochs to train our model so that the precision can be optimal and the model can have efficient learning. Our batch size is 32 based on the data available in the dataset. It takes samples from the dataset and works in iteration. After each iteration, it evaluates the values and updates the model.

5.6 Model Optimization

To optimize our model we have used several Hyperparameter to tune our model training. A model learn these value as parameter when we train a machine learning model. Upon learning these values During the training period a model, these value controls how the model will behave.
The parameters that we modified to optimize our model to keep away from model over-fitting and under-fitting are given below:

```
model.compile(loss=”categorical_crossentropy”,
optimizer=keras.optimizers.Adam(learning _rate=0.001),
metrics = [’accuracy’] )
```

Batch normalization: Batch normalization is basically used in neural networks to increase accuracy as well as speed of training data by normalizing activation inside intermediate layers [47]. Among many characteristics of batch normalization, one of the most efficient features is boosting up the speed training data which occurs by reducing Internal Covariate Shift(ICS) which works through distributing each activation layer value by forcely making mean value zero and unit variance in each layer’s distribution while performing deep neural network models [48].

Adam Optimizer: We have adopted the adam optimizer algorithm to optimize our model training. Adam refers to adaptive moment estimation [49]. Adam optimizer uses the stochastic gradient descent method which is different from the classical gradient descent method. It follows a learning rate which is (lr=0.0010) called alpha. It controls the learning rate for weight updates and keeps the alpha rate the same when the model runs. For multiple parameters, it calculates specific learning rates from evaluations which are first, second-order moments of the gradient. In terms of memory, Adam uses very small memory resources. The equation for the adam optimizer is shown below:

Here the hyperparameter that controls exponential decay rates of moving averages are β 1, β 2 [0, 1).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{5.1}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{5.2}$$

$$\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_t}{\sqrt{(\hat{v}_t + \epsilon)}} \tag{5.3}$$

Activation Function: The usage of this function is adding the non-linearity expression to a network so that the network can learn the complicated pattern of the data and can improve the performance as well as accuracy of the model [50]. The activation function solves nonlinear issues and improves the network's learning capabilities [51].

ReLu Activation Function: Rectified Linear Unit is one of the simplest nonlinear activation functions that is used for achieving non-linearity of the data [52].
i) Relu is fast to compute.
ii) It doesn't cause gradient loss of the input data.
iii) It speeds up the models' learning as well as training process.
iv) It generates the value between 0 to 1. If the output is positive it will generate the value for the output. Otherwise the output will be 0.

$$f(u) = \max(0, u) \tag{5.4}$$

$$ReLu(u) = \begin{cases} 0 & \text{if } u < 0, \\ u & \text{if } u >= 0, \end{cases} \tag{5.5}$$

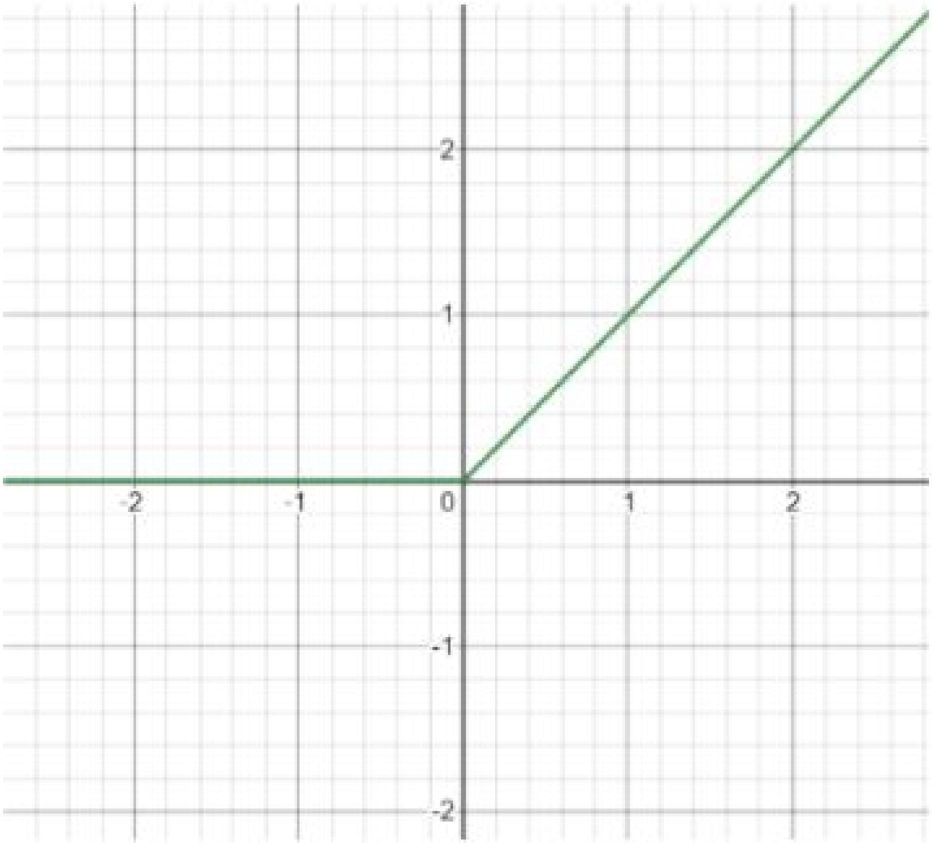


Figure 5.6: ReLU activation function graph

Sigmoid Activation Function: Sigmoid activation function is used to introduce nonlinearity in the network. Its shape is like an “S” in the graph. The sigmoid function is usually employed in the output layer of a neural network for binary classification since it has a range of values from 0 to 1 [53].

$$f(u) = \frac{1}{1 + e^{-u}} \tag{5.6}$$

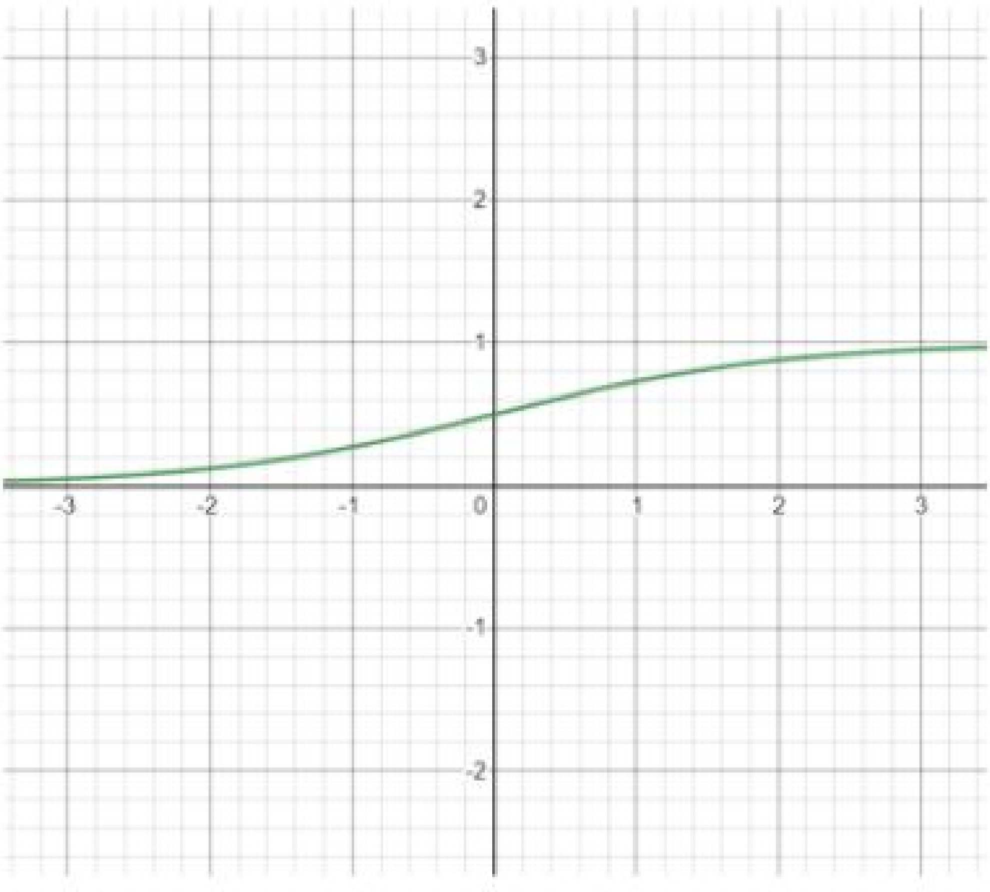


Figure 5.7: Sigmoid activation function graph

Chapter 6

Results And Analysis

6.1 Result

So, after performing our proposed model, this is the final output where after the tumor segmentation the tumor section was identified from the BRATS data set which indicates that our proposed model works properly.

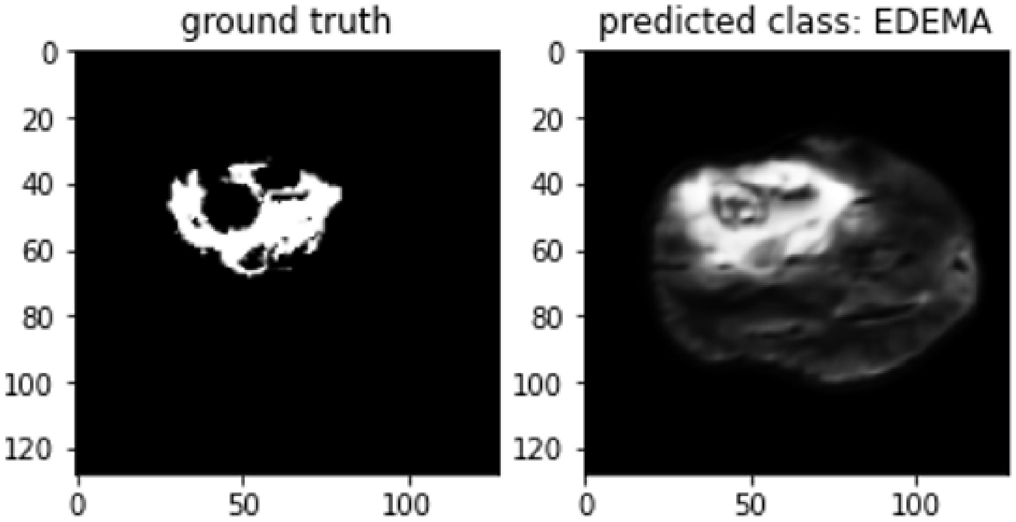


Figure 6.1: Predicted Tumor

6.2 Result Analysis

For getting results with more accuracy with less amount of data loss, model selection is the most important work to perform algorithms properly. Here, we've performed a combination of Unet and Res-Unet model in our dataset to identify the best accuracy with less data loss to make our model more effective in our selected area which is identifying tumors on the primary stage. As, one of the main problems of convolution neural network based models is gradient loss which hampers the performance of the image segmentation. Thus residual blocks take advantage of residual networks by using skip connections and identity connections to the network, so that the gradients can easily flow through the network which prevents gradients from vanishing as well as accelerate network fusion. That's why here, we tried to select a combination of both Unet and Res-Unet models which gives us more accuracy with less data loss using less epochs. At the same time,in our selected model,at epoch 5, 6, 7, 8, 9, 10, 15 the value of accuracy and data loss in varying a little bit but after 15 epochs, the model is over fitting and as a result, accuracy is decreasing and data loss is increasing. In comparison with Unet model where we get (95-96.11) percent accuracy and (22.8-16.58) percent data loss in between 5-20 epoch whereas in our selected model, we are obtaining (97-98.70) percent accuracy and (7.98-10.41) percent data loss for the same range of epochs. That's why we selected the combined Unet and Res-Unet model to get high accuracy and less data loss using less epochs.

Epochs	Accuracy (Percent)	Loss (Percent)
5	97.91	9.09
6	97.06	10.41
8	97.48	12.43
10	98.70	9.3
15	97.45	7.98
20	93.04	23.57

Table 6.1: Testing Epoch,Accuracy and Loss for our U-ResUnet model

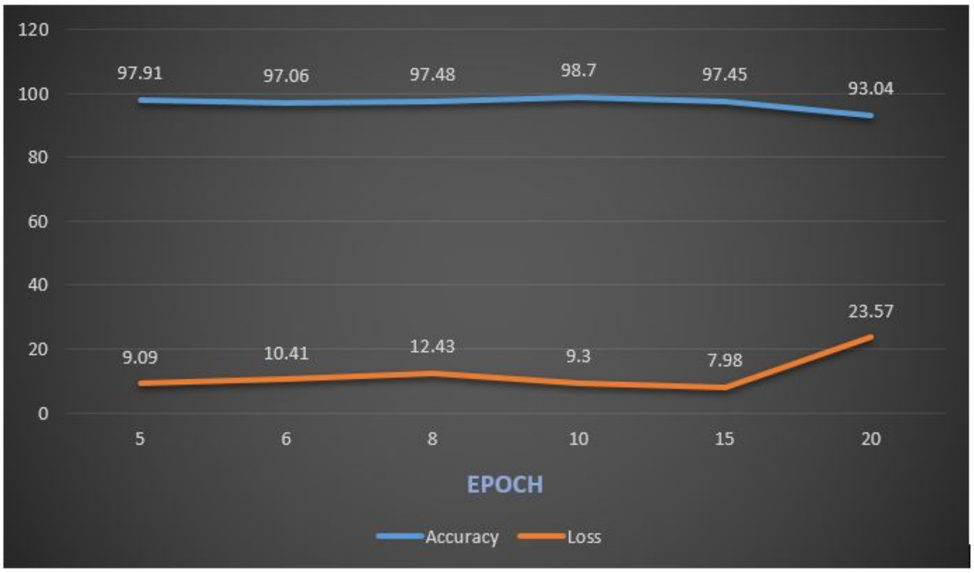


Figure 6.2: Epoch vs Accuracy-Loss Graph of Testing data for Our U-ResUnet Model

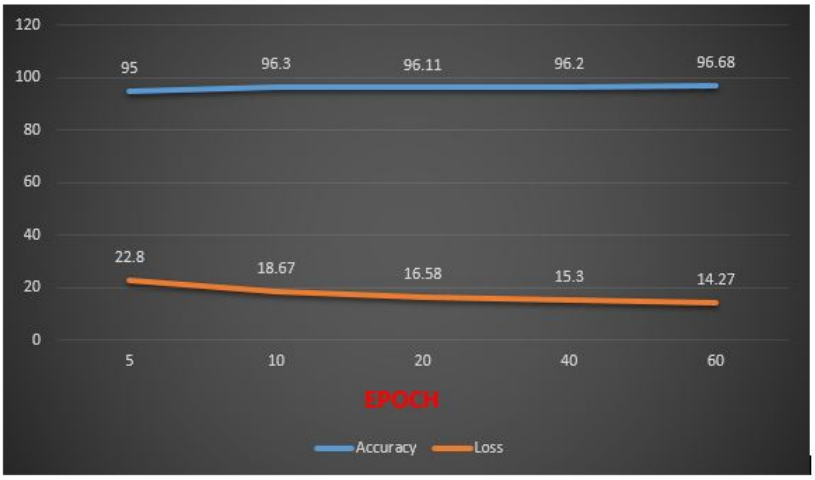


Figure 6.3: Epoch vs Accuracy-Loss Graph for Unet Model in testing data

In Table 6.2, we have shown the training accuracy and loss values in each 10 epoch. We have shown the result for every 2 epoch. In the 2nd epoch the accuracy is 98.43 and loss is 6.49 percent. In the 4th epoch the accuracy came to 98.36 and the loss decreased from 6.49 to 5.07 percent. In 6th and 8th epoch the accuracy came 98.61 and 98.48 as well as loss decreases to 4.14 and 4.33 respectively. Finally in the 10th epoch the accuracy increased to 98.70 with 3.69 percent loss.

Epochs	Accuracy (Percent)	Loss (Percent)
2	98.43	6.49
4	98.36	5.07
6	98.61	4.14
8	98.48	4.33
10	98.70	3.69

Table 6.2: Training Epoch,Accuracy and Loss for our U-ResUnet model

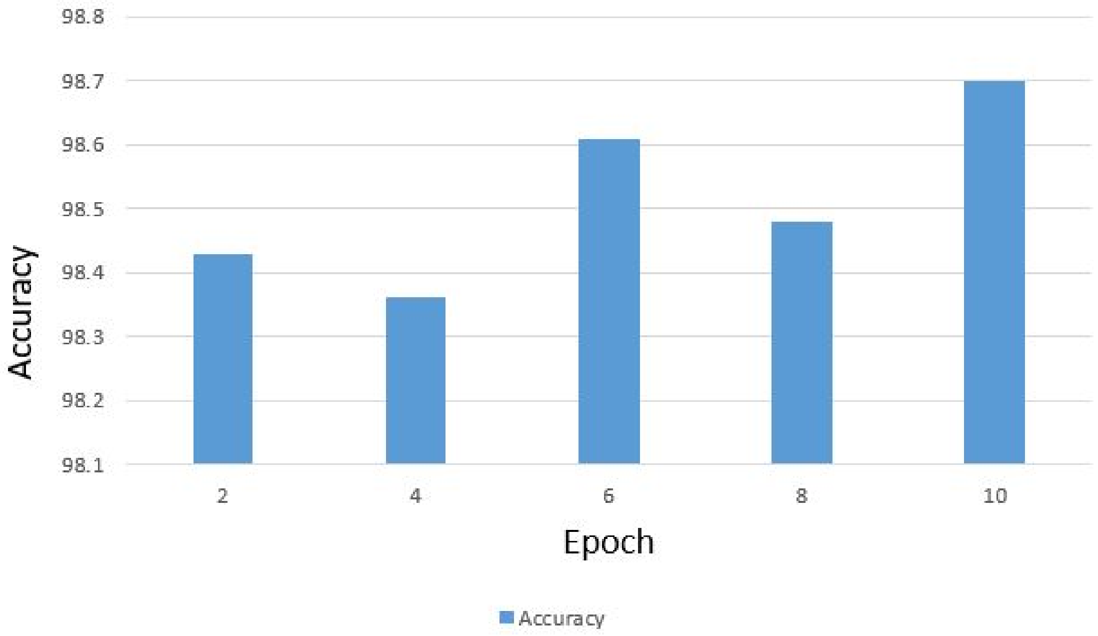


Figure 6.4: Epoch vs Accuracy Graph of Training data for Our U-ResUnet Model

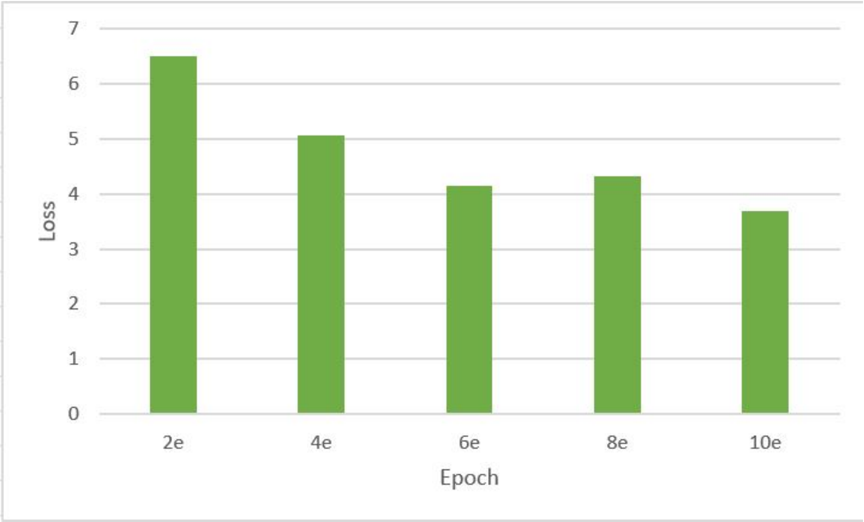


Figure 6.5: Epoch vs Loss Graph of Training data for our U-ResUnet Model

Chapter 7

Conclusion

7.1 Future Work

In our proposed model, pre-processing and segmentation are done using the Res-Unet approach on the BraTS 2020 dataset and we have obtained a good accuracy with less amount of data loss. So, in future work, to upgrade the outcome of the our suggested model we are planning for utilizing the LSTM model for feature extraction and classification of brain tumors for further processing so that it can bring a great revolution in medical sector technology.

7.2 Conclusion

In our research paper, we have implemented a model focused on a combination of Unet and Res-Unet model for precise identification and segmentation of brain tumors using the MRI images to identify brain tumors in the early stages with better efficiency. While implementing the model, at first after taking the MRI input image, preprocessing is done with BraTS2020 data and after that, segmentation is done utilizing the our propoder U-ResUnet model approach whereas while evaluating the training set and validation set we have achieved 98.70 percent accuracy and 9.3 percent data loss for 10 epoch on the test set till now which was better than all other results in comparison. So, to improve it's performance more we want to use LSTM model for further data classification of brain tumors in the next stages in future work.

Bibliography

[1] N. Penttila, *Cells of the brain*, 2019. [Online]. Available: https://www.dana.org/article/cells-of-the-brain/.

[2] *Brain tumor*, 2021. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/brain-tumor/symptoms-causes/syc-20350084.

[3] R. S. Oakes, T. J. Badger, E. G. Kholmovski, N. Akoum, N. S. Burgon, E. N. Fish, J. J. Blauer, S. N. Rao, E. V. DiBella, N. M. Segerson, *et al.*, "Detection and quantification of left atrial structural remodeling with 21 errors7 warnings delayed-enhancement magnetic resonance imaging in patients with atrial fibrillation," *Circulation*, vol. 119, no. 13, pp. 1758–1767, 2009.

[4] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, pp. 18–31, 2017.

[5] A. Işın, C. Direkoğlu, and M. Şah, "Review of mri-based brain tumor image segmentation using deep learning methods," *Procedia Computer Science*, vol. 102, pp. 317–324, 2016.

[6] T. Navamani, "Efficient deep learning approaches for health informatics," in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, Elsevier, 2019, pp. 123–137.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[8] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, "Httu-net: Hybrid two track u-net for automatic brain tumor segmentation," *IEEE Access*, vol. 8, pp. 101 406–101 415, 2020.

[9] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[10] G. A. Francia, C. Pedraza, M. Aceves, and S. Tovar-Arriaga, "Chaining a u-net with a residual u-net for retinal blood vessels segmentation," *IEEE Access*, vol. 8, pp. 38 493–38 500, 2020.

[11] W. Xu, H. Liu, X. Wang, and Y. Qian, "Liver segmentation in ct based on resunet with 3d probabilistic and geometric post process," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, IEEE, 2019, pp. 685–689.

[12] Q. Jin, Z. Meng, C. Sun, H. Cui, and R. Su, "Ra-unet: A hybrid deep attention-aware network to extract liver and tumor in ct scans," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 1471, 2020.

[13] J. Ding, Z. Huang, M. Shi, and C. Ning, "Automatic thyroid ultrasound image segmentation based on u-shaped network," in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, IEEE, 2019, pp. 1–5.

[14] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, and et al., "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015. DOI: 10.1109/tmi.2014.2377694. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/25494501/.

[15] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific Data*, vol. 4, no. 1, Sep. 2017. DOI: 10.1038/sdata.2017.117. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28872634/.

[16] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, and et al., *Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge*, 2018. [Online]. Available: https://arxiv.org/abs/1811.02629.

[17] A. Kermi, I. Mahmoudi, and M. T. Khadir, "Deep convolutional neural networks using u-net for automatic brain tumor segmentation in multimodal mri volumes," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 37–48.

[18] W. Chen, B. Liu, S. Peng, J. Sun, and X. Qiao, "S3d-unet: Separable 3d u-net for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 358–368.

[19] S. Puch, I. Sánchez, A. Hernández, G. Piella, and V. Prckovska, "Global planar convolutions for improved context aggregation in brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 393–405.

[20] M. Bhalerao and S. Thakur, "Brain tumor segmentation based on 3d residual u-net," in *International MICCAI Brainlesion Workshop*, Springer, 2019, pp. 218–225.

[21] K. Yan, Q. Sun, L. Li, and Z. Li, "3d deep residual encoder-decoder cnns with squeeze-and-excitation for brain tumor segmentation," in *International MICCAI Brainlesion Workshop*, Springer, 2019, pp. 234–243.

[22] J. Sun, W. Chen, S. Peng, and B. Liu, "Drrnet: Dense residual refine networks for automatic brain tumor segmentation," *Journal of medical systems*, vol. 43, no. 7, pp. 1–9, 2019.

[23] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An efficient brain tumor image segmentation based on deep residual networks (resnets)," *Journal of King Saud University-Engineering Sciences*, 2020.

[24] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, "Httu-net: Hybrid two track u-net for automatic brain tumor segmentation," *IEEE Access*, vol. 8, pp. 101 406–101 415, 2020.

[25] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, 2020.

[26] N. Sheng, D. Liu, J. Zhang, C. Che, and J. Zhang, "Second-order resu-net for automatic mri brain tumor segmentation," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, pp. 4943–4960, 2021.

[27] H. Chen, Q. Dou, L. Yu, and P.-A. Heng, "Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation," *arXiv preprint arXiv:1608.05895*, 2016.

[28] X. Chen, L. Yao, and Y. Zhang, "Residual attention u-net for automated multi-class segmentation of covid-19 chest ct images," *arXiv preprint arXiv:2004.05645*, 2020.

[29] N. A. Shah, D. Gupta, R. Lodaya, U. Baid, and S. Talbar, "Colorectal cancer segmentation using atrous convolution and residual enhanced unet," in *International Conference on Computer Vision and Image Processing*, Springer, 2020, pp. 451–462.

[30] K. Xie and Y. Wen, "Lstm-ma: A lstm method with multi-modality and adjacency constraint for brain image segmentation," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 240–244.

[31] M. Hamghalam, B. Lei, and T. Wang, "Convolutional 3d to 2d patch conversion for pixel-wise glioma segmentation in mri scans," in *International MICCAI Brainlesion Workshop*, Springer, 2019, pp. 3–12.

[32] F. Özyurt, E. Sert, E. Avci, and E. Dogantekin, "Brain tumor detection based on convolutional neural network with neutrosophic expert maximum fuzzy sure entropy," *Measurement*, vol. 147, p. 106 830, 2019.

[33] H. Kutlu and E. Avcı, "A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks," *Sensors*, vol. 19, no. 9, p. 1992, 2019.

[34] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *2014 13th international conference on control automation robotics & vision (ICARCV)*, IEEE, 2014, pp. 844–848.

[35] A. Rehman, M. A. Khan, T. Saba, Z. Mehmood, U. Tariq, and N. Ayesha, "Microscopic brain tumor detection and classification using 3d cnn and feature selection architecture," *Microscopy Research and Technique*, vol. 84, no. 1, pp. 133–149, 2021.

[36] F. Wang, R. Jiang, L. Zheng, C. Meng, and B. Biswal, "3d u-net based brain tumor segmentation and survival days prediction," in *International MICCAI Brainlesion Workshop*, Springer, 2019, pp. 131–141.

[37] M. Hasan, S. D. Barman, S. Islam, and A. W. Reza, "Skin cancer detection using convolutional neural network," in *Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence*, 2019, pp. 254–258.

[38] F. Xu, H. Ma, J. Sun, R. Wu, X. Liu, and Y. Kong, "Lstm multi-modal unet for brain tumor segmentation," in *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 2019, pp. 236–240.

[39] I. Shahzadi, T. B. Tang, F. Meriadeau, and A. Quyyum, "Cnn-lstm: Cascaded framework for brain tumour classification," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, IEEE, 2018, pp. 633–637.

[40] M. K. Aditi and E. Poovammal, "Image classification using a hybrid lstm-cnn deep neural network," *Int. J. Eng. Adv. Technol.(IJEAT)*, vol. 8, no. 6, pp. 1342–1348, 2019.

[41] T. Balodi, *Convolutional neural network with python code explanation — convolutional layer — max pooling in cnn*, 2019. [Online]. Available: https://www.analyticssteps.com/blogs/convolutional-neural-network-cnn-graphical-visualization-code-explanation.

[42] A. Bhandari, J. Koppen, and M. Agzarian, "Convolutional neural networks for brain tumour segmentation," *Insights into Imaging*, vol. 11, pp. 1–9, 2020.

[43] A. A. Nitnaware and S. Gore, "Brain tumor segmentation using unet deep neural network," *BRAIN*, vol. 7, no. 19, p. 2020, 2020.

[44] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[45] J. Wang, H. Zhu, S.-H. Wang, and Y.-D. Zhang, "A review of deep learning on medical image analysis," *Mobile Networks and Applications*, vol. 26, no. 1, pp. 351–380, 2021.

[46] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94–114, Apr. 2020, ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2020.01.013. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2020.01.013.

[47] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," *arXiv preprint arXiv:1806.02375*, 2018.

[48] J. Kohler, H. Daneshmand, A. Lucchi, M. Zhou, K. Neymeyr, and T. Hofmann, "Towards a theoretical understanding of batch normalization," *stat*, vol. 1050, p. 27, 2018.

[49] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 `[cs.LG]`.

[50] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-classification of brain tumor images using deep neural network," *IEEE Access*, vol. 7, pp. 69 215–69 225, 2019.

[51] W. Hao, W. Yizhou, L. Yaqin, and S. Zhili, "The role of activation function in cnn," in *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, IEEE, 2020, pp. 429–432.

[52] Y. Wang, Y. Li, Y. Song, and X. Rong, “The influence of the activation function in a convolution neural network model of facial expression recognition,” *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020.

[53] T. Wood, *Sigmoid function*, Sep. 2020. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function.