Brain Tumor classification and detection from MRI images using CNN based on ResUNet Architecture

Sanyukta Suman

30002794

Advanced Project -II

MSc Data Engineering

Jacobs Univeristy Bremen

2021

Abstract

Brain tumor is a serious disease occurring in human being. Medical treatment process mainly depends on tumor types and its location. The final decision of neurospecialists and radiologist for the tumor diagnosis mainly depend on evaluation of Magnetic Resonance Imaging (MRI) Images. The manual evaluation process is time-consuming and needs domain expertise to avoid human errors. To overcome this issue Convolution Neural Network (CNN) deep learning algorithm based on ResUNet architecture is proposed for detecting the tumor and marking the area of their occurrence. There are several advantages to using these proposed architectures for segmentation tasks. First, a residual unit helps when training deep architectures. Second, feature accumulation with recurrent residual convolutional layers ensures better feature representation for segmentation tasks.The automatic brain tumor classification is very challenging task in large spatial and structural variability of surrounding region of brain tumor. The method proposed accuracy 96% on the test data.

Table of Contents

List of Figures

Chapter 1

Introduction

A Brain tumor is considered as one of the aggressive diseases, among children and adults. Brain tumors account for 85 to 90 percent of all primary Central Nervous System(CNS) tumors. Every year, around 11,700 people are diagnosed with a brain tumor. The 5-year survival rate for people with a cancerous brain or CNS tumor is approximately 34 percent for men and36 percent for women. Brain Tumors are classified as: Benign Tumor, Malignant Tumor, Pituitary Tumor, etc. Proper treatment, planning, and accurate diagnostics should be implemented to improve the life expectancy of the patients. The best technique to detect brain tumors is Magnetic Resonance Imaging (MRI). A huge amount of image data is generated through the scans. These images are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties.

Brain Tumors are complex. There are a lot of abnormalities in the sizes and location of the brain tumor(s). This makes it really difficult for complete understanding of the nature of the tumor. Also, a professional Neurosurgeon is required for MRI analysis. Often times in developing countries the lack of skillful doctors and lack of knowledge about tumors makes it really challenging and time-consuming to generate reports from MRI'. So an automated system on Cloud can solve this problem.

1.1 Problem Definition

Given the MRI images, the model aims to detect the point in the image which contains the tumor.

1.2 Motivation and objective

The motivation of the proposed application is to aid neurosurgeons and radiologists in detecting brain tumors in an inexpensive and non-invasive manner.

The objective of the proposed project are as follows:

- to capture MRI images to analyze the presence of tumor using CNN and U-Net classification model.
- to perform segmentation process on MRI images to separate the tumor from the normal brain tissues
- to extract features of from the image data set to find the tumor
- to detect the presence of tumor.

The objective of the proposed project is to capture MRI images to analyze the presence of tumor using CNN and U-Net classification model.

1.3 System Architecture

The system architecture of the proposed system is illustrated in figure 1.1.

1.3.1 Preprocessing

Preprocessing of image before segmentation is a critical step for accurate detectio of tumor. In this stage, CLAHE (Zuiderveld, 1994) algorithm is used to enhance the contrast of the images.The images are cropped by obtaining the extreme points in the images.

1.3.2 Classification

In this section, Convolution Neural Network (CNN) model based on ResNet50 architecture is used to classify tumor in the MRI Brain scans.

1.3.3 Segmentation

The segmentation is a process that partitions an image into regions, which allows to separate objects and texture in images. In this study, CNN based on ResUNet architecture is used for image segmentation.

1.3.4 Prediction

Finally, in this step image is passed through the classification network which predicts whether the image has tumor or not.

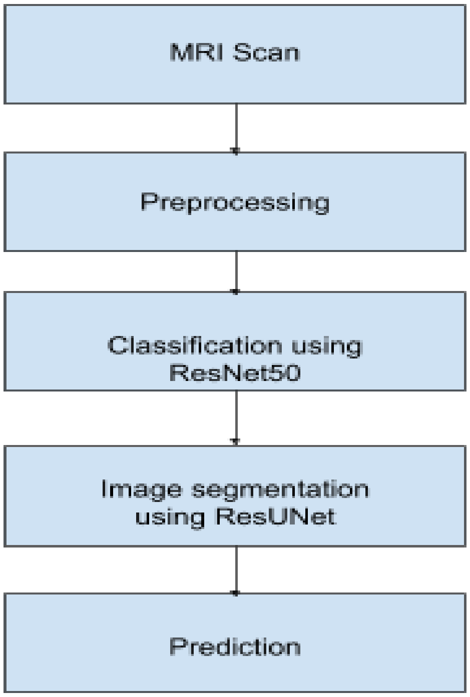

MRI Scan
Preprocessing
Classification using ResNet50
Image segmentation using ResUNet
Prediction

Figure 1.1: System Architecture of a proposed system

1.4 Description of Data

MRI images collected from The Cancer Imaging Archive (TCIA)https://wiki.cancerimagingarchive.net/display/Public/TCGA-LGG along with manual fluid-attenuated inversion recovery (FLAIR) segmentation masks are used in this project. FLAIR sequence produces high quality digital tumor (Rucco, Viticchi and Falsetti, 2020). Images of 109 patients included in The Cancer Genome Atlas (TCGA) LGG collection https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga with at least FLAIR sequence and genomic cluster data available.There are over 20 pairs of images and corresponding mask data for each patient. The dataset is open source and available free of charge.https://www.kaggle.com/mateuszbuda/lgg-mri-segmentation#TCGA_CS_4943_20000902_11_mask.tif. Each MR image is an RGB three-channel image with 256×256 pixels in each channel. Each image also has a corresponding mask image with the same size but only one channel(Yang et al., 2021).

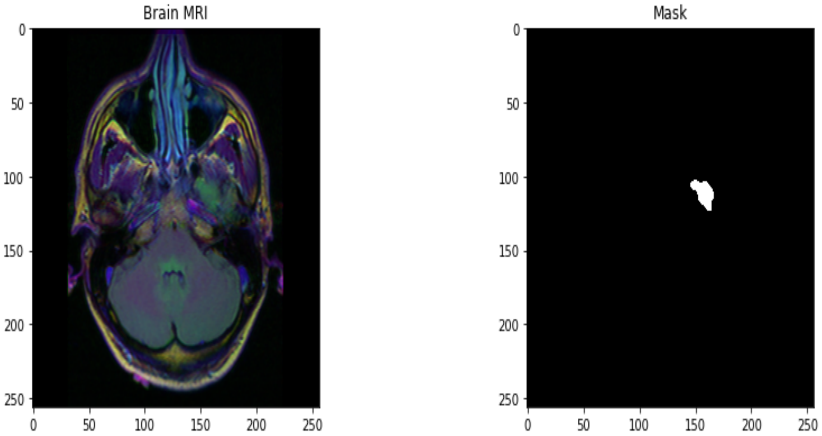


Figure 1.2: Brain tumor MRI images and corresponding masks from the patient.

Here in 1.3 is the distribution of brain with tumor and without tumor images in the dataset. In the below picture, 1 indicates tumor and 0 indicates no tumor. We have total of 2556 non-tumor brain images and 1373 with tumor in the brain in the dataset.

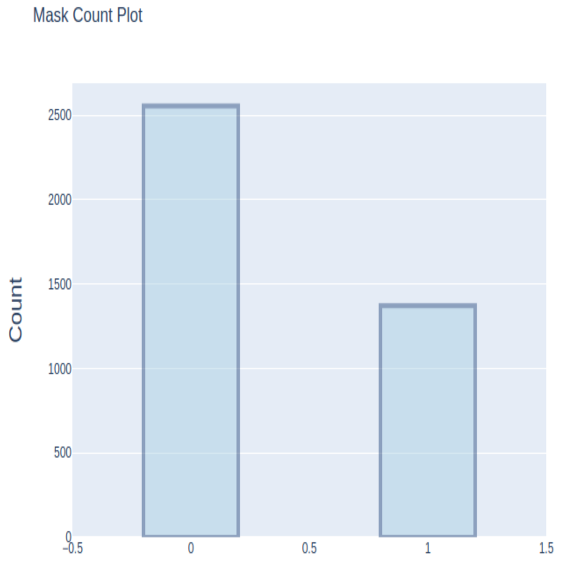


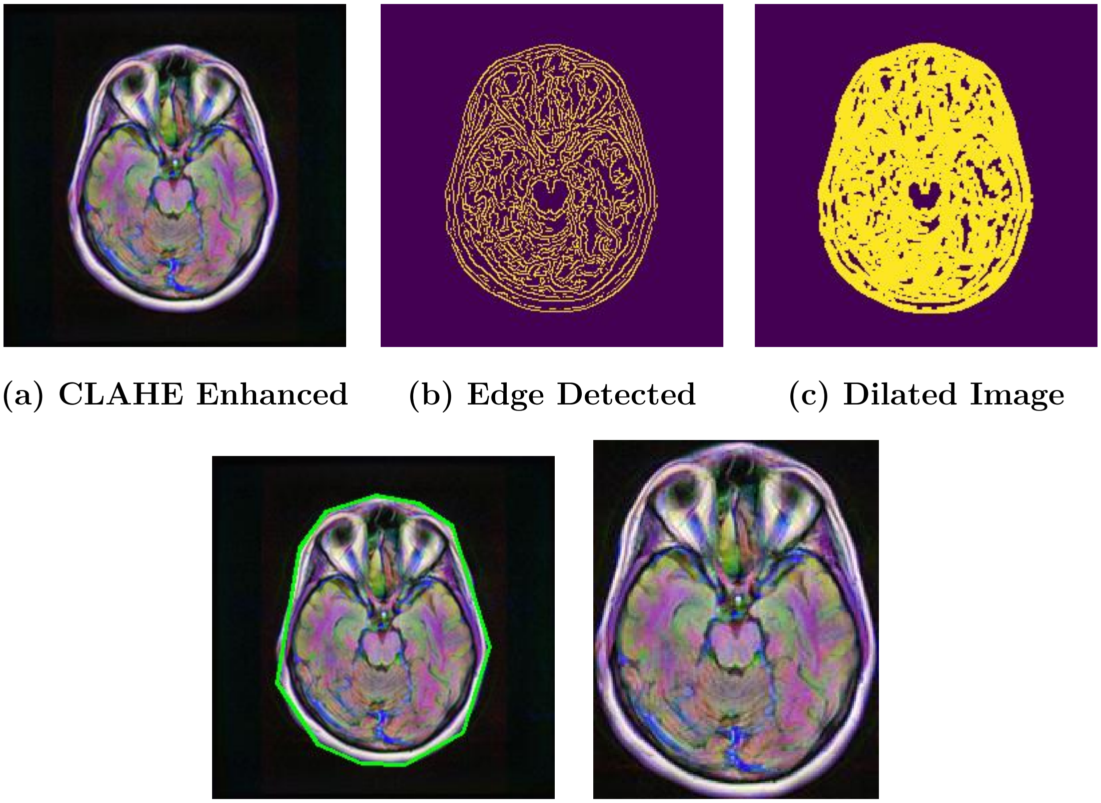
Figure 1.3: Distribution of brain with and without tumour

Chapter 2

Methodology

2.1 Data Preprocessing

Preprocessing is the term used for all the steps taken to improve the data and prepare for the analysis. (Bakas et al., 2017). The given images were initially skull-tripped and co-registered. The tumor segmentation labels were produced by an automated hybrid generative method. These segmentation labels were manually corrected by an expert board-certified neuroradiologist. The final images have rich imaging features, including intensity, volumetric, morphologic, histogram based and textural parameters.The brain data is measured in voxels. They are like the pixels used to display images on the computer screen in 3D. The Brain MRI image dataset are color maps with three RGB channels. In order to perform pixel level segmentation, that is, predicting if the pixel is 0 or 1 (tumor does not exist or exist) all the images along with their respective mask is cropped to remove any noise that is present in the images. Before cropping these images, since the contrast of some of the images are poor, CLAHE (Zuiderveld, 1994) algorithm is used to enhance the contrast of the images. Once the enhanced contrast images are generated; Canny edge detection (**green2002canny**) algorithm is used to detect the edges in the images. Furthermore, dilate operation is performed to remove any small regions of noises present in the images. Finally, in this step the images are cropped by obtaining the extreme points in the images and the contours (Rai and Chatterjee, 2020). The MRI image dataset after the cropping process applied is shown in 2.1.



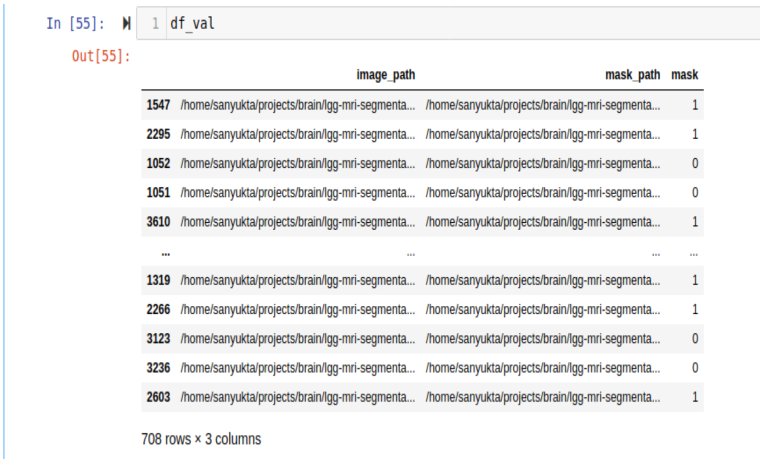
(a) CLAHE Enhanced **(b) Edge Detected** **(c) Dilated Image**

(d) Contoured Image **(e) Cropped Image**

Figure 2.1: **(a)** CLAHE Enhanced **(b)** Edge Detected **(c)** Dilation Image **(d)** Contoured Image **(e)** Cropped Image

2.2 Creating Train, Test and Validation set

The dataset is divided into three part, i.e. train set, validation set and test set. Training dataset is the actual dataset that is used to model train the model (Shah, 2017). validation dataset is used to evaluate a given model that has trained the dataset using the dataset from the training dataset. The validation dataset is used to fine-tune the model parameters (Shah, 2017). The result from the validation set affect the model indirectly. Test set is the sample of data used to rpovide an unbiased evaluation of a final model fit on the training dataset (Brownlee, 2017). Here in the project, there are in total 3929 numbers of images, where 2828 numbers of images are used for train set, 708 images are used for validation set and remaining 393 images are used for test set. The part of data from the validation set is shown in the 2.2



```
In [55]:  1 df_val
Out[55]:
```

	image_path	mask_path	mask
1547	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1
2295	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1
1052	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	0
1051	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	0
3610	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1
...	...	...	...
1319	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1
2266	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1
3123	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	0
3236	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	0
2603	/home/sanyukta/projects/brain/lgg-mri-segmenta...	/home/sanyukta/projects/brain/lgg-mri-segmenta...	1

708 rows × 3 columns

Figure 2.2: Part of validation set shown in the table form

2.3 Data Augmentation

Image data augmentation generate new training samples from the original dataset by adding random alteration in the arrangement of the images, but at the same time ensuring that the class labels of the data are not differed (Gu, Pednekar and Slater, 2019). Generally, the common methods to generate new images are:

- flip horizontally or vertically.
- rotate the image at some degree.
- crop randomly.
- shuffle and transform.

When applying data augmentation, our goal is to increase the generalizability of the model. Data augmentation is not applied in test dataset, therefore, it increases accuracy. Data augmentation is implemented by using ImageDataGenerator in Keras (Gulli and Pal, 2017). An input batch of image is presented to the ImageDataGenerator. This transforms each images in the batch by series of random transformation. Lastly, the randomly transformed batch is then returned to the calling function. Here, the ImageDataGenerator is not returning both the original data and the transformed data, it only returns the randomly transformed data.

2.4 Feature Extraction

Feature extraction is the process of collecting information of an image such as texture, shape, contrast and color. It is used to improve the accuracy of a system. (Haralick, Shanmugam and Dinstein, 1973) introduced one of the most used image analysis for feature extraction - Gray Level Co occurrence Matrix (GLCM). The GLCM is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image (Hall-Beyer, 2017). GLCM texture considers the relation between two pixels at a time, called the reference and the neighbour pixel. GLCM is calculated by creating a framework matrix and describing the spatial relation between the reference and neighbour pixels. Textural finding and analysis improves the detection of tumor. Some of the useful statistic features are listed below (Hall-Beyer, 2017):

- **Correlation** feature describes the spatial dependencies between the pixels.
- **Contrast** is a measure of intensity of a pixel and its neighbor over the image tissues.
- **Energy** can be defined as the quantifiable amount of the extent of pixel. pair repetitions. Energy is a parameter to measure the similarity of an image.
- **Homogeneity** is a scaled local changes of image texture. It returns value that measures the closeness of the distribution of pixel in the GLCM.
- **Dissimilarity** is a measure of comparison between segmentation created by different algorithms.

2.5 Classification

2.6 Convolution Neural Network (CNN)

In this study, Convolution Neural Network model is used to classify tumor in the MRI Brain scans. CNN models are commonly used in object recognition applications. (O'Shea and Nash, 2015) CNNs are comprised of three types of layers. these are convolution layers, pooling layers, and fully-connected layers. When these layers are stacked, a CNN architecture is formed. A simplified CNN architecture for classification is illustrated in 2.3

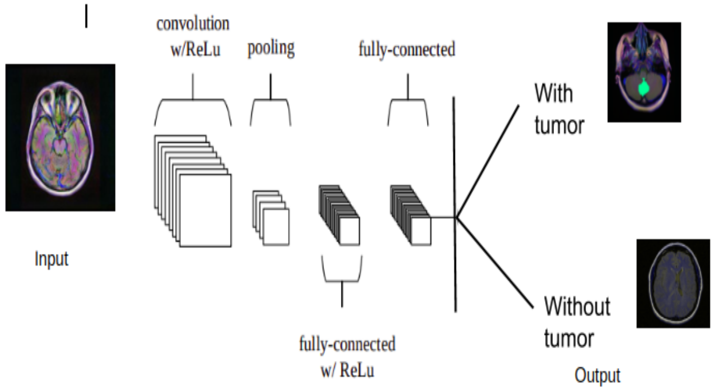


Figure 2.3: A simple CNN architecture, comprised of layers (O'Shea and Nash, 2015).

The basic functionality of CNN can be broken down into four key areas:

- The **input layer** will hold pixel values of the MRI scan of brain .
- The **Convolution layer** will determine the weather the brain consist of tumor or not. These convolution layers are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume.

  The **The rectified linear unit (ReLu)** focuses to apply an activation function such as sigmoid to the output of the activation produced by the previous layer.
- The **pooling layer** will perform the same task down sampling along the spatial dimensionality of the given input; reducing the parameter number within that activation.
- The **fully connected layers** will perform the task to produce class score from the activation, to be used for classification.

Through this method of transformation, CNN are able to transform the original input layer by layer using convolution and down sampling methods to generate class scores for the classification of MRI scans of brain.

2.7 ResNet50

In this project, instead of building a new network, the Resnet50 model is used as the base to train the model. The residual network is a deep convolution neural network model that was introduced by Microsoft in 2015 (Szegedy et al., 2015). The reason for using Resnet50 model at the base is that it is desirable to benefit from the collection of previously trained network, and also that Resnet50 model has achieved successful results in biomedical data. Furthermore, in residual network rather than learning features from the layers, the model learns from the residual that are result of the subtraction of features learned from the input from the layers (H. A. Khan et al., 2020). In 2.4, the architecture of Resnet50 shown.

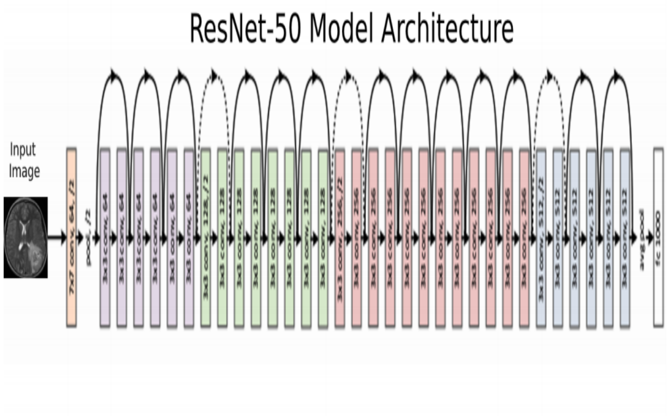
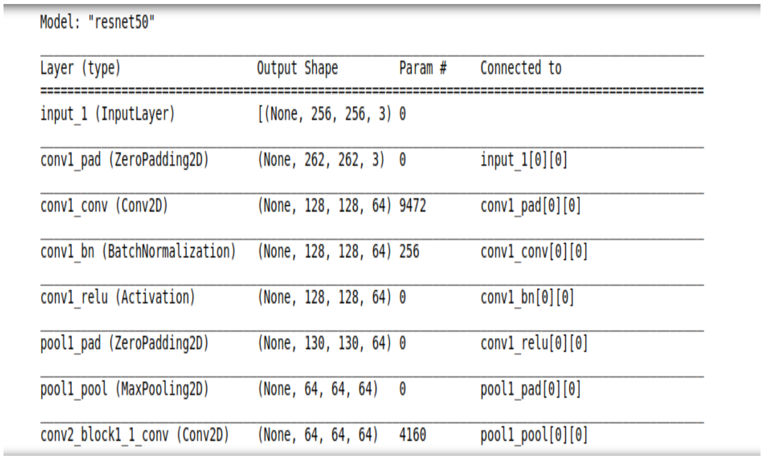


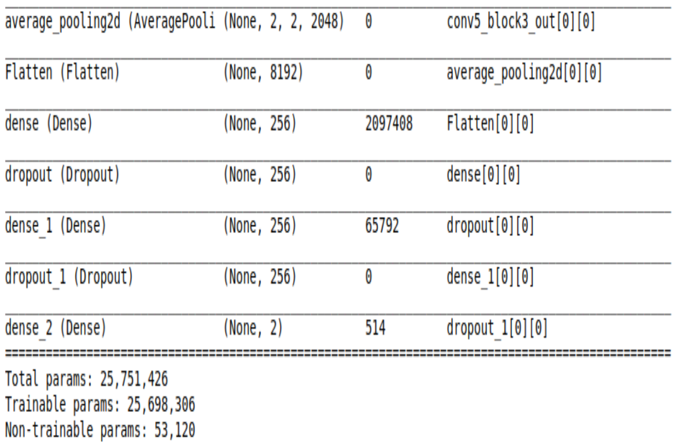
Figure 2.4: Resnet50 Model Architecture.

In the project, the layers present in the residual network are -Input layer, Padding layer, Convolution layer, Batch normalization, Activation (ReLu), and Max pooling. Apart from these additional layers are added to the model. These additional layers are - Average Pooling, Flatten, Dense, Dropout to increase the total number of parameters, consequently, increasing the number of trainable parameters. The result after performing Resnet50 is illustrated in the 2.5 and the result after adding the additional layers is shown in 2.6



```
Model: "resnet50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 256, 256, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 262, 262, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 128, 128, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 128, 128, 64) 256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 128, 128, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 130, 130, 64) 0           conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 64, 64, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
conv2_block1_1_conv (Conv2D)    (None, 64, 64, 64)   4160        pool1_pool[0][0]
```

Figure 2.5: Result after performing Resnet50.



```
__________________________________________________________________________________________________
average_pooling2d (AveragePooli (None, 2, 2, 2048)   0           conv5_block3_out[0][0]
__________________________________________________________________________________________________
Flatten (Flatten)               (None, 8192)         0           average_pooling2d[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 256)          2097408     Flatten[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 256)          0           dense[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          65792       dropout[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 256)          0           dense_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 2)            514         dropout_1[0][0]
==================================================================================================
Total params: 25,751,426
Trainable params: 25,698,306
Non-trainable params: 53,120
```

Figure 2.6: Added layers in Resnet.

2.8 Segmentation using ResUNet to localize the tumor

From the paper, (Ronneberger, Fischer and Brox, 2015), the typical use of convolution networks is a classification tasks, where the output to an image is single class label. However, in many biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel . The segmentation is a process that partitions an image into regions, which allows to separate objects and texture in images.

U-Net is a successful architecture that allow to perform pixel- based image segmentation. U-Net takes its name from the architecture, which when visualized, appears similar to the letter U. as shown in the 2.7.

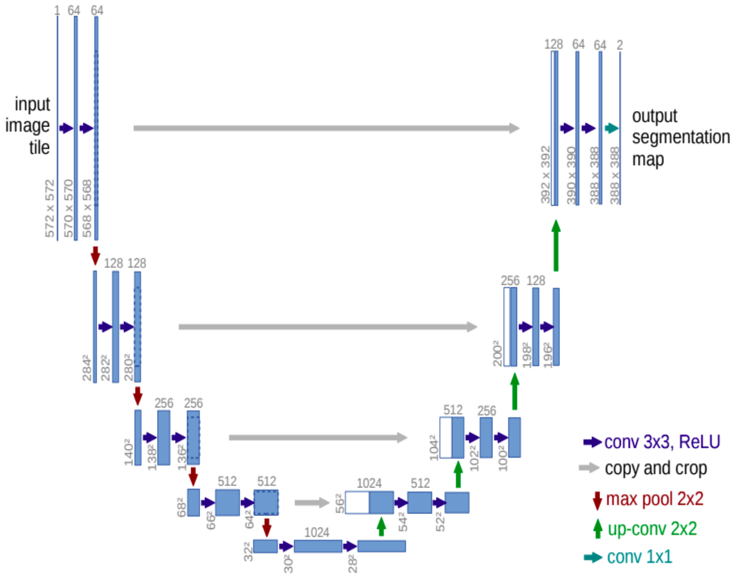


Figure 2.7: ResUNet Architecture.

(Ronneberger, Fischer and Brox, 2015) ResUNet consists of two part - contraction path and expansion path. The contraction path consist of several contraction blocks, each block takes an input that passes through res-blocks followed by 2x2 max pooling. Feature maps after each block doubles, which helps the model learn complex feature effectively. The most significant aspect of this architecture lies in expansion or decoder section. Each block takes in the up-sampled input from the previous layer and concatenates with the corresponding output features from the res-blocks in the contraction path. This feature ensures that the features learned while contracting are used while reconstructing the image. In the last layer of expansion path, the output from the res.block is passed through the 1x1 convolution layer to produce the desired output with the same size as the input.

2.9 Performance Analysis

In this section, performance metrics i.e. accuracy and loss for evaluation of the performance of ResUNet model is discussed.

2.9.1 Accuracy, Precision and Recall

Accuracy is one metric for evaluating the classification models. The accuracy of the model can be easily be calculated by the confusion matrix using the formula (Suganthe et al., 2020).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2.1}$$

where,

- TP : True Positive
- FP : False Positive
- TN: True Negative
- FN : False Negative

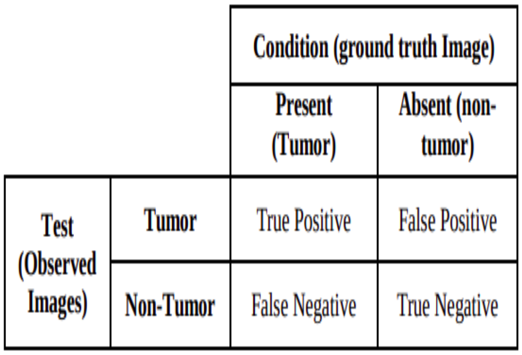
Precision is related to random errors. Precision effectively describes the purity of our positive detections relative to the ground truth. The high prediction means that an algorithm returned more appropriate results than irrelevant. The precision of the model can also be calculated by the confusion matrix using the formula (Anithadevi and Perumal, 2016).

$$Precision = \frac{TP}{(TP + FP)} \tag{2.2}$$

Recall effectively describes the completeness of the positive predictions relative to the ground truth. High recall means that model returned most of the relevant result. Recall can be calculated using the formula.

$$Recall = \frac{TP}{(TP + FN)} \tag{2.3}$$

Test statistic of the model is shown in 2.8



		Condition (ground truth Image)	
		Present (Tumor)	Absent (non-tumor)
Test (Observed Images)	Tumor	True Positive	False Positive
	Non-Tumor	False Negative	True Negative

Figure 2.8: Performance Evaluation

Finally, in statistical analysis, the F-score is a measure of a test accuracy. The F_1 is constructed as weighted average of the precision and recall. It can be defined as (Anithadevi and Perumal, 2016)

$$F_1 = \frac{precision.recall}{(precision + recall)} \tag{2.4}$$

2.9.2 Loss function

Deep learning algorithm uses stochastic gradient descent approach to optimize and learn the objective from the model (Jadon, 2020). In order for model to learn the objective faster and accurately, it is important to ensure that the mathematical representation of objectives are able to cover the edge cases. In addition, class imbalance dataset is a frequent problem experienced when performing segmentation. Therefore, to deal with the class imbalance problem the loss function are used for updating the weight vector by using labelled output and calculated output of the model. In the medical community, the Dice score coefficient (DSC) is widely used to asses segmentation. The Dice Coefficient is well known for being the go to evaluation metric for image segmentation, but it can also serve as a loss function. Dice coefficient only consider the segmentation class and not the background class. Therefore, a novel approach proposed by (Abraham and N. M. Khan, 2019), the focal Tversky loss function (FTL) is used to evaluate the model. The Focal Tversky Loss (FTL) is defined as:

$$FTL_c = \sum_c (1 - TI_c)^{\frac{1}{\gamma}} \tag{2.5}$$

In practice, if a pixel is misclassified with high Tversky index (an asymmetric similarity measures on sets that compares a variant to a prototype), the FTL is unaffected. However, if the Tversky index is small and the pixel is misclassified, the FTL will decrease significantly.

In the project, the Focal Tversky loss of the segmentation model using ResUNet is shown in the 2.9

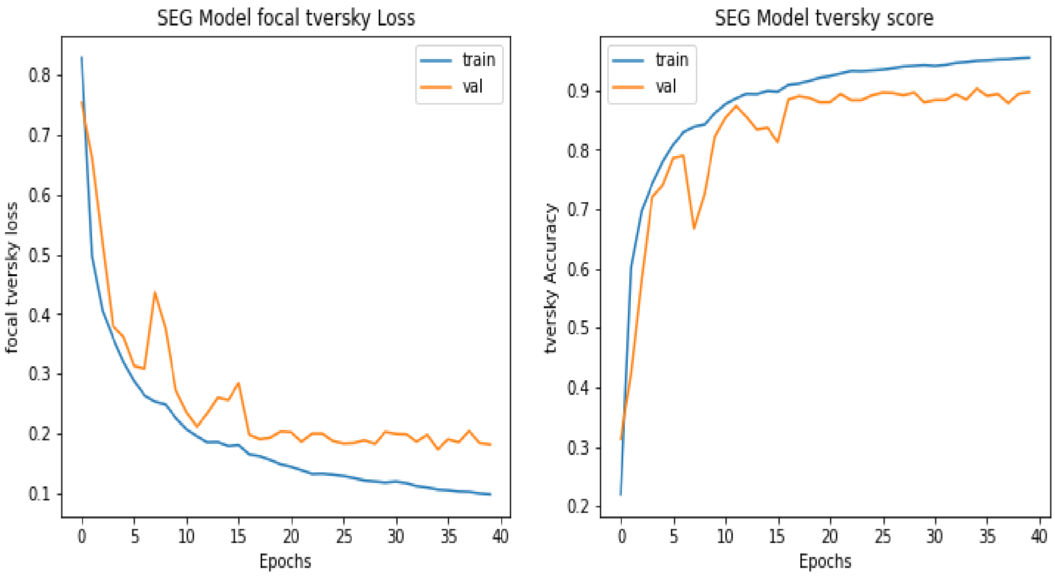


Figure 2.9: Focal Tversky Loss

Chapter 3

Implementation

3.1 Implementation details

This project is based on implementation using the Keras library with TensorFlow (Abadi et al., 2016) as the backend. They are open-source machine learning library capable of performing deep learning algorithm. There are total of 2556 images with no tumor and 1373 images with tumor.

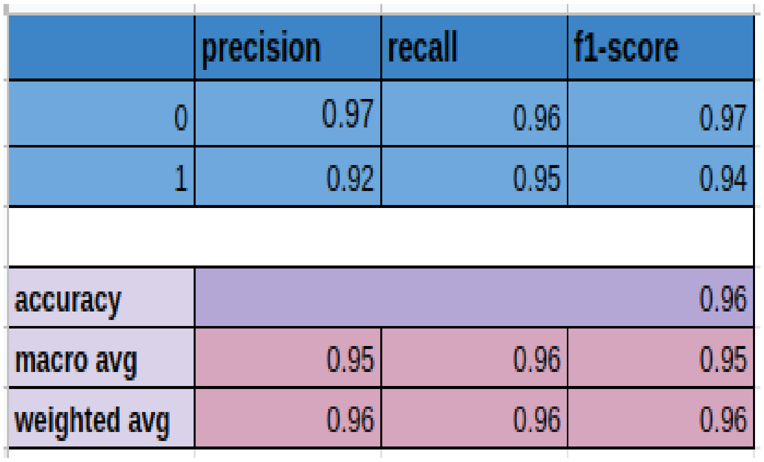
Since CNN's are able to learn useful features from scratch, minimal preprocessing is applied. Initially, the preprocessing follows five steps: enhancing the contrast of the images using CLAHE (Zuiderveld, 1994), detecting edges using Canny edged detection (**green2002canny**), dilation of images and finally cropping the images. Since the number of images with tumor and with no tumor is imbalanced, data augmentation is used to improve the balance in the dataset.The brain images are augmented using ImageDataGenerator with Keras library. After the successful preprocessing step, dataset is split into three set - Train set, test set and validation set using train-test-split function from the SkLearn library The classification model to classify if the tumor exist or not is build via transfer learning method called ResNet50. The shape of input taken by the ResNet50 is (256 x 256 x 3). Segmentation model -ResUNet is built to localize the tumor. Each brain tumor segmentation is trained with standard back-propagation with Rectified Linear Unit (ReLU) as activation function on 40 epochs (number of passes of entire training dataset the algorithm has completed).

Prediction function takes data frame containing ImageID as Input and perform 2 type of prediction on the image. Initially, image is passed through the classification network which predicts whether the image has tumor or not. Next, if the model is sure that the image has no tumor, then the image is labeled as no-tumor. Again, if the model is not sure, it passes the image to the segmentation network, it again checks if the image has tumor or not, if it has tumor, then the type and location of defect is found.

3.2 Result and Analysis

3.2.1 Analysis of Confusion Matrix

The CNN managed to accurately categorize the images into tumor patient and normal patient with a test accuracy of 96%. As shown in 3.1, the accuracy of the CNN obtained by ResUNet used to classify images by the process of image segmentation is 91.21%. Also, the precision of the CNN obtained by ResUNet is 96%. In the propoesed CNN based classification does not require feature extraction steps separately, The feature value is taken from CNN itself.



	precision	recall	f1-score
0	0.97	0.96	0.97
1	0.92	0.95	0.94
accuracy			0.96
macro avg	0.95	0.96	0.95
weighted avg	0.96	0.96	0.96

Figure 3.1: Precision, recall and accuracy

The diagram of the classification loss and accuracy is also shown on the 3.2.

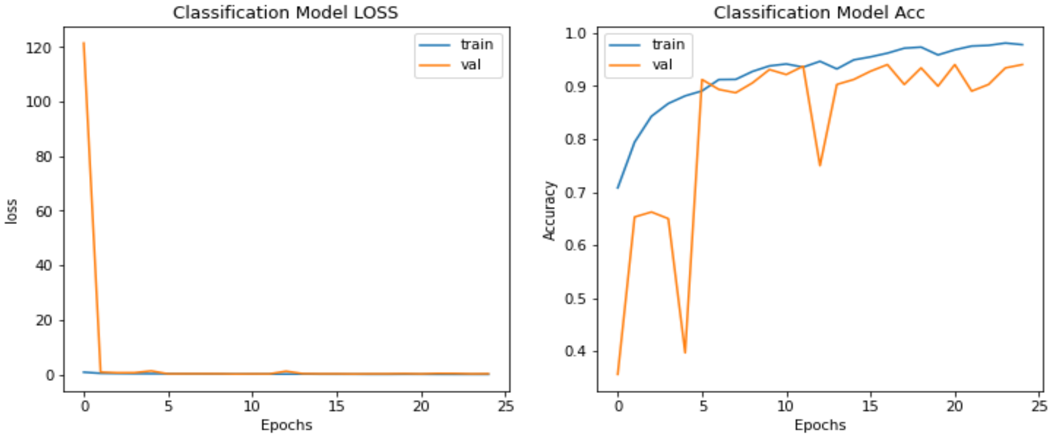


Figure 3.2: Classification model loss and accuracy.

3.2.2 Data Visualization

In this section, finally the classification of images categorized by the ResUNet is shown below in 3.3

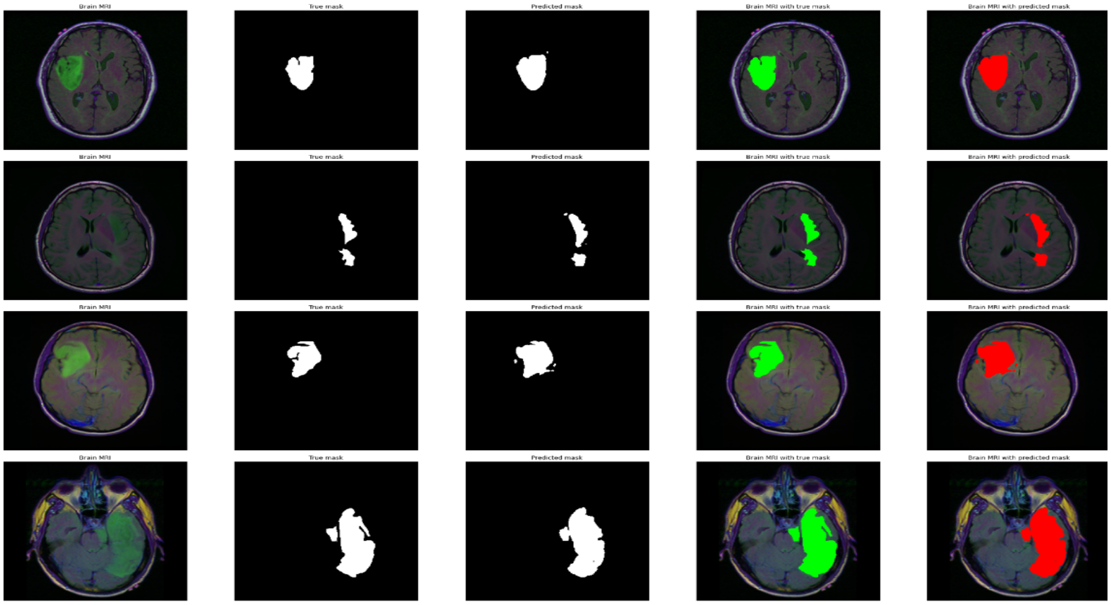


Figure 3.3: Images by the CNN are correctly classified

Chapter 4

Conclusions

4.1 Conclusion

The main goal of this project work is to design efficient automatic brain tumor classification and localization of tumor with high accuracy, performance and low complexity. First, the conventional brain tumor classification is performed by using CNN based on ResNet50. Here, the classification results if the tumor exist or not. The complexity is low, computational time is high meanwhile accuracy is also high. Further to localize the tumor in the given image and to draw an edge around the tumor another convolution neural network based classification i.e. ResUNet based segmentation is introduced to carry out the localization of tumor in the proposed scheme. The result gives the brain MRI images with the predicted position of the tumor in the form of mask. Finally, the focal Tversky loss function is applied to achieve high accuracy. The training accuracy is 96%. Due to the importance of the diagnosis given by the physician, the accuracy of the doctors help in diagnosing the tumor and treating the patient increased high medical accuracy of the proposed method.

4.2 Future Scope

A careful attention should be paid to the unclassified and misclassified samples. The presence of unclassified samples is something related to detections with low scores.

Such samples may be thought to be classified as “no tumours” even though they contain a tumour. In the future, this situation may be overcome by adding healthy images. The pre-processing step may include algorithms that will emphasize the unobvious features in the given images. In the future, some optimization algorithms (such as genetic algorithm, particle swarm optimization, simulated annealing, etc.) may be utilized to find the best parameter set giving the highest classification accuracy (Salçin et al., 2019).

References

Abadi, Martın et al. (2016). 'Tensorflow: Large-scale machine learning on heterogeneous distributed systems'. In: *arXiv preprint arXiv:1603.04467* (cit. on p. 17).

Abraham, Nabila and Naimul Mefraz Khan (2019). 'A novel focal tversky loss function with improved attention u-net for lesion segmentation'. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019).* IEEE, pp. 683–687 (cit. on p. 16).

Anithadevi, D and K Perumal (2016). 'A hybrid approach based segmentation technique for brain tumor in MRI Images'. In: *arXiv preprint arXiv:1603.02447* (cit. on pp. 14, 15).

Bakas, Spyridon et al. (2017). 'Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection'. In: *The cancer imaging archive* 286 (cit. on p. 6).

Brownlee, Jason (2017). 'What is the difference between test and validation datasets'. In: *Machine Learning Mastery* 14 (cit. on p. 7).

Gu, Shanqing, Manisha Pednekar and Robert Slater (2019). 'Improve Image Classification Using Data Augmentation and Neural Networks'. In: *SMU Data Science Review* 2.2, p. 1 (cit. on p. 8).

Gulli, Antonio and Sujit Pal (2017). *Deep learning with Keras.* Packt Publishing Ltd (cit. on p. 8).

Hall-Beyer, Mryka (2017). 'GLCM texture: a tutorial v. 3.0 March 2017'. In: (cit. on p. 9).

Haralick, Robert M, Karthikeyan Shanmugam and Its' Hak Dinstein (1973). 'Textural features for image classification'. In: *IEEE Transactions on systems, man, and cybernetics* 6, pp. 610–621 (cit. on p. 9).

Jadon, Shruti (2020). 'A survey of loss functions for semantic segmentation'. In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB).* IEEE, pp. 1–7 (cit. on p. 15).

Khan, Hassan Ali et al. (2020). 'Brain tumor classification in MRI image using convolutional neural network'. In: *Mathematical Biosciences and Engineering* 17.5, pp. 6203–6216 (cit. on p. 11).

O'Shea, Keiron and Ryan Nash (2015). 'An introduction to convolutional neural networks'. In: *arXiv preprint arXiv:1511.08458* (cit. on p. 10).

Rai, Hari Mohan and Kalyan Chatterjee (2020). 'Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images'. In: *Machine Learning with Applications* 2, p. 100004 (cit. on p. 6).

Ronneberger, Olaf, Philipp Fischer and Thomas Brox (2015). 'U-net: Convolutional networks for biomedical image segmentation'. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241 (cit. on pp. 12, 13).

Rucco, Matteo, Giovanna Viticchi and Lorenzo Falsetti (2020). 'Towards Personalized Diagnosis of Glioblastoma in Fluid-Attenuated Inversion Recovery (FLAIR) by Topological Interpretable Machine Learning'. In: *Mathematics* 8.5, p. 770 (cit. on p. 4).

Salçin, Kerem et al. (2019). 'Detection and classification of brain tumours from MRI images using faster R-CNN'. In: *Tehnički glasnik* 13.4, pp. 337–342 (cit. on p. 21).

Shah, Tarang (2017). 'About train, validation and test sets in machine learning'. In: *Towards Data Science* 6 (cit. on p. 7).

Suganthe, RC et al. (2020). 'Deep Learning Based Brain Tumor Classification Using Magnetic Resonance Imaging'. In: *Journal of Critical Reviews* 7.9, pp. 347–350 (cit. on p. 14).

Szegedy, Christian et al. (2015). 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9 (cit. on p. 11).

Yang, Qifan et al. (2021). 'Evaluation of magnetic resonance image segmentation in brain low-grade gliomas using support vector machine and convolutional neural network'. In: *Quantitative Imaging in Medicine and Surgery* 11.1, p. 300 (cit. on p. 4).

Zuiderveld, Karel (1994). 'Contrast limited adaptive histogram equalization'. In: *Graphics gems IV*. Academic Press Professional, Inc., pp. 474–485 (cit. on pp. 2, 6, 17).

.1 Appendix A

Code Repository

All the code implemented and utilized during the execution of the process described in this report is available at the GitHub repository `https://github.com/sumansanyukta/Brain-tumor-detection-using-ResUNet`