BRAIN TUMOUR SEGMENTATION USING CNN

Project Submitted to

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

In partial fulfilment of the requirements for the degree of

MASTER OF COMPUTER APPLICATION

BY

Aritra Chakravarty

Examination Roll: MCA2360023

Registration No.: 154211 of 2020-2021

Under the guidance of

Dr. Jamuna Kanta Sing

Professor, Department of Computer Science and Engineering

Jadavpur University

2023

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

To whom it may concern

I hereby recommend that the Project "Brain tumour segmentation using CN" has been carried out by ARITRA CHAKRAVARTY (Reg. No.: 154211 of 2020-2021, Roll No: 002010503003, Examination Roll: MCA2360023) under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of MASTER OF COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY, during the academic year 2022-2023.

Prof. (Dr.) Jamuna Kanta Sing
Project Supervisor
Dept. of Comp. Science & Engineering
Jadavpur University, Kolkata 700032

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Comp. Science and Engineering
Jadavpur University, Kolkata 700032

Prof. (Dr.) Ardhendu Ghoshal
DEAN, Faculty of Engineering and Technology
Jadavpur University, Kolkata 700032

CERTIFICATE OF APPROVAL

This is to certify that the project entitled "**Brain Tumour Segmentation using CNN"** is a bonafide record of work carried out by **ARITRA CHAKRAVARTY** in partial fulfilment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATION** in the **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, JADAVPUR UNIVERSITY.** It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project work only for the purpose for which it has been submitted.

……………………….	……………………….
Signature of Examiner 1	Signature of Examiner 2
Date:	Date:

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC PROJECT

I hereby declare that this thesis work holds literature survey and original research work by the undersigned candidate, as a student of MASTER OF COMPUTER APPLICATION. All the information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all the material results that are not original to this work.

NAME: ARITRA CHAKRAVARTY

EXAMINATION ROLL NUMBER: MCA2360023

PROJECT TITLE: Brain Tumour Segmentation Using CNN

SIGNATURE WITH DATE:

ACKNOWLEDGEMENT

With my most sincere and gratitude, I would like to thank **Prof. (Dr.) Jamuna Kanta Sing,** Department of Computer Science and Engineering, my supervisor, for his overwhelming support throughout the duration of this project. His motivation always gave me the required inputs and momentum to continue with my work without which it would not have taken its current shape. His valuable suggestion and numerous discussions have always inspired new ways of thinking. I feel deeply honoured that I got this opportunity to work under him. I would like to express my sincere thanks to all my teachers for providing sound knowledge base and co-operation.

I would like to thank all the faculty members of the Department of Computer Science and Engineering of Jadavpur University for their continuous support. Last, but not the least, I would like to thank my batchmates for staying by my side when I needed them the most.

Aritra Chakravarty

Examination Roll Number: MCA2360023

Registration Number :154211 of 2020-2021

CONTENTS

Topic **Page Number**

1. ABSTRACT

The determination of tumour extent is a major challenging task in brain tumour and quantitative evaluation. Magnetic Resonance Imaging (MRI) is one of the non-invasive techniques that has emanated as a front-line diagnostic tool for brain tumour without ionizing radiation.

Among brain tumours, gliomas are the most aggressive, leading to a very short life expectancy in their highest grade. In the clinical practice manual segmentation is a time-consuming task and their performance is highly depended on the operator’s experience.

This project proposes fully automatic segmentation of brain tumour using CNN. It uses brain MR images together with manual FLAIR abnormality segmentation masks. Using RESNET-50 tumour detection is first done and then using RESUNET, brain tumour segmentation is done.

Hence, this project detects brain tumour and if it exists, segments the brain tumour. This is an essential step in diagnosis and treatment planning, both of which is needed to be done quickly to maximise the likelihood of successful treatment.

2. INTRODUCTION

Brain tumour is the abnormal growth of cells within the brain. Generally, tumour is classified into primary and secondary tumour. primary tumour starts within the brain and secondary tumour will spread to the other parts of the body. There are many medical imaging methods available like X-Ray, CT (Computed Tomography) and MRI (Magnetic Resonance Imaging).

This project uses MRI brain images, because of its high resolution and good quality of image. After capturing MRI brain image, it is necessary to separate the tumour region from the MRI brain image. This presented work segments lower grade gliomas if it exists.

The manual segmentation of brain tumours is labour intensive and the segmentation results obtained depends on the operator's experience and subjective decision making. Thus, we need a fully automatic, objective and reproducible segmentation method.

This work uses CNN, ResNet-50 to detect tumours and ResUNet to segment them if it is detected by the ResNet-50 model, implemented using Python. The training of the MRI images is done using tensorflow. The performance of the segmentation is then compared against the ground truth images. I have used Tversky score to calculate accuracy of automatic segmentation.

3. MATERIALS AND METHODS

The project uses Convolutional Neural Network for MRI brain tumour segmentation. It uses tensor-flow based RESNET50 model to first detect if tumour exists or not and if exists, it segments it using RESUNET model. After segmentation, the results are compared using Tversky score.

PYTHON BASED CNN

This project work utilises the python-based programming to implement the segmentation of MRI brain tumour. The features are listed below in order because python programming was chosen to implement the project work.

- Python code is more compact and readable.
- The Python data structure is superior.
- It is open source and provides lots of packages, libraries and datasets which make using it easier.

4. FLOW DIAGRAM OF PROPOSED WORK

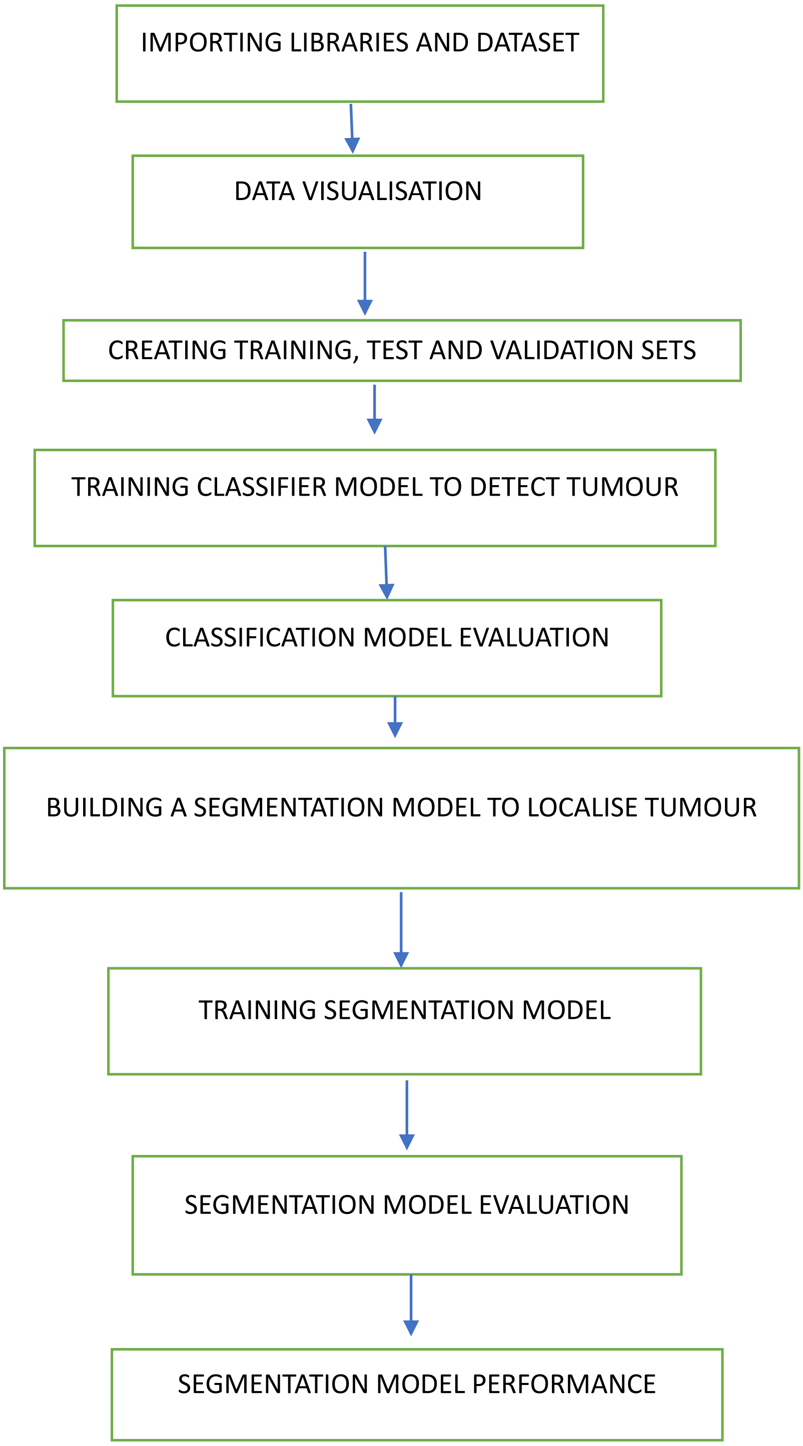

IMPORTING LIBRARIES AND DATASET
DATA VISUALISATION
CREATING TRAINING, TEST AND VALIDATION SETS
TRAINING CLASSIFIER MODEL TO DETECT TUMOUR
CLASSIFICATION MODEL EVALUATION
BUILDING A SEGMENTATION MODEL TO LOCALISE TUMOUR
TRAINING SEGMENTATION MODEL
SEGMENTATION MODEL EVALUATION
SEGMENTATION MODEL PERFORMANCE

Fig. 1. Flow Diagram of Proposed Work

5. MRI BRAIN DATA ACQUISITIONS

This recommended method is tested and implemented on “The Cancer Imaging Archive (TCIA)” which has MRI scans of 110 patients with lower grade gliomas with at least fluid attenuated inversion recovery (FLAIR) sequence and genomic data cluster data.

The dataset contains brain MR images together with manual FLAIR abnormality segmentation masks. Few images of sample dataset are shown below.

Tumour genomic clusters and patient data provided in data.csv file.

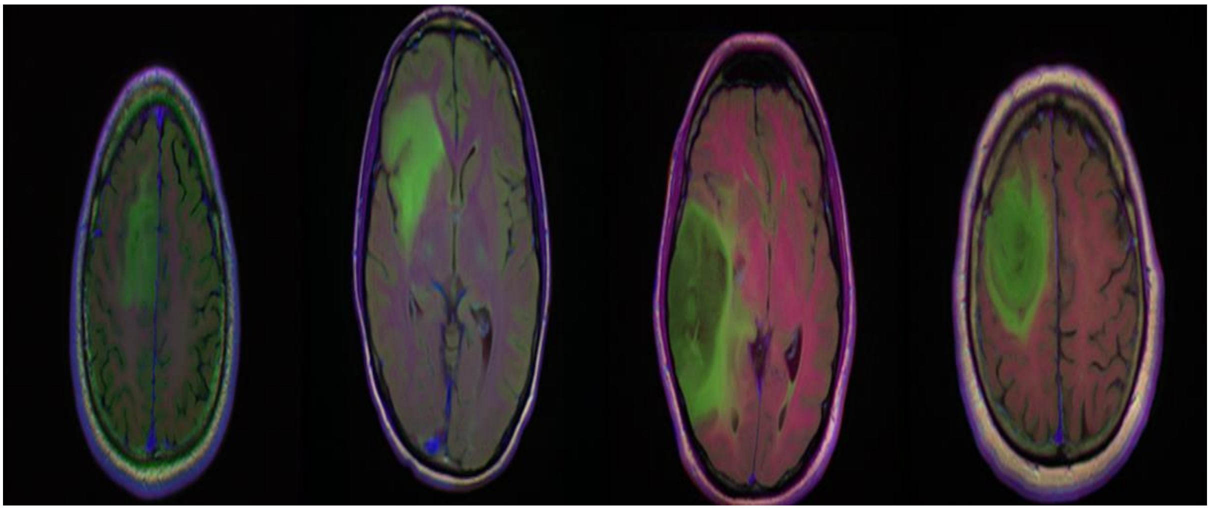


Fig. 2. Brain MRI Dataset Input Images

6. CONVOLUTIONAL NEURAL NETWORK

A CNN [1] is a type of deep learning algorithm that is particularly effective in analysing visual data. It is widely used for tasks such as image classification, object detection, and segmentation.

CNNs are designed to mimic the visual processing mechanism of the human brain. They consist of multiple layers of interconnected artificial neurons, which perform convolution operations to extract features from input images. These features are then passed through non-linear activation functions, pooled, and fed into subsequent layers for further processing. The final layers of the network typically involve fully connected layers that generate the desired output.

When it comes to brain tumour segmentation, CNNs have shown great promise. Segmentation refers to the process of identifying and delineating the tumour region within medical images, such as MRI scans. CNNs can be trained to automatically learn the complex patterns and structures associated with tumours, enabling accurate and efficient segmentation.

In this project, I have used ResNet-50 and ResUNet which are CNN models for detection and segmentation of brain tumours respectively.

7. ResNet-50 Model

ResNet-50 Model [2] is a network with skip connections that perform identity mappings, merged with the layer outputs by addition. This enables deep learning models with tens or hundreds of layers to train easily and approach better accuracy when going deeper. It has 50 layers hence the name and was first introduced in 2015.

Step by step working principle of ResNet-50 Model [3] for detection of brain tumour:

Step 1: Input Image

The model takes an input image of the brain, typically in the form of a 2D matrix of pixel values.

Step 2: Convolutional Layers

The input image is passed through a series of convolutional layers. These layers consist of filters that convolve across the image, extracting local features through a dot product operation.

Step 3: Activation Function

After each convolutional layer, an activation function such as the Rectified Linear Unit (ReLU) is applied element-wise to introduce non-linearity, enabling the model to learn complex relationships between the features.

ReLU function is given by: $f(x) = \max(0, x)$

Step 4: Residual Blocks

The ResNet architecture introduces residual blocks to address the problem of vanishing gradients in deep networks. A residual block contains multiple convolutional layers with skip connections that bypass one or more layers. This allows the network to learn residual mappings, making it easier to optimize deep models.

Step 5: Pooling

Periodically, pooling layers like max pooling are applied to reduce the spatial dimensions of the feature maps, decreasing computational complexity and providing some translational invariance.

Step 6: Fully Connected Layers

After several convolutional and pooling layers, the feature maps are flattened into a 1D vector. This vector is then passed through fully connected layers that perform a series of matrix multiplications and apply non-linear activations to capture higher-level abstractions.

Step 7: Output Layer

The final fully connected layer is typically followed by a softmax activation function, which assigns probabilities to each possible class label. In the case of brain tumour detection, the model is trained to classify images as either tumour-present or tumour-absent.

Softmax function is given by:

$$\sigma\overrightarrow{(z)}_i = \frac{e^{zi}}{\sum_{j=1}^{K} e^{zj}}$$

σ = softmax

$\vec{z}$ = input vector

e^{zi} = standard exponential function for input vector

e^{zj} = standard exponential function for output vector

K = number of classes in the multi-class classifier.

Step 8: Loss Function

During training, a loss function such as cross-entropy is used to measure the difference between the predicted probabilities and the ground truth labels. The model aims to minimize this loss by adjusting its internal parameters.

Step 9: Backpropagation and Optimization

The gradients of the loss with respect to the model parameters are computed using backpropagation. An optimizer, such as stochastic gradient descent or Adam, is then used to update the model parameters, making them more effective at classifying brain tumour images.

Step 10: Inference

During inference, the trained model takes an input brain image, processes it through the network, and produces a probability indicating the likelihood of a brain tumour being present. The class label with the highest probability is considered the model's final output, indicating whether a tumour is detected or not.

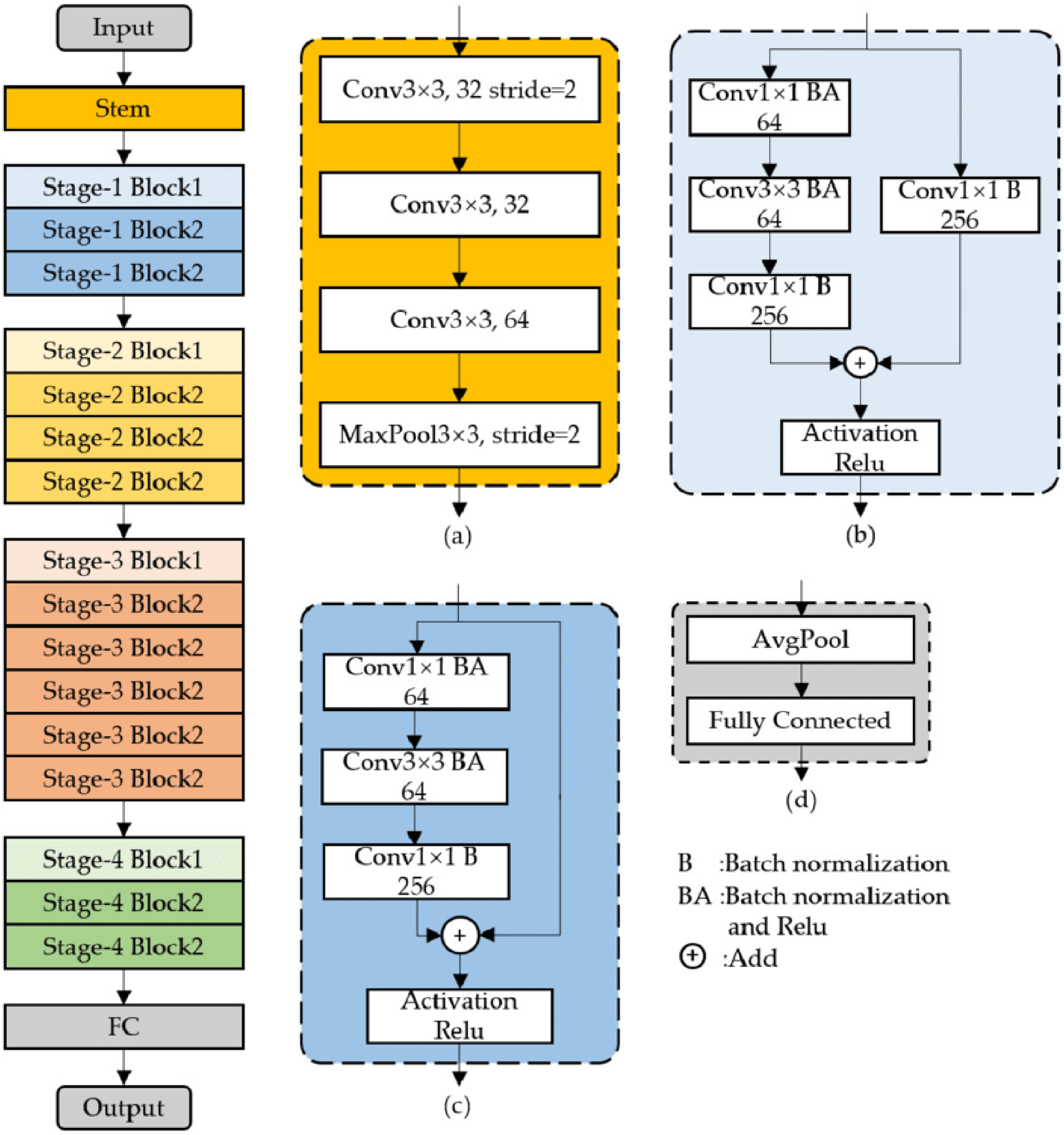


Fig. 3. ResNet-50 Architecture

8. RESUNET MODEL

The ResUNet model, which is short for Residual U-Net [4], is a popular deep learning architecture used for various computer vision tasks, including medical image segmentation such as brain tumour segmentation.

Step-by-step explanation of how the ResUNet model works for brain tumour segmentation:

Step 1: Input Preparation

The input to the ResUNet model is a brain MRI scan, typically in 3D or 2D format. The MRI scan captures different intensities of the brain tissues, including healthy tissues and tumour regions. The input scan is preprocessed to normalize the intensity values and prepare it for further processing.

Step 2: Encoding Stage

The ResUNet model begins with an encoding stage, which consists of several down-sampling steps to extract hierarchical features from the input image. It uses a U-Net architecture, which is composed of an encoder and a decoder. The encoder part of the network consists of multiple convolutional layers with max-pooling operations. Each convolutional layer learns increasingly complex features as the spatial resolution decreases.

Step 3: Residual Connections

One key aspect of the ResUNet model is the incorporation of residual connections. These connections enable the model to learn residual features, capturing the differences between the input and output of a particular layer. This helps in mitigating the vanishing gradient problem and facilitates better gradient flow during training.

Step 4: Bottleneck Layer

After the down-sampling steps, the ResUNet model reaches a bottleneck layer. This layer captures the most abstract and compressed representation of the input image. It typically consists of multiple convolutional layers, allowing the model to capture high-level features related to tumour regions.

Step 5: Decoding Stage

The decoding stage of the ResUNet model involves up-sampling the feature maps obtained from the bottleneck layer. This stage helps recover the spatial information lost during the encoding stage and reconstruct the output segmentation map. It uses transposed convolutional layers or upsampling followed by convolutional layers to increase the spatial resolution.

Step 6: Skip Connections

To enhance the segmentation performance, skip connections are introduced in the ResUNet model. These connections enable the fusion of feature maps from earlier encoding stages with the up-sampled feature maps in the decoding stage. By doing so, the model can combine low-level and high-level features, capturing both fine-grained and contextual information.

Step 7: Output Prediction

The final layer of the ResUNet model is a convolutional layer with a softmax activation function. This layer produces a probability map where each pixel represents the likelihood of being a tumour or a specific tumour class. The model is trained using labeled data, where the ground truth segmentation masks are available, and the loss function is computed between the predicted output and the ground truth.

Step 8: Training and Optimization

The ResUNet model is trained using a large dataset of brain MRI scans with corresponding tumour segmentations. The model parameters are optimized using backpropagation and gradient descent algorithms, minimizing the loss between predicted and ground truth segmentations. Techniques like data augmentation and regularization may be employed to improve generalization and prevent overfitting.

Step 9: Inference

During inference, the trained ResUNet model takes a new brain MRI scan as input and generates a segmentation map, highlighting the tumour regions. Post-processing steps such as thresholding, morphological operations, and connected component analysis may be applied to refine the segmentation map and obtain the final tumour segmentation.

Thus, following these steps, the ResUNet model can effectively segment brain tumours from MRI scans, providing valuable insights for diagnosis and treatment planning.

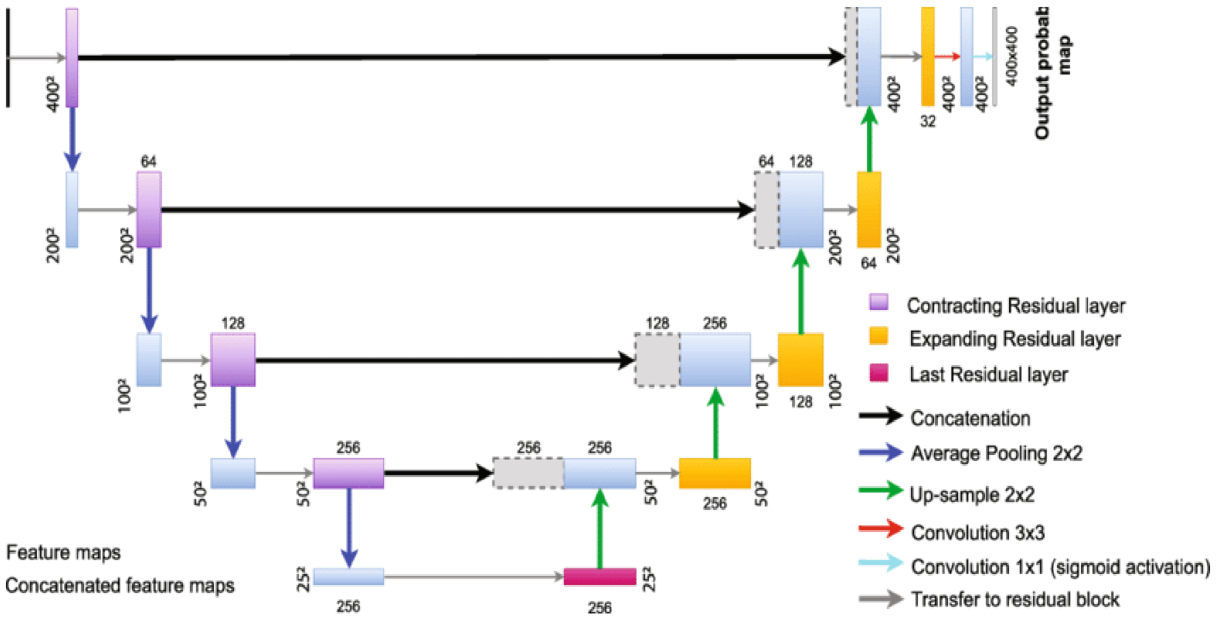


Fig. 4. ResUNet Architecture

9. BINARY CROSS ENTROPY LOSS

Binary cross-entropy loss [6] is a commonly used loss function in machine learning, particularly in binary classification tasks. It measures the dissimilarity between the predicted probability distribution and the true distribution of the target variable.

In brain tumour detection using resnet-50, target variable takes value of 0 for “no tumour” and 1 for “tumour”.

Binary cross-entropy loss is defined as follows:

$$L(y, \hat{y}) = -[ylog(\hat{y}) + (1 - y)\log(1 - \hat{y})]$$

L(y, ŷ) is the binary cross-entropy loss

y is the true label (0 or 1)

ŷ is the predicted probability of the positive class (usually between 0 and 1)

The formula consists of two terms, each corresponding to one of the possible class labels. When y = 1, the first term -y * log(ŷ) measures the loss when the true label is positive. Conversely, when y = 0, the second term (1 - y) log (1 - ŷ) measures the loss when the true label is negative.

Intuitively, the loss penalizes incorrect predictions that are far from the true label. When the predicted probability ŷ approaches the true label y, the loss tends to zero. The loss is maximized when the predicted probability is far from the true label, approaching either 0 or 1.

During training, the goal is to minimize this loss function using optimization algorithms like gradient descent. By adjusting the model's parameters, the algorithm attempts to find the optimal values that minimize the binary cross-entropy loss and improve the model's predictive performance on the binary classification task .

10.TVERSKY SCORE

The Tversky score [7] is a similarity metric commonly used for evaluating image segmentation tasks such as brain tumour segmentation. It measures the spatial agreement between the predicted segmentation mask and the ground truth mask.

Tversky(P, Y) is defined as :

$$\text{Tversky(P, Y)} = \frac{|\,\mathrm{P}\cap\mathrm{Y}\,|}{|\,\mathrm{P}\cap\mathrm{Y}\,| + \alpha|\,P\backslash Y\,| + \beta|\,Y\backslash P\,|}$$

- $|P \cap Y|$ = number of pixels that are correctly classified as tumour in both masks
- $|P \setminus Y|$ = number of pixels classified as tumour in the predicted mask but not in the ground truth mask (false positives)
- $|Y \setminus P|$ is the number of pixels classified as tumour in the ground truth mask but not in the predicted mask (false negatives).
- α and β are weighting factors that control the balance between false positives and false negatives. Typically, they are set to 0.7 and 0.3, respectively, to give slightly more weight to false positives.

The Tversky Score allows for adjusting the balance between false positives and false negatives by modifying the weighting factors. By doing so, it provides flexibility in emphasizing certain types of errors based on specific requirements of the segmentation task. The advantage of the Tversky score over the Dice score is its ability to adjust the balance between false positives and false negatives. By tuning the alpha and beta parameters, we can control the trade-off between under-segmentation and over-segmentation. This flexibility can be particularly useful in scenarios where the emphasis on false positives and false negatives needs to be adjusted based on the specific applications.

11. FOCAL TVERSKY LOSS

For segmentation, I have used the Focal Tversky loss [8] which is an extension of the Tversky loss, which is a similarity measure commonly used in image segmentation tasks. The Focal Tversky loss aims to address the issue of class imbalance by emphasizing the importance of “hard-to-segment” regions during the training process.

Why I used Focal Tversky loss instead of Dice Loss:

1. Imbalanced Class Distribution:

In medical imaging, class imbalance is a common issue, where the background class greatly outweighs the target class (e.g., tumour). The dice coefficient is sensitive to class imbalance, which means that it may not effectively capture the performance of the model in segmenting the smaller class. The Focal Tversky loss addresses this issue by allowing for the adjustment of sensitivity to false positives and false negatives, enabling better handling of imbalanced datasets.

2. False Positive and False Negative Variations:

The Focal Tversky loss provides a mechanism to tune the loss function's sensitivity to false positives and false negatives independently. This can be beneficial in scenarios where false positives or false negatives have different consequences or where the emphasis on one type of error is desired over the other. By adjusting the focal parameter, the loss function can be customized to prioritize minimizing false positives or false negatives, depending on the specific requirements of the segmentation task.

3. Robustness to Small Lesions:

Brain tumours can vary in size, and small lesions can be challenging to accurately segment. The Focal Tversky loss, by adjusting the focal parameter, can help improve the model's performance in identifying and segmenting smaller tumour regions. It allows the loss function to assign more importance to these regions, thereby potentially enhancing the model's sensitivity to smaller tumour lesions.

The formula for the Focal Tversky Loss (FTL) is as follows:

FTL = $(1 - \mathrm{Tversky}(\mathrm{P}, \mathrm{Y})^{gamma}$

- **P:** predicted Segmentation
- **Y:** ground truth segmentation
- **Tversky**: measures the similarity between the predicted and target segmentation masks.
- **gamma**: a tuneable parameter that controls the degree of emphasis on hard examples. Typically, gamma is set to a value between 0 and 5.

The Focal Tversky loss penalizes false positives and false negatives differently based on their difficulty. By using the (1 - Tversky) term raised to the power of gamma, the loss function assigns higher weights to hard examples, making the model focus more on correctly segmenting challenging regions.

12. ALGORITHM OF PROPOSED METHOD

Step 1: Data Preparation

- We gather the dataset of brain MRI images with corresponding FLAIR masks, where the masks highlight the tumour regions.
- Then we split the dataset into training, testing, and validation sets.

Step 2: Training ResNet-50 Model for Tumour Detection

- We then initialize the ResNet-50 model with pre-trained weights on a large image dataset such as ImageNet.
- After initialising the model, we need to modify the model's output layer to have two classes: tumour and non-tumour.
- We then compile the model using an appropriate loss function, here I used binary cross-entropy, and an optimizer, such as Adam.
- Then we train the model on the training set using the following steps:

1. A batch of brain MRI images and their corresponding FLAIR masks are loaded.
2. The images are then fed into the ResNet-50 model and the predicted probabilities for each class is obtained.
3. We then compare the predictions with the ground truth labels (tumour or non-tumour) from the FLAIR masks.
4. The loss between the predictions and ground truth are then computed using binary cross entropy loss.
5. Backpropagation of the gradients are done and the model's weights using the optimizer are updated.
6. Repeat the training process for multiple epochs until the model converges.

Step 3: Testing ResNet-50 Model for Tumour Detection

- Evaluation of the trained ResNet-50 model on the testing set is done using the following steps:

1. A batch of brain MRI images and their corresponding FLAIR masks are loaded.
2. Feed the images into the ResNet-50 model and obtain the predicted probabilities for each class.
3. Compare the predictions with the ground truth labels (tumour or non-tumour) from the FLAIR masks.
4. Compute metrics such as accuracy, precision, recall, and F1-score to assess the model's performance.

Step 4: Validation of ResNet-50 Model for Tumour Detection

- Evaluate the trained ResNet-50 model on the validation set using the same steps as in the testing phase.
- Fine-tune the model, if necessary, based on the validation results.

Step 5: Segmentation [5] using ResUNet Model

- We pass the tumour-positive images from the ResNet-50 model through the ResUNet model for tumour segmentation.
- Initialize the ResUNet model with appropriate architecture for segmentation.
- Compile the model using an appropriate loss function, such as binary cross-entropy, and an optimizer, such as Adam.
- Train the model on the training set using the following steps:
  1. Load a batch of tumour-positive brain MRI images and their corresponding FLAIR masks.
  2. Feed the images into the ResUNet model and obtain the predicted segmentation masks.
  3. Compare the predicted segmentation masks with the ground truth masks.
  4. Compute the loss between the predictions and ground truth.

5. Backpropagate the gradients and update the model's weights using the optimizer.
6. Repeat the training process for multiple epochs until the model converges.

Step 6: Testing ResUNet Model for Tumour Segmentation

Evaluate the trained ResUNet model on the testing set using the following steps:

- Load a batch of tumour-positive brain MRI images and their corresponding FLAIR masks.
- Feed the images into the ResUNet model and obtain the predicted segmentation masks.
- Compare the predicted segmentation masks with the ground truth masks.
- Compute metrics such as Tversky Loss to assess the model's performance.

Step 7: Validation of ResUNet Model for Tumour Segmentation

- Evaluate the trained ResUNet model on the validation set using the same steps as in the testing phase.
- Fine-tune the model, if necessary, based on the validation results.

Step 8: Final Tumour Detection and Segmentation

- Apply the trained ResNet-50 model on the input brain MRI images.
- Identify tumour-positive images based on the model's predictions.
- Pass the tumour-positive images through the trained ResUNet model for tumour segmentation.
- Obtain the final segmentation masks for the tumours.

Step 9: Output

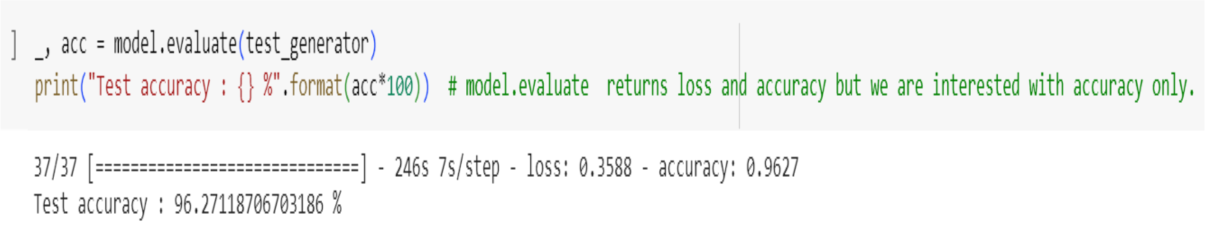
- We visualize and/or save the tumour detection results and the corresponding tumour segmentation masks for each input brain MRI image.

13. RESULTS AND ANALYSIS

The project uses either colour or grey scale intensity images with 256×256 size. First Tumour is detected in MR Image of Brain. If it exists then it is passed to the Segmentation Model which segments the Brain Tumour. The detection is done using the value of predicted mask which is compared against actual mask.

The main objective of segmentation is to cluster pixels into image region and help identify region of interest i.e., to locate the tumour and its size. The pixels' segmented portion is compared with normal brain image. This kind of comparison helps to locate the abnormal parts of brain tumour.

First, we need to detect tumour using Resnet-50 model. The accuracy of detection is shown below along with binary cross entropy loss vs epoch and accuracy vs epoch plots respectively.



```
] _, acc = model.evaluate(test_generator)
  print("Test accuracy : {} %".format(acc*100))  # model.evaluate  returns loss and accuracy but we are interested with accuracy only.

  37/37 [============================] - 246s 7s/step - loss: 0.3588 - accuracy: 0.9627
  Test accuracy : 96.27118706703186 %
```

Fig. 5. Brain tumour detection accuracy of RESNET-50 is 96.27%

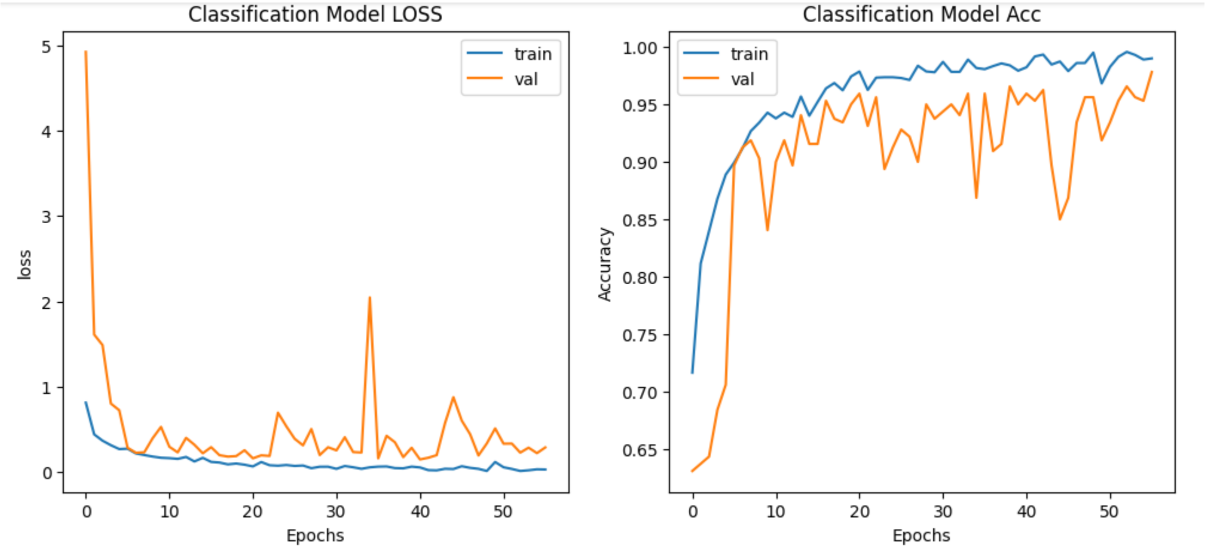
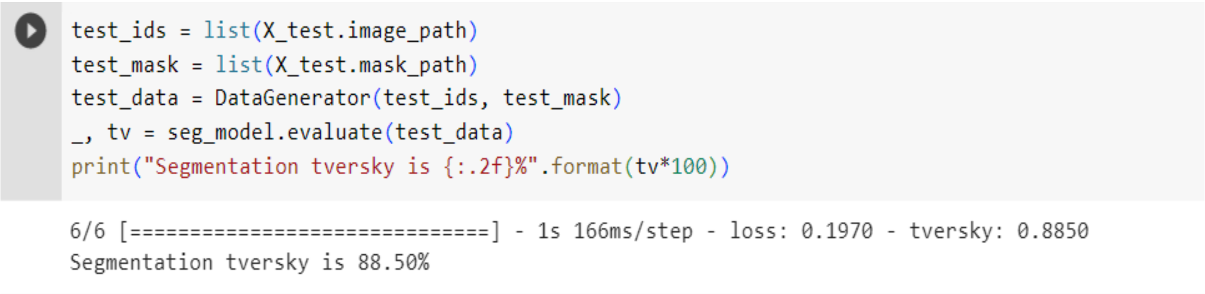


Fig. 6. binary cross entropy loss vs epoch and accuracy vs epoch plots

Next, of the MR images which have tumour, we pass them to the ResUNet model for segmentation. The accuracy of segmentation, model's focal Tversky loss vs epoch and Tversky score vs epoch plots are shown below respectively.



```
test_ids = list(X_test.image_path)
test_mask = list(X_test.mask_path)
test_data = DataGenerator(test_ids, test_mask)
_, tv = seg_model.evaluate(test_data)
print("Segmentation tversky is {:.2f}%".format(tv*100))
```

```
6/6 [==============================] - 1s 166ms/step - loss: 0.1970 - tversky: 0.8850
Segmentation tversky is 88.50%
```

Fig. 7. Brain tumour segmentation accuracy of ResUNet

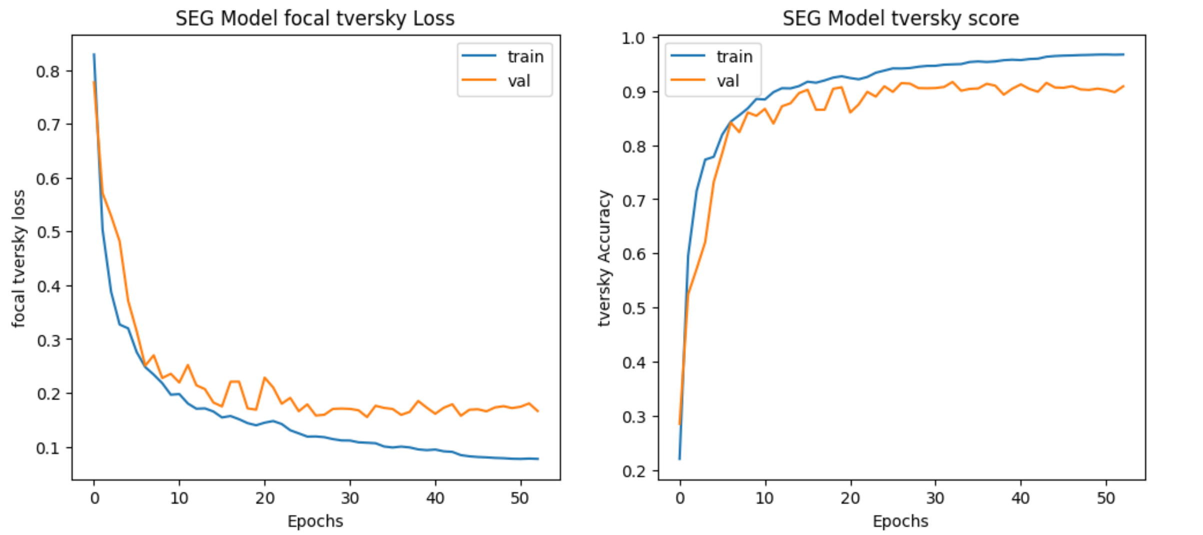


Fig. 8. focal Tversky loss vs epoch and segmentation Tversky score vs epoch plots

Final Predictions:

The images and their corresponding descriptions are given respectively. Red portion is actual mask while green portion is predicted mask. Also, the actual and the predicted masks are shown separately.

Output 1:

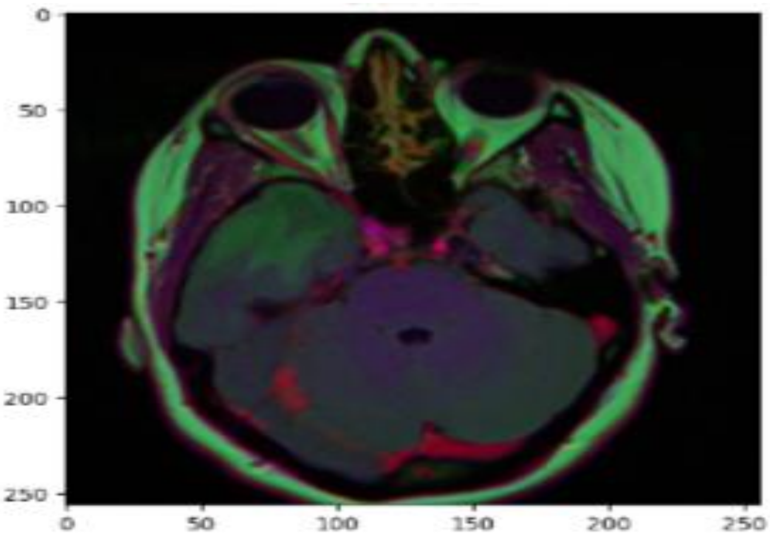
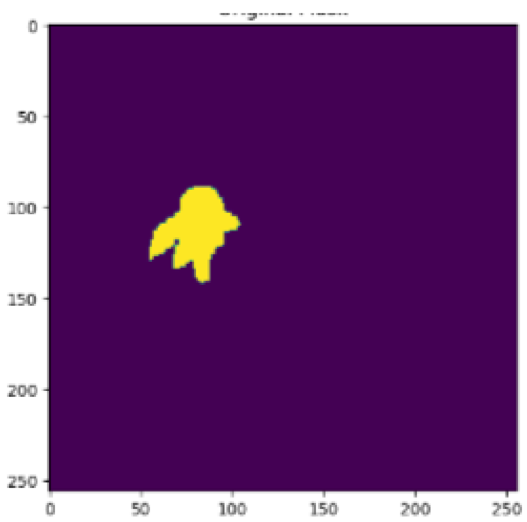


Fig. 9.1 Original MRI



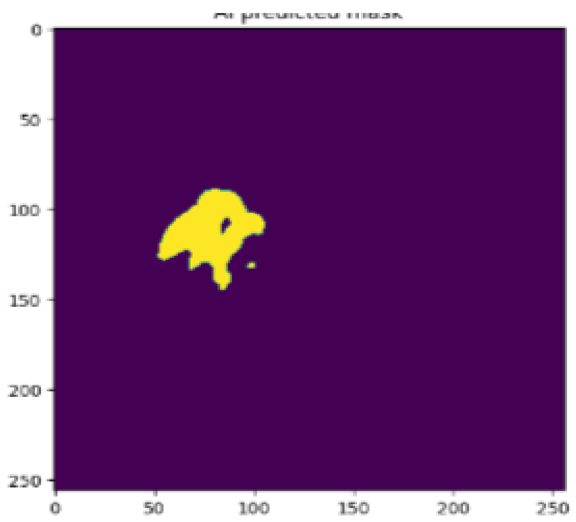


Fig. 9.2 Original mask

Fig. 9.3 Predicted mask

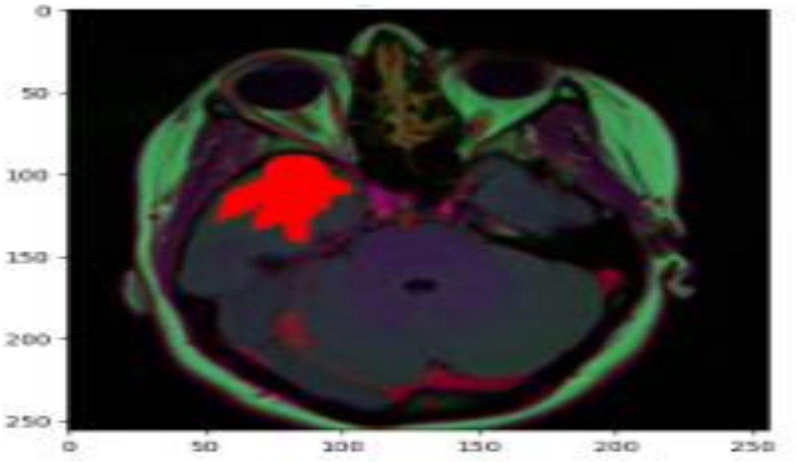


Fig 9.4 Brain MRI with original mask

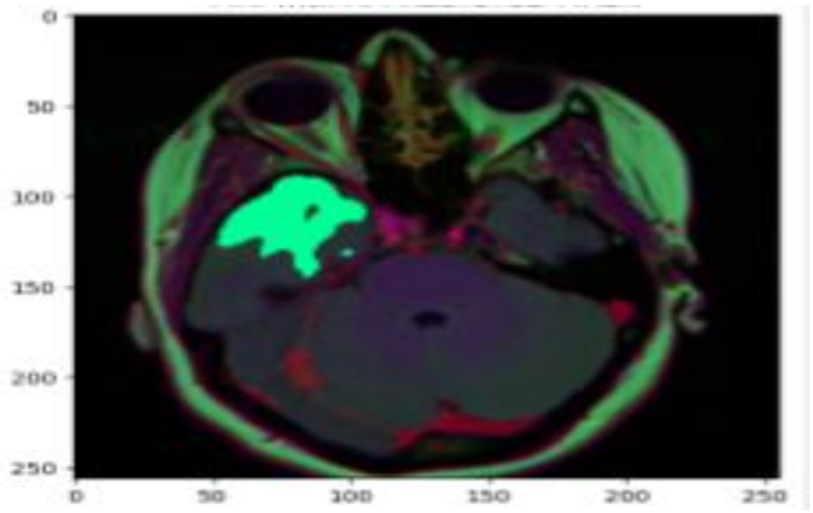


Fig 9.5 Brain MRI with predicted mask

Output 2:

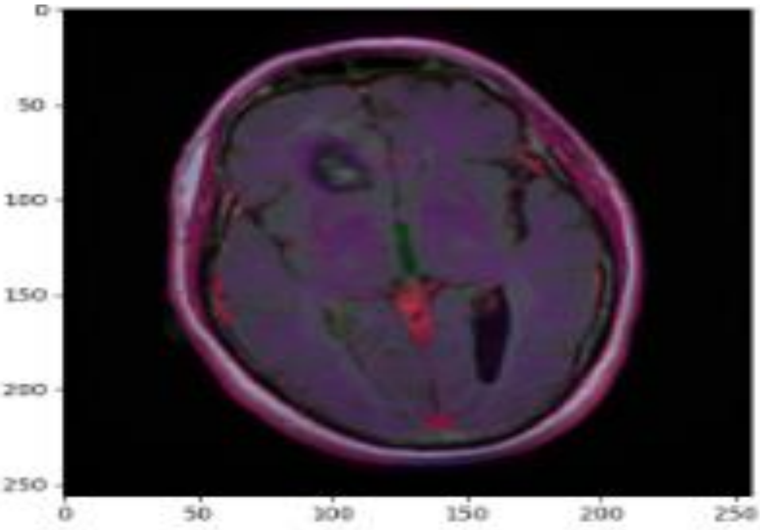


Fig. 10.1 Original MRI

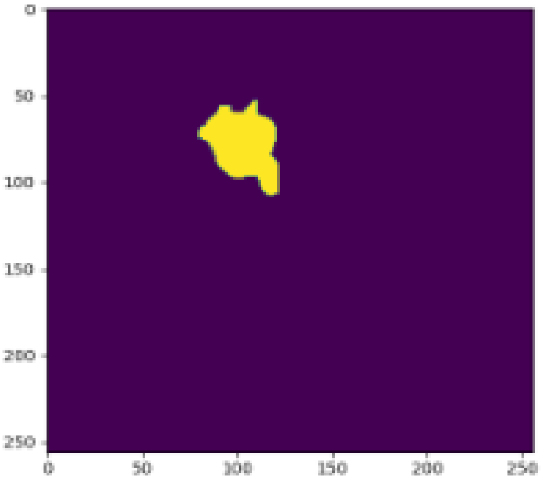


Fig 10.2 Original mask

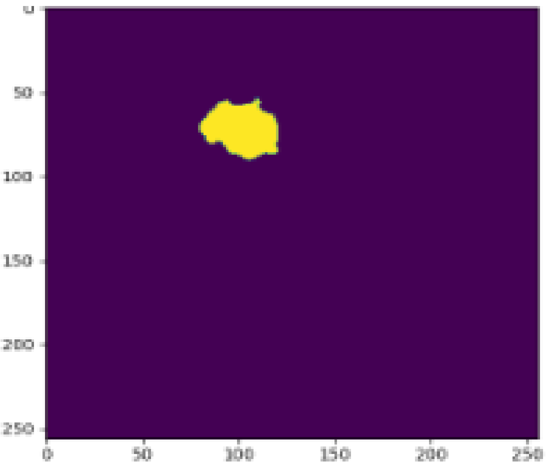


Fig 10.3 Predicted mask

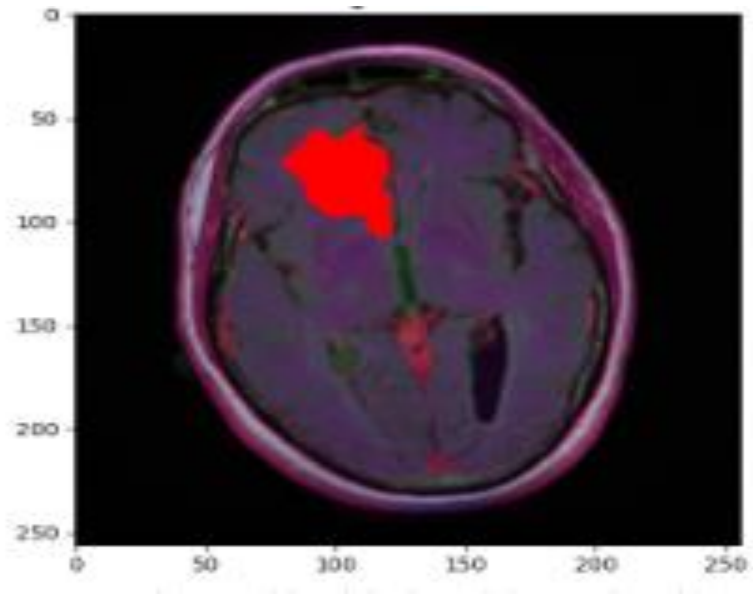


Fig. 10.4 Brain MRI with original mask

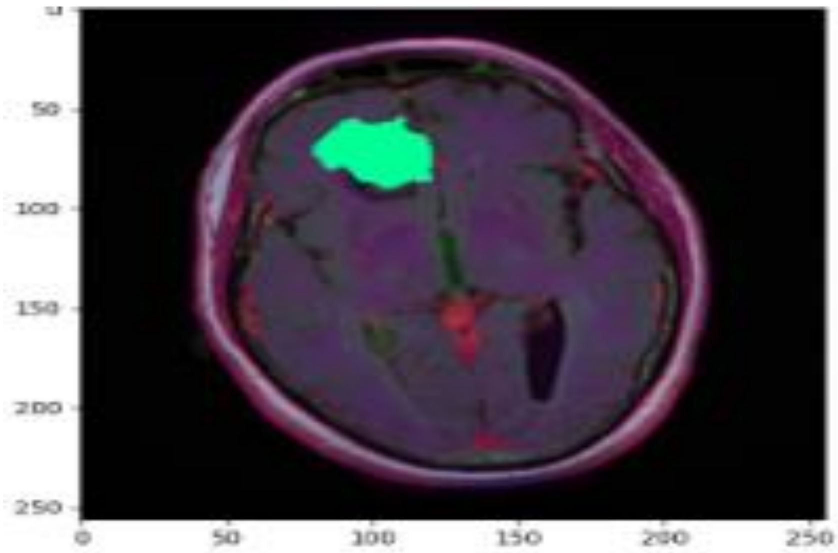


Fig 10.5 Brain MRI with predicted mask

Output 3:

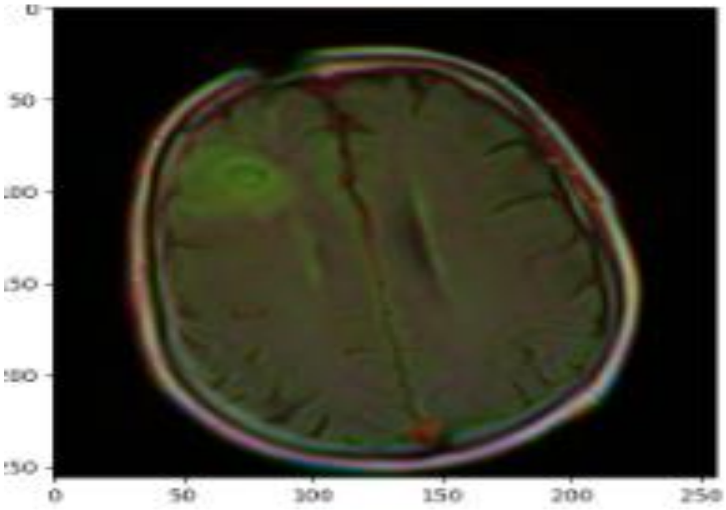


Fig. 11.1 Original MRI

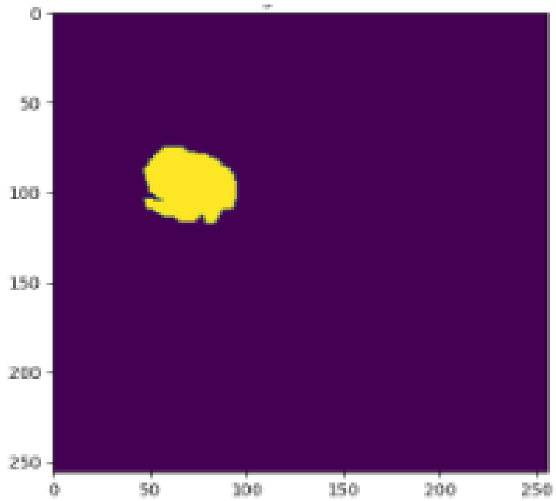


Fig. 11.2 Original mask

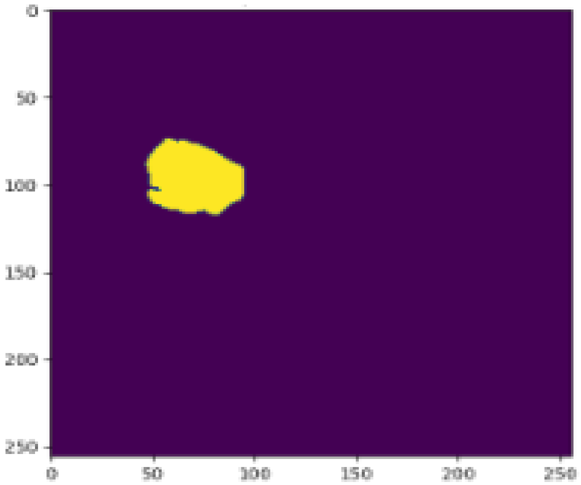


Fig. 11.3 Predicted mask

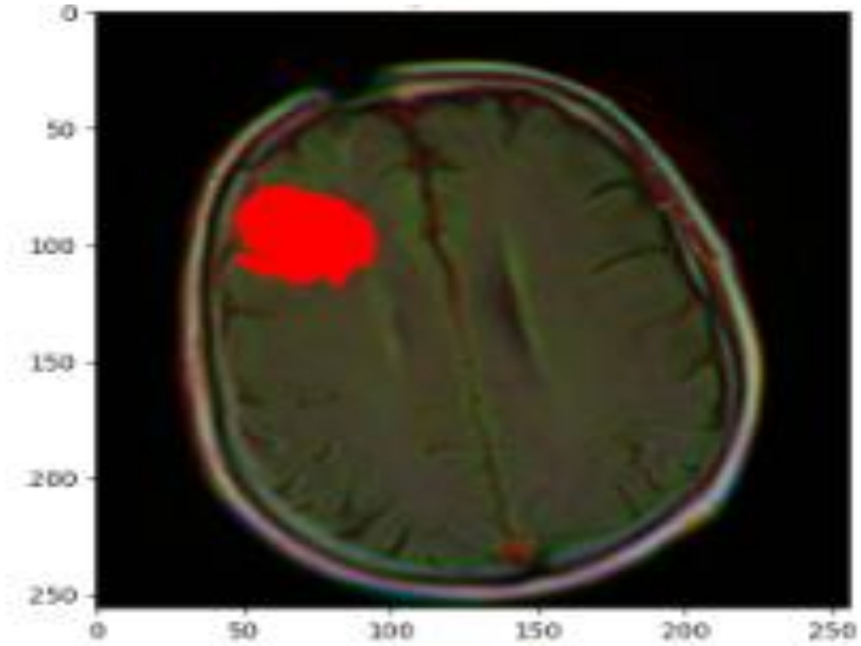


Fig. 11.4 Brain MRI with original mask

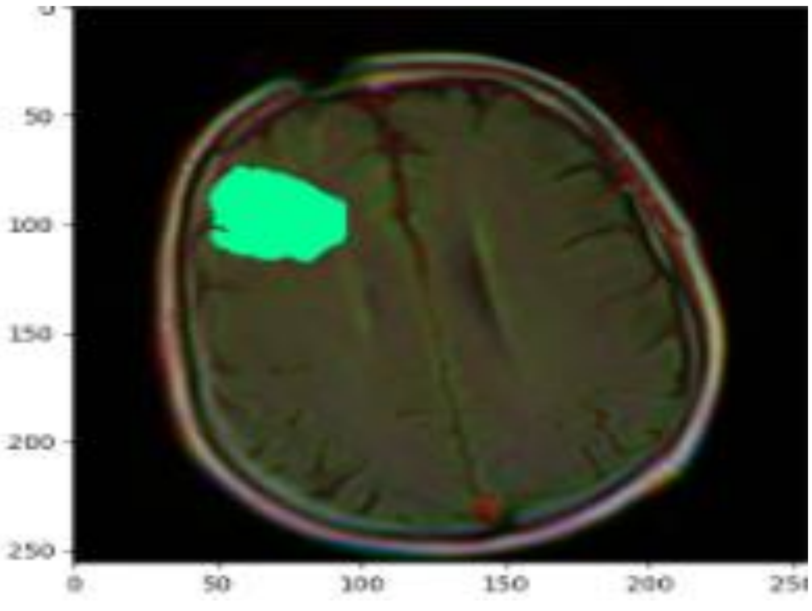


Fig. 11.5 Brain MRI with predicted mask

14. CONCLUSION

In conclusion, the segmentation of brain tumour plays an important role in diagnostic procedures. The accurate segmentation helps in clinical diagnostic, but also helps to increase the lifetime of the patient if detected early on.

In this project work detection of brain tumour is done using ResNet-50 architecture and the segmentation of brain tumour is implemented using ResUNet architecture. The tumour detection accuracy of the resnet-50 architecture was about 96.27%. The tumour segmentation Tversky score was about 88.50%.

The ResUnet model helps perform the segmentation task accurately. The training and testing speed is increased using max pooling, maxout and dropout. The speed is also increased by reducing the features in the fully connected layer. Reduction of parameters also causes reduction of over fitting. The result shows that implemented method helps in detection of tumour as well as specifying tumour to the actual tumour region only.

15. REFERENCES

[1] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, and N.I. Chervyakov. "Applications of the residue number system to reduce hardware costs of the convolutional neural network implementation". Mathematics and Computers in Simulation B.V. Elsevier, Volume 177, pp 232-243,2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren & Jian Sun, “Deep Residual Learning for Image Recognition”. 10.1109/CVPR.2016.90, pp 770-778, 2016.

[3] A. Zhang, Z.C. Lipton, M. Li, & A.J. Smola, “Dive into Deep Learning” *ArXiv Preprint ArXiv:2106.11342*, 2021.

[4] Debesh Jha, Pia Smedsrud, Michael Riegler, Dag Johansen, Thomas de Lange, Pal Halvorsen, Havard Dagenborg and Simulamet, “ResUNet++: An Advanced Architecture for Medical Image Segmentation”, 10.1109/ISM46123.2019.00049, 2019.

[5] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio & C. Pal (2017). “Brain tumor segmentation with deep neural networks. Medical image analysis, 35, 18-31”.

[6] George Cybenko, Dianne P. O'Leary, Jorma Rissanen, “The Mathematics of Information Coding, Extraction and Distribution”, pp 82 ,1999.

[7] Amos Tvsesky, "Features of Similarity", Psychological Review, 84(4), pp327-352, 1977.

[8] Nabila Abraham and Naimul Khan: “A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation”, 10.1109/ISBI.2019.8759329, pp 683-687,2019.