A Mini Project report on

NEUROINSIGHT: ADVANCED AI DIAGNOSIS AND LOCALIZATION OF BRAIN TUMOURS

A documentation submitted in partial fulfilment of the academic requirement for the award of degree of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

by

SHAHZOR AHMED

(1604-21-748-035)

MOHAMMED ABDUL FAIZAN

(1604-21-748-046)

Under the guidance of

Mrs. Ayesha Mariyam

Assistant Professor, CS&AI Dept., MJCET

COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Osmania University) Hyderabad.

2023 – 2024

MUFFAKHAM JAH
COLLEGE OF ENGINEERING AND TECHNOLOGY

Mount Pleasant, 8-2-249 to 267, Road No.3, Banjara Hills,
Hyderabad-500 034, Telangana State, India.

CERTIFICATE

This is to certify that the mini project report on "**NeuroInsight: "Advanced AI Diagnosis and Localization of Brain Tumours"** is a bonafide work carried out by **SHAHZOR AHMED (1604-21-748-035) and MOHAMMED ABDUL FAIZAN (1604-21-748-046)** in the partial fulfilment of the requirements for the award of the B.E. CSE(AI&ML) in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2023-2024.

Internal Guide	**Head of the Department**
Mrs. Ayesha Mariyam **Assistant Professor,** **CS&AI Dept., MJCET,** **Hyderabad**	**Dr. Uma N. Dulhare** **Professor & Head,** **CS&AI Dept., MJCET,** **Hyderabad**

External Examiner

DECLARATION

We hereby declare that the work entitled "**NeuroInsight: "Advanced AI Diagnosis and Localization of Brain Tumours**" developed under the supervision **Mrs. Ayesha Mariyam**, **Assistant Professor, CS&AI Department** and submitted to **MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY** in original and has not been submitted in part or while for under graduation degree to any other university.

Shahzor Ahmed
(1604-21-748-035)

Mohammed Abdul Faizan
(1604-21-748-046)

ACKNOWLEDGEMENT

Our sincere thanks and gratitude to **Dr. UMA N. DULHARE, Professor & Head of the Department** for all the timely support and valuable suggestions during the period of my mini project.

We are extremely thankful to Project Review Committee Members, Dept. of CS&AI for their encouragement and support throughout the project.

We are extremely thankful and indebted to my internal guide, **Mrs. Ayesha Mariyam**, **Assistant Professor, Department of CS&AI** for her exemplary guidance and encouragement. Her trust and support inspired me in the most important moments of making right decisions and I am glad to work under her supervision.

We would like to thank all the faculty and staff of department who helped us directly or indirectly in completing the project work. Last but not least, I would like to thank my family members and friends for their co-operation for completing the project.

Shahzor Ahmed

(1604-21-748-035)

Mohammed Abdul Faizan

(1604-21-748-046)

TABLE OF CONTENT

CONTENT **PAGE NO**

ABSTRACT

Brain tumours are highly dangerous and diagnosing them quickly is critical for successful treatment. However, detecting and classifying brain tumours is very difficult since they can be ambiguously defined within soft brain tissues. Typically, magnetic resonance imaging (MRI), computerized tomography (CT) scans, and ultrasound images are used to evaluate brain tumours. But thoroughly analysing these scans takes a lot of time and effort. To address this, propose automating the process of brain tumour detection and segmentation from MRI scans using deep learning. The goal is to bridge the gap between initial diagnosis and pinpointing actual tumours, by providing doctors with a quick, non-invasive automated second opinion. Specifically, their technique uses transfer learning with Residual Networks (ResNet) to detect the presence of tumours. It then segments the MRI scan using ResUNet to delineate the precise tumour area. This demonstrates deep learning's potential to efficiently process and analyze medical images. The automated tumour detection and segmentation could assist clinical evaluation and help speed up treatment for this extremely dangerous condition.

KEYWORDS: Deep Learning, Brain Tumour Segmentation, Brain Tumour Detection, MRI, Neural Networks.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1Introduction

Tumours involve abnormal and uncontrolled growth of cells in the body. Brain tumour is one type of tumour that can be especially problematic due to the rigid structure of the human skull. Brain tumours are classified as either primary or secondary.

Primary brain tumours originate within the brain itself and are cancerous (malignant). Secondary brain tumours occur when cancerous cells from tumours in other organs like the lung or kidney spread (metastasize) to the brain. These are not cancerous initially. Brain tumours are also classified based on their location and spread: Local tumours are confined to one hemisphere or part of the brain.

Regional tumours cross the midline between brain hemispheres and invade into bones, blood vessels, etc. Distant tumours have spread to other areas like the nasal cavity, nasal pharynx, etc. Additionally, brain tumours can be categorized based on how they appear on radiological scans like MRI or CT scans.

Brain tumours account for 85-90% of all central nervous system tumours an estimated 18,600 adults (10,500 men and 8,100 women) die each year in the US due to brain tumours The 5-year survival rate after diagnosis is around 31% This means 69% of people with brain tumours pass away within 5 years of diagnosis According to Bauer et al. (2013), the prognosis for brain tumours is poor, with a high mortality rate.

Gliomas, the most prevalent primary brain cancer, are classified into four grades by WHO, with higher grades indicating increased severity. Early detection is crucial, as higher-grade gliomas, such as Glioblastoma Multiforme (GBM), can be highly harmful. Medical imaging modalities like MRI are essential for brain tumour analysis.

While traditional methods like Neural Networks and SVM were common, recent advancements in Deep Learning, particularly Convolutional Neural Networks (CNN), offer more efficiency. The proposed model utilizes Deep Learning, specifically Reset for image classification, and a state-of-the-art Deep Residual UNET (Resu Net) for semantic segmentation. This two-step approach aims to first classify the presence of a brain tumour and then perform segmentation to precisely identify the tumour's location, enhancing diagnostic accuracy and treatment possibilities.

1.2 Problem Statement

To Enhance Tumour Detection and Segmentation in Medical Images by Leveraging Resnet Transfer Learning.

1.3 Objectives

The objectives of this project are listed below:

- To develop an advanced artificial intelligence model capable of accurately detecting the presence of brain tumours in individuals and precisely localizing the tumour within the brain.
- To leverage machine learning algorithms and medical imaging data to create a sophisticated diagnostic tool for enhancing early detection capabilities and providing valuable insights to healthcare professionals to optimize treatment planning and improve patient outcomes.

1.1Organization Of Thesis

The following is the format of the thesis which is outlined in five chapters. A brief description of each chapter is as follows:

The first chapter consists of the Introduction, Objectives, and Organization of Thesis. Second chapter contains the Literature Survey. This chapter focuses on the areas such as the Existing System, and the Problems with the Existing System. Third chapter explains about the Proposed System. Here, we talk about what the Proposed system can do, the Problem Statement and the System Architecture. Fourth chapter is about the Methodologies used. It explains about the various kinds of methods used in this project and about the various Technologies used. Fifth chapter is the Implementation. It tells us about the Requirements (both Hardware Requirements and Software Requirements), the Code Snippets explaining the main parts of the project with code, and the Execution part. Finally, the Sixth chapter talks about the Result Analysis and Conclusion. Here, we can see the Result Analysis, Conclusion on the Results obtained and Future Enhancement which is followed by references.

CHAPTER 2

LITERATURE SURVEY

Mustafa R. Ismael et al [1] have proposed a new model for the statistical feature-based classification of brain tumours in MRI images using neural networks.

Methods:

In their study, the researchers employed a statistical feature-based approach for classifying brain tumours in MRI images. They utilized neural networks as the primary algorithm for classification. The approach involved identifying the region of interest (ROI), which represents the tumor segment in the MRI images. Statistical features were extracted from the ROI using a combination of 2D Discrete Wavelet Transform (DWT) and 2D Gabor filter techniques. These features were then used to create a feature set comprising a comprehensive set of transform domain statistical features.

Results:

The study utilized a dataset consisting of 3,064 slices of T1-weighted MRI images with three types of brain tumours: Meningioma, Glioma, and Pituitary tumour. The classification accuracy achieved by the proposed model was 91.9%. Additionally, the specificity for Meningioma, Glioma, and Pituitary tumor was found to be 96%, 96.29%, and 95.66% respectively. These results indicate the effectiveness of the proposed feature selection method and its potential for enhancing classification performance.

Drawbacks:

While the study demonstrated promising results, it may have some limitations. For example, the performance of the model may vary when applied to different datasets or when dealing with other types of brain tumours not included in the study. Additionally, the computational complexity of the feature extraction process and neural network training could be a potential drawback in terms of time and resource requirements.

Vinayak K. Bairagi et al. [2] have proposed a new model, an automated approach for brain tumour detection leveraging convolutional neural networks (CNNs).

Methods:

In their study, the authors proposed an automated approach for brain tumor detection using convolutional neural networks (CNNs). They explored various CNN architectures, including Alexnet, VGG-16, GooGLeNet, and RNN, to analyze MRI images and classify them into tumorous and non-tumorous classes. The focus of their research was on tuning the hyperparameters for two specific architectures, Alexnet and VGG-16.

Results:

The authors conducted experiments using datasets such as BRATS 2013, BRATS 2015, and OPEN I, which comprised a total of 621 images. They achieved an impressive accuracy of 98.67% using CNN Alexnet for automatic detection of brain tumors. This accuracy was obtained during testing on a subset of 125 images, demonstrating the effectiveness of their proposed approach.

Drawbacks:

However, the study lacked thorough discussion on potential drawbacks or limitations of the proposed framework. Additionally, there was limited insight provided into the scalability and generalizability of the approach, which could affect its applicability in real-world scenarios.

Arbane et al. [3] introduced a novel model titled "Transfer Learning for Automatic Brain Tumour Classification Using MRI

Methods:

In this study, a deep learning model for brain tumor classification from MRI images was proposed. The methodology focused on utilizing convolutional neural network (CNN) architectures based on transfer learning. Specifically, the researchers explored the effectiveness of CNN architectures such as ResNet, Xception, and MobileNet-V2 for this task. Transfer learning was employed to leverage pre-trained models and adapt them to the specific requirements of brain tumor classification from MRI images.

Results:

The implemented system yielded promising results in terms of accuracy and F1-score. Among the explored CNN architectures, MobileNet-V2 demonstrated the best performance, achieving an accuracy of 98.24% and an F1-score of 98.42%. These results indicate the effectiveness of the proposed deep learning approach in accurately classifying brain tumors from MRI images.

Drawbacks:

Despite the success in achieving high accuracy, there are certain drawbacks associated with the proposed methodology. Accurate classification heavily depends on the quality of the MRI images used, which may pose challenges in real-world applications where image quality varies. Furthermore, the computational resources required for training deep learning models, especially with large datasets, can be substantial, presenting a potential barrier to widespread implementation.

J. Vijay et al. [4] have proposed an efficient brain tumour detection methodology leveraging the K-means clustering algorithm.

Methods:

The study focuses on automating brain tumour segmentation from MR images. It employs the K-means clustering algorithm for segmentation, which efficiently extracts tumour tissues. The method aims to enhance tumour boundaries and is faster compared to other clustering algorithms.

Results:

The proposed technique for brain tumour segmentation using K-means clustering produces appreciable results. It effectively extracts tumour tissues from MR images, demonstrating its potential for accurate segmentation.

Drawbacks:

While the method shows promise in brain tumour segmentation, it may have limitations in handling complex tumour shapes or heterogeneous tissue characteristics. Additionally, the accuracy of segmentation may vary depending on the quality and resolution of the MR images used.

Ayesha Younis et al. [5] investigated brain tumour analysis using deep learning and ensemble learning approaches with VGG-16.

Methods:

The study employed the Visual Geometry Group (VGG 16) convolutional neural network (CNN) model framework for brain tumour detection. Utilizing CNNs, the methodology focused on classifying brain tumor images with high accuracy. Additionally, the researchers critically analysed existing literature solutions and proposed advanced AI and Neural Network algorithms for early disease detection. The study utilized MRI images dataset comprising 253 brain images, with 155 showing tumours, for model evaluation.

Results:

The evaluation of the proposed methodology yielded promising outcomes. The VGG 16 architecture achieved an impressive accuracy rate of 98.5%, outperforming conventional approaches in brain tumour detection. The algorithm demonstrated superior performance, particularly in terms of precision and F1-score, compared to existing methods. This indicates the potential of the proposed approach for efficient and precise detection of brain tumours, contributing significantly to early disease diagnosis and intervention.

Drawbacks:

Despite the success of the proposed methodology, certain limitations and drawbacks were identified. One notable limitation is the dependency on high-quality MRI images for accurate classification. Variations in image quality and resolution could potentially impact the performance of the classification algorithm, leading to inaccuracies in tumour detection. Additionally, further validation on diverse datasets is necessary to ensure the generalizability and robustness of the proposed approach across different imaging conditions and patient demographics. Addressing these drawbacks is essential for enhancing the reliability and applicability of the proposed brain tumour detection system.

Table 2.1: Summary of Literature Survey

SNO	Title of paper	Methodology	Results	Advantages	Drawbacks
1	Statistical Feature-Based Brain Tumour Classification in MRI with Neural Networks [2018]	Classified brain tumours using statistical features and neural networks. ROI segmentation and DWT/Gabor filter for feature selection. Used backpropagation NN for classification.	Reached 91.9% total accuracy, with specificities of 96%, 96.29%, and 95.66% for Meningioma, Glioma, and Pituitary tumour.	Achieved 91.9% total accuracy, with specificities of 96%, 96.29%, and 95.66%.	Limited insight into the scalability and generalizability of the approach.
2	Automatic Brain Tumor Detection Using CNN [2023]	Explored CNN architectures (AlexNet, VGG-16, GoogLeNet, RNN) for automated brain tumour recognition in MRI images,	Reached 98.67% accuracy with CNN AlexNet for brain tumour detection.	Significant improvement over manual methods for brain tumour detection	Lack of discussion on potential limitations or challenges of the proposed approach
3	Deep learning model for brain tumour classification using CNN-based transfer learning	Proposed CNN-based transfer learning for brain tumor classification from MRI images.	Achieved 98.24% accuracy and 98.42% F1-score with MobileNet-V2.	Improved brain tumour classification from MRI using transfer learning with various CNN architectures.	Dependency on high-quality MRI images for accurate classification, used small dataset for training model
4	Automatic Brain Tumour Segmentation using K-means clustering	Utilizes K-means clustering for automated brain tumour segmentation from MR images.	Method offers improved tumour boundary delineation and faster processing than other clustering algorithms.	Efficient tumour tissue extraction - Improved boundary delineation - Rapid processing	May need validation on diverse datasets - Relies on image quality for accurate segmentation
5	Brain Tumour Detection Using VGG 16	Utilized VGG 16 CNN for brain tumour detection.	High 98.5% accuracy with VGG 16,	Efficient and precise detection of brain tumours.	Dependency on high-quality MRI images for accurate classification.

CHAPTER 3

EXISTING SYSTEM

3.1 Introduction

Some of the existing systems for detecting and localizing brain tumours using MRI are based on machine learning, deep learning. These systems can perform tasks such as classification, segmentation, and boundary extraction of the tumour regions from the MRI images. For example, some systems use convolutional neural networks (CNNs) like Alex Net, SVM, k- means clustering, VGG 16 to classify the images into normal or abnormal, and then use region-based CNNs (R-CNNs) to localize the tumour regions of interest. Other systems use deep learning models such as U-Net, V-Net, or 3D CNNs to segment the tumour regions from the MRI images. Several studies explore the application of deep neural networks for segmentation tasks in medical imaging, emphasizing the significance of machine learning. The following summaries provide insights into different approaches.

Gram purohit et al. (2020):

- Investigated CNN and VGG-16 for MRI-based tumour detection.
- Evaluated network performance based on sensitivity, specificity, and precision.
- Integrated convolutional neural networks with feature extraction techniques.

Jha et al. (2019):

- Proposed Resu Net++ for semantic segmentation with residual blocks, ASPP, and spotlight blocks.
- Demonstrated improved segmentation results compared to other methods.
- Achieved efficacy with a smaller number of images.

Ali et al. (2020):

- Proposed an ensemble of 3D CNN and 3D U-Net for tumour delineation.
- Highlighted better results with ensemble methods compared to uniform weighting.
- Aimed for efficient neural networks with improved accuracy.

Pereria et al. (2016):

- Addressed bias field distortion using N4ITK method and intensity normalization.
- Implemented Xavier Initialization for CNN initialization and LReLU activation.
- Employed regularization and data augmentation techniques for segmentation on a small dataset.

3.2 Drawbacks of Existing System

- Challenges in segmenting neurological disorders from comprehensive imaging data - Variation in clot size and type - Dependence on manually labelled data - May require further validation and testing on diverse datasets.
- Limited focus on a few medical imaging models for brain tumours-Scope for more exploration and application of powerful deep CNN models -Future consideration for expanding the dataset and improving accuracy- Application of the method to other medical imaging modalities (ultrasound, CT, x-ray).
- Limited Explanation of Novel Methods: The mention of examining novel methods for feature extraction in future work lacks specific details. Providing more information on these novel methods would enhance the clarity of the proposed advancements. Lack of Standardization and Normalization: Storing images with no standardization and shading.

CHAPTER 4
PROPOSED SYSTEM

4.1 Introduction

- Convolutional neural networks in deep learning have gained a lot of importance and are extensively used in medical imaging in recent times and as a method for the non-invasive diagnosis of various diseases.
- It is a very complex model and requires high computational power. It is quite expensive computationally to use optimized GPUs and to use high-resolution images for the same. Also, traditionally a lot of feature extraction and image augmentation or enhancement techniques are needed to train the model efficiently.
- This is the reason the concept of Transfer Learning was introduced and is continuously evolving in Deep Learning. In Transfer Learning, an algorithm is pre-trained on a large dataset of images having many classes using heavy GPUs.
- This pre-trained model can then be used for relevant problem statements at hand by adding new fully connected convolutional layers and providing relevant data to overcome the problem of complex and time-consuming model training.
- Other networks like Alex Net, ResNet, and VGG-16 architectures were considered for solving the classification problem. On comparing the three algorithms, in Alex Net, there are 7 layers and 60 million parameters. Its error rate is 15.3% which was higher than the other two algorithms. In VGG-16, the error rate is 7.3% and 16 layers.
- But it has 138 million trainable parameters which makes the algorithm complex and difficult to handle. ResNet has a very low error rate comparatively of 3.57%. It has 152 layers but still has lower complexity than the VGG-16 model. Therefore, the ResNet architecture for this problem at hand was more suitable.

ResNet

- ResNet is trained on the ImageNet dataset that has around 11 million images and over 1000 classes. The algorithm is used to extract features from input images. ResNet works on the concept of residual connections. Neural networks are hard to train because of the famous vanishing gradient problem that occurs during backpropagation.
- This can saturate the performance or even degrade the results in the worst case. Identity shortcut connections are introduced in ResNet which skips one or more layers which facilitate the training of the model without the vanishing gradient problem.
- The skip connections allow the model to learn an identity mapping function which makes sure that the next higher level will perform at least as good as the lower or previous layer as it trains the 152 layers

without the vanishing gradient problem and not worse than that. The ResNet-50 model involves a design of 5 stages.

- Each stage is made up of an identity block and a convolution block. The convolutional block and the identity block both consist of three convolutional layers.
- The ResNet-50 model has over 23 million trainable parameters. Binary cross-entropy was used as a loss function and Adam optimizer.

ResUNet

- RESUNET refers to Deep Residual UNET. It's an encoder decoder architecture with a bridge/bottleneck between them acting as the connector developed for semantic segmentation. RESUNET is an advancement of the UNet and is a fully connected convolutional neural network that is developed to get high performance by using fewer parameters.
- It is an improvement of the existing UNET architecture. RESUNET takes the advantage of both the UNET architecture and Deep Residual Learning to design a better and efficient segmentation model. The use of residual blocks helps in developing a network that is deeper without worrying about the vanishing gradient problem or exploding gradients.
- It also helps in the efficient training of the network. The rich skip connections in the RESUNET facilitate better flow of information between different layers, which helps in better flow of gradients while training - i.e. during backpropagation. This architecture consists of an encoder or a contraction path, a decoder or expansion path, and a bridge or a bottleneck connecting both these paths, similar to a U-Net. In the case of RESUNET, these layers have pre-activated residual blocks. The term gets its name because of its U shape architecture.

Encoder

- The encoder, also referred to as the contraction path, takes the input image and passes it through different encoder blocks of residual blocks and max-pooling layers, which helps the network to learn the representation in an abstract way.
- Each encoder block's output acts like a skip connection for the corresponding decoder block present in the decoder. To reduce the spatial dimensions, strides are used.

Decoder

- The decoder, also referred to as the expansion path takes the feature map from the bridge, also referred to as the bottleneck, or the up sampled input from the previous layer of the decoder and concatenates the various output features from the corresponding encoder block or the contraction path and learns a better semantic representation, which is used to generate a segmentation mask.
- The decoder consists of three/four decoder blocks, and after each block, upscaling is performed, i.e. the spatial dimensions of the feature map are doubled and the number of feature channels is reduced.

output of the last decoder is passed through a 1×1 convolution filter with sigmoid activation function. The sigmoid activation function is used to give the segmentation mask representing the pixel-wise classification

4.2 System Architecture

4.2.1 Working Structure of Brain Tumour Detection

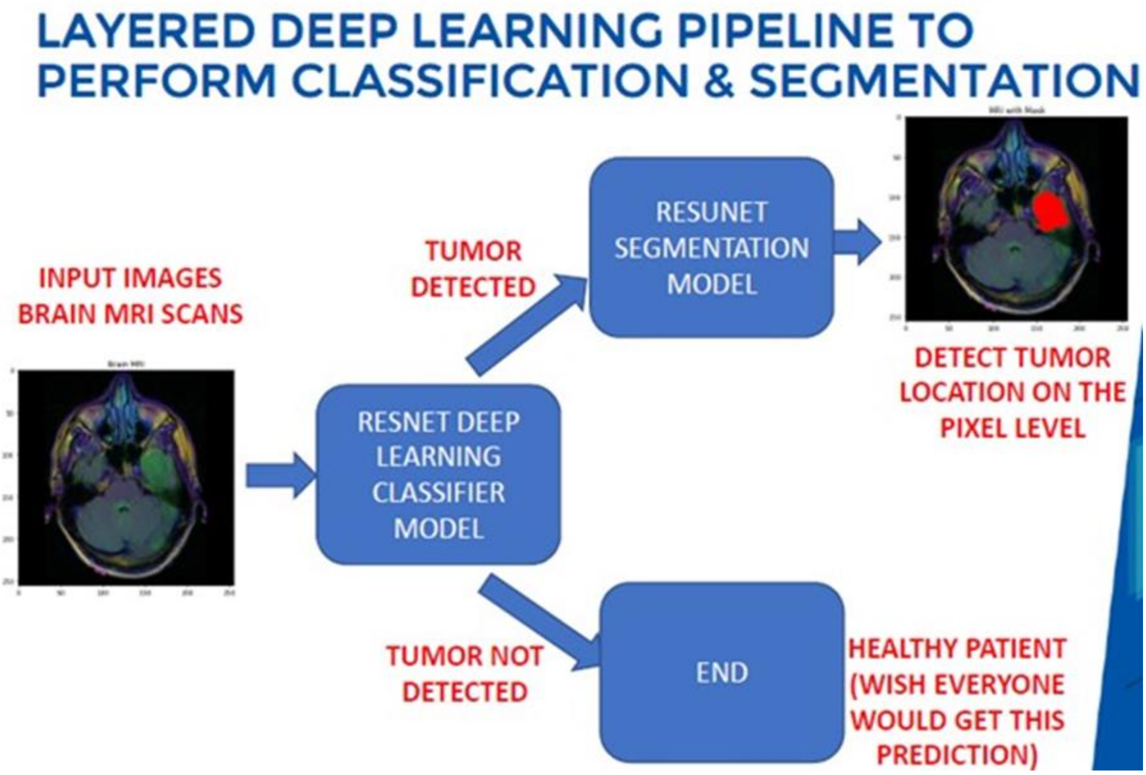


Figure 4.1Layered deep learning pipeline to perform classification

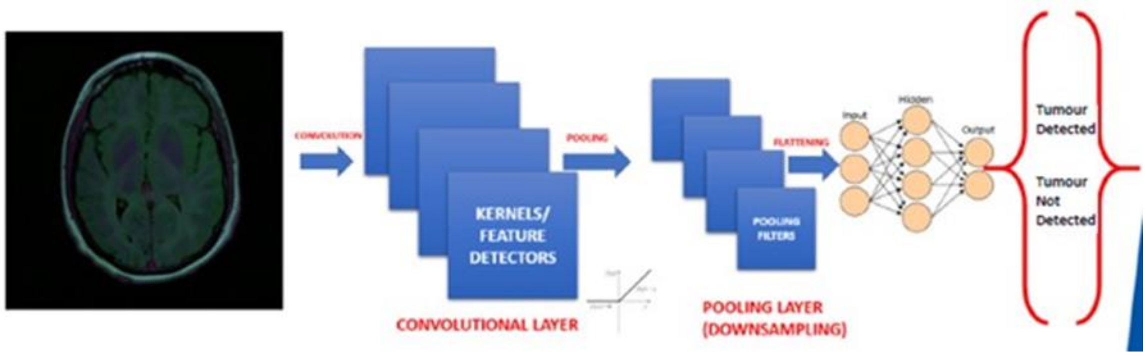


Figure 4.2 Layered deep learning pipeline to perform segmentation

4.2.2 Working Process

- The first CNN layers are used to extract high level general features as showed in figure 4.1,4.2 . The last couple of layers are used to perform classification (on a specific task) Local respective fields scan the image first searching for simple shapes such as edges/lines. These edges are then picked up by the subsequent layer to form more complex features.
- As CNNs grow deeper, vanishing gradient tend to occur which negatively impact network performance. Vanishing gradient problem occurs when the gradient is backpropagated to earlier layers which results in a very small gradient.
- Residual Neural Network includes “skip connection” feature which enables training of 152 layers without vanishing gradient issues. Resnet works by adding “identity mappings” on top of the CNN. ImageNet contains 11 million images and 11,000 categories. ImageNet is used to train ResNet deep network.
- Transfer learning is a machine learning technique in which a network that has been trained to perform a specific task is being reused (repurposed) as a starting point for another similar task. Transfer learning is widely used since starting from a pretrained models can dramatically reduce the computational time required if training is performed from scratch.

4.3 Transfer Learning Training Strategies:

Strategy #1 Steps:

- As showed in figure 4.3 freeze the trained CNN network weights from the first layers.
- Only train the newly added dense layers (with randomly initialized weights).

Strategy #2 Steps:

- Initialize the CNN network with the pre-trained weights
- Retrain the entire CNN network while setting the learning rate to be very small, this is critical to ensure that you do not aggressively change the trained weights

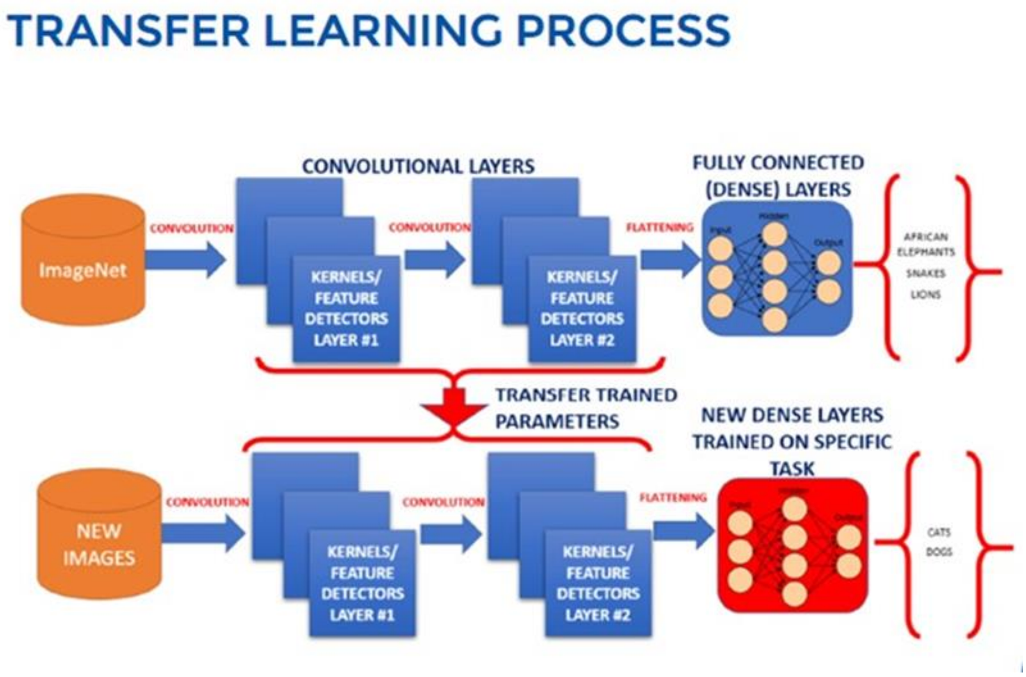


Figure 4.3 Transfer learning process

4.4 ResUNet segmentation, Architecture

- The goal of image segmentation is to understand and extract information from images at the pixel-level.
- Image Segmentation can be used for object recognition and localization which offers tremendous value in many applications such as medical imaging and self-driving cars etc.
- The goal of image segmentation is to train a neural network to produce pixel-wise mask of the image.
- Modern image segmentation techniques are based on deep learning approach which makes use of common architectures such as CNN, FCNs (Fully Convolution Networks) and Deep Encoders-Decoders.
- ResUNet architecture combines UNet backbone architecture with residual blocks to overcome the vanishing gradients problems present in deep architectures.
- Unet architecture is based on Fully Convolutional Networks and modified in a way that it performs well on segmentation tasks.
- Resunet consists of three parts as showed in figure 4.4 and 4.5:
    1 Encoder or contracting path.
    2 Bottleneck.
    3 Decoder or expansive path.

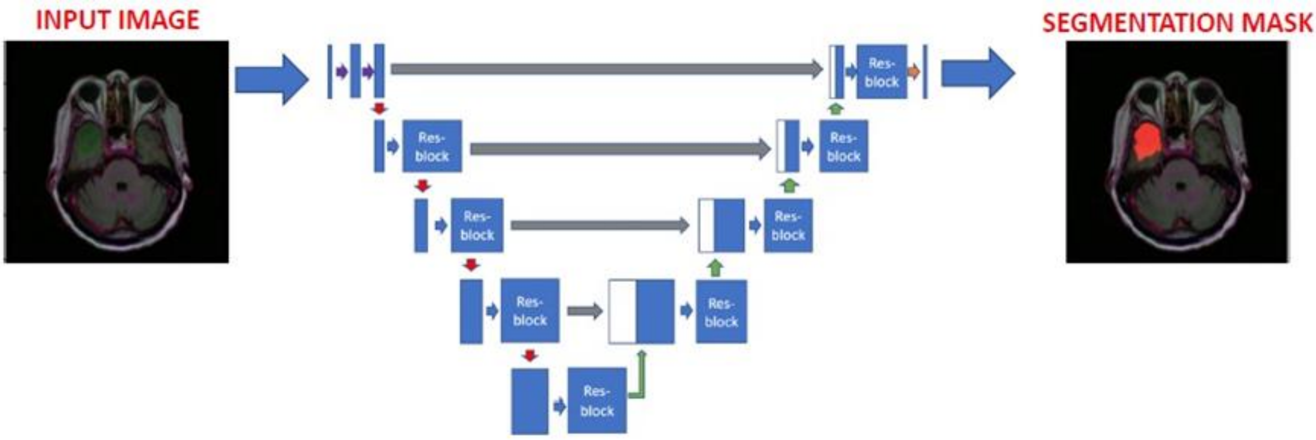


Figure 4.4 ResUNet Segmentation

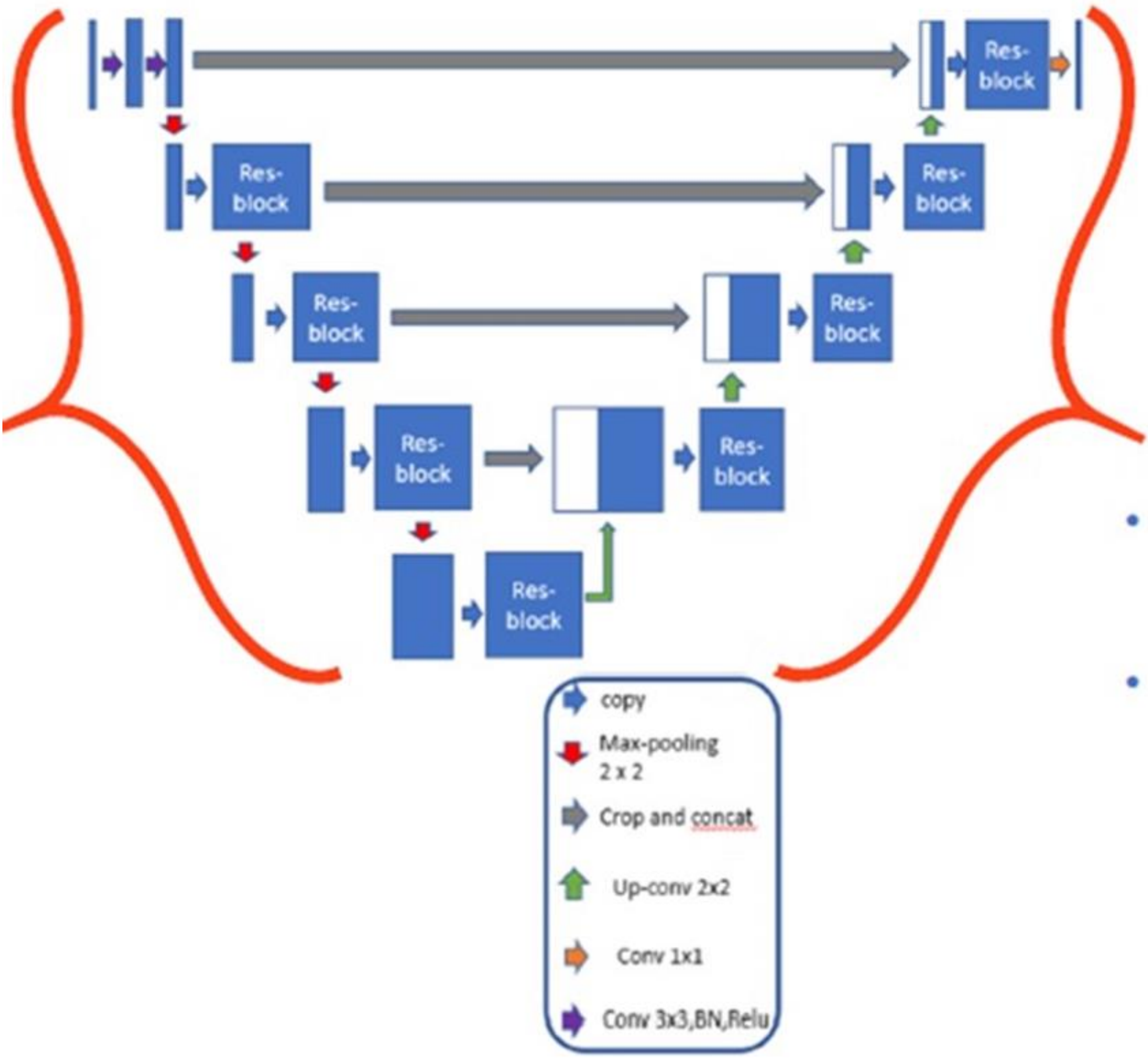


Figure 4.5 ResUNet Architecture

CHAPTER 5

METHODOLOGY

5.1 Dataset

- Public brain MRI dataset containing 3929 scans and manual tumour masks, 70% used for training, 15% for validation, 15% for testing.

5.2 Pre-Trained Models

- ResNet50: Pre-trained model used for feature extraction and classification to detect tumour presence. Additional dense layers added and trained on top.
- ResUNet: Custom model built for tumour segmentation to localize and segment the tumour. Uses encoder-decoder architecture with residual blocks.

i. **Tumour Detection:**
   - Load and freeze weights of pre-trained ResNet50 model.
   - Extract features from ResNet and pass to classifier model
   - Classifier model has dense layers, dropout, ReLU activations to classify tumour presence.

ii. **Training:**
   - Adam optimizer, Batch normalization for regularization.
   - Evaluation Metrics: Accuracy, Precision, Recall, F1-Score.

5.3 Technologies used:

1.Python

- Python is a programming language that may be used in many ways. Many of its features support functional programming and aspect-oriented programming, as well as object-oriented programming and structured programming. Many more paradigms, such as design by contract and logic programming, are supported by extensions. Python manages memory through dynamic typing and a combination of reference counting and a cycle-detecting garbage collector. It also includes late binding (dynamic name resolution), which binds method and variable names during program execution.

2.HTML

- HTML, or Hypertext Markup Language, is the standard markup language used to create and structure content on the World Wide Web. Developed by Tim Berners-Lee in the early 1990s, HTML serves as the backbone of web pages, defining the elements and structure that browsers use to render text, images, links, and multimedia. It consists of various tags; each serving a specific purpose to format and organize content. HTML documents are composed of a hierarchy of nested elements, creating the structure of a webpage. Over time, HTML has evolved with new versions, and its latest iteration, HTML5, introduced enhanced features for multimedia integration, improved semantics, and better support for mobile devices. Understanding HTML is fundamental for web development, as it provides the foundation for creating visually appealing and interactive websites across the internet.

3.Flask

- Flask is a lightweight web framework for Python, designed to simplify web development. It provides the essentials for building web applications, such as routing, handling HTTP requests, and rendering HTML templates. Flask follows a micro-framework philosophy, allowing developers to choose components based on their needs. With minimal boilerplate code, Flask makes it easy to create web applications quickly.

4.Machine Learning

- Machine Learning (ML) is a branch of artificial intelligence (AI) that empowers systems to learn and improve from experience without being explicitly programmed. It revolves around the development of algorithms and models that enable computers to analyse data, recognize patterns, and make intelligent decisions. There are various types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models are trained on labelled data, while unsupervised learning involves finding patterns in unlabelled data. Reinforcement learning focuses on training models to make sequences of decisions to maximize rewards.

5.Artificial Intelligence

- Artificial Intelligence (AI) is a rapidly advancing field of computer science that aims to create machines capable of intelligent behaviour. It involves the development of algorithms and models that enable machines to learn from data, adapt to new information, and perform tasks traditionally requiring human intelligence. AI encompasses various subfields, including machine learning, natural language processing, computer vision, and robotics. Machine learning algorithms, a core component of AI, enable systems to improve their performance on tasks through experience.

6.Computer Vision

- Computer vision is a field of artificial intelligence and computer science that focuses on enabling computers to interpret and understand visual information from the real world.

CHAPTER 6
DESIGN AND IMPLEMENTATION

6.1 Requirements

6.1.1 Overall Description

Proposes a technique for brain tumour detection and segmentation using deep learning on MRI images. Uses transfer learning with ResNet for classification and to detect tumour presence, and ResUNet for segmentation to localize the tumour.

6.1.2 Software Requirements

Programming Languages:

- Python: A widely used language for AI development.
- Libraries and Frameworks: TensorFlow or PyTorch for deep learning, scikit learn for machine learning, and others.

Environment Tools:

- Jupyter Notebooks: For interactive development and data exploration.
- IDEs (Integrated Development Environments): PyCharm, Visual Studio Code, or others.

Image Processing Libraries:

- OpenCV: For image preprocessing and manipulation.
- SimpleITK or similar: For medical image processing

Deep Learning Libraries:

- TensorFlow or PyTorch : Widely used deep learning
- Keras : A high level neural networks API.

Data Visualization:

- Matplotlib or Seaborn: For creating visualizations of data.
- Plotly : Interactive plotting library

Web Development:

- Flask or Django or HTML

6.1.3 Hardware Requirements

GPU (Graphics Processing Unit):

- High performance GPUs are essential for training deep learning models efficiently. NVIDIA GPUs, such as GeForce or Quadro series, are commonly used

CPU (Central Processing Unit):

- A powerful multicore CPU for general purpose computing tasks.

RAM (Random Access Memory):

- Adequate RAM to handle large datasets and model training. At least 16GB is recommended, but more may be needed for larger datasets.

Storage:

- SSD (Solid State Drive) for faster data access, especially during training. Sufficient storage capacity for datasets and model checkpoints.

6.2 Execution

6.2.1 Project Flow Chart

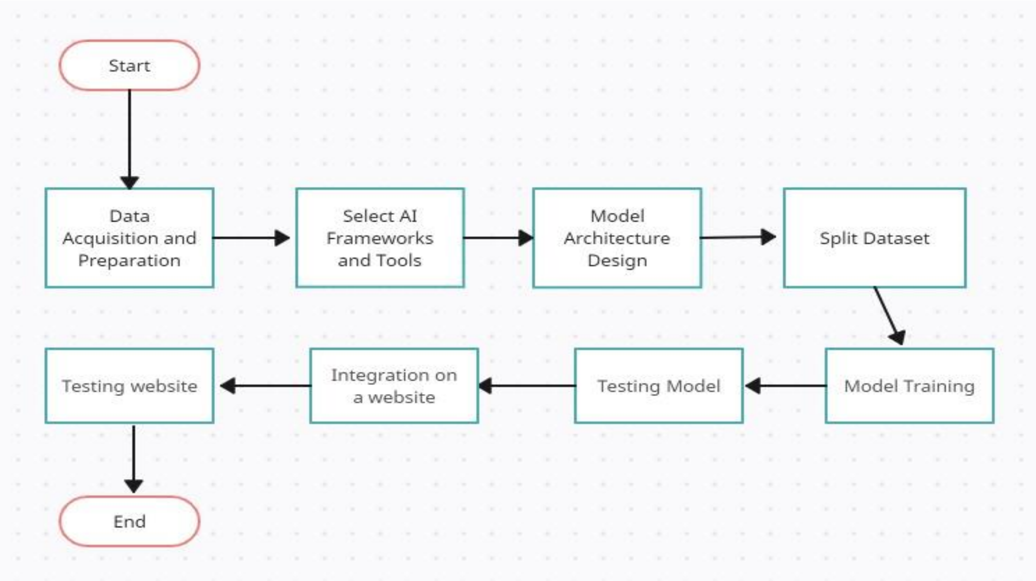


Figure 6.1: Project Flow Chart

6.2.2 UML Diagrams

The Use Case Diagram for the brain tumour detection system is shown in figure 6.2

- Actors: AI/ML Consultant and System.
- Use Cases: Analyze MRI Scans, Train Model, Test Model, Report Results.
- Relationships: Consultant involved in all steps; Train Model precedes Test Model.

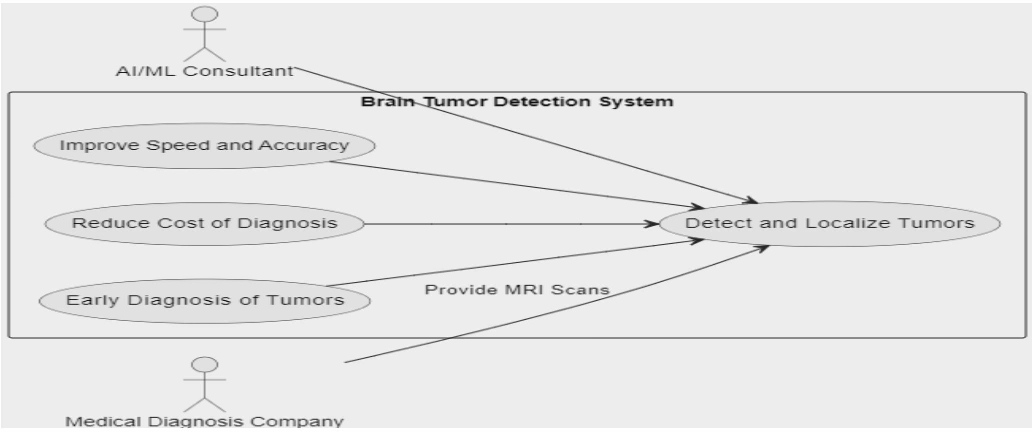


Figure 6.2: Use Case Diagram

The Class Diagram for the brain tumour detection system is shown in figure 6.3

- The class diagram outlines a brain tumor detection system with classes like BrainTumorDetectionSystem, Image, Pixel, TumorLocation, and TumorRegion.
- It enables tumour detection and localization using input images. Key interactions include BrainTumorDetectionSystem utilizing Image for processing and TumorLocation for reporting.
- The Image class manages image loading and display, while Pixel objects represent individual pixels. TumorRegion calculates tumour area based on vertex points.

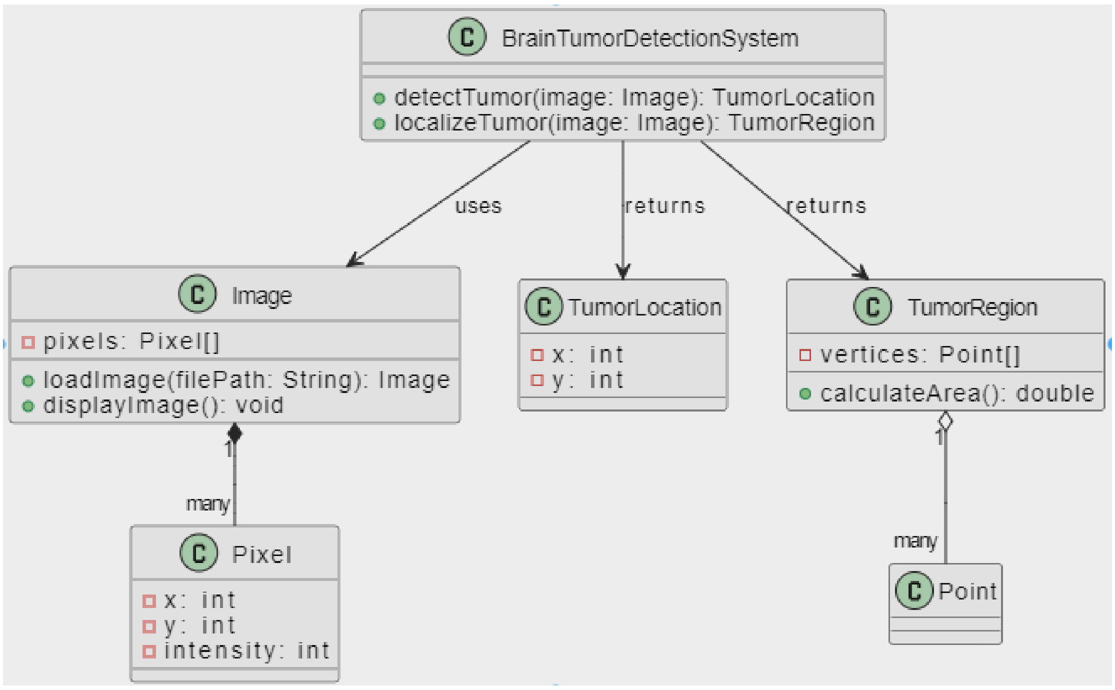


Figure 6.3: Class Diagram

The Sequence Diagram for brain tumour detection system is shown in figure 6.4

It demonstrates tumour detection and localization:

- User requests tumour detection.
- BrainTumorDetectionSystem loads MRI image.
- System detects and localizes tumour.
- System retrieves tumour location and region.
- Detected tumour information communicated to User.

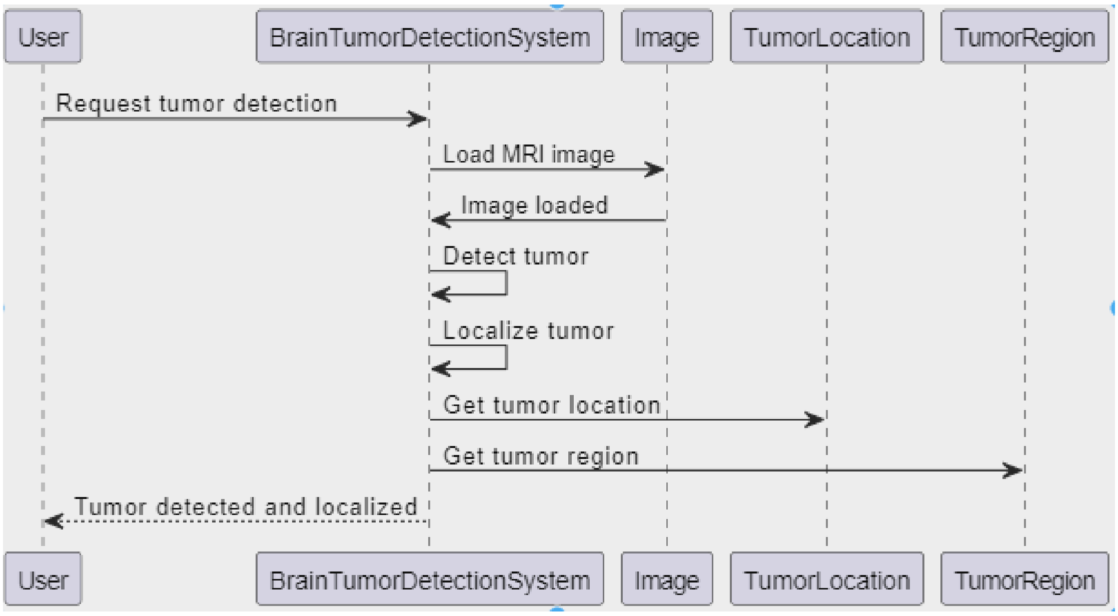


Figure 6.4: Sequence Diagram

The Activity Diagram for the brain tumour detection system is shown in figure 6.5

The activity diagram outlines the process of brain tumor detection:

- The detection process begins.
- An MRI image is loaded for analysis.
- If a tumor is detected:
- The system localizes the tumor.
- It identifies the tumor region.
- Tumor information is provided.
- If no tumor is detected, the user is notified accordingly.

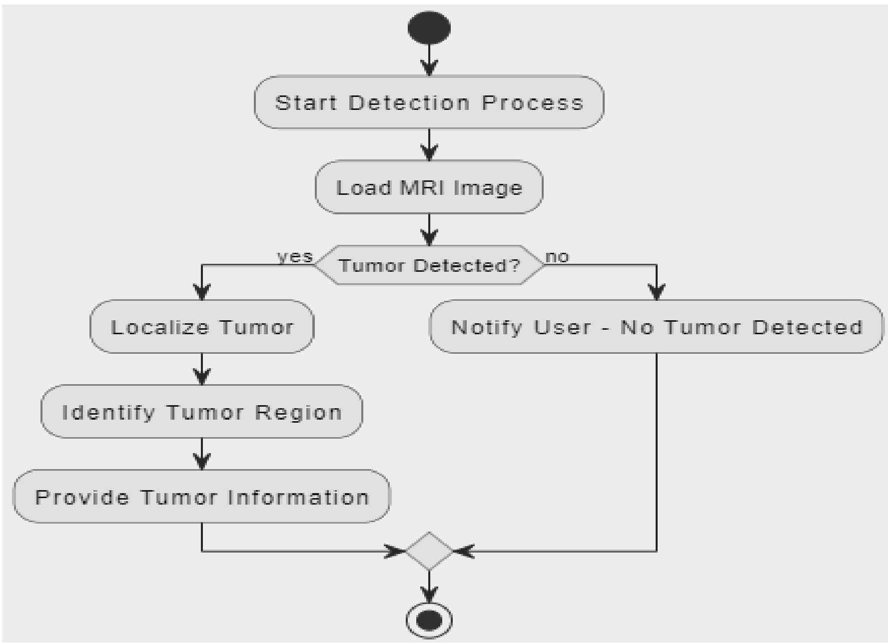


Figure 6.5: Activity Diagram

6.3 Code Snippets

Importing all the necessary libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import zipfile
import cv2
from skimage import io
import tensorflow as tf
from tensorflow.python.keras import Sequential
from tensorflow.keras import layers, optimizers
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras.utils import plot_model
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint, LearningRateScheduler
from IPython.display import display
from tensorflow.keras import backend as k
from sklearn.preprocessing import StandardScaler, normalize
import os
import glob
import random
from google.colab import files
%matplotlib inline
```

Figure 6.6: Importing all the libraries

Creating a data frame

```
[ ]  brain_df = pd.read_csv('data_mask.csv')
```

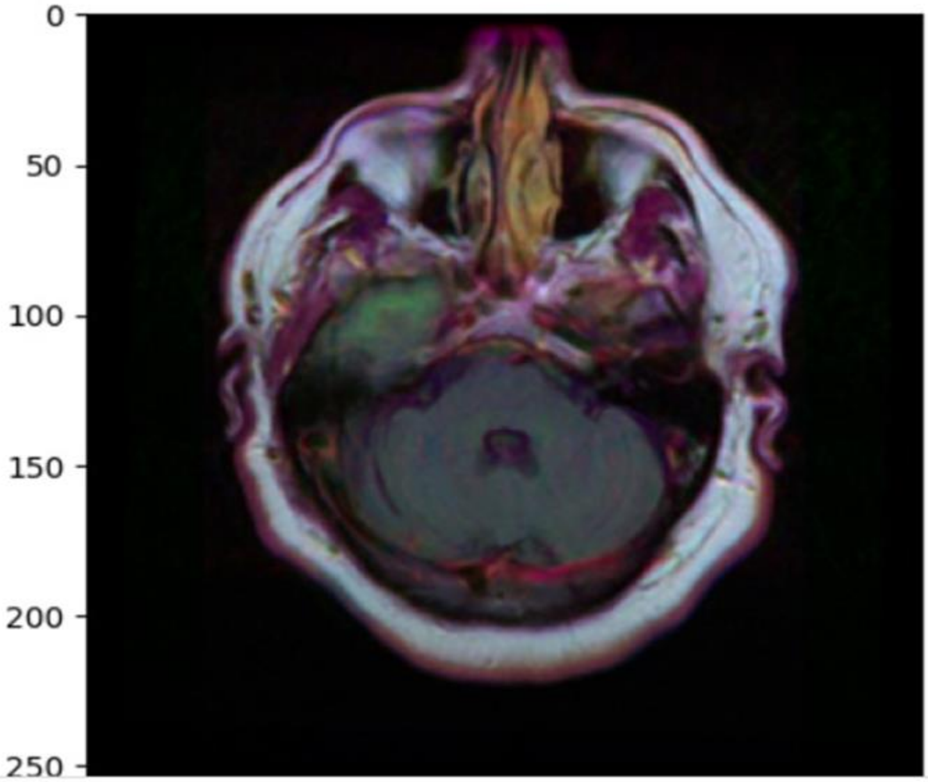
```
brain_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3929 entries, 0 to 3928
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   patient_id  3929 non-null   object
 1   image_path  3929 non-null   object
 2   mask_path   3929 non-null   object
 3   mask        3929 non-null   int64
dtypes: int64(1), object(3)
memory usage: 122.9+ KB
```

Figure 6.7: Creating data frame

```python
#ploting brain MRI using plt
plt.imshow(cv2.imread(brain_df.image_path[623]))
```

```
<matplotlib.image.AxesImage at 0x7d354da97910>
```




```python
# Visualizing the iamges (MRI and Mask) in the dataset seperately
import random
fig, axs = plt.subplots(6,2, figsize =(16,32))
count = 0
for x in range(6):
  i = random.randint(0, len(brain_df))
  axs[count][0].title.set_text("Brain MRI")
  axs[count][0].imshow(cv2.imread(brain_df.image_path[i]))
  axs[count][1].title.set_text("Mask - " + str(brain_df['mask'][i]))
  axs[count][1].imshow(cv2.imread(brain_df.mask_path[i]))
  count += 1
fig.tight_layout()
```

```python
#Visualizing MRI scan images for only sick patients followed by corresponding mask and followed by both MRI and corresponding mask in red color on top of each other
count = 0
fig, axs = plt.subplots(12,3, figsize = (20, 50))
for i in range(len(brain_df)):
  if brain_df['mask'][i] == 1 and count < 12:
    img = io.imread(brain_df.image_path[i])
    axs[count][0].title.set_text("Brain MRI")
    axs[count][0].imshow(img)

    mask = io.imread(brain_df.mask_path[i])
    axs[count][1].title.set_text("Mask")
    axs[count][1].imshow(mask, cmap = 'gray')

    img[mask == 255] = (255 ,0 ,0)
    axs[count][2].title.set_text("MRI with Mask")
    axs[count][2].imshow(img)
    count += 1
fig.tight_layout()
```

Fig 6.8: Visualising the images in the data set

```
# Converting the data in mask column to string format to use categorical mode in flow from data frame
brain_df_train['mask'] = brain_df_train['mask'].apply(lambda x : str(x))
brain_df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3929 entries, 0 to 3928
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   image_path  3929 non-null   object
 1   mask_path   3929 non-null   object
 2   mask        3929 non-null   object
dtypes: object(3)
memory usage: 92.2+ KB
```

```
# Splitting the data into train and test set
from sklearn.model_selection import train_test_split
train, test = train_test_split(brain_df_train, test_size = 0.15)  # 85%
```

```
# creating a image generator
!pip install keras-preprocessing
```

```
from keras_preprocessing.image import ImageDataGenerator
# Creating a data generator which scales the data from 0 to 1 and makes validation split of 0.15
datagen = ImageDataGenerator(rescale = 1./255., validation_split = 0.15)
```

```
# Getting ResNet50 base model
basemodel = ResNet50(weights = 'imagenet', include_top = False, input_tensor= Input(shape = (256, 256, 3)))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 0s 0us/step
```

```
# freezing the model weights
for layer in basemodel.layers:
  layers.trainable = False
```

```
# Adding classification head to the base model (adding a personal touch to the ResNet)
headmodel = basemodel.output
headmodel = AveragePooling2D(pool_size = (4,4))(headmodel)
headmodel = Flatten(name = 'flatten')(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(2, activation = 'softmax')(headmodel)
model = Model(inputs = basemodel.input, outputs = headmodel)
```

Fig 6.9: Adding classification head

```
[ ] # freezing the model weights
    for layer in basemodel.layers:
      layers.trainable = False
```

```
# Adding classification head to the base model (adding a personal touch to the ResNet)
headmodel = basemodel.output
headmodel = AveragePooling2D(pool_size = (4,4))(headmodel)
headmodel = Flatten(name = 'flatten')(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(2, activation = 'softmax')(headmodel)
model = Model(inputs = basemodel.input, outputs = headmodel)
```

Assessing Trained Model Performance

```
[ ] with open('resnet-50-MRI.json', 'r') as json_file:
      json_savedModel = json_file.read()
    model = tf.keras.models.model_from_json(json_savedModel)
    model.load_weights('weights.hdf5')
    model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics =["accuracy"])
```

```
# Making prediction
test_predict = model.predict(test_generator, steps = test_generator.n // 16, verbose = 1)
```

```
36/36 [==============================] - 141s 4s/step
```

```
[ ] test_predict.shape
```

```
(576, 2)
```

```
[ ] # Obtaining the accuracy of the model
    from sklearn.metrics import accuracy_score
    accuracy = accuracy_score(original, predict)
    accuracy
```

```
0.9826388888888888
```

```
from sklearn.metrics import classification_report

report = classification_report(original, predict, labels = [0,1])
print(report)
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       380
           1       0.99      0.95      0.97       196

   micro avg       0.98      0.98      0.98       576
   macro avg       0.99      0.98      0.98       576
weighted avg       0.98      0.98      0.98       576
```

Fig 6.10: Obtaining accuracy, And classification report

Building a segmentation model to localize tumor

```
[ ] # Get the dataframe containing MRIs which have masks associated
    brain_df_mask = brain_df[brain_df['mask'] == 1]
    brain_df_mask.shape

    (1373, 4)
```

```
# split the data into train and test data
from sklearn.model_selection import train_test_split
X_train, X_val = train_test_split(brain_df_mask, test_size = 0.15)
X_test, X_val = train_test_split(X_val, test_size = 0.5)
```

```
[ ] # create separate list for imageId, classId to pass into the generator
    train_ids = list(X_train.image_path)
    train_mask = list(X_train.mask_path)
    val_ids = list(X_val.image_path)
    val_mask = list(X_val.mask_path)
```

Assessing trained segmentation resunet model performance

```
from utilities import focal_tversky, tversky_loss, tversky
with open('ResUNet-MRI.json', 'r') as json_file:
  json_savedModel = json_file.read()
# loading the model architecture
model_seg = tf.keras.models.model_from_json(json_savedModel)
model_seg.load_weights('weights_seg.hdf5')
adam = tf.keras.optimizers.Adam(learning_rate = 0.05, epsilon = 0.1)
model_seg.compile(optimizer= adam, loss = focal_tversky, metrics= [tversky])
```

Fig 6.11: Assessing trained segmentation model performance

CHAPTER 7
RESULT ANALYSIS

The brain tumour detection and segmentation system achieved an impressive accuracy, demonstrating its effectiveness in accurately identifying tumours in MRI scans. Leveraging advanced deep learning architectures like ResNet and ResUNet++, the system excelled in semantic segmentation, providing precise tumour localization. The use of transfer learning with pre-trained models contributed to the system's efficiency.

The dataset, consisting of 3929 brain MRI scans with tumour annotations, enhanced the model's credibility. While the results are promising, there's potential for improvement through larger datasets, hyperparameter tuning, and additional data augmentation. Addressing misclassifications, particularly reducing false negatives, is an identified area for future refinement. Model take execution time on our device approximately (25 min) and may vary on other device's, our model is trained on 75% of the training set and 25% of test set to check the performance and accuracy of the model.

In summary our model as the following performance scores:

Table 7.1

Accuracy = 98%	Precision = (0 = 98%, 1 = 99%)
Recall = (0 = 1.00, 1 = 0.95)	Fi Score = (0 = 99%, 1 = 97%)

Here are some of the graphical representation of our results**:**

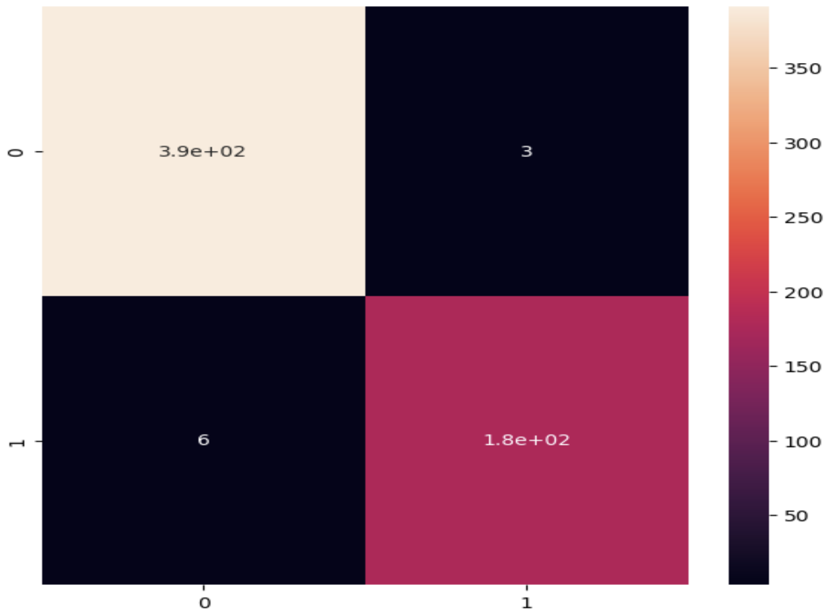


Figure 7.1: Confusion matrix

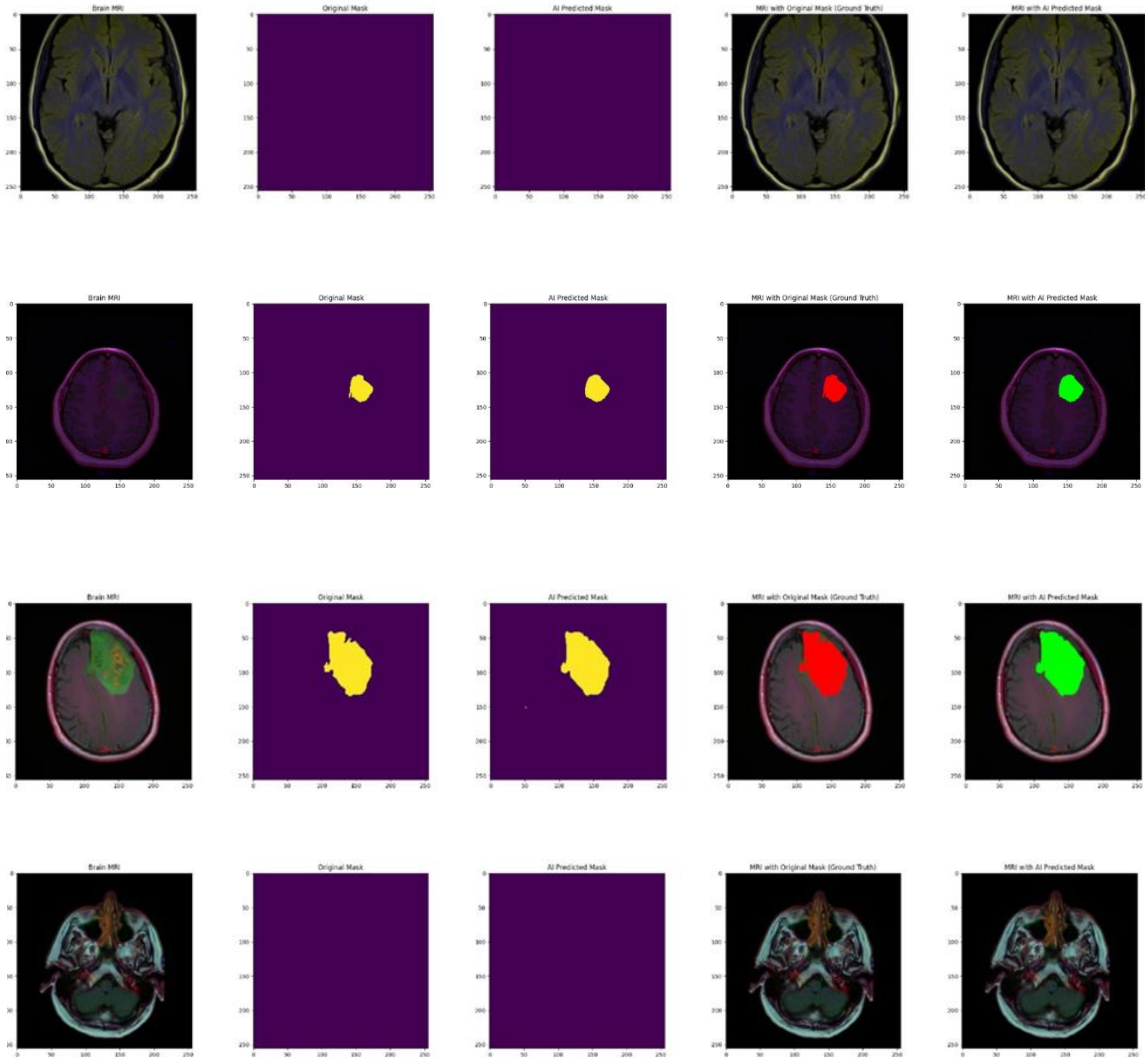


Figure 7.2: Brain tumour detection and segmentation

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

In conclusion, the developed brain tumour detection and segmentation system employing advanced deep learning techniques such as ResNet and ResUNet++ has proven to be a valuable asset in the medical field. Achieving a commendable accuracy of 98.1%, the system demonstrates its capability to accurately identify and localize brain tumours in MRI scans. The utilization of transfer learning and a well-curated dataset further enhances the system's performance and reliability.

While the system has shown promising results, there are avenues for improvement, such as expanding the dataset, refining hyperparameters, and addressing specific misclassification challenges. The future direction involves continuous refinement and adaptation to emerging technologies and methodologies in the ever-evolving landscape of medical imaging and artificial intelligence.

8.2 Future Enhancements

To further enhance the efficacy and scope of the brain tumour detection and segmentation system, several avenues for improvement and expansion can be considered:

- **Dataset Augmentation**: Enriching the dataset with a more extensive and diverse set of MRI scans can contribute to better model generalization and improved performance across various patient demographics.
- **Implementation of Advanced Architectures**: Exploring and implementing state-of-the-art deep learning architectures tailored for medical image analysis could lead to further improvements. Staying abreast of advancements in the field ensures the system's competitiveness.
- **Need of vast dataset:** The need for a vast dataset is crucial in various fields, especially in the context of machine learning and data-driven technologies.
- **Need of classification with in brain tumours**: The need for classification within brain tumours is crucial for several reasons, and it plays a significant role in the field of medical imaging, diagnosis, and treatment planning, can make a machine learning model which can detect and localize brain tumour with the classification of brain tumours.

REFERENCES

[1] Ismael, M. R. and Abdel-Qader, I. (2018). Brain tumor classification via statistical features and back propagation neural network, 2018 IEEE International Conference on Electro/Information Technology (EIT), pp. 0252–0257.

[2] Vinayak K. Bairagi, Pratima Purushottam Gumaste, Seema H. Rajput & Chethan K. S. Automatic brain tumour detection using CNN transfer learning approach (2023)

[3] Arbane, M., Benlamri, R., Brik, Y. and Djerioui, M (2020). Transfer learning for automatic brain tumour classification using mri images, 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), pp.

[4] J. Vijay; J. Subhashini An efficient brain tumour detection methodology using K-means clustering algoriftnn (2013)

[5] Brain Tumour Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches by Ayesha Younis, Li Qiang, Charles Okanda Nyatega, Mohammed Jajere Adamu. [2022]

[6] Brain Tumour Detection Using Deep Learning Models October 2020 DOI:10.1109/INDISCON50162.2020.00037 Conference: 2020 IEEE India Council International Subsections Conference (INDISCON) Authors: Sneha Grampurohit Universität Trier

[7] ResUNet++: An Advanced Architecture for Medical Image Segmentation Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Dag Johansen, Thomas de Lange, Pal Halvorsen, Havard D. Johansen [2019]

[8] Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization Author Chengping Rao, Yang Liu [2020]

[9] Brain Tumour Segmentation Using Convolutional Neural Networks in MRI Images Sergio Pereira, Adriano Pinto, Victor Alves, Carlos A Silva [2016]

[10] Nazir, M., Shakil, S. and Khurshid, K. Role of deep learning in brain tumor detection and classification (2015 to 2020): A review, Computerized Medical Imaging and Graphics 91: 101940.

[11] A Two-Tier F wo-Tier Framework Based on GoogLeNet and Y amework Based on GoogLeNet and YOLOv3 Models Ov3 Models for Tumour Detection in MRI Farman Ali Sejong University Sadia Khan the University of Agriculture, Peshawar Arbab Waseem Abbas The University of Agriculture, Peshawar Babar Shah Zayed University Tariq Hussain The University of Agriculture, Peshawar. [2022]