



**Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department**

Brain Tumors Classification and Segmentation

**This documentation submitted as required for
the degree of bachelors in
Mainstream Program
Computer and Information Sciences
Ain Shams University**

**By:
Ehab Mahmoud
Bassel Islam
Islam Mohamed
Eiad Amr
Ahmed Yehia
Abd Alrahman Emad**

Under Supervision of:

**Dr. Sally Saad
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.**

**T.A. Mohammad Essam
Bioinformatics Department,
Faculty of Computer and Information Sciences,
Ain Shams University.**

Acknowledgments

All praise and thanks to ALLAH, who provided me with the ability to complete this work. I hope to accept this work from me.

I am grateful to *my parents* and *my family* who are always providing help and support throughout the whole years of study. I hope I can give that back to them.

I also offer my sincerest gratitude to my supervisors, *Dr. Sally Saad, and T.A. Mohammad Essam*. who have supported me throughout my thesis with their patience, knowledge and experience.

Finally, I would like to thank my friends and all the people who gave me support and encouragement.

Abstract

Brain tumors are among the most challenging medical conditions to diagnose and treat, often requiring accurate classification and precise identification of the tumor's size and location. This project addresses this challenge by leveraging deep learning techniques to classify brain MRI images into four categories: glioma, meningioma, pituitary tumor, and no tumor. Additionally, the project involves tumor segmentation to estimate the tumor's size and 3D reconstruction of brain images to aid in further analysis.

The proposed solution utilizes convolutional neural networks (CNNs) for classification and a U-Net architecture for segmentation of the tumor region. For segmentation, mask images highlighting the tumor areas are used alongside MRI images. Post-processing techniques are applied to calculate the tumor's volume based on the segmented area. Furthermore, the project explores 3D reconstruction techniques to provide a comprehensive view of the tumor in a three-dimensional context, aiding in surgical planning and treatment assessment.

The results show promising accuracy in classification and segmentation, offering a reliable method for tumor detection and size estimation. The 3D reconstruction provides a detailed visualization of the tumor, enhancing the diagnostic process. This approach demonstrates the potential for AI-driven solutions in the medical field, improving early diagnosis and supporting clinical decisions.

Conclusively, the model developed in this project contributes to the growing field of medical image analysis, offering a potential tool for automatic brain tumor classification, segmentation, and 3D reconstruction. It is designed for future deployment in mobile applications, making advanced tumor analysis more accessible to healthcare professionals.

تعد أورام الدماغ من بين أكثر الحالات الطبية تحديًا في التشخيص والعلاج، وغالبًا ما تتطلب تصنيفًا دقيقًا وتحديدًا دقيقًا لحجم الورم وموقعه. يتناول هذا المشروع هذا التحدي باستخدام تقنيات التعلم العميق لتصنيف صور الرنين المغناطيسي للدماغ إلى أربع فئات: الورم الدبقي، الورم السحائي، ورم الغدة النخامية، وعدم وجود ورم. بالإضافة إلى ذلك، يتضمن المشروع تقسيم الورم لتقدير حجمه وإعادة بناء ثلاثية الأبعاد لصور الدماغ للمساعدة في التحليل الإضافي.

لتقسيم U-Net للتصنيف، وهندسة CNN يستخدم الحل المقترح شبكات الالتفاف العصبية منطقة الورم. بالنسبة للتقسيم، يتم استخدام الصور القناعية التي تبرز مناطق الورم جنبًا إلى جنب مع صور الرنين المغناطيسي. يتم تطبيق تقنيات المعالجة اللاحقة لحساب حجم الورم بناءً على المنطقة المقسمة. علاوة على ذلك، يستكشف المشروع تقنيات إعادة البناء ثلاثي الأبعاد لتوفير رؤية شاملة للورم في سياق ثلاثي الأبعاد، مما يساعد في التخطيط الجراحي وتقييم العلاج.

أظهرت النتائج دقة واعدة في التصنيف والتقسيم، مما يوفر طريقة موثوقة لاكتشاف الأورام وتقسيم حجمها. يوفر إعادة البناء ثلاثي الأبعاد تصورًا مفصلاً للورم، مما يعزز عملية التشخيص. يوضح هذا النهج إمكانيات الحلول المدعومة بالذكاء الاصطناعي في المجال الطبي، مما يحسن التشخيص المبكر.

في الختام، يساهم النموذج المطور في هذا المشروع في مجال تحليل الصور الطبية المتنامي، حيث يقدم أداة محتملة للتصنيف التلقائي لأورام الدماغ، تقسيمها، وإعادة بنائها ثلاثيًا. وهو مصمم ليتم نشره مستقبلاً في تطبيقات الهواتف المحمولة، مما يجعل التحليل المتقدم للأورام أكثر سهولة للمختصين في الرعاية الصحية.

Table of Contents

Acknowledgement	i
Abstract.....	ii
List of Figures	vi
List of Tables	vi
List of Abbreviations	vii
1- Introduction	1
1.1 Motivation.....	1
1.2 Problem Definition.....	2
1.3 Objective	3
1.4 Time Plan.....	4
1.5 Document Organization.....	5
2- Background	6
2.1 Field Of The Project.....	6
2.2 Scientific Background	8
2.2.1 Neural Networks	8
2.2.2 Convolutional Neural Netwroks.....	12
2.3 Survey Of Work Done.....	15
2.4 Description of Any Technology Used	17
3- Analysis and Design.....	22
3.1 System Overview.....	22
3.1.1 System Architecture.....	22
3.1.2 System Users	23
3.2 Design Diagrams.....	24
3.2.1 Use Case Diagram	24
3.2.2 Sequence Diagram	25
4- Implementation and Testing.....	28
4.1 Dataset Description.....	28
4.1.1 Dataset For Brain Tumor Classification.....	28
4.1.2 Dataset For Brain Tumor Segmentation	29
4.2 Phases Description.....	31
4.3 Technologies	40

4.4 UI Design And Wireframes.....	44
4.5 Experimental Results.....	49
5- User Manual.....	51
6- Conclusion and Future Work	55
6.1 Conclusion.....	55
6.2 Future Work.....	56
References	57

List of Figures

Figure 1.1: Time plan Gantt chart.....	4
Figure 2.1: Neural Network	8
Figure 2.2: Convolutional Neural Networks:.....	12
Figure 2.3: U-Net Architecture.....	21
Figure 3.1: System Architecture.....	22
Figure 3.2: Use Case Diagram.....	24
Figure 3.3: Classification Sequence Diagram.....	25
Figure 3.4: Segmentation Sequence Diagram.....	26
Figure 3.5: 3D Reconstruction Sequence Diagram.....	27
Figure 4.1: Feature Extraction.....	32
Figure 4.2: Image Classification.....	34
Figure 4.3: Image Segmentation.....	36
Figure 4.4: 3D Brain.....	38

List of Tables

Table 2.1: Classification models related work.....	15
Table 2.2: Segmentation models related work.....	16
Table 4.1: Classification Dataset.....	29
Table 4.2: Segmentation Dataset.....	30
Table 4.3: Classification Results.....	49
Table 4.4: Segmentation Results.....	50

List of Abbreviations

ANNs: Artificial Neural Networks
CINR: Consistency Implicit Neural Representation
CNN: Convolutional Neural Network
DNNs: Deep Neural Networks
GPU: Graphics Processing Unit
HDLN: Hybrid Deep Learning Network
IOU: Intersection Over Union
MIOU: Mean Intersection Over Union
MRI: Magnetic Resonance Imaging
NERF: Neural Radiance Field
PA: Pixel Accuracy
RDGM: Radiation Diffusion Generation Model
SGDM: Stochastic Gradient Descent with Momentum
VDSG: Volume Diffusion Super-Resolution Generation

1- Introduction

1.1 Motivation

The motivation behind this project arises from the critical need for leveraging deep learning in neuro-oncology to enhance brain tumor diagnosis and treatment. Manual MRI analysis by radiologists is not only time-intensive but also prone to variability in interpretation, especially when identifying and segmenting tumors. By developing a neural network-based model that integrates both classification and segmentation, this project addresses these challenges. The model's ability to classify brain MRI images into tumor categories (glioma, meningioma, pituitary) and perform precise segmentation for tumor size estimation provides a robust tool for early diagnosis and treatment planning. Furthermore, the inclusion of 3D reconstruction brings a new dimension to visualization, allowing healthcare professionals to better understand the spatial relationships between the tumor and surrounding brain tissue. Deploying this system as a Flutter-based mobile application enhances its accessibility and usability, offering a real-time solution to streamline medical workflows and reduce diagnostic delays. This project not only pushes the boundaries of deep learning in medical imaging but also demonstrates the potential of AI in transforming healthcare.

1.2 Problem Definition

Brain tumors, whether benign or malignant, pose a critical challenge in healthcare due to their complexity, variability, and potential to threaten life. Accurate diagnosis and effective treatment rely heavily on the ability to identify tumor types, locations, and extents from medical imaging, particularly MRI scans. However, manual interpretation of these scans is time-intensive, prone to errors, and highly dependent on the expertise of radiologists, which can vary across regions and institutions. This often leads to delays in diagnosis, inconsistent outcomes, and increased pressure on medical professionals.

Our project aims to address this challenge by developing an AI-driven system for brain tumor classification, segmentation, and 3D reconstruction. The classification module will identify tumor types, while the segmentation module will delineate tumor regions with high precision. Additionally, the 3D reconstruction feature will provide an interactive, visual representation of tumors, enabling doctors to gain a deeper understanding of their structure and spread.

This solution will assist doctors in making quicker, more accurate diagnoses and help medical organizations standardize tumor analysis. By automating repetitive tasks and offering advanced visualizations, our project enhances efficiency and supports better decision-making, ultimately improving patient care and outcomes.

1.3 Objective

The project aims to develop a deep learning classification model capable of categorizing brain MRI images into four distinct classes: normal brain, glioma tumor, meningioma tumor, and pituitary tumor. In addition to classification, a tumor segmentation algorithm will be implemented to accurately identify and delineate tumor regions within the MRI images, enabling precise measurement of tumor size. This measurement will assist in clinical assessments and treatment planning. The project also involves reconstructing the brain from 2D MRI slices into a 3D model, providing a comprehensive visual representation of both the tumor and the surrounding brain structures. To make the solution accessible and practical, the entire model will be deployed as a Flutter-based mobile application, allowing for real-time classification, segmentation, and visualization of brain tumors directly on mobile devices. The system will be optimized to ensure high accuracy, fast performance, and the ability to handle new MRI images with reliable predictions and minimal latency.

1.4 Time Plan



Figure 1.1: Time Plan Gannt Chart

1.5 Document Organization

Chapter 2: Background

This chapter introduces brain tumor detection using classification, segmentation, and 3D reconstruction. It provides an overview of medical imaging, deep learning techniques, and previous research in the field.

Chapter 3: Analysis and Design

This chapter shows the system architecture and workflow. It explains how the system processes MRI scans for classification, segmentation, and 3D reconstruction.

Chapter 4: Implementation and Testing

This chapter explains how the system was implemented, which datasets were used, and how the system was tested. It also summarizes the results and evaluation metrics.

Chapter 5: User Manual

This chapter explains how to use the application, including how to upload MRI scans, run the system, and view classification, segmentation, and 3D reconstruction output.

Chapter 6: Conclusion and Future Work

This chapter highlights the project results, challenges faced, and possible future improvements.

2- Background

2.1 Field of the Project

This project operates at the intersection of Medical Imaging Analysis and Artificial Intelligence (AI) in healthcare, specifically within the specialized domains of Neuroradiology and Computational Neuroscience. It leverages advanced computational techniques to address critical challenges in brain tumor diagnosis and management.

Medical Imaging Analysis forms the cornerstone of modern diagnostics. Techniques like Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and Positron Emission Tomography (PET) provide invaluable visual insights into the human body. However, interpreting these complex images, particularly for intricate conditions like brain tumors, presents significant challenges. Manual analysis is time-consuming, prone to inter-observer variability, and can miss subtle details crucial for accurate diagnosis and treatment planning. The sheer volume and complexity of medical image data necessitate more efficient and reliable analytical methods.

This is where Artificial Intelligence in Healthcare steps in, revolutionizing how medical data is processed and understood. AI, especially through Machine Learning and Deep Learning, offers powerful tools to automate and enhance various healthcare processes. In the context of medical imaging, AI algorithms can identify patterns, detect anomalies, and make predictions with a precision that often rivals, or even surpasses, human capabilities. For brain tumor cases, AI can significantly assist in accurate diagnosis, optimize treatment strategies, and improve overall patient management by providing rapid, consistent, and data-driven insights.

The project specifically delves into Neuroradiology, focused on diagnosing diseases of the brain using advanced imaging techniques. In this field, the accuracy and timeliness of diagnosis are paramount. Brain tumors demand precise localization and characterization to inform critical decisions regarding surgical intervention, radiation therapy, and chemotherapy. Early and accurate identification directly impacts patient prognosis and quality of life.

Finally, the project draws upon principles from Computational Neuroscience, a field that applies computational methods to understand the brain's functions and pathologies. By employing sophisticated algorithms to analyze brain images, this project aims to computationally model and interpret the complexities of brain tumors, aligning with the broader goal of using computational approaches to gain deeper insights into neurological conditions.

2.2 Scientific background

2.2.1: Neural networks

Neural networks are a class of machine learning models inspired by the structure and functioning of the human brain. They are used to recognize patterns, classify data, and make predictions, and have become a core part of deep learning, enabling tasks like image and speech recognition, language translation, and game playing.

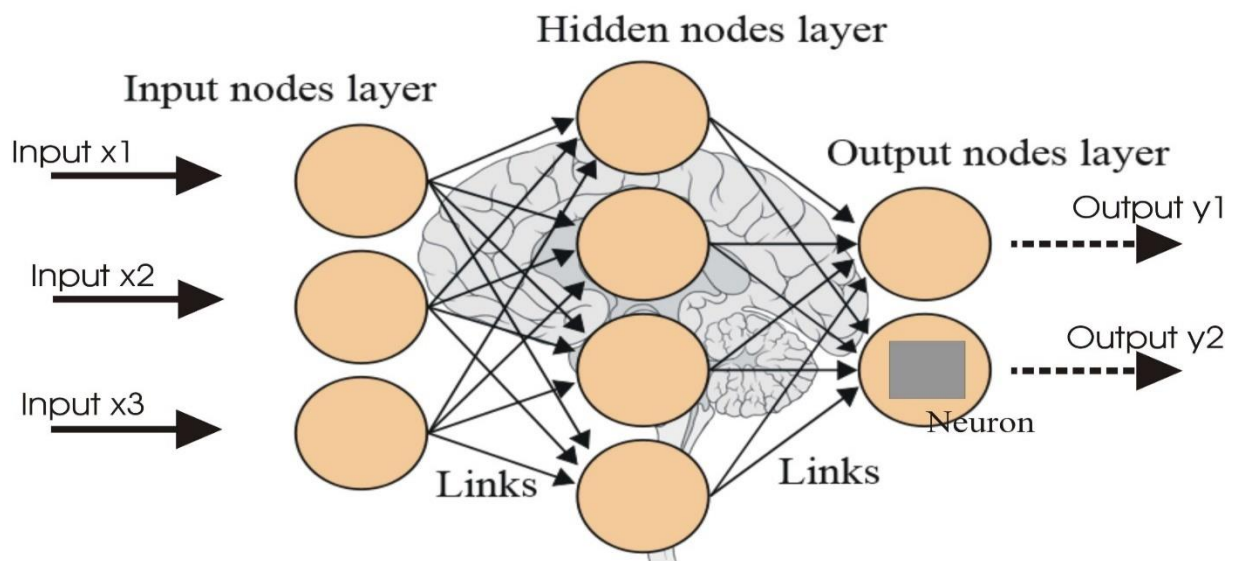


Figure 2.1: Neural Network

Key Concepts of Neural Networks

1. Neurons:

Neurons, also known as nodes, are the fundamental units of a neural network. A neural network consists of multiple layers formed by these interconnected neurons. Each neuron receives input, processes them, and passes the resulting output to the next layer. The typical operation of a neuron involves computing a weighted sum of its inputs, applying a non-linear activation function to the result, and forwarding the output to the subsequent layer.

2. Layers:

The input layer is the layer that receives the raw data, such as pixels of an image or words of a sentence. Hidden layers lie between the input and output layers and perform computations and transformations. Deep neural networks have multiple hidden layers, which allow the model to learn complex patterns. The output layer is the final layer that produces the network's output, such as classification labels or predicted values.

3. Weights and Biases:

Weights are parameters that control the strength of the connection between neurons. These are adjusted during training. Biases are additional parameters that allow the network to shift the activation function, enabling the model to learn patterns better.

4. Activation Function:

The activation function determines whether a neuron should be activated or not. It introduces non-linearity into the network, allowing it to learn complex patterns. Common activation functions include ReLU (Rectified Linear Unit), Sigmoid, and Tanh.

5. Forward Propagation:

In forward propagation, data flows from the input layer through the hidden layers to the output layer. Each neuron computes its output and passes it to the next layer.

6. Loss Function:

The loss function, or cost function, measures how far the network's predictions are from the actual target values. The goal during training is to minimize this loss.

7. Backpropagation:

Backpropagation is the process used to update the weights and biases of the network during training. It calculates the gradient (partial derivatives) of the loss function with respect to each weight and bias and uses this information to adjust the parameters through an optimization algorithm like gradient descent.

8. Optimization:

Optimization algorithms, such as Gradient Descent or its variants (e.g., Adam, RMSprop), adjust the weights and biases to minimize the loss function iteratively.

2.2.2: Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a specialized type of neural network designed to process and analyze visual data, such as images and videos. CNNs have become the backbone of modern computer vision tasks due to their ability to automatically learn spatial hierarchies and patterns in image data. They differ from traditional fully connected neural networks by using convolutional layers that apply filters (kernels) to local regions of an image, making them more efficient for image processing.

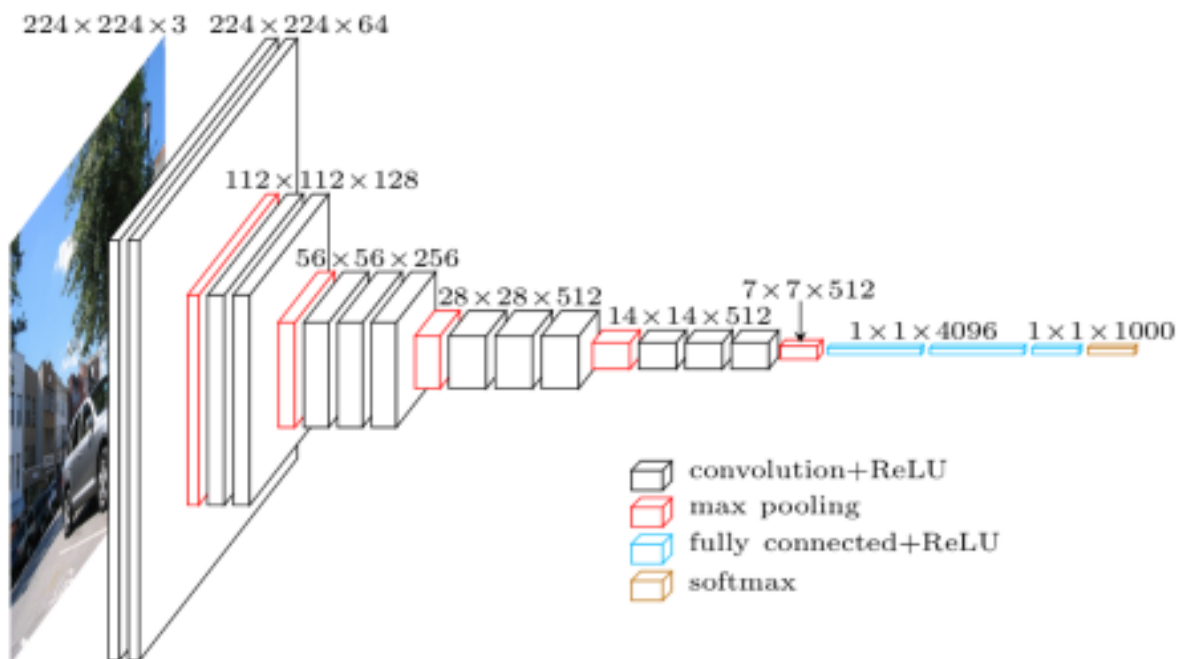


Figure 2.2: Convolutional Neural Networks

Key Components of CNNs:

1. Convolutional Layer:

The convolutional layer is the core building block of a CNN. It applies a series of filters (or kernels) to the input image, performing convolutions and mathematical operations that combine the filter with the image data. Each filter detects different features like edges, textures, and patterns. The result of applying the filters is a set of feature maps that represent the presence of these features in different parts of the image. Convolution reduces the spatial dimensions of the image while preserving important information.

2. Activation Function (ReLU):

After each convolution operation, an activation function (typically ReLU - Rectified Linear Unit) is applied to introduce non-linearity into the network. This helps the model learn complex patterns and decision boundaries. ReLU is a simple function that sets all negative values in the feature maps to zero and keeps positive values unchanged, making it computationally efficient.

3. Pooling Layer:

The pooling layer is used to downsample the feature maps and reduce their spatial dimensions, helping to lower computational cost and reduce overfitting. Max pooling is the most common pooling operation, where the maximum value from a local region (often 2x2 or 3x3) is selected. This retains the most important features while reducing the size of the representation. Pooling makes the network more invariant to small translations or distortions in the input image.

4. Fully Connected Layer:

After several convolutional and pooling layers, the network typically ends with one or more fully connected layers. These layers are similar to the layers in traditional neural networks, where each neuron is connected to every neuron in the previous layer. The purpose of the fully connected layer is to integrate the features learned by the convolutional layers and make final predictions, such as classifying the image into a specific category. The final output layer is typically a SoftMax or sigmoid activation, depending on the task (e.g., SoftMax for multi-class classification, sigmoid for binary classification).

5. Output Layer:

The output layer provides the final predictions. In a classification task, this layer typically uses a SoftMax activation function to output a probability distribution over the possible classes.

2.3 A survey of Work Done

Classification

<u>Title</u>	<u>Date</u>	<u>Data set</u>	<u>Models</u>	<u>Accuracy (%)</u>	<u>Precision (%)</u>	<u>Recall (%)</u>
A robust MRI-based brain tumor classification via a hybrid deep learning technique	2023	T1-weighted contrast enhanced brain MRI	1- AlexNet 2- Squeezenet 3- Nas net 4- Majority	1- 96.08 2- 97.17 3- 97.50 4- 99.31	1- 96.22 2- 96.44 3- 97.07 4- 99.92	1- 96.11 2- 97.41 3- 97.40 4- 98.30
Classification of MRI Brain Tumor Images using Deep Learning Segment Anything Model for segmentation and Deep Convolution Neural Network	2024	Brain Tumor MRI Kaggle Dataset	1- inceptionv3 2- DenseNet 3 - EfficientNetV2 B3 4- CoCa	1- 96.43 2- 94.83 3- 96.87 4- 97.60	1- 97.32 2- 98.21 3- 98.97 4- 99.04	1- 93.47 2- 94.56 3- 96.50 4- 97.26
Deep Learning in Medical Image Classification from MRI-based Brain Tumor Images	2024	Brain Tumor Detection 2020 dataset	1- EfficientNet-B0 2- VGG16 3- MobileNet-BT	1- 89.33 2- 94.97 3- 99.24	1- 89.61 2- 94.95 3- 99.24	1- 89.33 2- 94.97 3- 99.24
An l2 normalized spatial attention network for accurate and fast classification of brain tumors in 2d t1-weighted ce mri images	2023	2D T1-weighted CE-MRI dataset	1- CNN (Custom) 2- Baseline+CBAM 3- Proposed L2-SA	1- 95.27 2- 95.60 3- 96.57		
A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network	2021		1-CNN Transfer Learning	1- 93	1- 90	1- 87

Table 2.1: Classification models related work

Segmentation

<u>Title</u>	<u>Date</u>	<u>Dataset</u>	<u>Models</u>	<u>Mean IOU (%)</u>	<u>Mean Dice Score (%)</u>	<u>Dice Score(%)</u>
Tumor Segmentation for Brain Tumor Using Combination of Deep Learning and Machine Learning Algorithms.	2024	Image-Net	1-UNet - VGG16 2-UNet - VGG19	1-78.31 2-79.31	1- 95.21 2-96.21	
Medical Image Analysis for Tumor Segmentation Using U-Net Variants	2025	BraTS	1-Vanilla U Net 2-Attention U Net 3-Residual U Net	1-76 2-83 3-79		1-83 2-89 3-86
Improved Brain Tumor Segmentation in MR Images with a Modified U-Net	2024	Figshare Brain	1-deeplab v3-ResNet-18 2-deeplab ResNet-50 3-deep lab Mobile Net	1-83.4 2-81.4 3-84		1-87.8 2-86.7 3-88.2
Brain Tumor Segmentation from MRI Images using Deep Learning Techniques	2023	Figshare Brain	1-UNET (Densenet201 backbone). 2-Attention UNET (Dense net backbone). 3-Recurrent Residual UNET	1-85.07 2-85.53 3-86.65		1-82.88 2-83.49 3-84.95

Table 2.2: Segmentation models related work

2.4 Description of Any Technology Used

1. Programming Language:

Python: Chosen as the core programming language due to its extensive ecosystem of scientific computing and deep learning libraries, its ease of use, and strong community support within the machine learning and medical imaging fields.

2. Deep Learning Framework:

PyTorch: Served as the primary deep learning framework for building, training, and evaluating the neural network models, including Convolutional Neural Networks (CNNs) and U-Net variants. PyTorch was selected for its dynamic computation graph, which offers exceptional flexibility during model debugging, research, and rapid prototyping, alongside its intuitive API and active community.

3. Libraries for Data Handling and Image Processing:

NumPy: An essential Python library for efficient numerical operations and array manipulation, fundamental for handling and processing large multi-dimensional medical image data.

Albumentations: Utilized for fast and flexible image augmentations. This library provided a wide range of transformation techniques (e.g., rotations, flips, and brightness adjustments) crucial for increasing the diversity of the training dataset, enhancing model robustness, and mitigating overfitting, especially vital with limited medical image data.

MONAI (Medical Open Network for AI): This open-source, PyTorch-based framework was instrumental for domain-specific operations in medical imaging. MONAI provided specialized functionalities for handling medical image formats, a rich collection of medical-specific data transforms and augmentation techniques (such as those for intensity and spatial adjustments), and optimized implementations of common medical imaging components like loss functions (e.g., Dice Loss). Its modular design streamlined the data loading, preprocessing, and model evaluation pipeline, ensuring robust and reproducible medical AI development.

Segmentation Models PyTorch (SMP): The SMP library was employed for constructing the core segmentation model. SMP offers a high-level API to access a wide array of popular segmentation architectures and, critically, provides a vast selection of pre-trained convolutional encoders (backbones) from natural image datasets (e.g., ImageNet). Leveraging these pre-trained encoders (e.g., ResNet, EfficientNet) enabled the effective use of transfer learning, significantly accelerating model convergence and enhancing feature extraction capabilities, which were then fine-tuned for the specific task of brain tumor segmentation.

4. Deep Learning Architectures:

U-Net and its Variants: The U-Net architecture served as the foundational deep learning model for the 2D binary brain tumor segmentation task. U-Net is a cornerstone in biomedical image segmentation, characterized by its symmetric encoder-decoder structure and vital skip connections. The encoder (contracting path) systematically reduces spatial dimensions while increasing feature depth, capturing hierarchical contextual information. Conversely, the

decoder (expansive path) upsamples these features to reconstruct the segmentation map at the original image resolution. The unique skip connections directly transfer high-resolution feature maps from the encoder to corresponding layers in the decoder, thereby preserving fine-grained spatial details that are crucial for accurate boundary delineation of irregular tumor shapes and for combating the loss of information during downsampling. This fully convolutional nature of U-Net also allowed for flexible input image sizes and proved highly effective in achieving precise pixel-level segmentations even with the limited, typically available medical datasets.

5. Computational Hardware Environment:

Kaggle Notebooks with NVIDIA Tesla P100 GPU: The intensive training and experimentation for this deep learning project were performed within Kaggle's cloud-based notebook environment. This platform provided access to NVIDIA Tesla P100 Graphics Processing Units (GPUs), which are highly optimized for parallel computing and demanding deep learning workloads. Key features of the P100 that were instrumental include:

Its powerful Pascal Architecture which delivers high single-precision floating-point performance (up to 10.6 TFLOPS), significantly accelerating complex deep learning calculations.

16 GB of High Bandwidth Memory (HBM2) (up to 732 GB/s), critical for efficiently managing large 3D medical image volumes and enabling larger batch sizes during training, thereby preventing out-of-memory errors.

3584 CUDA Cores enabling massive parallel processing, which dramatically reduced the computational time required for model training and inference.

6. U-Net and its Variants:

U-Net is a convolutional neural network architecture originally developed for biomedical image segmentation. Introduced by Ronneberger et al. in 2015, U-Net is distinguished by its encoder-decoder structure with skip connections. The encoder progressively reduces the spatial dimensions to capture context, while the decoder reconstructs the spatial resolution to enable precise localization. The skip connections directly link corresponding layers of the encoder and decoder to preserve fine-grained features that are crucial for accurate segmentation. Due to its effectiveness and simplicity, U-Net has become a foundational model in medical image analysis. However, to address its limitations such as handling complex boundaries, multi-scale features, or training data scarcity several variants have been proposed. One widely used extension is the Attention U-Net, which incorporates attention gates into the skip connections, allowing the network to focus on the most relevant spatial regions and suppress irrelevant or noisy features. This mechanism significantly enhances segmentation performance, especially in medical images with cluttered backgrounds or small tumor regions.

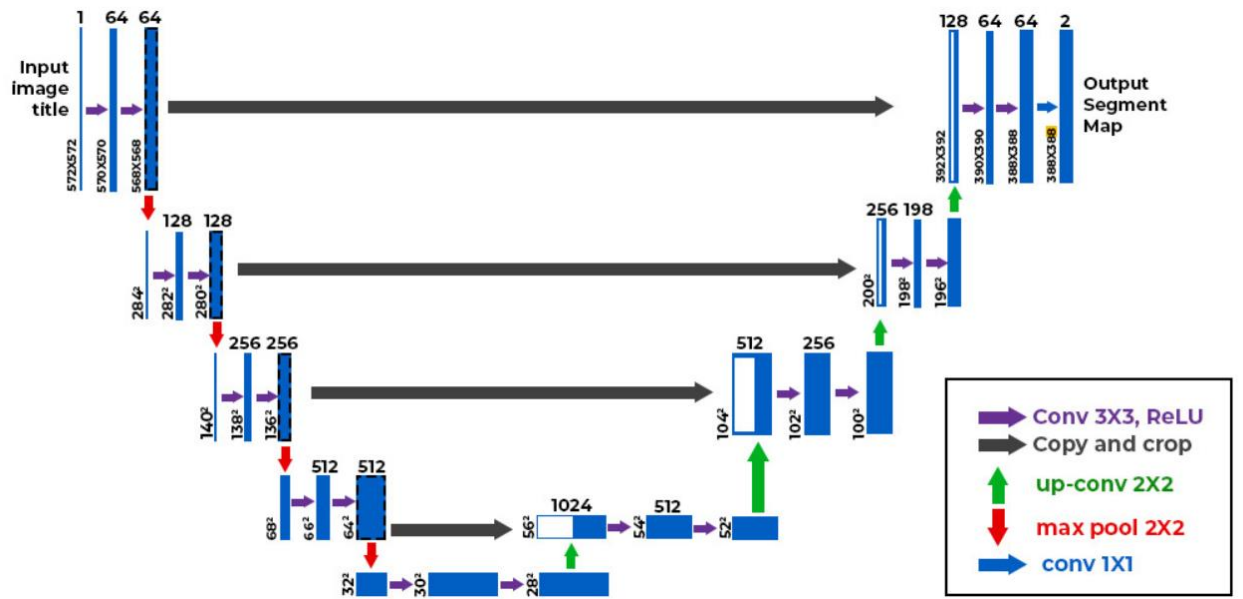


Figure 2.3: U-Net Architecture

3- Analysis and Design

3.1 System Overview

3.1.1 System Architecture

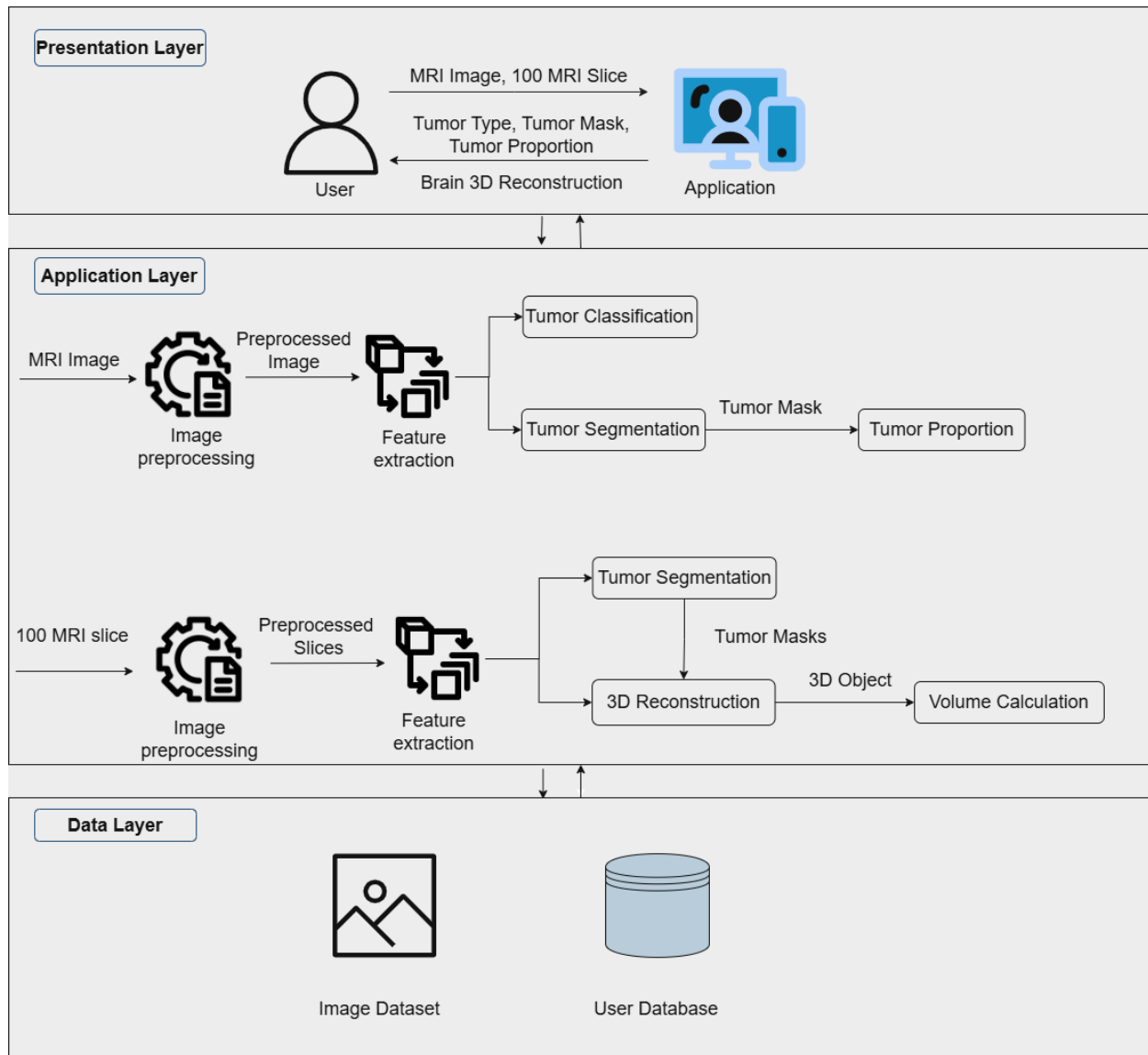


Figure 3.1: System Architecture

3.1.2 System Users

A. Intended Users:

Medical Staff: Upload images then the system classifies and outlines tumors. This aids quick diagnosis and reporting.

B. User Characteristics

Medical Expertise: Users must understand brain anatomy, radiology, and tumor types. They need to interpret image results clinically.

3.2 Design Diagrams

3.2.1 Use Case Diagram

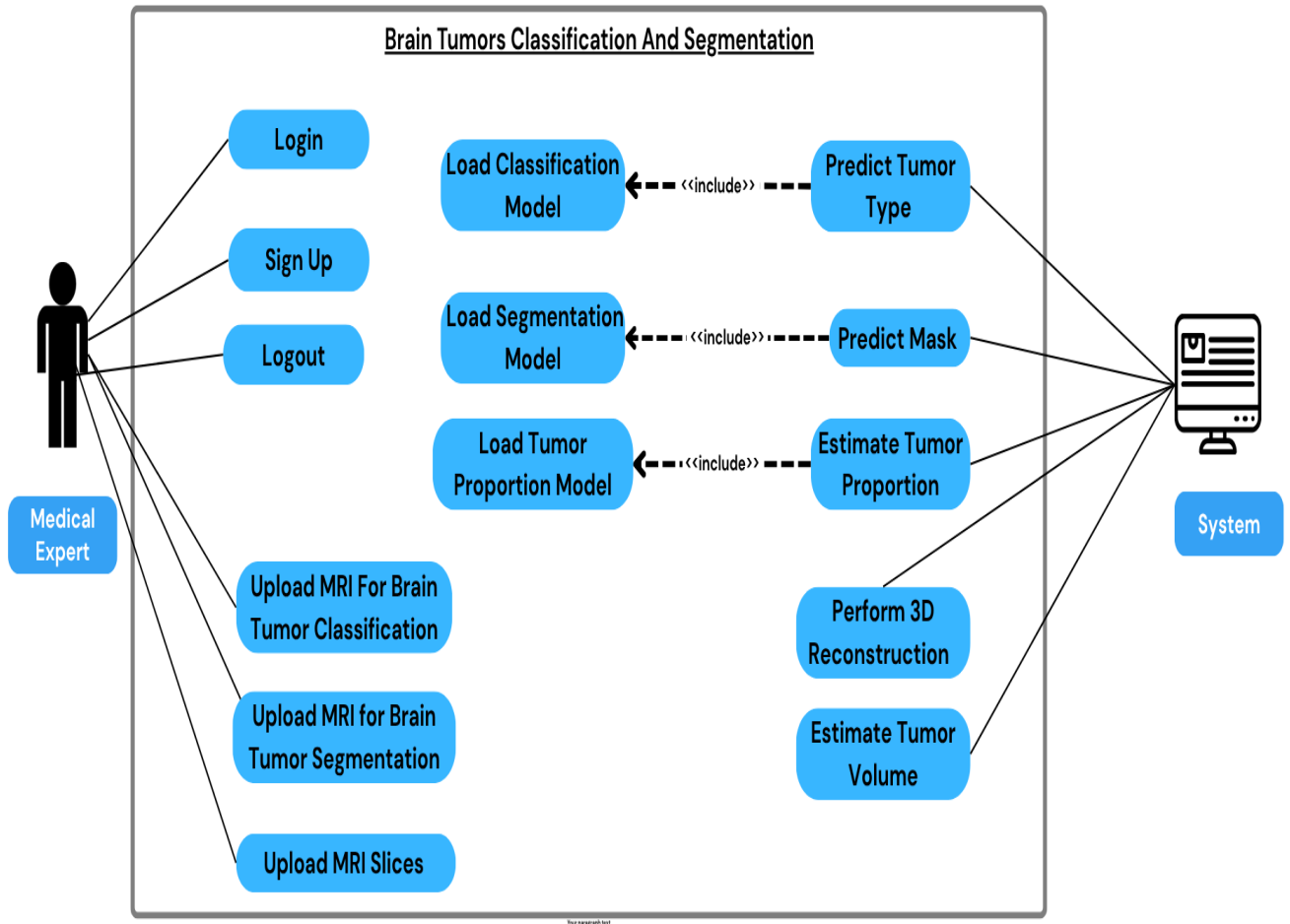


Figure 3.2: Use Case Diagram

3.2.2 Sequence Diagram

Brain Tumor Classification

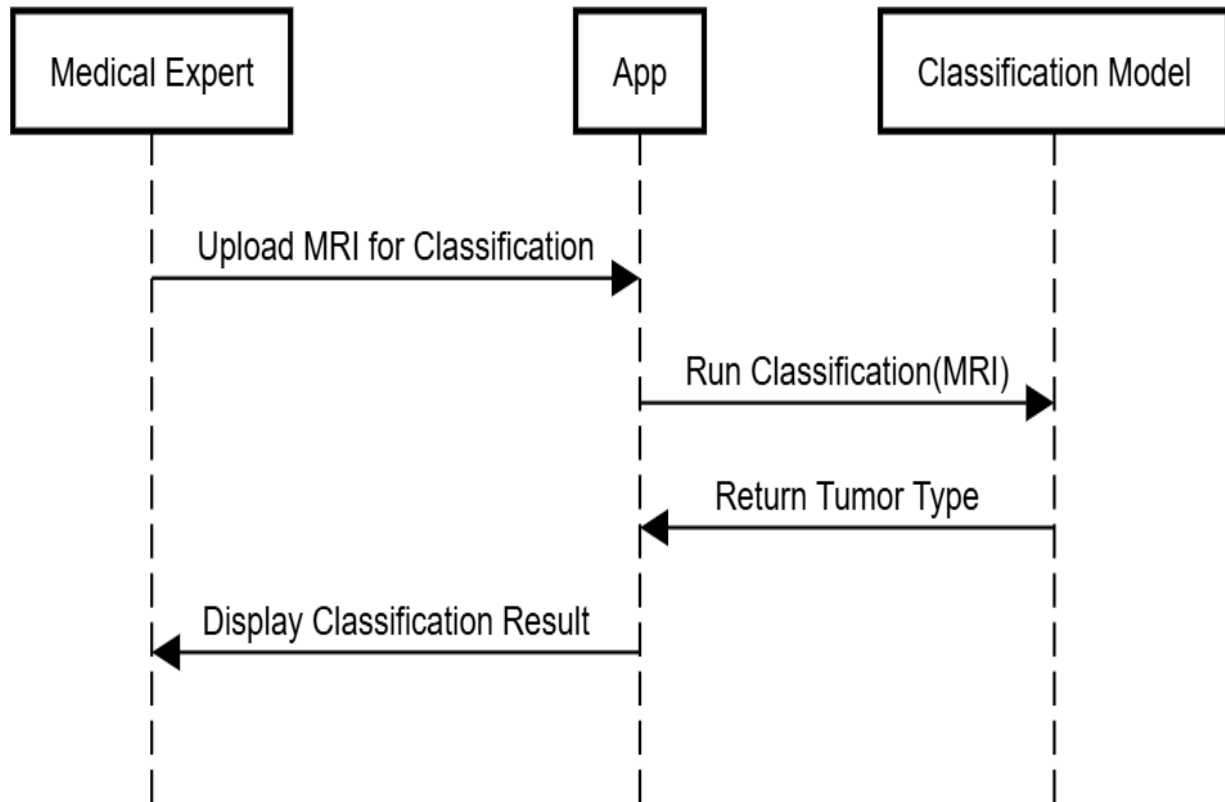


Figure 3.3: Classification Sequence Diagram

Brain Tumor Segmentation

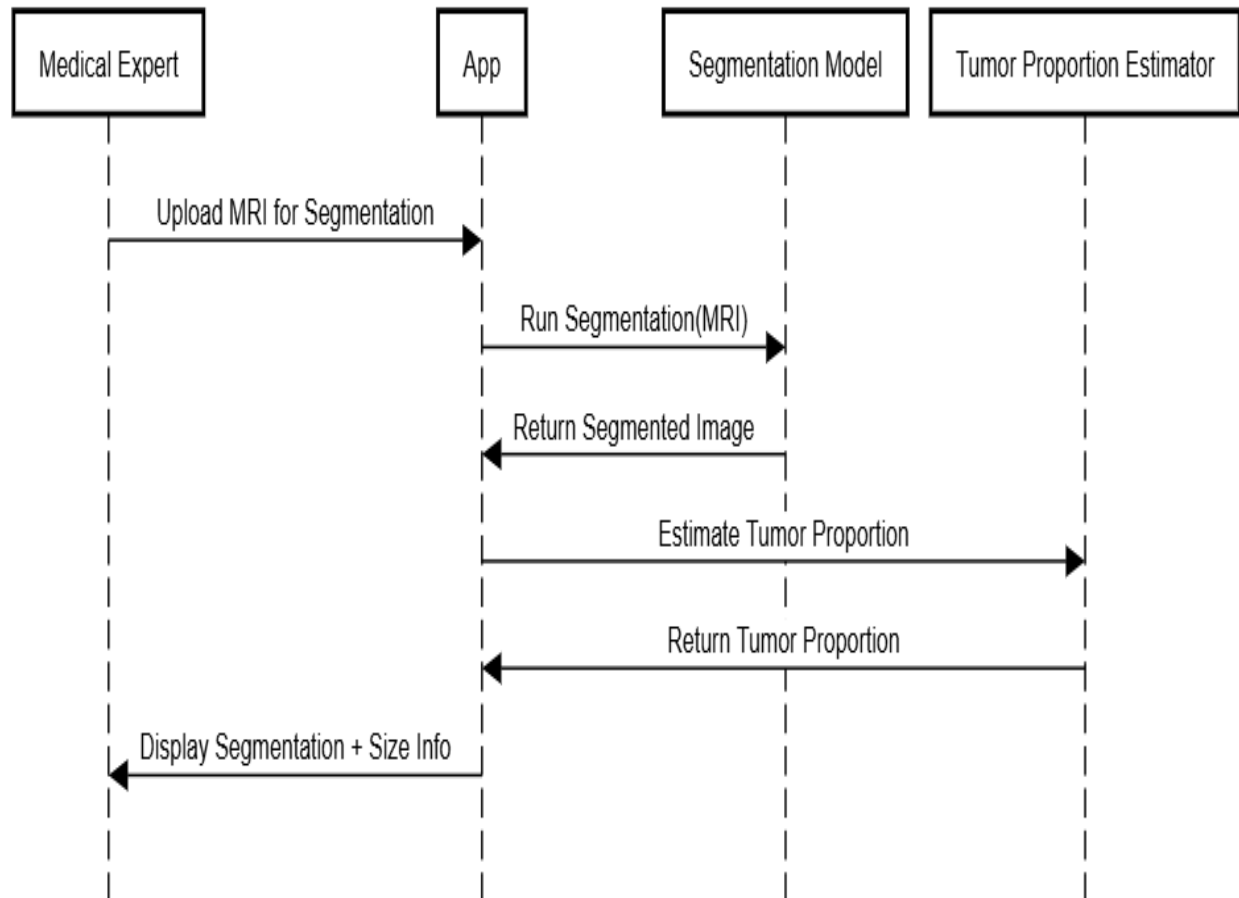


Figure 3.4: Segmentation Sequence Diagram

Brain 3D Reconstruction

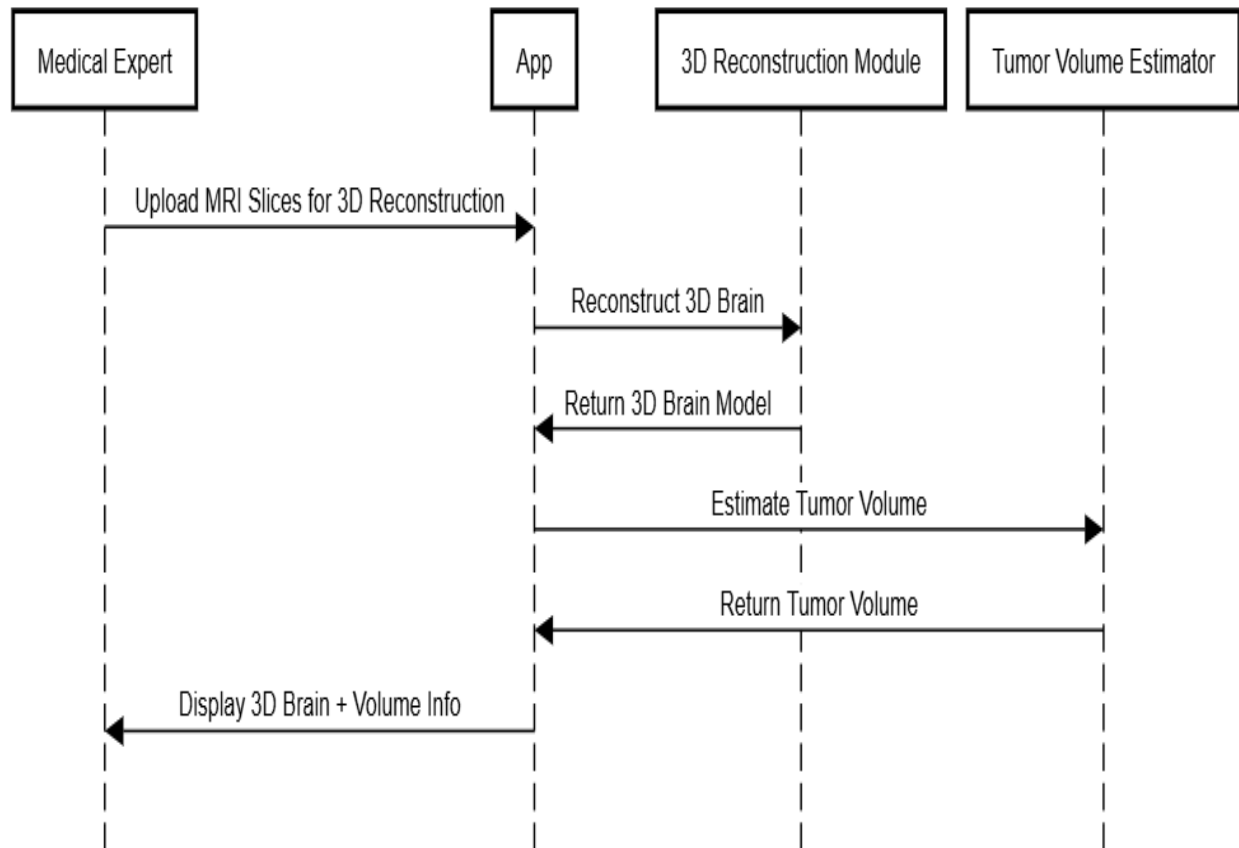


Figure 3.5: 3D Reconstruction Sequence Diagram

4- Implementation and Testing

4.1 Dataset Description

4.1.1 Dataset for Brain Tumor Classification

- **Source:** This dataset is obtained from Kaggle (commonly found as "Brain Tumor MRI Dataset" or "Brain MRI Scans for brain tumor classification"). It is a collection of pre-processed MRI brain images.
- **Composition:** The dataset comprises 2D MRI brain images, each explicitly labeled into one of four distinct classes:
 - **Glioma:** Representing images containing glioma tumors.
 - **Meningioma:** Representing images containing meningioma tumors.
 - **No Tumor:** Images depicting healthy brains without any tumor presence.
 - **Pituitary Tumor:** Representing images containing pituitary gland tumors.
- **Data Distribution:** The dataset is systematically divided into training/validation and testing sets to ensure an unbiased evaluation of the model's generalization capabilities. The distribution of images across these classes and splits is detailed below:

<u>Class</u>	<u>Training & Validation images</u>	<u>Testing images</u>
Glioma	1312	300
Meningioma	1339	306
No Tumor	1595	405
Pituitary	1457	300

Table 4.1: Classification Dataset

4.1.2 Dataset for Brain Tumor Segmentation

- **Source:** This dataset is also sourced from Kaggle specific Kaggle Given its focus on multiple tumor types with masks; it is highly suitable for learning precise tumor boundaries.
- **Composition:** This dataset is comprised of pairs of MRI images of the brain and their corresponding expert-annotated masks. The cases within this dataset include three specific tumor types:
 - **Glioma:** MRI images with corresponding masks delineating glioma tumors.
 - **Meningioma:** MRI images with corresponding masks delineating meningioma tumors.
 - **Pituitary:** MRI images with corresponding masks delineating pituitary tumors. Each entry includes a raw MRI image (likely 2D slices, given typical Kaggle segmentation datasets) and a pixel-wise mask that precisely highlights the tumor region.
- **Data Distribution:** The distribution of MRI images and their associated masks across the classes is as follows:

<u>Class</u>	<u>Number of MRI Images and Masks</u>
Glioma	1426 MRI images and masks
Meningioma	708 MRI images and masks
Pituitary	930 MRI images and masks

Table 4.2: Segmentation Dataset

4.2 Phases Description

1. Image Preprocessing

The primary objective of image preprocessing is to enhance the quality of raw MRI images and standardize their characteristics, thereby optimizing them for subsequent analysis by deep learning models. This step is crucial for ensuring consistency across varied input data and improving model robustness. Two main methods are employed: Normalization and Resizing. Normalization scales pixel intensity values across all MRI images to a consistent range, typically $[0,1]$, ensuring stable model training by mitigating variations in signal intensity. Resizing uniformly scales all images to a standard dimension, such as 256×256 pixels, which is necessary for input consistency required by deep learning architectures and to manage computational load. The procedure for image preprocessing involves first loading the raw MRI images into the system. Following this, each image's pixel intensity values are transformed into the $[0,1]$ range. Finally, the images are resized to the predefined standard dimensions.

2. Feature Extraction

The goal of feature extraction is to derive meaningful and discriminative features from the preprocessed MRI images that are essential for accurate tumor classification. Two main approaches are considered. Shape Features, such as tumor area and perimeter, can be extracted from the delineated tumor masks after segmentation, providing geometric insights into the tumor's morphology. More prominently, Deep Learning Features are automatically extracted by leveraging the internal representations learned by Convolutional Neural Networks (CNNs). This approach utilizes pre-trained CNN backbones such as VGG16 or

ResNet, which have learned to recognize a wide array of visual patterns from large image datasets (e.g., ImageNet), and their early and mid-layers serve as robust feature extractors for medical images, especially through transfer learning. The procedure for deep learning feature extraction involves inputting the preprocessed image into a feature extraction pipeline. The preprocessed image is then passed through a pre-trained network to extract high-level features, encapsulating complex patterns relevant to tumor characteristics.

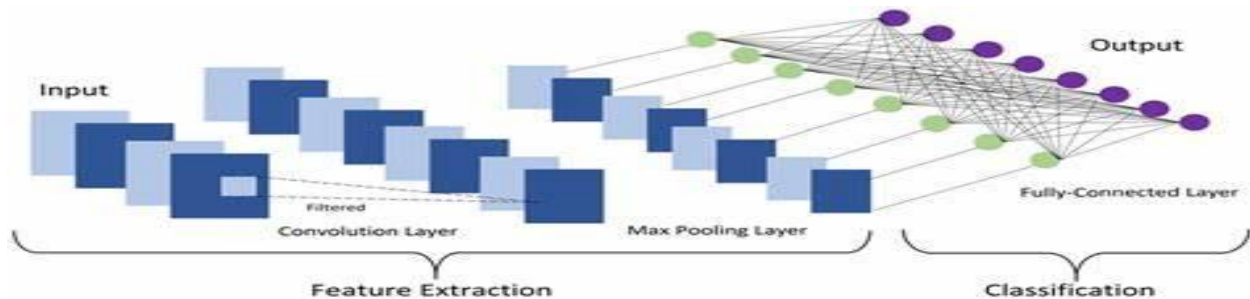


Figure 4.1: Feature Extraction

3. Image Classification

The purpose of this phase is to accurately determine the presence of a brain tumor and classify its specific type (glioma, meningioma, pituitary tumor) from input MRI images, or identify if no tumor is present. For classification, state-of-the-art deep learning models such as MobileNetV2, Xception, and VGG16 are employed, leveraging their proven effectiveness in image classification. The primary approach is to utilize transfer learning, where models pre-trained on vast general image datasets are fine-tuned on the specific brain MRI classification dataset. This strategy significantly enhances accuracy and efficiency, especially given the typically smaller size of medical imaging datasets. To further improve model robustness and generalization, extensive data augmentation techniques (e.g., rotation, scaling, flipping, brightness adjustments) are applied during training.

The classification procedure involves several key steps. For Model Training, the chosen classification model (e.g., MobileNetV2) is trained using the labeled dataset of MRI images. Data augmentation techniques are applied on-the-fly during this process. The model is optimized by tuning hyperparameters like learning rate and batch size, and by employing regularization techniques such as early stopping and dropout to prevent overfitting and ensure better generalization. Regarding Feature Extraction within the classification flow, MRI images are implicitly processed by the pretrained backbone of models like MobileNetV2 or VGG16, and the high-level features generated are fed into the model's classification head. For Classification, the model then uses these extracted features to predict the tumor type, assigning the image to one of the predefined categories: Glioma, Meningioma, Pituitary tumor, or No tumor. Validation and Testing are crucial for evaluating the model's performance on unseen data using metrics such as Accuracy, Precision, Recall, and a Confusion Matrix. Accuracy provides the overall proportion of correct classifications, while Precision and Recall measure the model's ability to correctly identify specific tumor types. The Confusion Matrix helps analyze misclassifications and guides further fine-tuning. Finally, upon satisfactory validation, the trained model is prepared for Deployment for real-world application as a diagnostic aid.

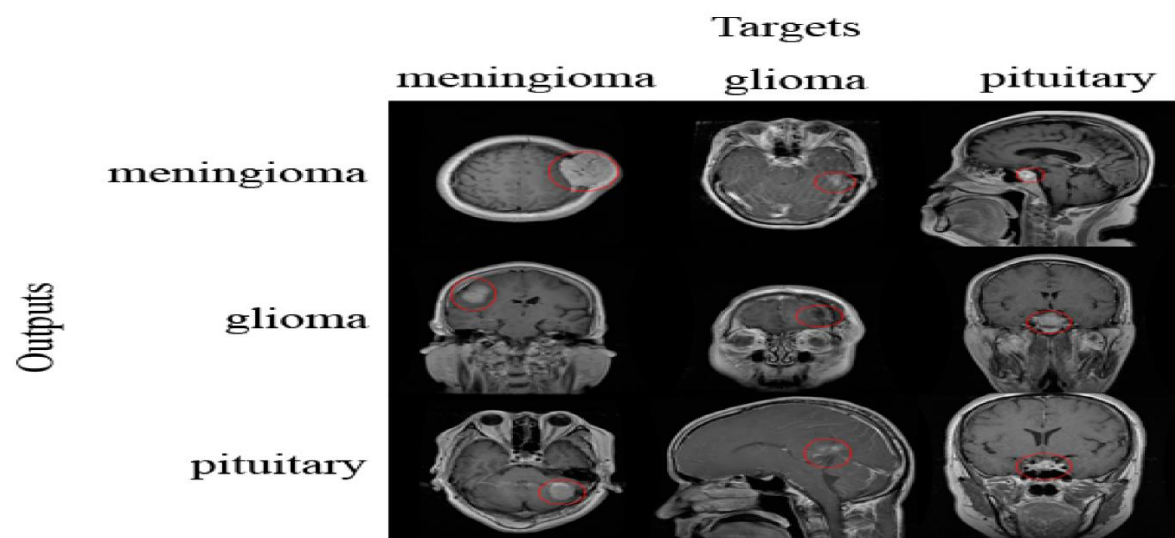


Figure 4.2:Image Classification

4. Image Segmentation

The purpose of image segmentation is to precisely delineate tumor regions within MRI images, generating accurate masks that are crucial for detailed diagnosis, surgical planning, and treatment monitoring. This phase provides the pixel-level localization required for subsequent 3D reconstruction and volumetric analysis.

Several advanced deep learning architectures are leveraged for robust tumor segmentation. ResNet (Residual Network) is employed as a powerful backbone to extract hierarchical features from MRI images. Its skip connections mitigate vanishing gradients, enabling deep feature learning that enhances tumor boundary detection. Attention U-Net, an extension of the U-Net architecture, integrates attention gates to enable the model to selectively focus on relevant tumor regions while suppressing irrelevant background noise. This approach enhances feature propagation by dynamically emphasizing salient features at different spatial locations, leading to more precise delineation. Dynamic U-Net (often referring to frameworks like U-Net) offers flexible, encoder-decoder architecture designed to adapt to varying input sizes and medical imaging domains. This approach combines powerful feature extraction with dynamic upsampling, making it ideal for precise delineation of irregular tumor shapes. Furthermore, EfficientNet is utilized, applying a compound scaling method to balance network depth, width, and resolution for optimal efficiency and performance. This architecture achieves high accuracy with fewer parameters, proving effective for detecting and segmenting tumors in high-resolution images. The segmentation procedure begins by inputting the preprocessed MRI images into the chosen segmentation model. These images are normalized to a consistent intensity range, and the dataset is further augmented with techniques like rotations, flips, and scaling to improve model generalization. The segmentation model then processes the image

to generate a binary mask, precisely highlighting the tumor region. Each model outputs such a binary mask. The final step involves Validation and Testing of the segmentation quality using key performance metrics. The Dice Score measures the overlap between the predicted and ground truth masks, providing a robust indication of segmentation accuracy, while IoU (Intersection over Union) quantifies the accuracy of segmentation by comparing the area of overlap to the area of union between predicted and true masks.

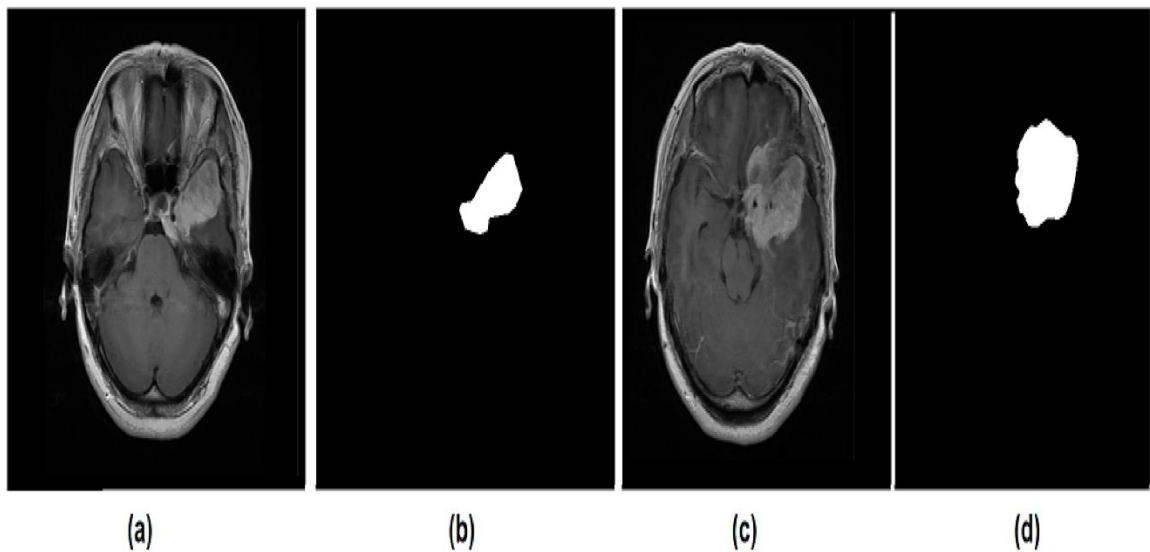


Figure 4.3:Image Segmentation

5. Tumor Proportion Calculation

Tumor proportion estimation is used to assist medical experts by determining the proportion of the tumor in a single 2D MRI scan. This is done by calculating the number of white pixels (foreground) and dividing it by the total size of the tumor mask (width \times height).

6. 3D Reconstruction

To reconstruct the brain and tumor in 3D, the user first uploads at least 100 MRI slices representing different cross-sections of the brain. These slices are sent to a trained segmentation model, which generates a binary tumor mask for each individual slice. Once segmentation is complete, both the original MRI slices and the corresponding tumor masks are stacked along the z-axis, forming two separate 3D volumes: one representing the brain and the other the tumor region. These volumes are then color-coded differently to distinguish between healthy tissue and tumor. Next, the Marching Cubes algorithm is applied server-side to both volumes. This algorithm extracts the surface mesh of each volume by identifying isosurfaces and converting them into triangular mesh representations. The resulting meshes are combined to form a single 3D object with overlapping brain and tumor regions. The server outputs the mesh as a 3D NumPy array or standard .glb format, which is sent back to the Flutter frontend. In the application, the 3D object is rendered interactively using packages such as `model_viewer_plus`, allowing the user to explore and visualize the tumor's shape and location inside the brain.

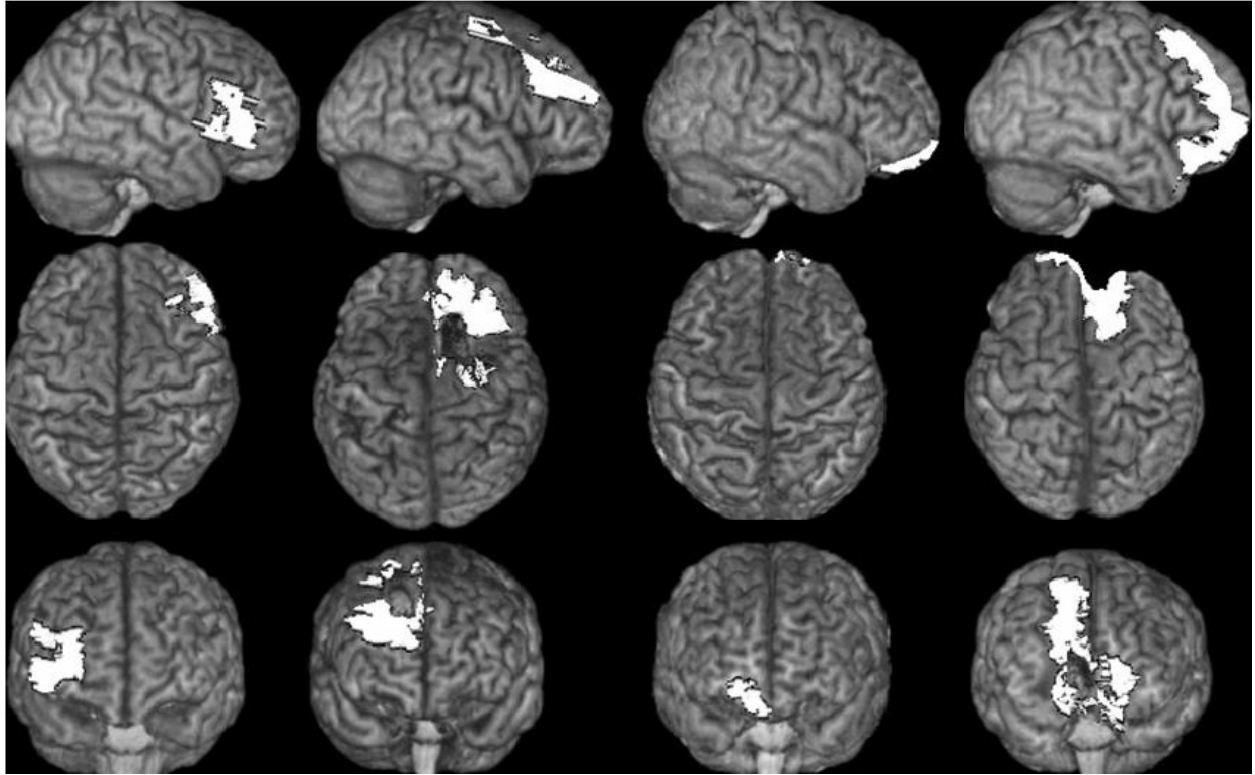


Figure 4.4: 3D Brain

7.Decision-Making and Output

The purpose of this final phase is to deliver actionable and easily interpretable outputs to the end-users (medical professionals), thereby directly assisting their diagnostic and treatment planning processes.

The primary method for output delivery is a User Interface (UI), which serves to display all the analytical results, including the classified tumor type, its calculated size/volume, and the interactive 3D model. The procedure involves evaluating the outcome of the tumor detection from the classification phase. If a tumor is detected, the UI prominently displays its classification, size, and 3D visualization. If no tumor is detected, the user is clearly notified.

4.3 Technologies

1. Kaggle

Kaggle served as the primary cloud-based development environment and a crucial repository for the datasets utilized in this project. It is an online platform widely recognized in data science and machine learning, providing essential resources such as free access to Jupyter notebooks, powerful GPUs, and a vast collection of public datasets. Its purpose in this project was to provide a robust, scalable, and accessible environment for developing, training, and rigorously evaluating our artificial intelligence models. The decision to choose Kaggle was driven by its provision of free access to powerful hardware, facilitating computationally intensive deep learning tasks. Furthermore, its features enable easy collaboration, and it offers seamless integration with the Python libraries and frameworks central to our workflow.

2. PyTorch

PyTorch is the core deep learning framework adopted for the implementation of this project. It is a widely used, open-source library for deep learning, distinguished by its dynamic computational graphs and efficient GPU acceleration capabilities. In this project, PyTorch serves as the fundamental framework for designing, implementing, training, and testing all deep learning models, including those for image classification and segmentation. The framework was chosen for its exceptional flexibility, which significantly aided in rapid prototyping and debugging. Its strong community support ensures access to extensive resources and updates, and its seamless integration with specialized libraries like MONAI and `segmentation_models.pytorch` further streamlined the development process.

3. MONAI (Medical Open Network for AI)

MONAI is a specialized PyTorch-based framework explicitly designed for deep learning applications within healthcare imaging. It provides a comprehensive suite of tools specifically tailored for medical image preprocessing, advanced augmentation techniques, and the development of robust deep learning models. Within this project, MONAI was instrumental in facilitating precise medical image segmentation, handling of complex medical datasets (such as NIfTI and DICOM formats), and incorporating loss functions that are specifically optimized for medical AI tasks. Its selection was based on its direct optimization for medical imaging tasks, offering robust support for both 2D and 3D data, and providing access to pretrained models that can serve as strong baselines.

4. Segmentation Models PyTorch (SMP)

Segmentation_models.pytorch (SMP) is a high-level PyTorch library that significantly accelerates the development of image segmentation solutions. It offers ready-to-use, efficient implementations of popular semantic segmentation architectures, including U-Net, DeepLabV3, FPN, and others. The primary purpose of SMP in this project was to enable quick implementation and training of advanced segmentation models, providing flexibility to experiment with various encoder backbones (such as ResNet or EfficientNet). This library was chosen because it simplifies segmentation model development by providing access to pretrained weights, a standardized model structure, and straightforward customization options, thereby reducing development time and effort.

5. Albumentations

Albumentations is a crucial data augmentation library employed in this project. It is known for its speed and flexibility in applying image transformations, making it highly compatible with deep learning workflows. Its purpose in this project was to apply a diverse range of data augmentations, including geometric transformations (e.g., flipping, rotation, scaling, cropping) and photometric adjustments (e.g., brightness, contrast changes). These augmentations are vital for artificially expanding the training dataset, enhancing model robustness against variations in real-world MRI scans, and significantly improving generalization capabilities to unseen data. The library was chosen for its high-performance execution and ease of integration with PyTorch and OpenCV.

6. NumPy

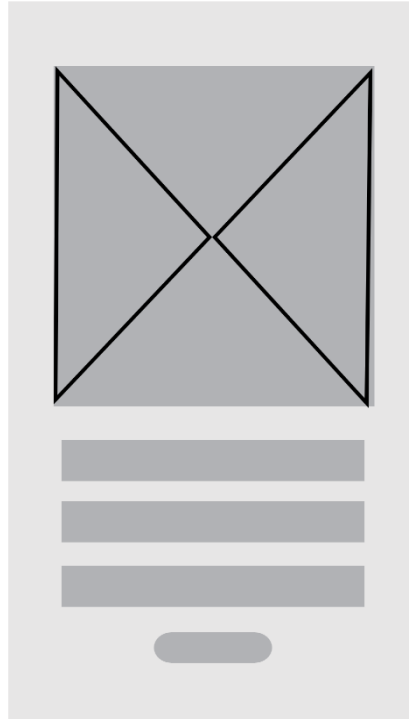
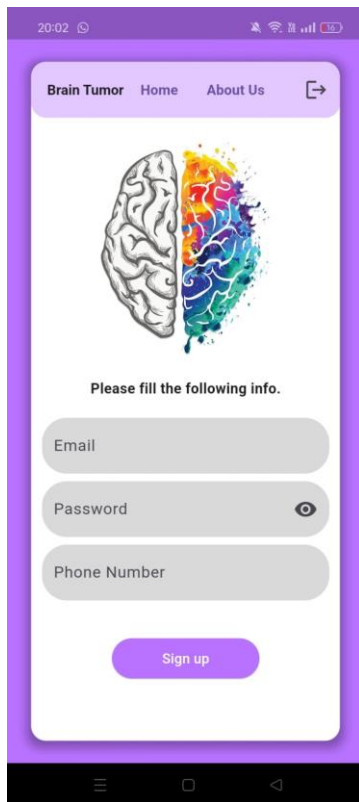
NumPy is a fundamental and indispensable library for numerical operations in Python, serving as the bedrock for efficient data manipulation in this project. It provides robust support for multi-dimensional arrays and matrices, which are the primary data structures for handling image pixel data. Its purpose in the project involved extensive array manipulation, performing mathematical operations on image arrays, and facilitating seamless interfacing between different data formats, such as converting image arrays into PyTorch tensors. NumPy was selected for its exceptional efficiency, reliability, and its universal adoption in scientific computing and artificial intelligence workflows, making it a ubiquitous tool for data processing.

7. OpenCV (Open-Source Computer Vision Library)

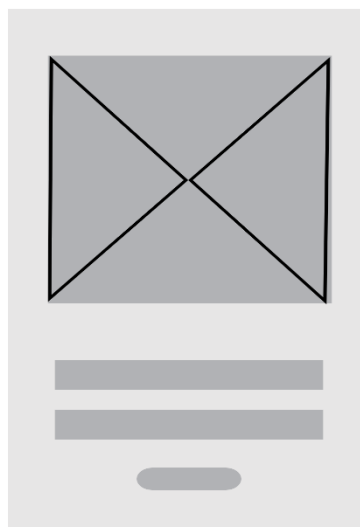
OpenCV is a widely recognized and robust open-source library focused on real-time computer vision applications. It provides a comprehensive set of tools for various image processing and transformation tasks. In this project, OpenCV was utilized for fundamental image preprocessing operations such as efficient image resizing, intensity normalization, and other basic transformations before the data is fed into the deep learning models. Its choice was based on its established reliability, broad community support, and its highly efficient implementation of core image processing capabilities.

4.4 UI Design and Wireframes

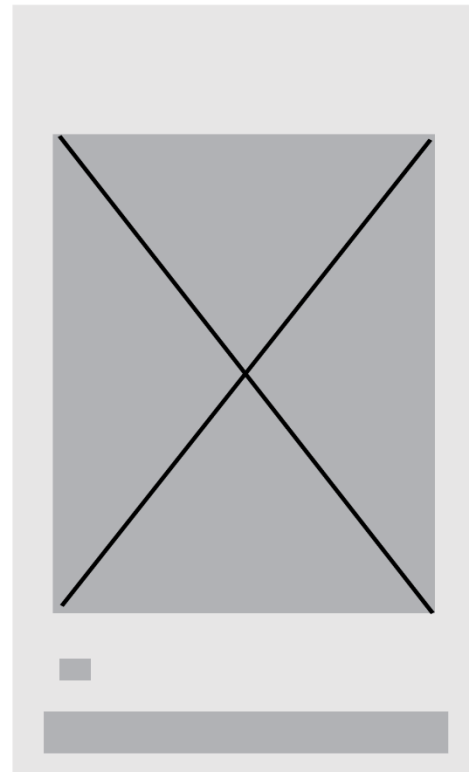
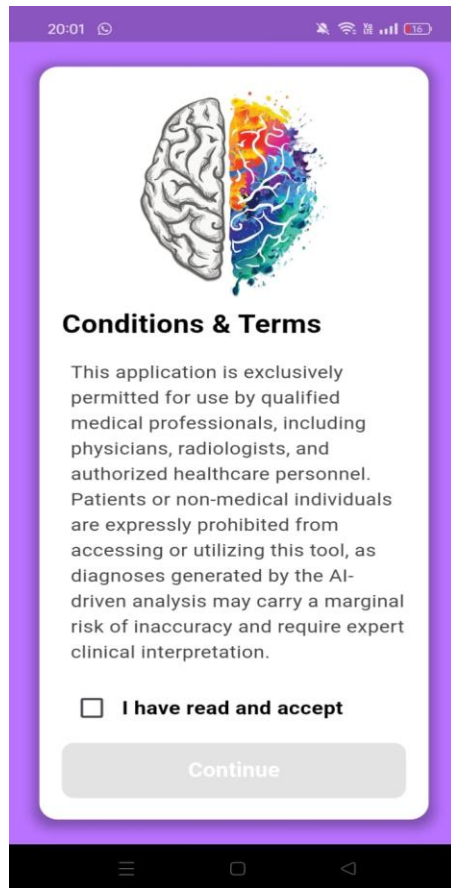
1. Sign-Up Screen:



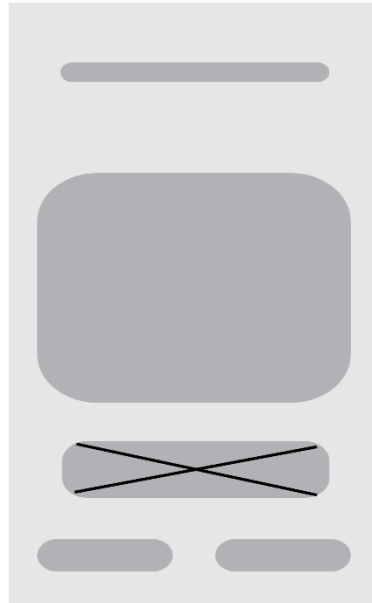
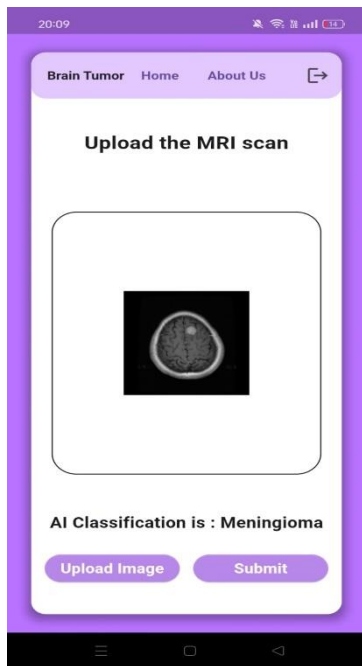
2. Login Screen:



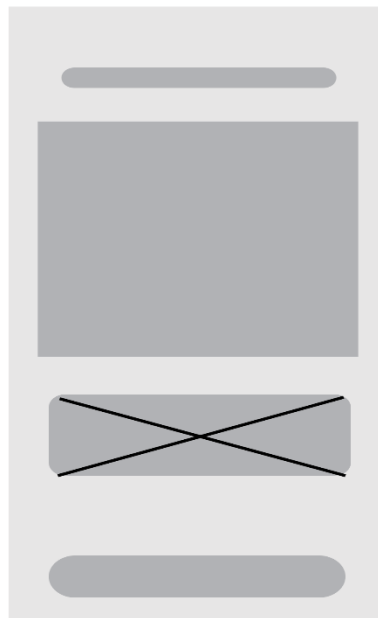
3. Conditions and Terms Screen:



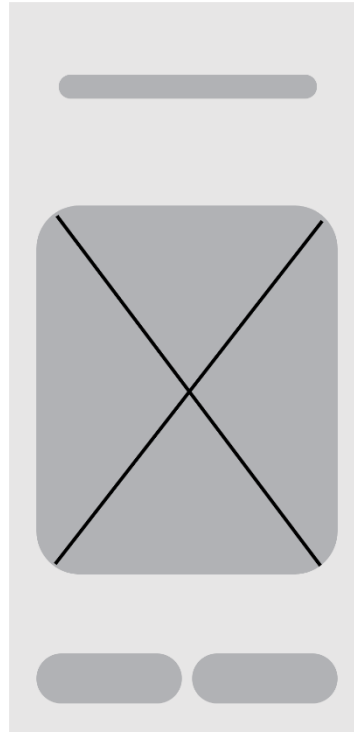
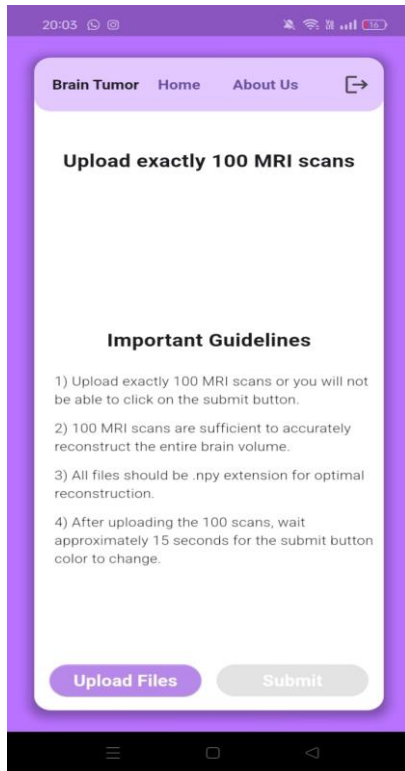
4. Tumor Classification Screen:



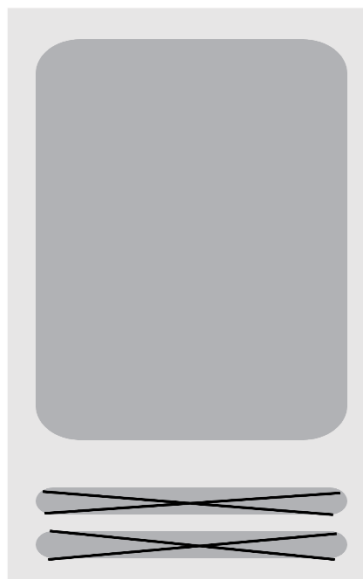
5. Tumor Segmentation Screen:



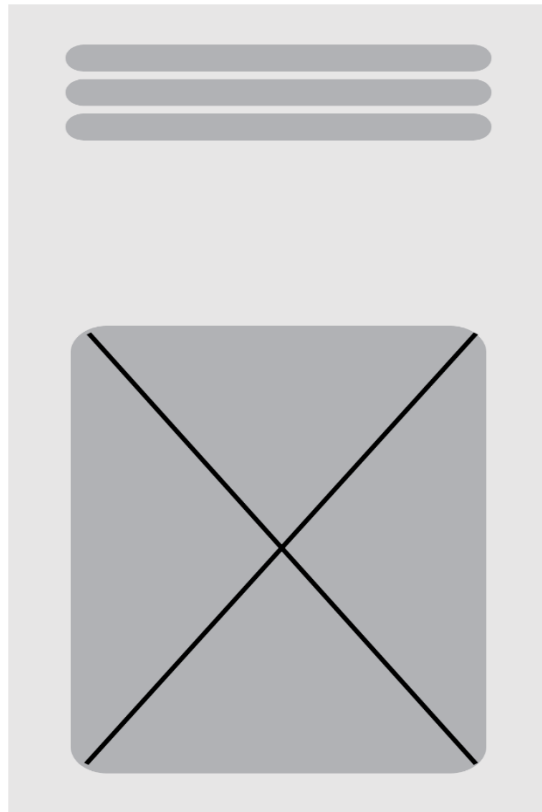
6. 3D Submit Files Screen:



7. 3D Rendering Screen:



8. About Us Screen:



4.5 Experimental Results

1. Classification Results:

This table presents the performance of five different model architectures: VGG16, MLP Mixer b16, Resnet50, Densenet121, and a custom-built CNN from scratch on a classification task, evaluated by Training Accuracy, Validation Accuracy, Testing Accuracy, and Testing F1-score. Based on these results, Densenet121 and the CNN from scratch models are the top performers, demonstrating superior accuracy and F1-scores on the validation and testing datasets.

<u>Model Architecture</u>	<u>Training Accuracy (%)</u>	<u>Validation Accuracy (%)</u>	<u>Testing Accuracy (%)</u>	<u>Testing F1-score (%)</u>
VGG16	27.92	30.87	30.89	47
MLP Mixer b16	91.71	91.23	91.23	90.75
Resnet50	97.71	96.04	96.03	96
Densenet 121	97.57	98.17	98.17	98.5
CNN from scratch	99.23	98.32	98.32	98.32

Table 4.3: Classification Results

2. Segmentation Results:

This table presents the performance of six different model architectures: Timm-effnet-b8, Resnet152, Segformer1, DynUnet, Attention U-Net, and Deeplab v3_Resnet101 on a segmentation task, evaluated by Validation Dice (%), Validation IOU (%), Test Dice (%), and Test IOU (%). Based on these results, DynUnet and Attention U-Net models are the top performers, demonstrating superior Dice and IOU scores on the validation and testing datasets.

<u>Model Architecture</u>	<u>Val Dice (%)</u>	<u>Val IOU (%)</u>	<u>Test Dice (%)</u>	<u>Test IOU (%)</u>
Timm-effnet-b8	85.72	75.17	85.91	78.25
Resnet 152	83.61	72.09	83.31	75.06
Segformer1	83.32	73.59	84.05	74.08
DynUnet	87.54	77.95	86.53	78.93
Attention U- Net	86.47	76.34	86.95	79.47
Deeplab v3_Resnet 101	82.73	70.83	83.4	71.38

Table 4.4: Segmentation Results

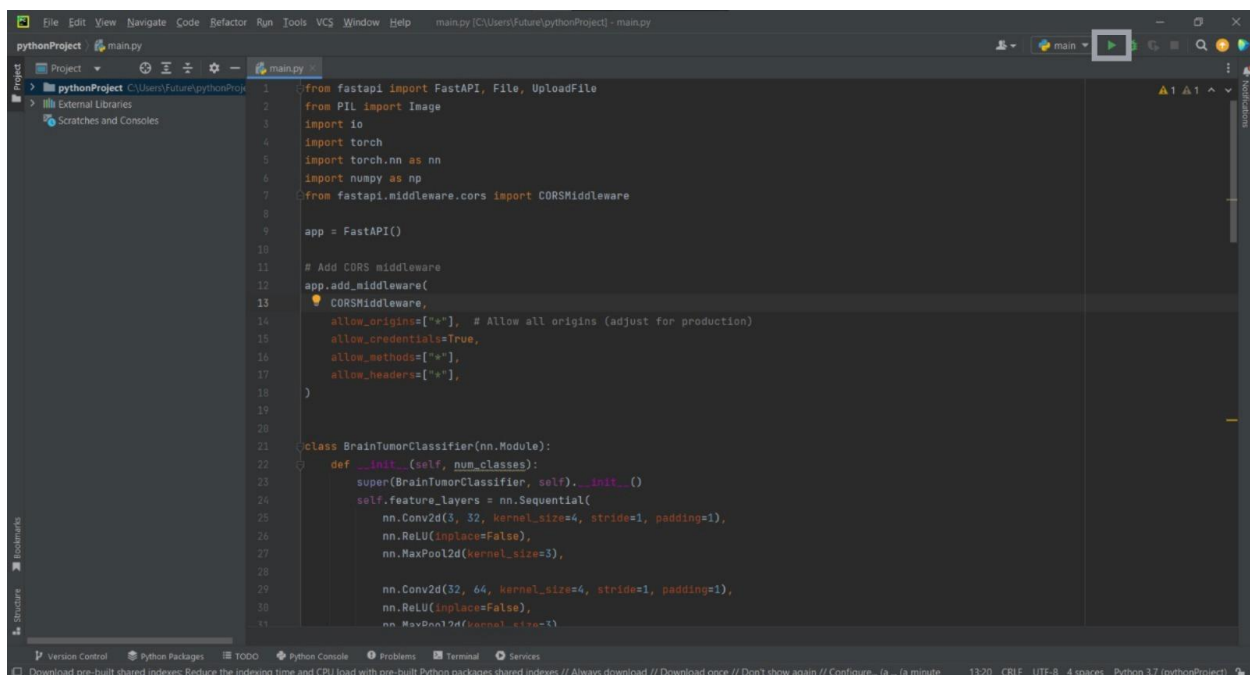
5- User Manual

In this chapter, we provide a detailed description of how to be able to use our flutter application. We used a local host technique to host our python backend on the laptop, and then we make a connection between the interface and the python backend using Fast API framework.

1. Create a network:

Connect your mobile phone and laptop to the same network via Wi-Fi or mobile hotspot.

2. Run all local hosts:

A screenshot of a Visual Studio Code editor window. The editor is open to a file named 'main.py' in a project called 'pythonProject'. The code is a FastAPI application that uses PyTorch for brain tumor classification. It includes imports for FastAPI, PIL, io, torch, torch.nn, numpy, and CORS middleware. The application sets up CORS middleware to allow all origins and defines a 'BrainTumorClassifier' class with a 'feature_layers' attribute. The 'feature_layers' is a sequential model with three layers: a Conv2d layer (32, kernel_size=4, stride=1, padding=1), a ReLU layer (inplace=False), and a MaxPool2d layer (kernel_size=3). The code is as follows:

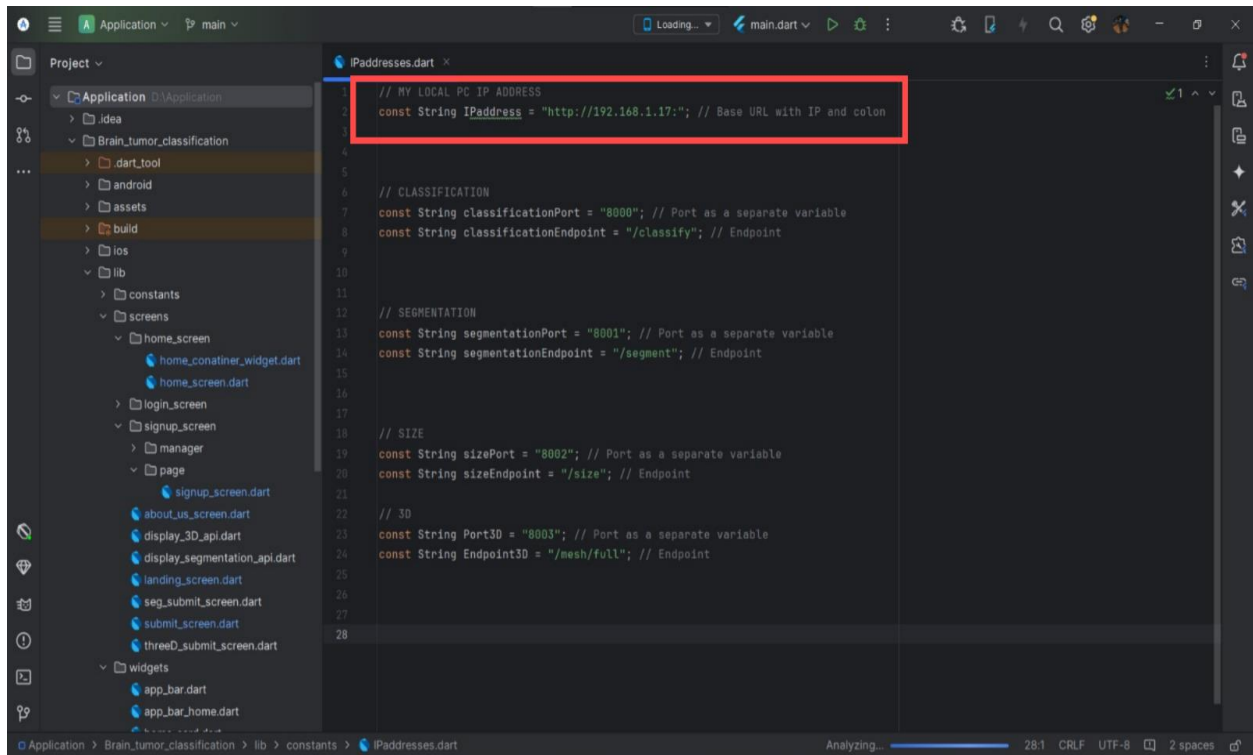
```
1 from fastapi import FastAPI, File, UploadFile
2 from PIL import Image
3 import io
4 import torch
5 import torch.nn as nn
6 import numpy as np
7 from fastapi.middleware.cors import CORSMiddleware
8
9 app = FastAPI()
10
11 # Add CORS middleware
12 app.add_middleware(
13     CORSMiddleware,
14     allow_origins=["*"], # Allow all origins (adjust for production)
15     allow_credentials=True,
16     allow_methods=["*"],
17     allow_headers=["*"],
18 )
19
20
21 class BrainTumorClassifier(nn.Module):
22     def __init__(self, num_classes):
23         super(BrainTumorClassifier, self).__init__()
24         self.feature_layers = nn.Sequential(
25             nn.Conv2d(3, 32, kernel_size=4, stride=1, padding=1),
26             nn.ReLU(inplace=False),
27             nn.MaxPool2d(kernel_size=3),
28
29             nn.Conv2d(32, 64, kernel_size=4, stride=1, padding=1),
30             nn.ReLU(inplace=False),
31             nn.MaxPool2d(kernel_size=3),
```

```
1 # ...
2
3 import ...
4
5 app = FastAPI()
6
7 STATIC_DIR = "static"
8 os.makedirs(STATIC_DIR, exist_ok=True)
9 app.mount("/static", StaticFiles(directory=STATIC_DIR, name="static"))
10
11 # Define the number of classes (use the same as before)
12 num_classes = 4 # Replace with your actual number of classes
13
14 # Load the entire model
15
16 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
17 segment_model = DynUNet(
18     spatial_dims=2,
19     in_channels=1, # Grayscale images
20     out_channels=1,
21     kernel_size=[3, 3, 3, 3, 3],
22     strides=[1, 2, 2, 2, 2],
23     upsample_kernel_size=[2, 2, 2, 2],
24     filters=[16, 32, 64, 128, 256],
25 )
26 segment_model.load_state_dict(
27     torch.load("D:\API\API\dynunet_unet_model-best.pth", map_location=device))
28 segment_model.to(device)
29 segment_model.eval()
30
```

```
1 from fastapi import FastAPI, File, UploadFile
2 from PIL import Image
3 import io
4 from pydantic import BaseModel
5 from fastapi.responses import JSONResponse, StreamingResponse
6 import torch
7 import torch.nn as nn
8 from torchvision import transforms
9 import numpy as np
10 import base64
11
12 app = FastAPI()
13
14
15 class TumorSizeCalculator(torch.nn.Module):
16     def __init__(self):
17         super().__init__()
18         self.transform = transforms.Compose([
19             transforms.ToTensor(),
20         ])
21
22     def forward(self, x):
23         if isinstance(x, str):
24             x = Image.open(x)
25         if isinstance(x, Image.Image):
26             x = x.convert('L')
27             x = self.transform(x)
28         return torch.mean(x).item()
29
30
```

3. IP configuration:

Replace this IP address with your current IP.

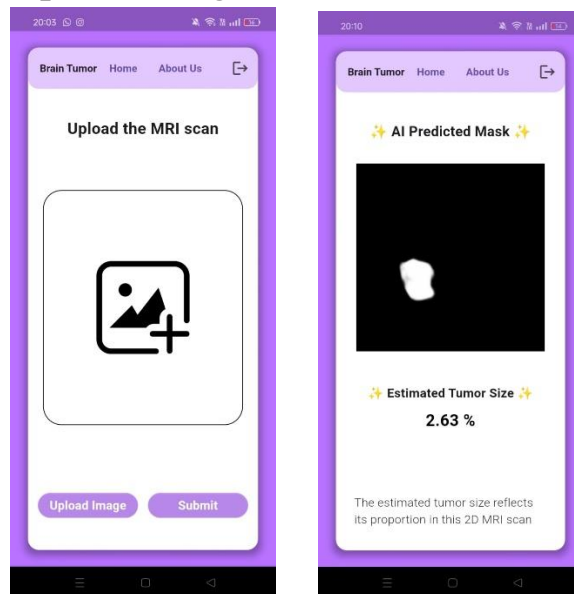


4. Run the flutter application:

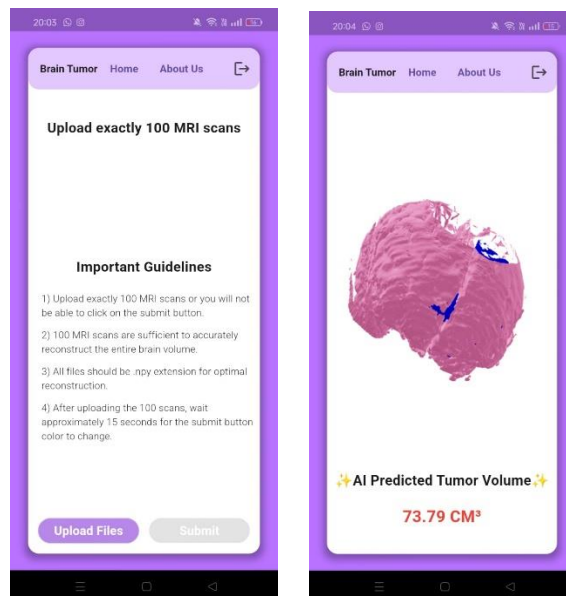
- Agree on the terms and conditions for your role as a medical expert.
- Log in if you have an account; otherwise, sign up.
- Upload image to do brain tumor classification:



- Upload image to do brain tumor segmentation:



- Upload 100 MRI scans in .npy extension to do brain 3D reconstruction:



6. Conclusion and Future Work

6.1 Conclusion

This project successfully developed and implemented a comprehensive system for brain tumor classification and segmentation, addressing a critical need for efficient and accurate diagnostic support in medical imaging. The system's primary objective was to assist medical staff by automating the analysis of MRI brain slices, providing capabilities for tumor type classification, precise tumor region segmentation, quantitative tumor proportion calculation, and 3D brain reconstruction for enhanced visualization.

The system's architecture revolved around the singular interaction point of Medical Staff, who can initiate various analytical tasks. Key functionalities include:

- **MRI Slice Classification:** Upon uploading a single MRI slice, the system accurately identifies and classifies the tumor type (e.g., Glioma, Meningioma, Pituitary Tumor, or no tumor). This classification capability was achieved using a Convolutional Neural Network (CNN) built from scratch. The results obtained demonstrated a Training Accuracy of 99.23%, Validation Accuracy of 98.32%, Test Accuracy of 98.32%, and a Test F1-Score of 98.32%. These strong metrics highlight the model's robustness in distinguishing between various tumor types, significantly aiding in preliminary diagnosis.
- **MRI Slice Segmentation and Proportion Calculation:** For individual slices, the system performs robust segmentation of tumor regions. This process leverages an Attention U-Net architecture. Following segmentation, the system automatically calculates the proportion of the tumor within the given slice. The segmentation results exhibited strong performance, with Validation

Dice of 86.47%, Validation IoU of 76.34%, Test Dice of 86.95%, and Test IoU of 79.47%. These metrics indicate highly accurate delineation of tumor boundaries, and the quantitative output provides valuable information for treatment planning and monitoring tumor progression.

- **3D Brain Reconstruction:** The system allows medical staff to upload a series of MRI slices to reconstruct a detailed 3D model of the brain. This 3D visualization facilitates a holistic understanding of tumor size, shape, and spatial relationship to surrounding brain structures.

6.2 Future Work

- Overlaying Segmented Tumors onto Original MRI Slices.
- Compute tumor sphericity.

References

- [1] Cheng, J., Zhang, L., Ma, S., Wu, J., & Li, J. (2023). Classification of MRI Brain Tumor Images Using a Deep Learning Segment Anything Model. *World Journal of Advanced Research* <https://doi.org/10.1155/2023/1224619> and *Reviews*, 23(1), 14-25.
- [2] Barik, R. K., & Das, A. (2023). A Comprehensive Review on Brain Tumor Detection and Classification Using Deep Learning Techniques. *The Journal of Supercomputing*. <https://doi.org/10.1007/s11227-023-05549-4>
- [3] Bashir, A., Ali, F., & Shah, F. (2023). A Review of Medical Imaging Techniques for the Detection of Brain Tumors. *Scientific Programming*, 2023, Article ID 2088. <https://scpe.org/index.php/scpe/article/view/2088>
- [4] Saha, P., & Ghosh, M. (2023). Classification of MRI Brain Tumor Images Using Deep Learning: Segment Anything Model. *World Journal of Advanced Research and Reviews*, 23(1), 1-13. https://wjarr.com/content/classification_mri-brain-tumor-images-using-deep-learning-segment-anything-model
- [5] Suh, S. Y., Park, S. H., Kim, D. H., & Yoo, S. H. (2023). Brain Tumor Classification Using Multiscale Convolutional Neural Network and Transfer Learning. *Brain Sciences*, <https://doi.org/10.3390/brainsci13091320> 13(9), 1320.
- [6] Liu, Z., Zhang, X., Zhang, T., & Zhang, Y. (2023). An L2 Normalized Spatial Attention Network for Medical Image Classification. *Papers with Code*. <https://paperswithcode.com/paper/an-l2-normalized-spatial-attention-network>
- [7] Awan, M. A., & Iqbal, J. (2022). Brain Tumor Detection Using Deep Learning Techniques: A Systematic Review. *Computational Intelligence and Neuroscience*, 2022, <https://doi.org/10.1155/2022/8104054> Article ID 8104054.
- [8] Nandha, S. S., & Kumar, S. (2023). A Novel Approach for Brain Tumor Detection Using Deep Learning Techniques. *Journal of Medical Imaging and Health Informatics*, 13(3), 635-643.
- [9] Kumar, P., Singh, S., Pal, S., Manav, M., & Tyagi, A. (2024). Tumor Segmentation for Brain Tumor Using Combination of Deep Learning and Machine Learning Algorithms. *JK-Practitioner*, 29(4), 48-54.

- [10] Preetha, R., Priyadarsini, M. J. P., & Nisha, J. S. (2025). Brain Tumor Segmentation Using Multi-Scale Attention U-Net with EfficientNetB4 Encoder for Enhanced MRI Analysis. *Scientific Reports*, 15, 9914.
- [11] Pasupuleti, M. K. (2025). Medical Image Analysis for Tumor Segmentation Using U-Net Variants. *International Journal of Academic and Industrial Research Innovations (IJAIRI)*, 5(5), 506–517.
- [12] Alquran, H., Alslatie, M., Rababah, A., & Mustafa, W.A. (2024). Improved Brain Tumor Segmentation in MR Images with a Modified U-Net. *Appl. Sci.*, 14, 6504.