TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A

PROJECT REPORT

ON

LEVERAGING MACHINE LEARNING AND COMPUTER VISION FOR MEDICAL IMAGING ANALYSIS

SUBMITTED BY:

NEEKA MAHARJAN (PUL077BCT050)

PUNAM ADHIKARI (PUL077BCT062)

AAYUSH PATHAK (PUL077BCT101)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

APRIL, 2024

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled **"LEVERAGING MACHINE LEARNING AND COMPUTER VISION FOR MEDICAL IMAGING ANALYSIS"** submitted by **Aayush Pathak**, **Punam Adhikari**, **Neeka Maharjan**, in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

.....
Supervisor
Full Name
Designation
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

.....
Internal examiner
Full Name
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

.....
External examiner
Full Name
Assistant Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

Date of approval:

Copyright

Acknowledgments

Firstly, we want to express our sincere appreciation to Assistant Professor **Bibha Sthapit** and Assistant Professor **Santosh Giri**. Their exceptional guidance, profound expertise, and consistent support have played a crucial role in shaping the path of this challenging research endeavor. Additionally, we extend our heartfelt thanks to Associate Professor **Dr. Jyoti Tandukar**, the Head of the Department of Electronics and Computer Engineering (DoECE) at Pulchowk Campus, for generously sharing insights and offering valuable feedback for our project. Our genuine thanks go to the Department of Electronics and Computer Engineering at Pulchowk Campus for providing us with the valuable opportunity to undertake this project. Their unwavering support, vigilant oversight, and insightful direction have been vital to the successful completion of this venture. Moreover, we deeply appreciate the meaningful interactions with our respected colleagues, friends, and peers. Their valuable contributions have consistently served as a driving force and source of inspiration throughout this project proposal. Their constructive feedback, intellectually stimulating discussions, and collaborative approach have significantly elevated the overall quality of our work.

Abstract

This project explores the application of machine learning (ML) and computer vision (CV) techniques to enhance medical imaging processes. Leveraging ML algorithms, particularly convolutional neural networks (CNNs), we developed a system capable of accurately detecting and classifying anomalies in medical images. The integration of CV methodologies enables automated analysis and interpretation of MRI data, facilitating quicker diagnosis and treatment planning. Through extensive experimentation and validation on diverse medical datasets, our approach demonstrates promising results, showcasing its potential to revolutionize diagnostic imaging practices in healthcare. In addition to the development of machine learning (ML) and computer vision (CV) algorithms, this project also prioritized the creation of a user interface tailored for healthcare professionals. The interface, designed with simplicity and efficiency in mind, allows users to interact seamlessly with the system for image analysis and diagnosis. Featuring intuitive navigation and real-time feedback, the interface enhances the accessibility and usability of the ML and CV functionalities, ultimately empowering clinicians to make informed decisions efficiently and effectively.

Keywords: *Computer Vision,Machine Learning, CNN, MRI*

Contents

List of Figures

List of Tables

List of Abbreviations

VGG-16	Visual Geometry Group
CNN	Convolutional Neural Network
MRI	Magnetic Resonance Imaging
ReLu	Rectified Linear Unit
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
FPR	False Positive Rate
AI	Artificial Intelligence
ML	Machine learning
CV	Computer Vision

1. Introduction

The main goal of our project is to develop a system for categorizing brain tumors. CNNs provide a promising method for precisely identifying and classifying various types of brain tumors from medical scans like MRI in the cutting-edge fields of medical imaging and machine learning. CNNs can evaluate intricate patterns and structures in the images by utilizing deep learning, which helps doctors with early detection, accurate diagnosis, and treatment planning.

1.1 Background

In the past, the majority of brain tumor classification techniques were manual, requiring skilled radiologists to decipher medical imaging scans and classify tumors according to visual inspection. But this method was subject to subjectivity and human error, which resulted in limitations and inconsistencies, particularly in high-volume clinical settings.

In an effort to address these issues, technological advancements have brought about the development of automated brain tumor classification systems. These systems use state-of-the-art methods like Convolutional Neural Networks (CNNs) to digitally analyze medical imaging data by using deep learning algorithms, pattern recognition, and image processing..

1.2 Problem statements

Before the development of technology, brain tumor classification was difficult because of manual techniques. Different radiologists had different interpretations, which resulted in different diagnosis and treatment strategies. Time-consuming procedures, a risk of human error, and limited access to expertise made the problem worse. The absence of standardization led to disparities amongst healthcare facilities. These difficulties brought to light the necessity for technological advancements to improve the precision and effectiveness of brain tumor classification.

1.3 Objectives

The primary objectives of this project are:

- Enable early detection of brain tumors.
- Build a user-friendly interface or application that allows healthcare professionals to interact with and interpret the results of the automated image analysis

- Predict prognosis based on tumor classification.
- Monitor tumor response to treatment over time.

1.4 Scope

The scope of this project includes the development and implementation of a multiclass brain tumor classification system using CNN and VGG16. It involves multiple phases like data collection, preprocessing, model training and validation.Evaluation metrics such as accuracy, precision, recall, and F1 score is used to assess the performance of each model. The preprocessing phase includes standardizing image sizes and pixel values, as well as converting categorical tumor classes into numerical labels. The model training and validation is the main portion of the project in which we trained and tested two separate models to address variability in our brain tumor classification system: one using a custom Convolutional Neural Network (CNN), and the other utilizing the VGG16 architecture.Consequently, we integrated the sequential model into our system for its superior performance.

2. Literature Review

2.1 Related work

Emrah Irmak's work on "Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework" is reviewed in the literature and offers a thorough summary of the approaches currently used in brain tumor classification[1]. The study looks at different methods for using CNNs and deep learning to analyze MRI images and classify different kinds of brain tumors. To enable precise tumor classification, Emrah Irmak investigates how well CNNs automatically extract discriminative features from MRI scans. Additionally, the study looks into methods like data augmentation and transfer learning to improve model performance, especially in situations where there isn't a lot of annotated data. The study does, however, recognize certain issues that require more research, such as the need for model predictions to be interpreted and class imbalance. It can be concluded from the literature that several techniques have been implemented for applying multiclassification. Though much has been accomplished in this area, there is still potential to improve. In addition, the performance parameters can be increased.

2.2 Related theory

Deep Learning for Computer Vision

CNNs were inspired by the visual system's structure, and in particular by the models of it proposed. The first computational models based on these local connectivities between neurons and on hierarchically organized transformations of the image are found, which describes that when neurons with the same parameters are applied on patches of the previous layer at different locations, a form of translational invariance is acquired. Yann LeCun and his collaborators later designed Convolutional Neural Networks employing the error gradient and attaining very good results in a variety of pattern recognition tasks.[2]

CNN Based Multiclass Brain Tumor Detection Using Medical Imaging

The suggested deep CNN model, which supports automated feature learning from brain MRIs, is made up of six learnable layers. The major purpose of developing such a network was to get a higher classification result while learning at a quicker rate than traditional DL models. Despite the lesser quantity of training data, the experiment results suggest that this model is successful.[3]

VGG-16

The Visual Geometry Group's convolutional neural network (CNN) architecture, VGG-16, has shown remarkable results when used for brain tumor multiclassification. By utilizing its 16-layer deep architecture, which includes convolutional and fully connected layers, VGG-16 is able to extract discriminative features from brain tumor MRI images with ease. Through the use of transfer learning techniques, which involve pre-training the network on large datasets like ImageNet, VGG-16 is able to accurately classify gliomas, meningiomas, and metastatic tumors among other types of brain tumors. Because of this, VGG-16 is a useful tool that helps doctors diagnose and treat brain tumors, which eventually improves patient outcomes.

3. Methodology

This project aims to develop Brain Tumor Multiclassification system using CNN and transfer learning.

3.1 Project Phases

1. Feasibility Study:

Among the array of methodologies available for multiclass brain tumor classification, we were drawn to CNN due to its capacity to autonomously discern discriminative features from images. Consequently, we opted to integrate CNN into our project via the creation of two distinct models: one crafted with a custom sequential design leveraging CNN, and another utilizing the established VGG16 architecture. We evaluated both models based on various metrics and compared their performance. This comprehensive analysis allowed us to determine the strengths and weaknesses of each approach, guiding our decision-making process for selecting the most effective model for our classification task. Among the array of methodologies available for multiclass brain tumor classification, we were drawn to CNN due to its capacity to autonomously discern discriminative features from images. Consequently, we opted to integrate CNN into our project via the creation of two distinct models: one crafted with a custom sequential design leveraging CNN, and another utilizing the established VGG16 architecture. Given the constraints of limited data and relatively modest performance requirements for the neural network, the training of this model will be conducted on google collab. Moreover, legal and ethical considerations are addressed to ensure compliance with data privacy and security regulations.

2. Requirement Analysis:

The requirement for this project are mentioned below:

a. Functional Requirements

I. Real-time Detection:

- The system should be capable of detecting multi-class brain tumors in real-time from medical images such as MRI scans or CT scans.

II. User Interface (UI):

- Provide a user-friendly interface for uploading medical images of brain scans.

III. Documentation:

- Include comprehensive documentation within the application, covering its usage, features, and technical aspects.

b. Non Functional Requirement

I. Performance:

- The system should process brain scan images with minimal latency to ensure swift detection.

II. Accuracy:

- Ensure that the detection system achieves high accuracy in identifying different classes of brain tumors.

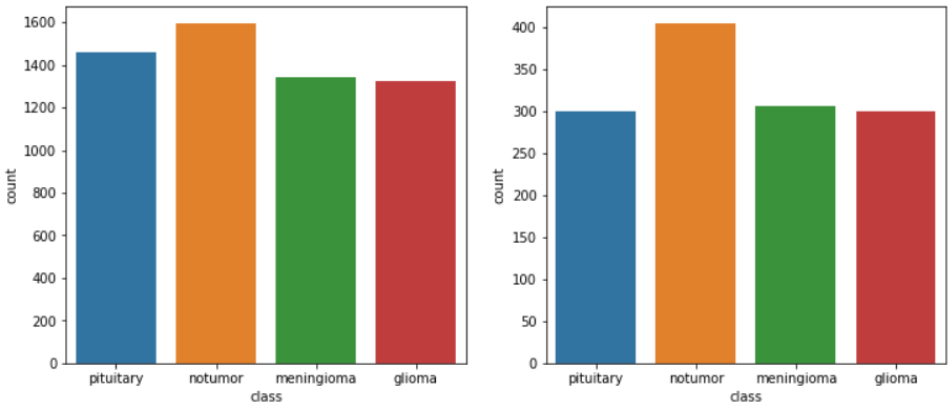
III. Interoperability and Portability:

- Design the system to be compatible with different operating environments commonly used in medical settings.

3. Data Collection:

In this project, we utilized the "BRAIN TUMOR MRI DATASET" sourced from IEEE Dataport, which comprises 7023 MRI images of human brain tumors categorized into four classes: glioma, meningioma, no tumor, and pituitary.

```
Number of images in training:  5712
Number of images in testing:  1311
```



4. Training and Evaluation

The models are trained using the classified and labeled training data. Following training, the model's performance is evaluated using a separate validation or test dataset, where metrics such as accuracy, precision, and recall are computed to assess its effectiveness in accurately classifying tumor types.

5. Integration and Deployment:

The trained model is integrated into the Multi-class Brain Tumor Classification System taking into account hardware requirements and real time processing capabilities.

The methodology combines advanced neural network techniques with comprehensive data handling, offering a reliable solution for authenticating signatures in diverse applications.

4. Experimental Setup

The following activities are involved in setting up the system:

4.1 Data Acquisition

For the purpose of classification, we divided the dataset into training and testing sets. The training set consists of 5712 images, while the testing set comprises 1311 images.The testing set, includes 300 images of glioma, 306 images of meningioma, 405 images of no tumor, and 300 images of pituitary. Similarly, within the training set, there are 1321 images of glioma, 1339 images of meningioma, 1595 images of no tumor, and 1457 images of pituitary. To visualize the distribution of images across different classes, we plotted bar charts representing the quantity of images in both the training and testing sets for each class. Additionally, we utilized Python code to automate the process of counting images within each class and visualizing their distribution using the Seaborn library. This approach allowed us to efficiently manage and analyze the dataset, ensuring that it meets the requirements for our classification task.

4.2 Pre Processing

During the preprocessing phase, we established a batch size of 32 to facilitate effective training and standardized image dimensions to 150x150 pixels. To make numerical operations easier, we converted our dataset from a Python list to a numpy array and aligned labels with images. We rescaled pixel values to a range of [0, 1] in order to get ready for model input. To add more unpredictability and avoid the model overfitting to particular image sequences, we also shuffled the dataset.We encoded the class labels by assigning unique numerical labels to each class (0 for pituitary, 1 for notumor, 2 for meningioma, and 3 for glioma).By optimizing our dataset, these steps improve the robustness and accuracy of CNN models used to classify brain tumors.

4.3 Experimental Environment

The training, validation and testing were conducted using Google Colab to suffice the requirement of a high-performance GPU. The system was tested and run on our local device.

4.4 Training and Testing

We built our classification models after finishing the preprocessing and data collection stages. We developed two distinct models: one based on the VGG-16 architecture and the other on convolutional neural networks (CNN). We aimed to categorize the dataset into four groups: glioma, meningioma, pituitary tumor, and absence of tumor (no tumor). We trained the network to recognize and extract features from the preprocessed MRI images by utilizing the CNN model. This allowed for precise classification of the various tumor types and the absence of pathology. In a similar vein, we sought to leverage the deep layers of the VGG-16 architecture in order to identify complex patterns and subtleties in the images, thereby enabling accurate classification across the designated tumor categories.

4.4.1 Convolutional Neural Network

Based on mathematical operations supporting their architecture, Convolutional Neural Networks (CNNs) are fundamental to the advancement of brain tumor classification from MRI images. The convolution operation, which is the foundation of CNNs, involves a filter moving across the input image and calculating the dot product at each position to extract features. The sum of element-wise multiplications between the filter and the corresponding area of the input image is the mathematical representation of this operation. After convolution, non-linearity is added to the network by the Rectified Linear Unit (ReLU) activation function, which sets negative values to zero while maintaining positive values. The feature maps are downsampled by subsequent pooling layers, which usually employ max pooling to preserve the most prominent features within the local regions. The maximum value within each pooling is chosen by max pooling mathematically.The feature maps are flattened and then passed through one or more fully connected layers in the network, where they are trained to identify complex patterns by adjusting weights and biases. CNNs can learn hierarchical representations of data thanks to the sequential arrangement of convolutional, activation, pooling, and fully connected layers. This results in robust models that can reliably identify brain tumors from MRI images.

The convolution operation is defined as:

$$(f * g)(x, y) = \sum_i \sum_j f(i, j) \cdot g(x - i, y - j)$$

where f represents the input image, g is the filter, and $(f * g)(x, y)$ denotes the output of the convolution operation at pixel (x, y).

The Rectified Linear Unit (ReLU) activation function is defined as:

$$f(x) = \max(0, x)$$

where x represents the input value.

Max pooling is commonly used and is defined as:

$$\text{MaxPooling}(x, y) = \max_{i,j} f(x + i, y + j)$$

This operation selects the maximum value within each pooling window.

A fully connected layer is represented as:

$$z = Wx + b$$

where W represents the weight matrix, x is the input vector, b is the bias vector, and z is the output vector.

4.5 VGG-16

The VGG-16 architecture, proposed by the Visual Geometry Group at the University of Oxford, is a deep convolutional neural network (CNN) known for its simplicity and effectiveness in image classification tasks. In the context of brain tumor multiclassification, VGG-16 can be employed to extract features from MRI images and classify them into different tumor types.

The VGG-16 architecture consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. Each convolutional layer is followed by a Rectified Linear Unit (ReLU) activation function, introducing non-linearity to the network. The convolutional layers are responsible for learning hierarchical features from the input images, capturing patterns at different levels of abstraction.

Mathematically, the convolutional layers in VGG-16 perform the convolution operation, where a filter (kernel) is applied to the input image to extract features.

After several convolutional layers, the feature maps are downsampled using max pooling layers to reduce spatial dimensions while retaining important features. The max pooling operation selects the maximum value within each pooling window, preserving the most salient features.

Finally, the output of the convolutional layers is flattened and passed through one or more fully connected layers. These fully connected layers perform a linear transformation followed by a non-linear activation function (commonly ReLU), which allows the network to learn complex patterns and relationships in the data.

In summary, VGG-16 offers a powerful framework for brain tumor multiclassification by effectively learning and extracting features from MRI images through its convolutional layers and making predictions through its fully connected layers, thus enabling accurate classification of different tumor types.

5. System design

The design of the Brain Tumor Multiclassification using CNN involves careful consideration of both architectural and functional aspects. The following subsections detail the key components of the proposed system.

5.1 System Architecture

The system architecture for Brain Tumor Multiclassification comprises of following aspects.

1. Data Input Module

After collecting and organizing the datasets into four folders corresponding to different tumor types, we constructed a model, such as a convolutional neural network (CNN) or VGG-16, for tumor classification. The model was trained using the organized dataset, achieving accurate results during training. Subsequently, its performance was evaluated on a separate validation dataset to ensure its effectiveness in real-world scenarios. This systematic approach ensures the development of a reliable tumor classification system, essential for aiding healthcare professionals in diagnosis and treatment.

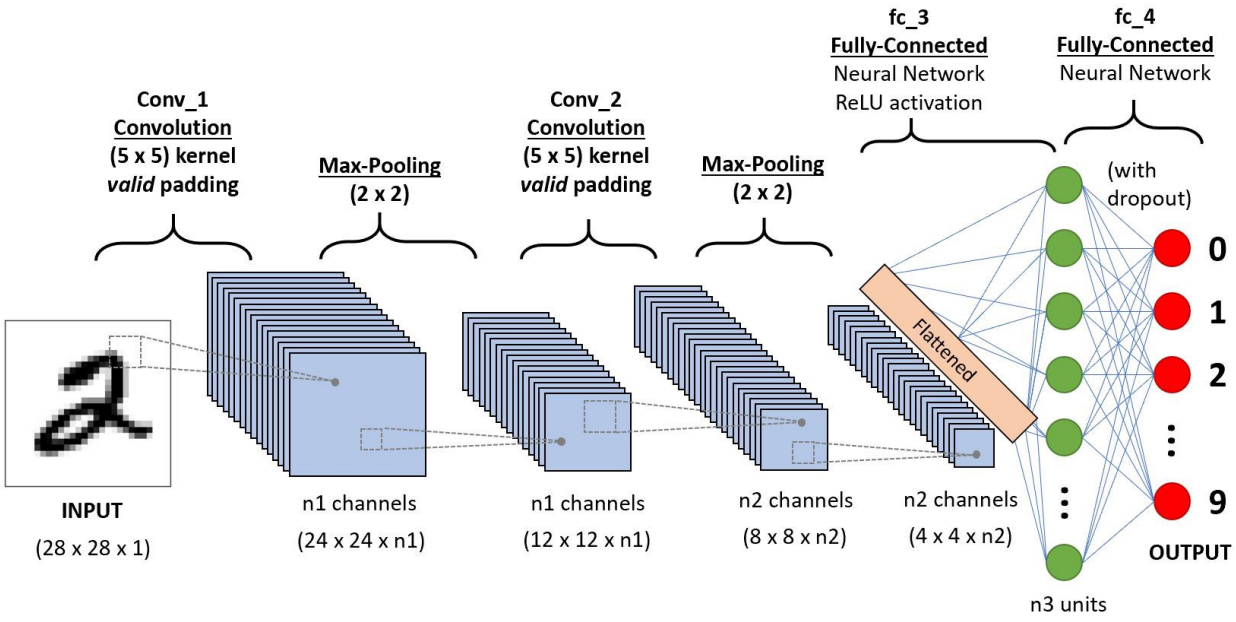
2. Preprocessing Module

To enable efficient training, we set a batch size of 32 during the preprocessing stage and standardized the image dimensions to 150x150 pixels. We aligned labels with images and transformed our dataset from a Python list to a numpy array to facilitate numerical operations. To prepare for model input, we rescaled pixel values to a range of [0, 1]. We also shuffled the dataset to increase unpredictability and prevent the model from overfitting to specific image sequences.By giving each class a distinct numerical label (0 for pituitary, 1 for notumor, 2 for meningioma, and 3 for glioma), we were able to encode the class labels.These steps enhance the robustness and accuracy of CNN models used in brain tumor classification by optimizing our dataset. A brief summary of this model is given below.

3. Convolutional Neural Network

It consists of three convolutional layers (*Conv1*, *Conv2*, and *Conv3*), each followed by max-pooling layers (*Max_Pool_1*, *Max_Pool_2*, and *Max_Pool_3*) for

downsampling. These convolutional layers employ rectified linear unit (ReLU) activation functions to introduce non-linearity and learn feature representations. The subsequent flattening layer converts the 2D feature maps into a 1D vector, preparing them for processing through two fully connected layers (*fc1* and *fc2*). The first dense layer (*fc1*) has 256 neurons with ReLU activation, facilitating the extraction of high-level features. Dropout regularization with a dropout rate of 0.5 is applied to mitigate overfitting during training. The final dense layer (*fc2*) with softmax activation produces class probabilities for the input images. With a total of 4,764,900 trainable parameters, the model is compiled using the Adam optimizer with categorical cross-entropy loss, making it suitable for multi-class classification tasks. This model achieved an accuracy of 97%.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Conv1 (Conv2D)               (None, 148, 148, 32)      896
_________________________________________________________________
Max_Pool_1 (MaxPooling2D)    (None, 74, 74, 32)        0
_________________________________________________________________
Conv2 (Conv2D)               (None, 72, 72, 32)        9248
_________________________________________________________________
Max_Pool_2 (MaxPooling2D)    (None, 36, 36, 32)        0
_________________________________________________________________
Conv3 (Conv2D)               (None, 34, 34, 64)        18496
_________________________________________________________________
Max_Pool_3 (MaxPooling2D)    (None, 17, 17, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 18496)             0
_________________________________________________________________
fc1 (Dense)                  (None, 256)               4735232
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
fc2 (Dense)                  (None, 4)                 1028
=================================================================
Total params: 4,764,900
Trainable params: 4,764,900
Non-trainable params: 0
```

4. VGG-16

The model architecture described is a variant of the VGG-16 CNN, modified for a specific classification task. It incorporates five blocks of convolutional layers, each followed by max-pooling layers, which progressively extract hierarchical features from input images. These features range from basic elements like edges and textures to more complex patterns like object shapes.

After the convolutional layers, a flatten layer reshapes the extracted features into a one-dimensional vector. Subsequently, two fully connected layers are added for further processing. The first dense layer has 256 neurons with ReLU activation, facilitating feature transformation and dimensionality reduction. A dropout layer with a rate of 0.5 is included to prevent overfitting during training. Finally, the output layer with 4 neurons and softmax activation generates class probabilities.

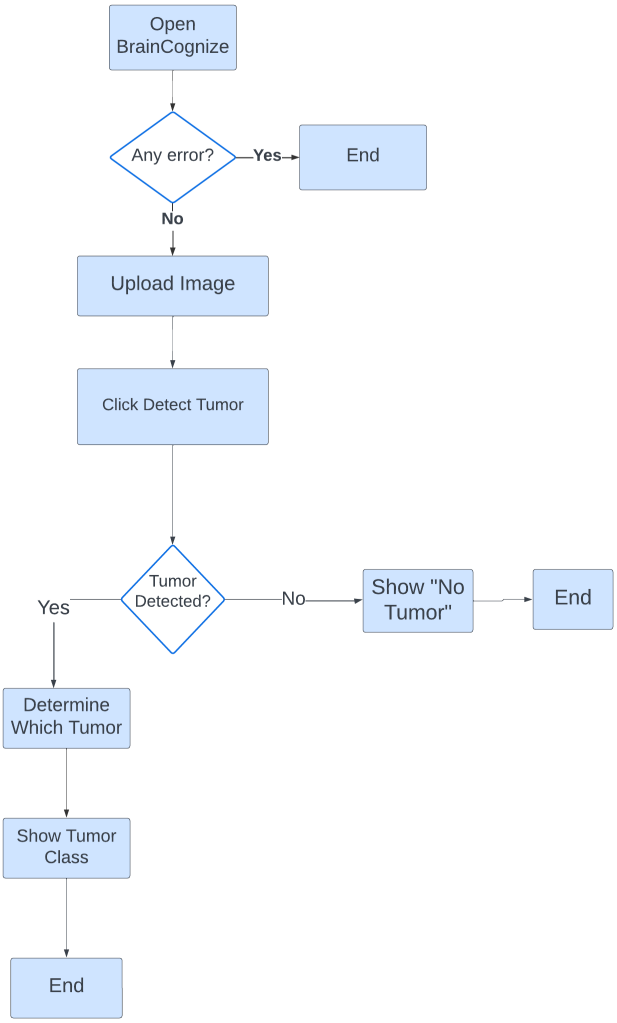
In terms of parameters, the model has a total of 16,813,124 parameters, of which 2,098,436 are trainable. The non-trainable parameters mainly consist of the pre-

trained weights in the convolutional layers. This architecture is well-suited for tasks requiring fine-grained classification of images into multiple classes. This model achieved an accuracy of 91%.

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 150, 150, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 150, 150, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 150, 150, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 75, 75, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 75, 75, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 75, 75, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 37, 37, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 37, 37, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
...
Total params: 16,813,124
Trainable params: 2,098,436
Non-trainable params: 14,714,688
```

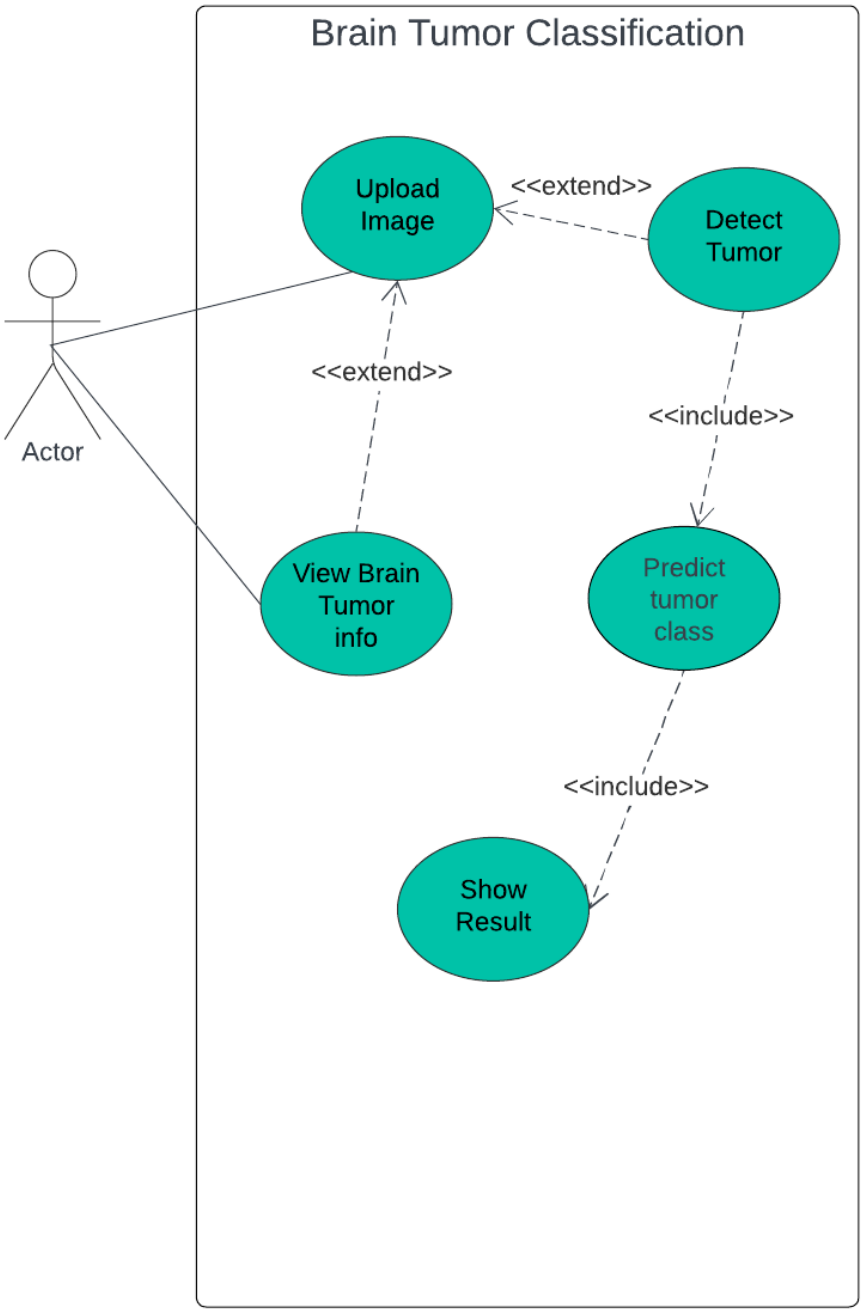
5.2 System Flow Diagram

The following diagram depicts the flow diagram of our system.


Open BrainCognize
Any error?
Yes
End
No
Upload Image
Click Detect Tumor
Tumor Detected?
No
Show "No Tumor"
End
Yes
Determine Which Tumor
Show Tumor Class
End

5.3 Use Case Diagram

The following figure depicts a use case diagram of our system.



5.4 Loss Function:

The categorical cross-entropy loss, denoted by $L(y, \hat{y})$, quantifies the disparity between the true probability distribution y and the predicted probability distribution $\hat{y}$, often derived from the softmax layer in neural networks, with y typically represented as a one-hot encoded vector. This loss function aims to minimize the difference between the distributions across all classes, where C signifies the number of classes. The probabilities y_i and $\hat{y}_i$ correspond to the i-th class in the true and predicted distributions, respectively.

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i)$$

5.5 Feature Extraction Module:

The convolutional layers are responsible for detecting patterns and features within the input images for both the models. As the data passes through successive convolutional layers and pooling layers, the network learns to identify increasingly complex features by combining and refining lower-level features.

5.6 Decision Module:

We utilize the softmax activation function to transform the raw output scores into probabilities. Subsequently, the class with the highest probability, determined by the 'argmax' operation provided by numpy, is selected as the predicted class representing the tumor category.

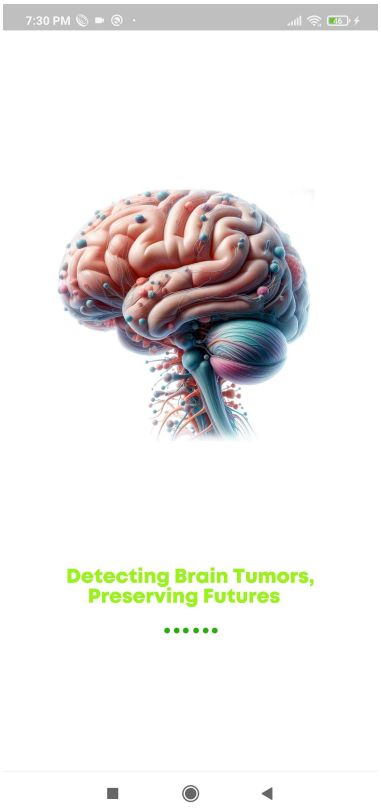
5.7 Integration Module:

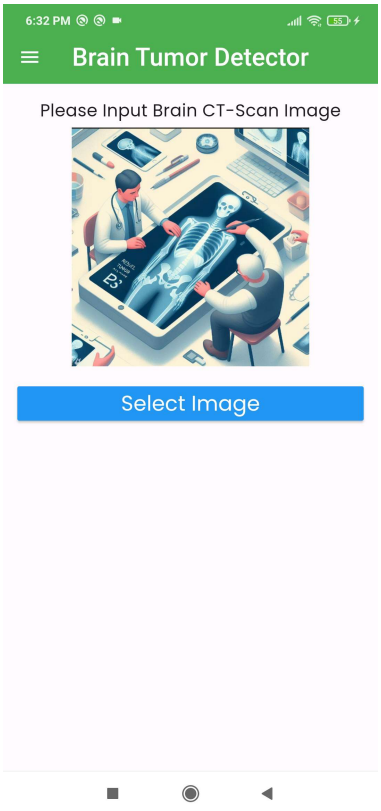
The sequential model, which achieved an accuracy of 97%, outperformed the VGG16-based model. Consequently, it was chosen to be integrated into the mobile application intended for use by the general public.

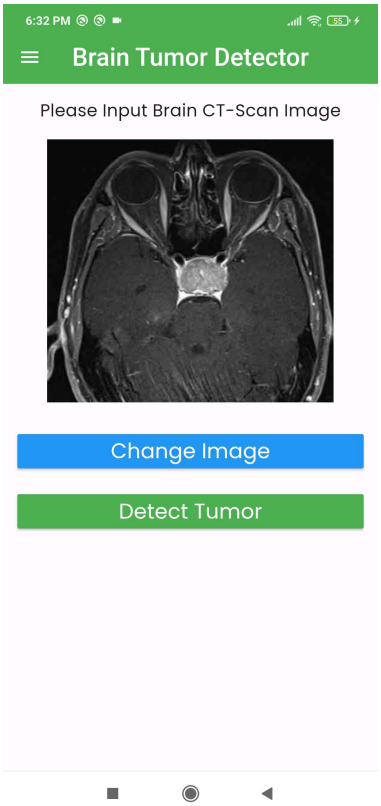
6. Results & Discussion

6.1 Software Implementation

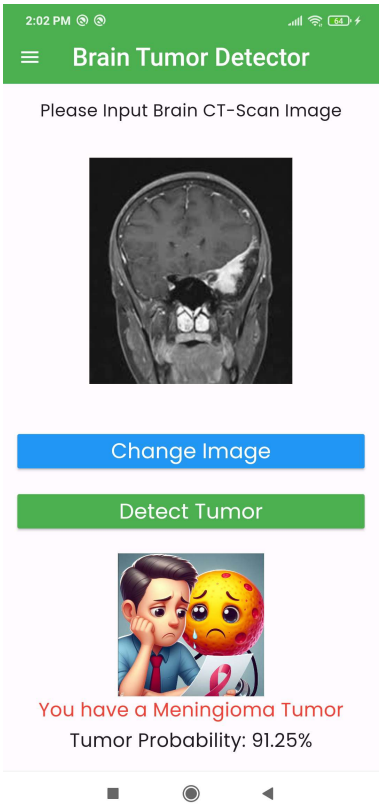
Our mobile application, built with Flutter for the frontend and Django for the backend, boasts a user-friendly interface tailored for effortless interaction. Users can easily upload MRI images for analysis, with the backend utilizing machine learning models to identify and classify brain tumors. Upon analysis completion, the application promptly displays results, indicating the presence and type of tumor detected (e.g., meningioma, pituitary, glioma) or confirming the absence of pathology. This streamlined process empowers users and healthcare professionals with accessible and accurate insights for diagnosis and treatment decisions.

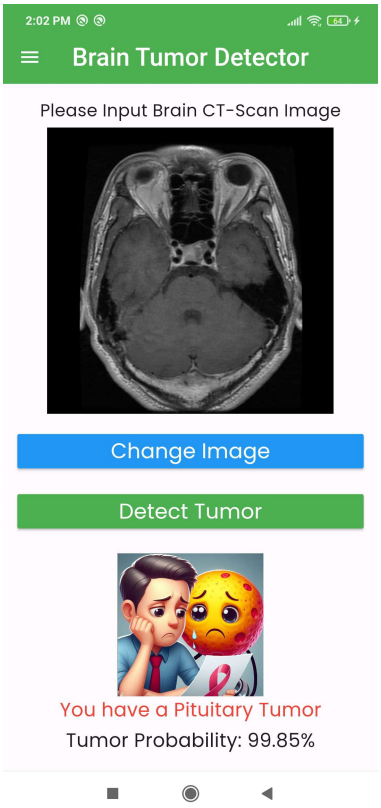


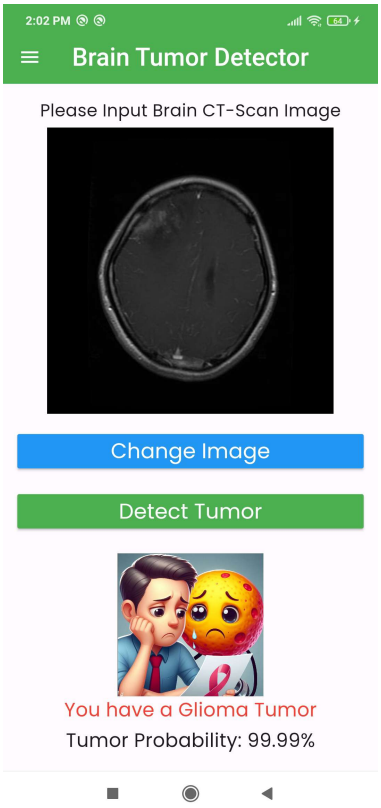


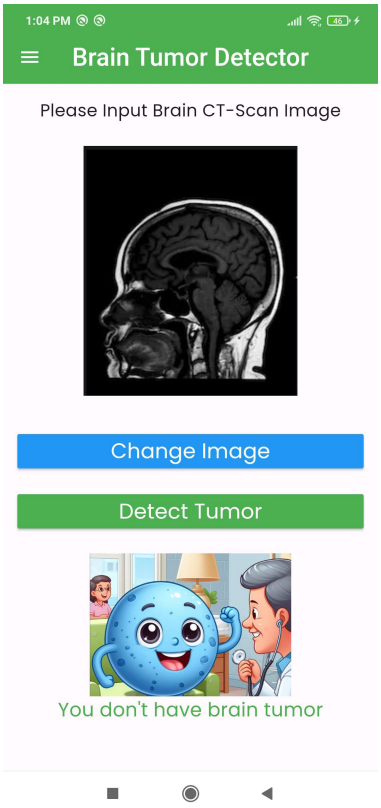


Then our system will predict output as shown below:









6.2 Sequential (CNN) Model vs VGG16-model

The two models are compared based on various evaluation metrics as the illustrations below will show. The table 6.1 below shows the hyperparameters and their corresponding values which are same for both the models.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	32
Number of Epoch 3	60
Optimizer	Adam
Activation Function	Softmax
Loss Function	Categorical Crossentropy

Table 6.1: Hyperparameters Tuning

	precision	recall	f1-score	support
pituitary	0.98	0.99	0.99	300
notumor	0.98	0.99	0.99	405
meningioma	0.94	0.97	0.95	306
glioma	0.99	0.94	0.96	300
accuracy			0.97	1311
macro avg	0.97	0.97	0.97	1311
weighted avg	0.98	0.97	0.97	1311

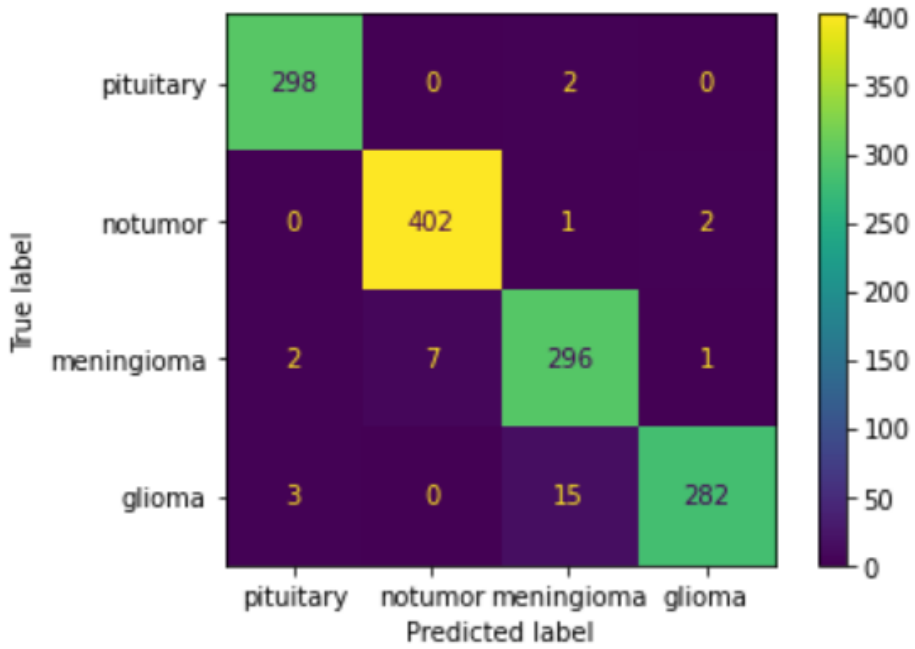


Figure 6.1: Sequential Model: Confusion Matrix and Evaluation Metrics

Above figure shows the confusion matrix and various evaluation metrics score for the Sequential Model.

	precision	recall	f1-score	support
pituitary	0.89	0.99	0.94	300
notumor	0.95	0.99	0.97	405
meningioma	0.82	0.86	0.84	306
glioma	0.98	0.77	0.86	300
accuracy			0.91	1311
macro avg	0.91	0.90	0.90	1311
eighted avg	0.91	0.91	0.91	1311

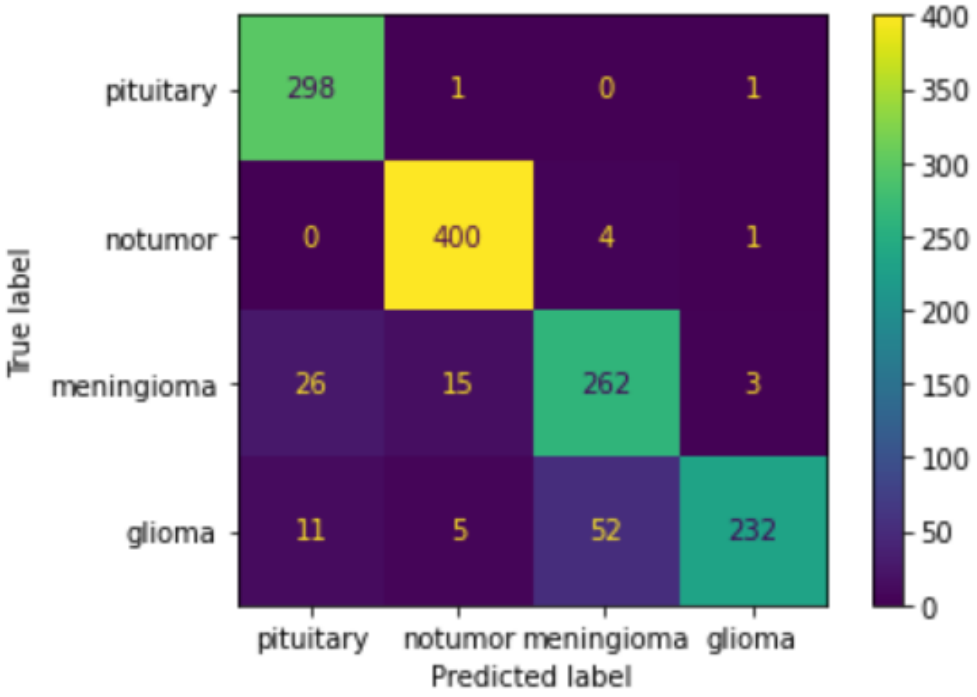


Figure 6.2: VGG16 Model: Confusion Matrix and Evaluation Metrics

Above figure shows the confusion matrix and various evaluation metrics score for the VGG16-based Model which shows lower accuracy than the Sequential (CNN) Model, above.

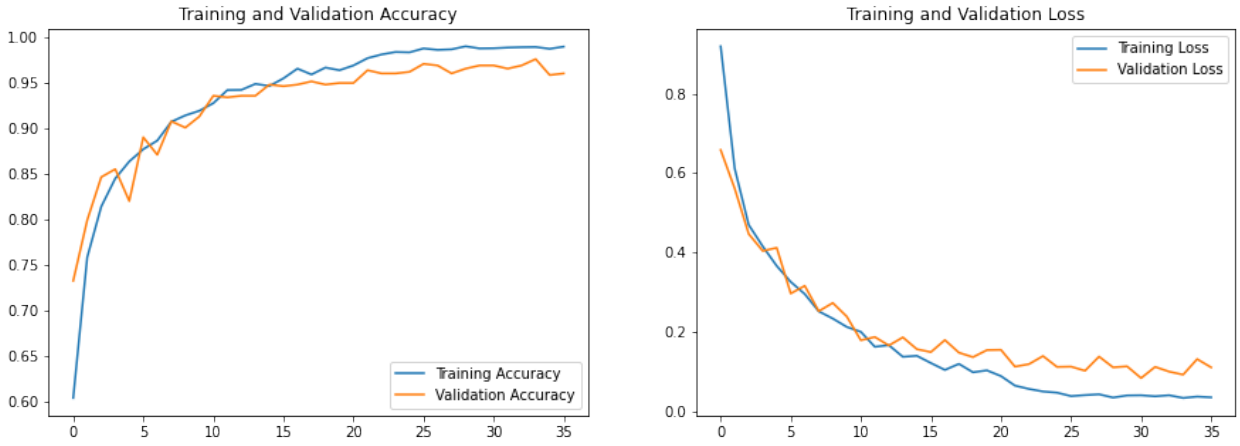


Figure 6.3: Loss and Accuracy for Sequential (CNN) Model

The figure above displays the training and validation accuracy, as well as the training and validation loss graphs for the Sequential Model

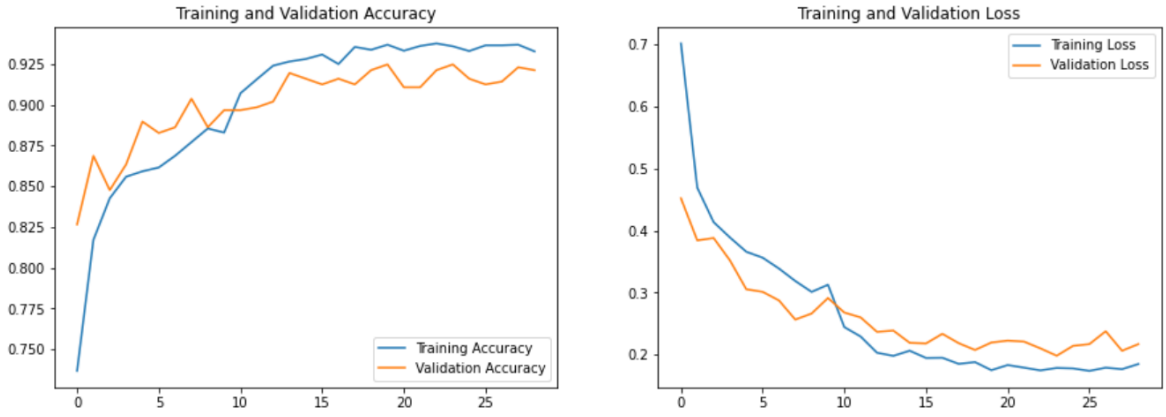


Figure 6.4: Loss and Accuracy for VGG16 Model

The figure above displays the training and validation accuracy, as well as the training and validation loss graphs for the VGG16 Model

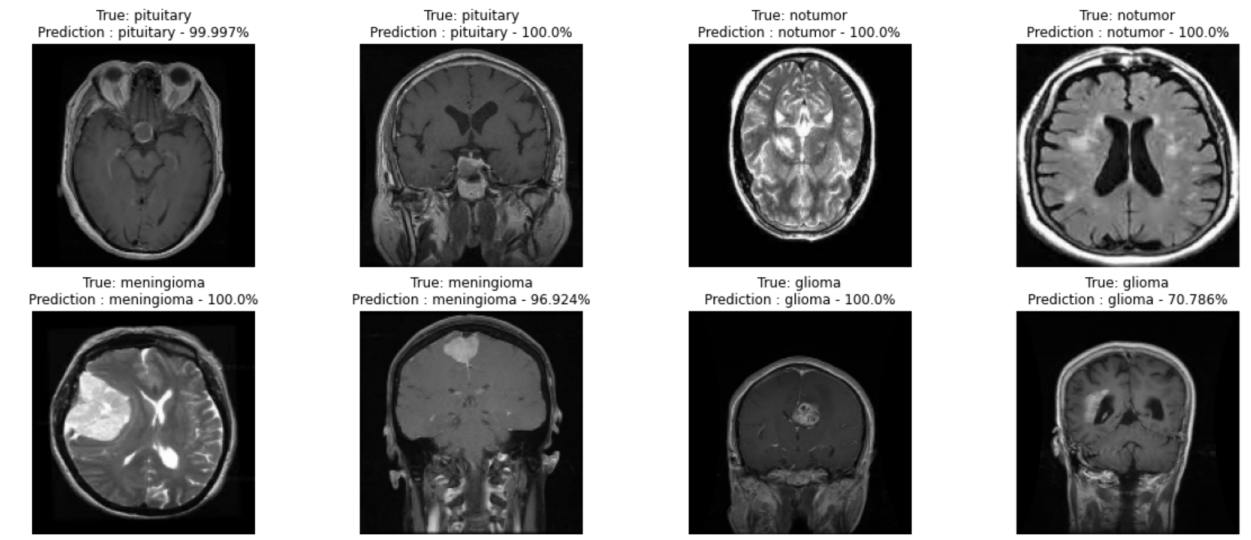


Figure 6.5: Sequential (CNN) Model Prediction

Above figure shows the predictions made by the Sequential (CNN) Model.

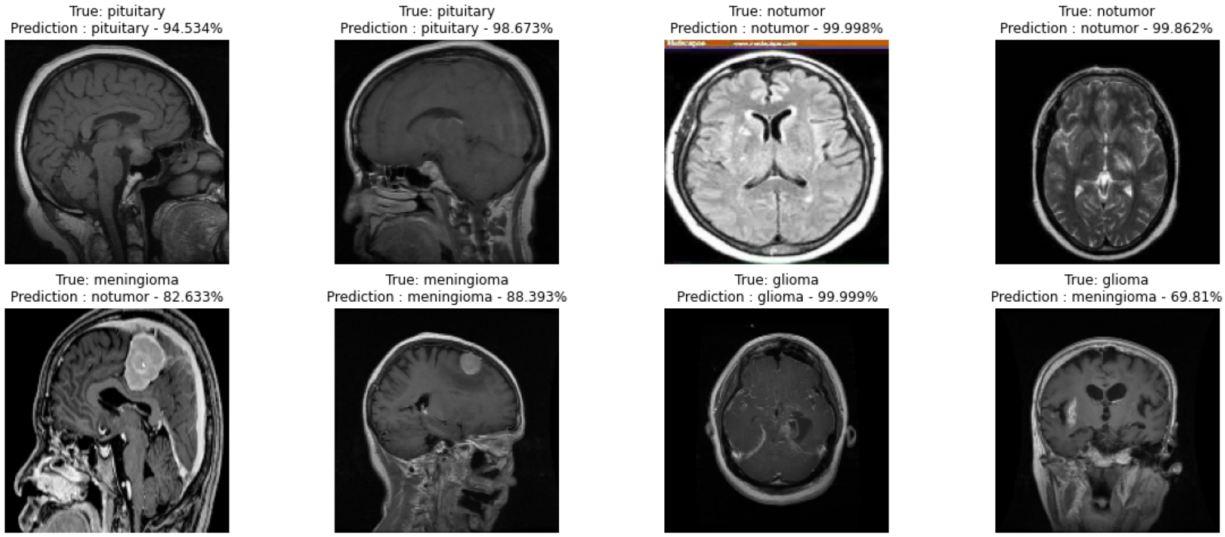


Figure 6.6: VGG16 Model Prediction

Above figure shows the predictions made by the VGG16 Model, which on comparision with the predictions made by the Sequential Model above, is slightly worse.

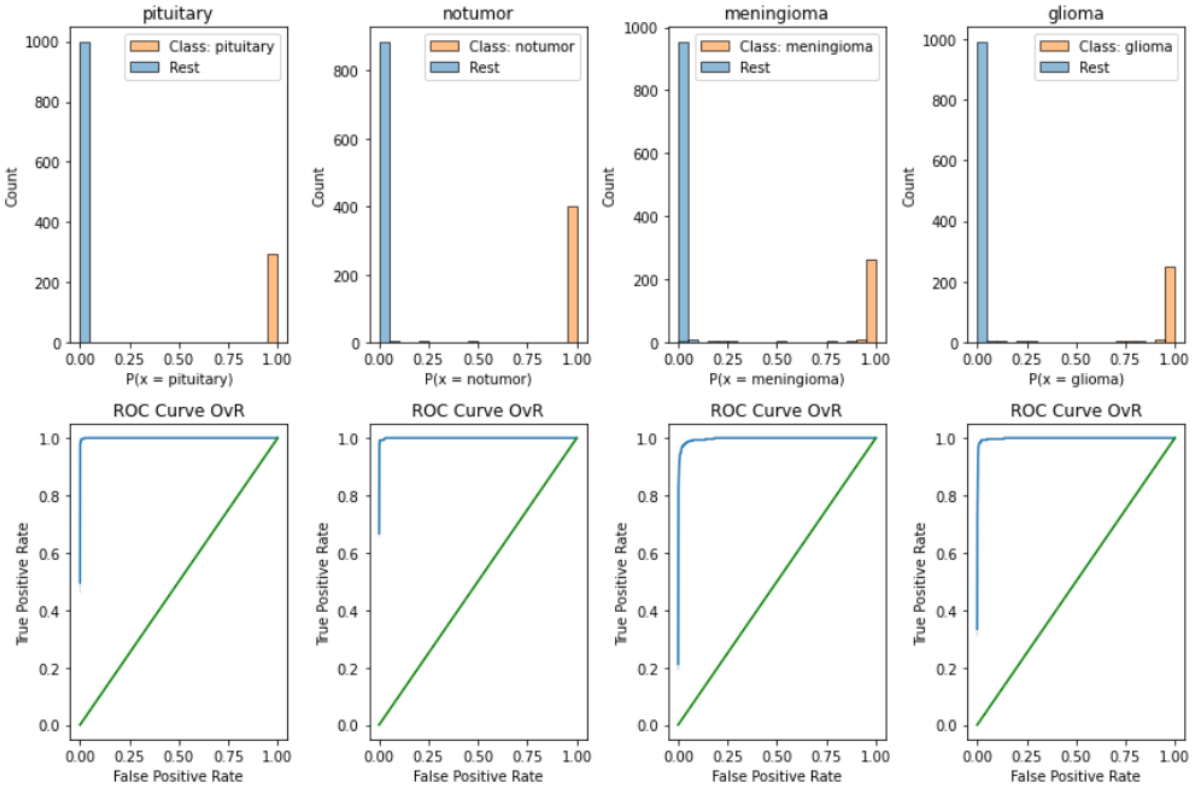


Figure 6.7: Probability Distribution and ROC Curves One vs Rest Plots for Sequential Model

Here, if the histograms overlap, then the model is struggling to classify that particular class. The more separate the histograms are, the better the ROC Curves are as well. From above, we can see that our CNN model is performing very well in classifying No Tumor and Pituitary Tumor.

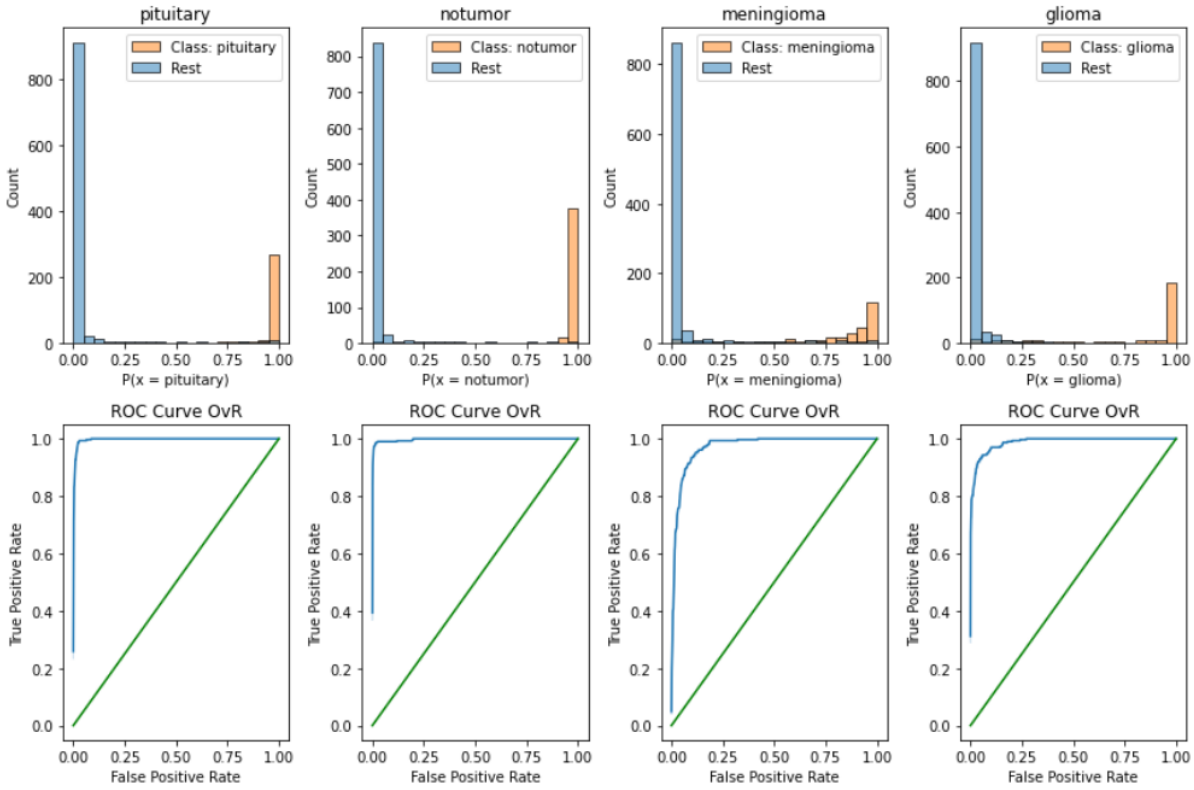


Figure 6.8: Probability Distribution and ROC Curves One vs Rest Plots for VGG16 Model

True Positive Rate(TPR) and False Positive Rate(FPR) are required for plotting ROC curves.

7. Conclusions

In conclusion, our project aimed to develop a brain tumor multiclassification system using advanced machine learning techniques, specifically convolutional neural networks (CNN) and the VGG-16 architecture. Leveraging Flutter for the frontend and Django for the backend, we created a mobile application that provides a user-friendly interface for uploading MRI images and receiving real-time predictions regarding the presence and type of brain tumor. Our model achieved impressive accuracies, with a 91% accuracy rate when utilizing the VGG-16 architecture and an even higher accuracy of 97% with a sequential CNN model. These results demonstrate the efficacy of our approach in accurately classifying four types of brain tumors: meningioma, pituitary, glioma, and no tumor. By combining cutting-edge machine learning algorithms with a streamlined mobile application, our project has the potential to significantly improve the efficiency and accuracy of brain tumor diagnosis, ultimately benefiting both healthcare professionals and patients alike.

8. Limitations and Future enhancement

Convolutional neural networks have shown great promise in brain tumor multiclassification, effectively distinguishing between various tumor types. But some limitations prevent it from operating at its best, so efforts must be made to address these issues and apply new improvements in the future. We can create a more reliable and efficient brain tumor multi-classification system by addressing these drawbacks and incorporating innovations, such as utilizing bigger and more varied datasets, improving model architectures, and investigating cutting-edge methods for feature extraction and classification.

8.1 Limitations

Data Dependency: The dataset obtained for brain tumor multiclassification was limited in sample size, which hindered the extraction of crucial features. Accessing real-time data was challenging due to confidentiality concerns, further complicating the dataset acquisition process. These constraints emphasize the need for alternative approaches, such as data augmentation and collaborations with medical institutions, to overcome dataset limitations and improve the system's effectiveness.

Computational Resources: Because of limited Resource constraints, such as limited computational power or memory of our computer system, we couldn't work with large datasets further increasing the complexity of the model architecture or the scale of the dataset that can be processed.

Ethical Considerations: Handling medical data requires strict adherence to ethical guidelines and privacy regulations, adding complexity and potential legal barriers to the project.

8.2 Future enhancement

Expansion of Dataset: Obtaining a larger and more diverse dataset could improve the model's ability to generalize and accurately classify brain tumors across different populations and tumor variations.

Real-Time Data Integration: Collaborating with medical institutions to access real-time data streams could enable the model to adapt to evolving tumor characteristics and improve its performance in clinical settings.

Advanced Model Architectures: Exploring more advanced neural network architectures, such as attention mechanisms or graph neural networks, could enhance the model's ability to capture complex relationships in brain tumor images.

Clinical Validation: Conducting rigorous clinical validation studies to assess the model's performance in real-world settings and its impact on clinical decision-making could validate its effectiveness and pave the way for clinical adoption.

User Interface Improvements: Enhancing the user interface of the mobile application to provide more intuitive features and functionalities could improve user experience and facilitate seamless interaction with the system.

References

[1]Irmak, E. (2021). Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework. Shiraz University.

[2]LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[3]Pallavi Tiwari, Bhaskar Pant, Mahmoud M. Elarabawy, Mohammed Abd-Elnaby, Noor Mohd, Gaurav Dhiman, Subhash Sharma, "CNN Based Multiclass Brain Tumor Detection Using Medical Imaging", Computational Intelligence and Neuroscience vol. 2022, Article ID 1830010, 8 pages, 2022

Appendices

Github Link: https://github.com/TheUndisput3d/Brain_Tumor_Multiclassification