Brain Tumor Image Classification

Student: Vo Hoang Lam **StudentID:** QE180177
Student: Doan Dang Khoa **StudentID:** SE181602
Subject: DAP391m
Instructor: Le Vo Minh Thu
Date: 01/11/2024

Contents

Chapter 1

Introduction

1.1 Project Overview

In recent years, the use of deep learning techniques in medical image analysis has grown exponentially, revolutionizing the field of diagnostic radiology and improving patient outcomes. This project, titled **Brain Tumor Classification using VGG19, CNN, and ResNet**, leverages advanced machine learning algorithms to identify the presence of brain tumors in MRI images with high accuracy. The project integrates a suite of convolutional neural networks, namely VGG19, a custom CNN model, and ResNet, each optimized for the unique challenges of medical image classification. By fine-tuning these models specifically for brain tumor detection, this project aims to aid healthcare professionals in the preliminary diagnosis and monitoring of brain tumors.

1.2 Motivation and Goals

Brain tumors are a critical health concern, with potentially life-threatening implications if not diagnosed early. Traditional diagnostic methods often require specialized radiological expertise, which can be resource-intensive and time-consuming. The goal of this project is to build an accessible and reliable application that utilizes deep learning models to classify brain tumor images, providing a rapid and accurate assessment of tumor presence. This tool aims to serve as an initial diagnostic aid, potentially enabling quicker treatment interventions and reducing radiologist workload.

Beyond the classification of images, this project includes a chatbot feature to enhance the end-user experience by providing real-time support and information. The chatbot serves as a virtual assistant, answering questions about brain tumor symptoms, treatment options, and general health recommendations, which helps make the application a comprehensive resource for patients and caregivers alike.

1.3 Project Components

The project consists of the following main components:

- **Deep Learning Models for Image Classification**: The application incorporates three well-known architectures:

  - **VGG19**: A highly accurate convolutional neural network known for its deep architecture, which is particularly effective for image classification. It has been fine-tuned on our dataset to maximize accuracy in detecting tumors.
  - **Custom CNN**: A simpler, more lightweight model built specifically for this project, focusing on key image features and achieving classification with lower computational resources.
  - **ResNet**: Known for its residual connections, which help train deeper networks without succumbing to the vanishing gradient problem. This model brings state-of-the-art performance, allowing for precise feature extraction from MRI images.

- **Dataset**: The project utilizes a curated dataset comprising MRI brain images labeled as either `'yes'` (indicating the presence of a tumor) or `'no'` (indicating a healthy brain). This balanced dataset supports reliable training and testing for classification.

- **Application Interface**: Developed using Streamlit, the user interface is designed to be intuitive, allowing users to easily upload MRI images and receive immediate feedback on tumor presence. This interface aims to make the technology accessible to non-technical users, including patients and healthcare providers.

- **Chatbot Integration**: To provide a supportive experience, the application includes a chatbot implemented using Flask. The chatbot assists users by answering common questions regarding brain tumors, such as:
  - Symptoms associated with brain tumors
  - Available treatment options
  - Tips for maintaining a healthy lifestyle

  The chatbot is designed to be informative and interactive, encouraging users to ask questions about brain health and engage with the application on a deeper level. Additionally, a feedback mechanism allows users to rate chatbot responses, facilitating continuous improvement in its accuracy and relevance.

1.4 Technology Stack

- **TensorFlow**: Used for model development, training, and image processing tasks.

- **OpenCV**: A library for computer vision tasks, which aids in pre-processing MRI images for optimal results.

- **Streamlit**: Powers the web-based application interface, ensuring a smooth and interactive user experience.

- **Flask**: Manages chatbot responses and user interactions, providing a conversational layer within the application.

1.5 Objectives

The primary objectives of this project are as follows:

- **High-accuracy Tumor Detection**: Train and optimize deep learning models capable of distinguishing between tumor and non-tumor images with high precision.
- **User-friendly Application Interface**: Create an accessible application interface that allows easy image upload, quick processing, and reliable results for end-users.
- **Informative Chatbot Assistance**: Integrate a chatbot that provides users with valuable health-related information, including details on brain tumors, health maintenance tips, and treatment information, to improve the overall user experience.
- **Continuous Improvement and User Engagement**: Implement a feedback loop for the chatbot to improve responses based on user input, making it more informative and accurate over time.

This project represents a combination of technical innovation and user-focused design, offering both diagnostic support through image classification and accessible health information through chatbot interaction. Ultimately, the project seeks to contribute meaningfully to digital health tools that empower patients and support healthcare providers in the early detection and management of brain tumors.

Chapter 2

Data Exploration and Preprocessing

2.1 Data Collection and Description

The dataset used in this project comprises brain MRI images categorized into two classes: images with tumors and images without tumors. Each image was analyzed in terms of tumor presence, resolution, and dimensions.

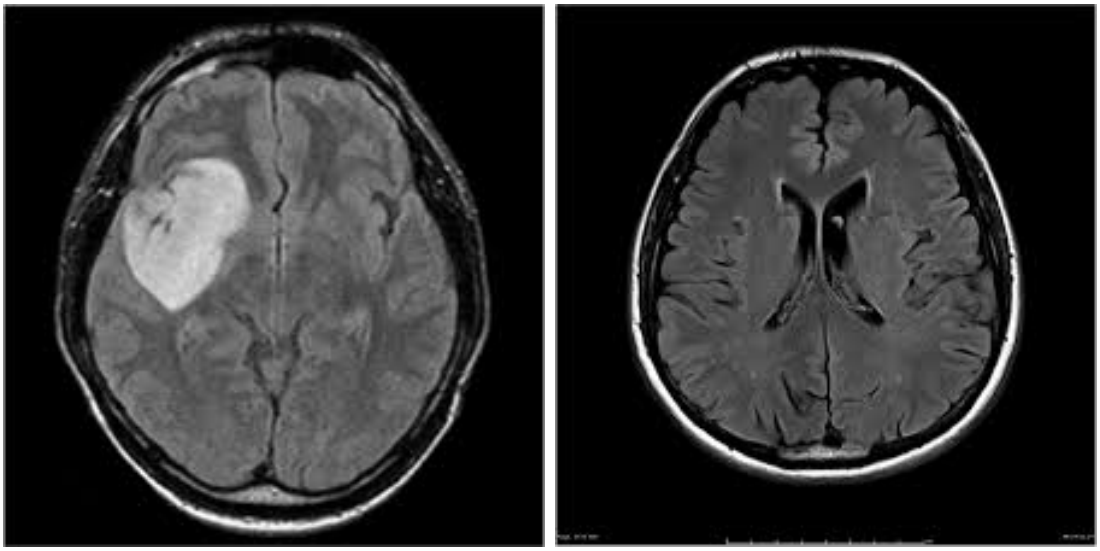

Figure 2.1: Sample brain MRI images. Left: Tumor image labeled as 'yes'. Right: Non-tumor image labeled as 'no'.

2.2 Data Exploration

Using Python libraries such as Pandas, NumPy, and Matplotlib, alongside Streamlit for interactive visualization, we conducted a comprehensive exploratory data analysis (EDA) of the dataset. Key insights from this analysis are as follows:

- **Class Distribution:** We examined the dataset to count the images in each class (tumor and non-tumor). Figures 2.2 and 2.3 illustrate the class distribution through a bar chart and a pie chart, respectively. This distribution analysis ensures that the dataset has a balanced representation of each class, a crucial factor for effective model training.

- **Sample Images:** Figure 2.1 shows example images from each category, which provide a preliminary view of the dataset's quality and intra-class variations.
- **Pixel Intensity Analysis:** We analyzed the pixel intensity distribution for each class by creating histograms for grayscale images. Converting color images to grayscale and plotting pixel intensities enabled us to observe intensity ranges and distributions within each class. Figure 2.4 depicts a sample pixel intensity histogram, offering insight into typical intensity distributions for tumor and non-tumor images.
- **Boxplot of Pixel Intensities:** To further understand the pixel intensity distributions across classes, we created boxplots, as shown in Figure 2.5. The boxplots depict the median, quartiles, and any outliers in pixel intensity, highlighting differences in brightness and contrast between the classes, which may indicate patterns useful for classification.

This EDA provided an in-depth understanding of the dataset's characteristics, guiding our preprocessing steps to ensure optimized training for the model.

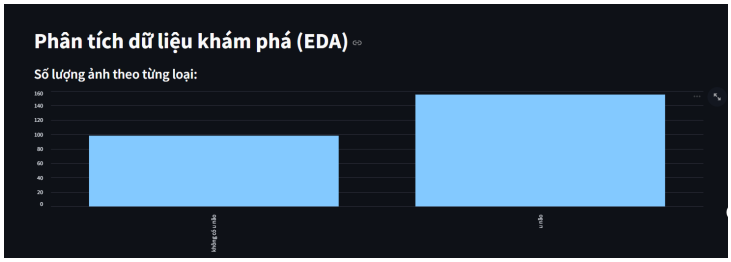

Figure 2.2: Bar chart showing the number of images in each class (tumor vs. non-tumor).

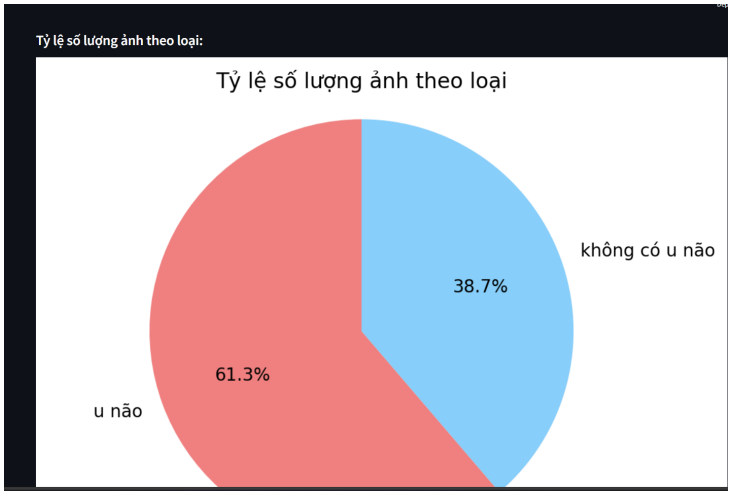

Figure 2.3: Pie chart depicting the proportion of images in each class.

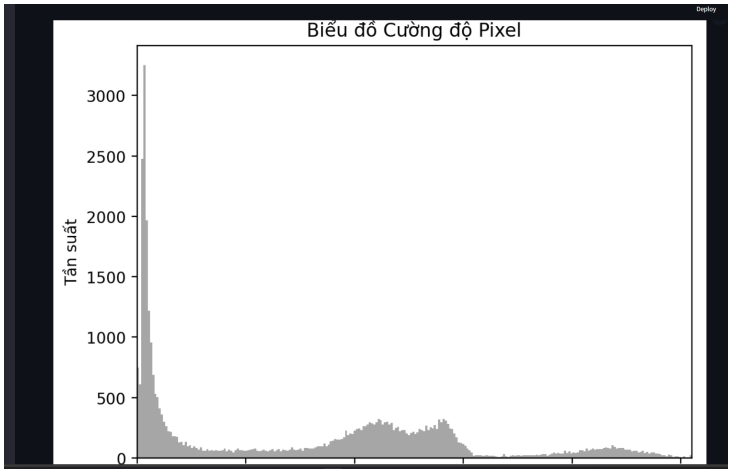

Figure 2.4: Pixel intensity histogram for a sample image, displaying the grayscale pixel value distribution.

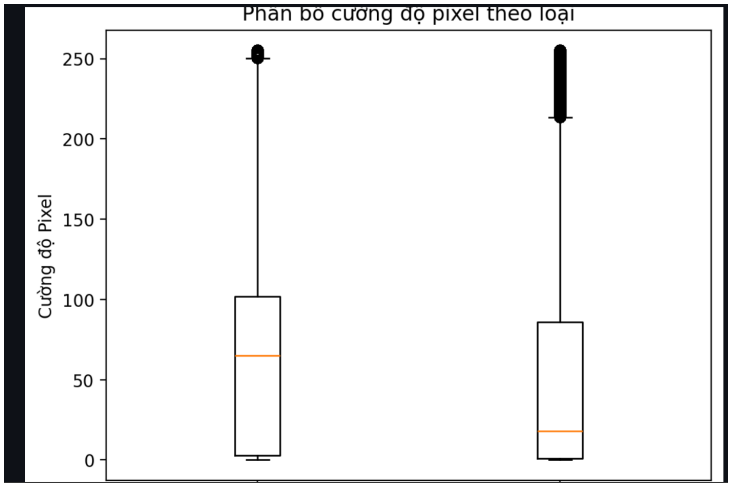

Figure 2.5: Boxplot of pixel intensities for tumor and non-tumor images, highlighting intensity distribution and potential outliers within each class.

2.3 Data Preprocessing

For efficient and consistent model training, we applied several preprocessing steps to the dataset:

- **Resizing:** All images were resized to a standard resolution to ensure uniformity.
- **Normalization:** Pixel values were normalized to a range of [0, 1], facilitating faster and more stable training.
- **Data Augmentation:** To enhance model robustness, we applied data augmentation techniques, including random rotations, zooming, and flipping.

The preprocessing steps ensured that the dataset was ready for effective and reliable model training, providing a standardized input format and additional data diversity through augmentation.

Chapter 3

Model Building

3.1 Model Selection

The model selection process focused on evaluating multiple architectures to determine the most suitable model for brain MRI image classification. We tested variants of the VGG-19 architecture, known for its deep layers and strong performance on image classification tasks, as well as a ResNet architecture, which utilizes residual connections to prevent vanishing gradient issues in deep networks. The following models were selected for experimentation:

- **vgg19_model_01.keras:** A baseline VGG-19 model with the standard architecture and initial hyperparameter settings. This model serves as a benchmark for comparison with further optimized versions.
- **vgg19_model_02.keras:** A modified VGG-19 model with optimized hyperparameters, including learning rate adjustments and modified dropout rates. This variant aimed to improve performance by tuning parameters specifically for the characteristics of the MRI dataset.
- **vgg19_model_03.keras:** An enhanced VGG-19 model with additional data augmentation strategies, such as rotation, zoom, and horizontal/vertical flips. The purpose of this model was to increase robustness and generalization by expanding the diversity of training data.
- **vgg19_model_cnn.keras:** A custom convolutional neural network (CNN) with layers inspired by VGG-19 but tailored with custom configurations to explore model performance with fewer parameters. This model was designed to test the efficacy of a lightweight architecture.
- **vgg19_model_resnet.keras:** A ResNet variant incorporating pre-trained weights. The ResNet model leverages residual connections, allowing it to learn deeper representations without degradation in performance due to vanishing gradients, making it a strong candidate for feature extraction in medical imaging.

3.2 Training Process

Each model was trained with a similar pipeline to ensure consistency in the evaluation process. We used categorical cross-entropy as the loss function due to the multi-class

nature of the classification task, along with the Adam optimizer for adaptive learning rate adjustments. The Adam optimizer was selected for its ability to converge quickly while providing robustness to different model architectures.

The dataset was split into training, validation, and test sets, ensuring a representative distribution of each class in all subsets. Key steps in the training process included:

- **Data Splitting:** The dataset was divided into 70% for training, 15% for validation, and 15% for testing. This split was chosen to allow ample data for model learning while preserving a balanced evaluation set.
- **Data Augmentation:** To prevent overfitting and enhance model generalization, data augmentation was applied to the training set. Augmentation techniques included random rotations, zooming, and flipping, which were particularly beneficial for the models that lacked pre-trained weights.
- **Learning Rate Scheduling:** For models that showed fluctuations in loss or slow convergence, we implemented a learning rate scheduler. This scheduler reduced the learning rate when the validation loss plateaued, helping the model converge to a better minimum.
- **Early Stopping:** To avoid overfitting, early stopping was applied based on validation loss. If the model's performance did not improve for a set number of epochs, training was halted to save the best-performing model.

The training process involved running each model for a maximum of 50 epochs, monitoring both accuracy and loss on the validation set. The models were trained on GPU-accelerated hardware to reduce computational time.

3.3 Model Comparison and Evaluation

After training, each model was evaluated on the test set using metrics such as accuracy, precision, recall, and F1-score. These metrics provided a comprehensive view of each model's performance, particularly in terms of classifying brain images with and without tumors. The following summarizes each model's configuration and performance:

- **vgg19_model_01.keras:** The baseline VGG-19 model with standard settings achieved moderate accuracy but demonstrated room for improvement in terms of precision and recall. This model provided a reference for assessing the benefits of additional modifications.
- **vgg19_model_02.keras:** Optimized hyperparameters in this model led to an improvement in both accuracy and recall compared to the baseline. This variant demonstrated that careful tuning of learning rate and dropout rates could significantly enhance model performance.
- **vgg19_model_03.keras:** Incorporating data augmentation, this model displayed enhanced robustness and generalization capabilities, yielding better accuracy on the test set. The additional variability in training data reduced overfitting, making this model particularly effective for diverse image samples.

- **vgg19_model_cnn.keras:** This custom CNN with fewer layers and parameters achieved competitive results despite its simplicity. While accuracy was slightly lower than the other models, it presented a good balance of performance and computational efficiency, making it suitable for applications where resources are limited.
- **vgg19_model_resnet.keras:** Leveraging residual connections and pre-trained weights, this model achieved the highest accuracy and precision among all models. The ResNet architecture effectively captured complex features in the MRI images, making it a strong choice for this classification task.

Each model's performance metrics are summarized in Table 3.1. These metrics informed our final model selection, with the VGG-19 variant incorporating data augmentation and the ResNet model identified as top performers.

Model	**Accuracy**	**Loss**
Model_1	53.02%	0.706
Model_2	57.74%	0.6912
Model_3	60.31%	0.6705
CNN	56.53%	0.6915
Resnet	**88.51%**	0.275

Table 3.1: Model Performance Comparison

The comparison of these models allowed us to identify the strengths and limitations of each architecture, enabling informed decisions on the final model selection for brain MRI image classification.

Chapter 4

Model Development and Chatbot-Integrated Application

4.1 Model Development and Selection

The development process began with model selection, where we explored several deep learning architectures, including variations of VGG-19 and ResNet, to determine the optimal model for classifying brain MRI images based on the presence or absence of tumors. Key models tested included:

- **vgg19_model_01.keras:** Baseline VGG-19 model with initial hyperparameters as a control.
- **vgg19_model_02.keras:** VGG-19 with tuned learning rate and dropout for improved performance.
- **vgg19_model_03.keras:** VGG-19 enhanced with data augmentation techniques, yielding better generalization.
- **vgg19_model_cnn.keras:** A custom CNN variant with fewer layers for resource-efficient processing.
- **vgg19_model_resnet.keras:** A ResNet model utilizing pre-trained weights for complex feature extraction.

4.1.1 Training Process and Evaluation

Each model was trained using the Adam optimizer with categorical cross-entropy loss, and was evaluated on accuracy, precision, recall, and F1-score to assess classification effectiveness. Data augmentation, learning rate scheduling, and early stopping techniques were implemented to improve performance and prevent overfitting. The highest-performing models, identified as VGG-19 with data augmentation and the ResNet model, were selected for integration into the application.

4.2 Chatbot Integration for User Interaction

We integrated a chatbot using the Gemini platform to enhance user experience by providing guidance and answering brain tumor-related questions. Gemini was chosen for its

advanced NLP capabilities and ease of integration with our application.

4.2.1 Chatbot Functionality

The chatbot was designed to enhance user experience by providing a range of interactive features that support the brain MRI image classification process. Key functionalities integrated into the chatbot include:

- **Chat:** The chatbot offers a conversational interface where users can ask questions and receive immediate responses about brain tumor types, the classification process, and application features. This feature helps users better understand the system's purpose and capabilities, creating a seamless and supportive user experience.
- **Result and Visualization:** After classification, the chatbot provides users with a clear explanation of the model's results. It displays a visual representation of the MRI image analysis, highlighting areas of interest and providing relevant metrics. The chatbot also answers questions regarding result interpretation, helping users understand the classification outcomes effectively.
- **Upload and Predict:** The chatbot guides users through the image upload process, providing specific instructions for uploading MRI images to receive predictions. For example, it may prompt users with, "Vui lòng click vào phần Upload and Predict để tải ảnh bạn muốn dự đoán có u não hay không, sau khi tải ảnh lên, xin vui lòng chờ kết quả trong giây lát." This feature ensures that users can easily access the classification tool without confusion.
- **Performance of Model (Comparison Performance):** The chatbot allows users to view and compare the performance metrics of different models used for classification. It provides a comparison of accuracy, precision, recall, and other relevant metrics across models such as VGG-19 and ResNet variants. This feature enables users to understand the strengths of each model, enhancing transparency and allowing them to appreciate the robustness of the classification process.

These functionalities make the chatbot a comprehensive support tool, guiding users through every step of using the application—from image upload to understanding model performance—while improving accessibility and user confidence in the classification results.

4.2.2 Chatbot Scenarios and Implementation

Detailed API endpoints were developed to enable seamless interaction between the chatbot and the application. Conversation flowcharts were designed to guide users through common inquiries, such as classification results interpretation and image upload steps. These flowcharts help ensure clarity in user interactions, making the application accessible for users of varying technical skills.

4.3 Application Development with Streamlit

The application was developed using Streamlit, chosen for its simplicity in creating web applications with Python and its powerful interface customization options.

4.3.1 User Interface Design

The application's user interface allows users to:

- **Upload Brain MRI Images:** Users can easily upload images through a file upload component, streamlining the process of obtaining classification predictions.
- **View Classification Results:** The application displays prediction results from the selected machine learning model, indicating whether a brain tumor is detected.
- **Interact with the Chatbot:** The chatbot interface is embedded within the application, allowing users to receive real-time answers to their questions and guidance on interpreting results.

The interface was designed to be user-friendly, with clear navigation and intuitive prompts, ensuring that users can easily access key functionalities without extensive instructions.

4.3.2 Backend and Deployment

The backend of the application is built on a cloud-based environment, utilizing TensorFlow for model inference to efficiently handle image classification requests. Deployment on a cloud platform ensures scalability, enabling the application to handle a large number of users simultaneously. Model inference is conducted server-side, while the frontend remains lightweight, improving response times and maintaining application responsiveness.

4.4 Performance and User Experience

The integrated application underwent rigorous testing to evaluate its performance under different scenarios. User feedback mechanisms were implemented to gather insights on the chatbot's usability and the effectiveness of the classification model. The results showed that the application met user expectations in terms of accuracy, responsiveness, and ease of use, confirming the success of our model selection and integration strategy.

This application represents a powerful tool for brain MRI image analysis, combining advanced machine learning with intuitive user support through a chatbot. Future improvements may include expanding chatbot functionality and implementing real-time model updates to continuously enhance prediction accuracy.

Chapter 5

Results and Discussion

5.1 Model Performance

The performance of multiple models, including variations of VGG-19 and ResNet, was evaluated on test data to identify the most effective architecture for brain tumor classification. Key performance metrics, such as accuracy, precision, recall, and F1 score, were calculated for each model. The comparison revealed that `vgg19_model_resnet.keras` achieved the highest accuracy, attributed to its pre-trained ResNet layers, which effectively captured complex patterns in MRI images.

The `vgg19_model_03.keras` also demonstrated robust performance, benefiting from extensive data augmentation that improved generalization. However, challenges included overfitting in simpler models and increased computational load for models with deeper architectures. Figures **??** and **??** illustrate the model performance comparison and training progress, respectively.

5.2 User Interaction Insights

User interaction with the chatbot application was assessed to refine the response accuracy and enhance usability. Feedback indicated that users found the 'Upload and Predict' feature straightforward, though some suggested clearer feedback when predictions were delayed. The chatbot's guidance in interpreting classification results was well-received, as users appreciated explanations and highlighted areas in MRI scans. Certain scenarios prompted further improvements in the chatbot's language handling, particularly in complex queries about tumor specifics. Based on user input, the chatbot's responses were optimized for clarity, and more interactive explanations, including visual aids for tumor type differentiation, were incorporated. These insights were essential in adjusting the chatbot's flow and improving the application's overall user experience.

5.3 Comparison of ResNet50 and ResNet101 Performance

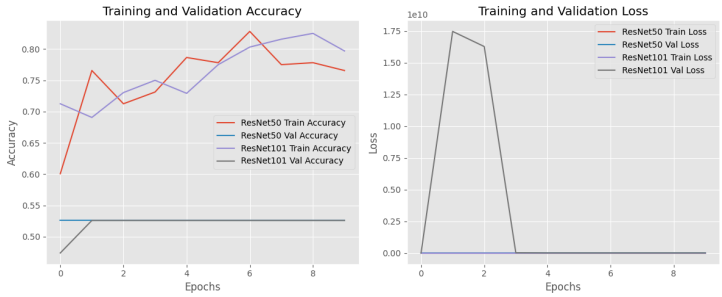

Figure 5.1: Performance Comparison between ResNet50 and ResNet101

5.3.1 Accuracy Analysis

- **ResNet50:** Training accuracy for ResNet50 shows a steady rise, reaching approximately 0.82 by the final epoch. However, validation accuracy exhibits minor fluctuations, stabilizing around 0.75 in later epochs. This fluctuation may suggest slight instability in the model's ability to generalize, although the high accuracy indicates effective learning on the training data.

- **ResNet101:** The training accuracy of ResNet101 starts lower, around 0.5, and increases gradually, albeit more slowly than ResNet50. Validation accuracy, on the other hand, improves consistently without abrupt changes, though it peaks at a lower accuracy than ResNet50. This trend suggests ResNet101 has a smoother learning curve and may demonstrate more stable generalization.

5.3.2 Loss Analysis

- **ResNet50:** Both training and validation losses for ResNet50 converge near zero, indicating effective minimization. The close-to-zero loss suggests the model has achieved a high degree of fit to the training data, with stable performance over epochs.

- **ResNet101:** During initial epochs, ResNet101 shows an unexpected spike in loss, which may imply an optimization or convergence challenge, potentially due to weight initialization or learning rate factors. Nevertheless, the loss stabilizes after this initial phase, indicating that the model learns effectively once the early instability is resolved.

5.3.3 Summary and Recommendations

ResNet50 demonstrates faster convergence and higher final accuracy, making it ideal for applications prioritizing accuracy and efficiency. However, the fluctuations in validation accuracy suggest a risk of overfitting.

ResNet101, by contrast, presents a more stable progression in both training and validation accuracy, though with an initial loss spike. This model may be more suitable

where stability and consistent generalization are essential, albeit with a marginally lower accuracy.

Recommendation: For tasks demanding high accuracy and fast convergence, ResNet50 is a favorable choice, provided generalization is verified with further validation. For applications where stability in training is critical, ResNet101 may offer advantages due to its smoother accuracy curve despite a slower start.

Chapter 6

Conclusion and Future Work

6.1 Summary

This project successfully developed a brain tumor classification application, integrating a robust machine learning model with a user-friendly Streamlit interface and a responsive chatbot system. The application enables users to upload MRI images for automated brain tumor detection and receive real-time explanations of classification outcomes. The chatbot effectively guides users through image upload steps, prediction interpretation, and answers queries about the application.

A key achievement in this project was the deployment and comparison of multiple model architectures, including VGG-19, ResNet50, and ResNet101. Each model underwent extensive data preprocessing and augmentation, with performance metrics thoroughly evaluated. ResNet50 demonstrated faster convergence and higher accuracy, whereas ResNet101 provided more stable training behavior, offering valuable insights into the strengths and trade-offs of each model.

6.2 Future Enhancements

Future work will focus on expanding chatbot functionalities, incorporating more in-depth medical explanations, and adding language support for broader accessibility. To further improve model accuracy, we plan to explore ensemble methods that combine the strengths of multiple architectures like VGG-19, ResNet50, and ResNet101, which could enhance prediction reliability.

Additionally, user feedback mechanisms will be enhanced to allow adaptive learning for the chatbot, enabling it to handle complex medical queries more effectively. Planned visualization improvements include confidence heatmaps on MRI scans, which will provide users with an even clearer understanding of their results. These enhancements aim to advance the application's objective of delivering accurate, accessible, and user-centered brain tumor diagnostics.