A Major Project Report On

Brain Hemorrhage Detection

Submitted to partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

M. SRI MANISH REDDY	**(21911A05G7)**
B. NILESH	**(21911A05D7)**
B. RAKESH KUMAR	**(21911A05D9)**
B. SRIKANTH	**(21911A05E2)**

Under the Guidance of

Mrs. A. LALITHA

Assistant. Professor

Department of Computer Science and Engineering

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, Accredited by NAAC, NBA & permanently Affiliated to JNTUH, Hyderabad)

Aziz Nagar Gate, C.B. Post, Hyderabad-500075

2024-2025

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

(Approved by AICTE, Accredited by NAAC, NBA & permanently Affiliated to JNTUH, Hyderabad)
Aziz Nagar Gate, C.B. Post, Hyderabad-500075

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report titled "**Brain Hemorrhage Detection**" is being submitted by **M. SRI MANISH (21911A05G7), B. NILESH (21911A05D7), B. RAKESH (21911A05D9), B. SRIKANTH (21911A05E2)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, is a record of Bonafide work carried out by them under my guidance and supervision. These results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide	Head of Department
Mrs. A. Lalitha	Dr. D. Aruna Kumari
Assistant. Professor	Professor

External Examiner

DECLARATION

We, **M. SRI MANISH, B. NILESH, B. RAKESH, B. SRIKANTH** hereby declare that the project entitled, **“Brain Hemorrhage Detection”** submitted for the degree of Bachelor of Technology in Computer Science and Engineering is original and has been done by us and this work is not copied and submitted anywhere for the award of any degree.

Date:	**M. SRI MANISH**	**(21911A05G7)**
Place: Hyderabad	**B. NILESH**	**(21911A05D7)**
	B. RAKESH	**(21911A05D9)**
	B. SRIKANTH	**(21911A05E2)**

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to the project guide **Mrs. A. Lalitha**, Assistant Professor, Vidya Jyothi Institute of Technology, Hyderabad for her timely cooperation and valuable suggestions while carrying out this work. It is her kindness that made us learn more from her.

We are grateful to **Dr. D Aruna Kumari**, Professor and HOD, Department of CSE, for her help and support during our academic year.

We would like to take this opportunity to express our gratitude to our Principal **Dr. A. Srujana** for providing necessary infrastructure to complete this project.

We would thank our parents and all the faculty members who have contributed to our progress through the course to come to this stage.

M. SRI MANISH (21911A05G7)
B. NILESH (21911A05D7)
B. RAKESH (21911A05D9)
B. SRIKANTH (21911A05E2)

ABSTRACT

Brain Hemorrhage Detection

Brain hemorrhage is a critical and potentially life-threatening medical condition that necessitates prompt detection and intervention to minimize the risk of permanent neurological damage or death. Conventional diagnostic methods primarily depend on the manual interpretation of imaging modalities such as computed tomography (CT) and magnetic resonance imaging (MRI), typically performed by radiologists. Although effective, this manual analysis is time-consuming, labor-intensive, and susceptible to human error. To overcome these limitations, this research explores the integration of deep learning and machine learning techniques for the automated detection of brain hemorrhages in CT scan images. Leveraging advancements in artificial intelligence, the study investigates the use of convolutional neural networks (CNNs), which have proven highly effective in the field of medical image analysis. Specifically, it focuses on two widely used CNN architectures: MobileNet and ResNet. MobileNet offers a lightweight design optimized for performance on mobile and low-resource devices, making it ideal for real-time clinical applications in decentralized or underserved settings. In contrast, ResNet employs deep residual connections that enable the training of substantially deeper networks, thereby enhancing the model's ability to capture complex patterns within medical images. The research involves the implementation and comparative analysis of these two architectures using a labeled dataset of brain hemorrhage CT scans. The models are evaluated based on key performance metrics, including accuracy, sensitivity, and specificity, to determine their diagnostic effectiveness and reliability. By identifying the strengths and limitations of each model, the study aims to provide meaningful insights into their practical deployment for medical image classification. Ultimately, the outcomes are expected to contribute to the development of advanced, efficient, and accessible diagnostic tools, thereby supporting faster and more accurate clinical decision-making and improving patient outcomes across diverse healthcare environments.

INDEX

FIGURES

CHAPTER-1
INTRODUCTION

1.1 Objective of the Project

This project's goal is to create and assess cutting-edge deep learning models, MobileNet and ResNet in particular, for the precise and quick identification of brain haemorrhages from medical images. The purpose of the study is to evaluate how well these architectures perform in terms of computational efficiency, sensitivity, specificity, and accuracy. In order to improve patient outcomes and shorten the time it takes for patients who have had brain haemorrhages to receive treatment, the project aims to develop a dependable and automated diagnostic tool that can help medical professionals make accurate and timely diagnoses.

1.2 Problem Statement:

The difficulty of quickly and precisely identifying brain haemorrhages using medical imaging is the issue this study attempts to address. The current diagnostic procedures, which mainly rely on radiologists' manual interpretation, are laborious and prone to mistakes, which could cause delays in necessary medical care. The purpose of this study is to assess and contrast the efficiency of cutting-edge deep learning models, MobileNet and ResNet in particular in automating the detection process. The study aims to improve patient outcomes and clinical decision-making by increasing the speed and accuracy of brain haemorrhage diagnosis, which will ultimately lower the associated rates of morbidity and mortality.

1.3 Motivation:

The motivation behind this study is driven by the critical need for accurate and quick identification of brain hemorrhages, which can seriously affect patient outcomes, drives this study. Conventional diagnosis techniques cause delays in treatment by requiring time and prone to human error. This work intends to improve diagnostic accuracy and efficiency by using the power of sophisticated deep learning architectures including MobileNet and ResNet. The ultimate aim is to create strong, automated systems that can help medical professionals in making timely and accurate diagnosis, so enhancing patient care and lowering the morbidity and death related with brain hemorrhages.

1.4 Scope:

The scope of this project encompasses the development, training, and evaluation of deep learning models, MobileNet and ResNet for brain hemorrhage detection using medical imaging data. Preprocessing of image datasets, neural network architecture implementation, and thorough performance evaluation using accuracy, sensitivity, and specificity all fall under here. The work also intends to evaluate the models' computational efficiency, so stressing their possible use in clinical settings. The project also investigates the viability of implementing these models in settings with limited resources, such mobile devices, to support generally available diagnostics solutions.

CHAPTER-2

LITERATURE SURVEY

[1] Md. Imdadul Haque Emon et al. (2023) – "Intracranial Brain Hemorrhage Diagnosis and Classification: A Hybrid Approach"

This study introduces a hybrid model that combines CNNs with classical ML classifiers to enhance the accuracy and reliability of brain hemorrhage diagnosis from CT images. The CNN is responsible for extracting high-dimensional features, capturing intricate spatial patterns from raw pixel data. These extracted features are then passed into machine learning models such as Support Vector Machines (SVMs) and Random Forests (RF) for final classification. This two-stage pipeline addresses a key limitation of standalone CNNs—overfitting on small datasets—by introducing a model generalization layer. The dataset used was a publicly available CT scan dataset, and the authors ensured a proper split for training and testing to avoid data leakage.

What makes this paper stand out is its emphasis on computational efficiency and clinical applicability. The authors suggest that their hybrid model could be embedded into hospital diagnostic workflows due to its lightweight inference time and high accuracy metrics. They also note that integrating domain-specific preprocessing steps like skull stripping and normalization significantly boosts model performance. This study serves as a strong foundation for developers looking to build clinically deployable hemorrhage detection systems, balancing accuracy with practical feasibility.

[2] Mia Daugaard Jørgensen et al. (2021) – "CNN Performance Compared to Radiologists in Detecting ICH: A Systematic Review and Meta-Analysis"

This paper provides a comprehensive review and meta-analysis of convolutional neural network (CNN) models used for detecting intracranial hemorrhages (ICH) on CT scans, comparing their diagnostic performance against expert radiologists. By aggregating results from multiple studies, the authors aim to evaluate the consistency, accuracy, and limitations of CNNs in real-world clinical environments. The findings suggest that well-trained CNNs can achieve sensitivity and specificity levels comparable to or even exceeding those of human radiologists.

One of the key contributions of this meta-analysis is its strong emphasis on model generalizability. It highlights that models trained on diverse and large datasets tend to perform better across different patient populations. However, the paper also warns of over-reliance on black-box models without clinical interpretability. Furthermore, the authors emphasize the importance of calibration and external validation before deployment in healthcare settings.

This study is crucial for understanding the clinical readiness of AI models. For any project involving automated hemorrhage detection, insights from this review offer benchmarks, cautionary advice on dataset bias, and a realistic picture of the capabilities of CNNs in a clinical workflow.

[3] Vincy Davis & Satish R. Devane (2017) – "Diagnosis & Classification of Brain Hemorrhage"

This work investigates machine learning-based classification methods to detect various types of brain hemorrhages using CT imaging data. The authors experimented with supervised learning techniques such as Decision Trees, k-NN, and Naïve Bayes classifiers on pre-processed CT scan datasets. Although the paper predates the widespread use of deep learning in medical imaging, it lays foundational concepts for pattern recognition and classification tasks in hemorrhage analysis.

The study focuses on identifying subtypes of hemorrhages like epidural, subdural, and intraparenchymal, highlighting that feature selection plays a crucial role in classification performance. Simple geometric and intensity-based features extracted from the CT images were used as input. While their reported accuracy metrics may not match modern deep learning models, the paper's structured approach to feature engineering, segmentation, and rule-based diagnosis is still relevant, especially for lightweight or embedded diagnostic tools.

This work is valuable from a historical and methodological standpoint. It provides insights into how hemorrhage classification was approached before the deep learning era and emphasizes the role of domain expertise in feature design—something still relevant when interpreting deep model outputs.

[4] Muhammad Faheem Mushtaq et al. (2021) – "BHCNet: Neural Network-Based Brain Hemorrhage Classification Using Head CT Scan"

BHCNet is a custom-built CNN designed specifically for brain hemorrhage classification. Unlike generic CNN architectures (e.g., ResNet, VGG), BHCNet incorporates domain-specific modifications like spatial attention blocks and customized convolution kernels suited for medical imaging. The model is trained on a large labelled CT scan dataset, with extensive data augmentation to combat class imbalance.

Key strengths of BHCNet include its high classification accuracy and generalizability. The paper also performs ablation studies to isolate the impact of each architectural component, making it easier for future developers to customize or improve the network. Furthermore, the authors implement Grad-CAM to visualize attention regions, enhancing interpretability—a vital factor in medical diagnostics.

For anyone developing a new deep learning architecture for hemorrhage detection, BHCNet offers a solid baseline. Its focus on interpretability, robustness, and class-wise accuracy breakdown makes it a go-to reference for both academic and clinical applications.

CHAPTER-3

SYSTEM ANALYSIS

3.1 Existing System

Existing methods for brain hemorrhage detection primarily rely on manual interpretation of medical images, such as CT scans and MRIs, by radiologists. This process, while effective, is time-consuming and subject to human error and variability in diagnostic expertise. Traditional computer-aided detection (CAD) systems have been employed to assist radiologists, using basic image processing and machine learning techniques. However, these systems often lack the sophistication required for high accuracy and are limited in their ability to generalize across diverse datasets. The need for more advanced, automated, and reliable diagnostic tools has prompted the exploration of deep learning models in this domain.

3.2 Disadvantages:

- Time-Consuming: Manual interpretation by radiologists is a slow process, leading to delays in diagnosis and treatment.
- Human Error: Diagnostic accuracy is subject to human error and variability, potentially resulting in missed or incorrect diagnoses.
- Limited Availability: Expert radiologists may not always be available, particularly in remote or resource-constrained settings.
- Basic CAD Systems: Traditional computer-aided detection systems often lack the sophistication needed for high accuracy, especially in complex cases.
- Generalization Issues: These systems struggle to generalize across different datasets and imaging conditions, limiting their robustness and reliability.
- Resource Intensive: Manual and traditional methods require significant human and computational resources, making them less efficient.

3.3 Proposed System

The proposed system leverages advanced deep learning models, specifically MobileNet and ResNet, to automate and enhance the detection of brain hemorrhages from medical images. MobileNet's lightweight architecture is optimized for deployment in resource-constrained environments, while ResNet's deep residual learning capabilities ensure high accuracy by effectively training very deep networks. This system aims to preprocess the image data, train the models, and evaluate their performance using metrics such as accuracy, sensitivity, and specificity. By providing a reliable and efficient diagnostic tool, the proposed system seeks to support healthcare professionals in making timely and precise diagnoses, ultimately improving patient outcomes.

3.4 Advantages:

- Enhanced Accuracy: Deep learning models like MobileNet and ResNet provide high accuracy in detecting brain hemorrhages, reducing the likelihood of missed or incorrect diagnoses.
- Automation: Automated analysis of medical images speeds up the diagnostic process, enabling quicker decision-making and treatment initiation.
- Consistency: Deep learning models provide consistent results, eliminating the variability and potential errors associated with human interpretation.
- Resource Efficiency: MobileNet's lightweight architecture allows deployment in resource-constrained environments, such as mobile devices, making advanced diagnostics more accessible.
- Scalability: The system can handle large volumes of data efficiently, making it suitable for use in busy clinical settings.
- Support for Radiologists: By providing a reliable second opinion, the system assists radiologists in making more informed decisions, enhancing overall diagnostic accuracy.
- Real-time Processing: Capable of rapid image analysis, the system supports real-time diagnostic applications, crucial for emergency medical situations.

CHAPTER-4

REQUIREMENT ANALYSIS

Functional and non-functional requirements

After the severe continuous analysis of the problems that arose in the existing system, we are now familiar with the requirements required by the current system. The requirements that the system needs are categorized into functional and non-functional requirements. These requirements are listed below:

4.1 Functional Requirements

Functional requirements specify the essential features and behaviours the system must perform to meet the user's needs. These include the inputs to be provided, the processing that occurs, and the expected outputs. For the brain hemorrhage detection system, these requirements are critical as they directly govern how the software will operate, especially in processing medical images, detecting abnormalities, and presenting results in an accessible format. The following functional requirements have been identified for the project:

1. Upload and Input Handling: The system must allow users to upload brain CT scan images in standard medical image formats. It should validate the format and quality of input images before processing.
2. Preprocessing of Input Images: The system must perform preprocessing steps such as resizing, normalization, and contrast enhancement to prepare images for analysis. It should remove noise and artifacts that could affect prediction accuracy.
3. Brain Hemorrhage Detection: The system must use trained deep learning models to detect the presence of brain hemorrhage in the input images.
4. Visualization of Results: The system should visually highlight the region of hemorrhage on the CT image using bounding boxes or heatmaps. It must display classification confidence scores to support decision-making.

5. Result Interpretation: The system must generate a summary report of the detection, including prediction outcome, confidence level, and any detected abnormalities.
6. User Interface and Interaction: • The application must provide a simple, user-friendly interface for medical staff or researchers to upload images, view results, and access reports.
7. Deployment Compatibility: The system must support execution both on local machines and cloud environments. A lightweight version of the application should be available for mobile or low-resource environments.

4.2 Non-functional requirements

Non-functional requirements define the quality attributes, system constraints, and operational characteristics that the proposed system must fulfil. Unlike functional requirements that describe specific system behaviours, non-functional requirements focus on how the system performs and operates under various conditions. These requirements play a critical role in shaping the system's architecture, usability, and long-term maintainability, often influencing decisions related to performance, cost, control, security, and user experience.

Based on the essential non-functional aspects relevant to the proposed system, the following criteria have been identified:

1. Reliability: The system must consistently reflect updates made by both users and developers, ensuring transparency, stability, and dependable operation over time.

2. Security: As technological threats continue to evolve, the system must implement robust security mechanisms to safeguard data integrity and prevent crashes or unauthorized access.

3. Maintainability: The system should support easy monitoring and maintenance, minimizing the need for complex background processes and reducing the time and effort required for updates or debugging.

4. Performance: It must deliver high-speed responsiveness and support concurrent user operations without experiencing significant lags or downtime.

5. Portability: The system should be easily deployable on different platforms or environments, allowing for seamless migration in case of server issues or hosting changes.

6. Scalability: It should be capable of accommodating future enhancements and increasing workloads, offering a flexible infrastructure for adding new features or services.

7. Flexibility: The architecture must be adaptable to evolving business logic, user requirements, and operational contexts, with minimal disruption to existing components or workflows.

4.3 Hardware Requirements

Processor	- I3/Intel Processor
Hard Disk	- 128GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	- SVGA
RAM	- 8GB

4.4 Software Requirements:

Operating System	: Windows 7/8/10/11
Server side Script	: HTML, CSS, Bootstrap & JS
Programming Language	: Python
Libraries	: Flask, Pandas, Tensorflow, Smtplib, Numpy
IDE/Workbench	: PyCharm or VS Code
Technology	: Python 3.10+
Server Deployment	: Xampp Server
Database	: MySQL

CHAPTER-5

METHODOLOGY

5.1 MobileNet

MobileNet is a class of efficient convolutional neural networks designed specifically for mobile and resource-constrained environments. It uses depthwise separable convolutions, which break down a standard convolution into a depthwise convolution and a pointwise convolution. This approach significantly reduces the number of parameters and computational cost, making MobileNet much lighter than traditional CNNs. Despite its lightweight nature, MobileNet maintains robust performance, making it suitable for various image classification tasks. In our brain hemorrhage detection project, MobileNet is chosen for its efficient architecture. Its reduced size and lower computational requirements allow for deployment on mobile devices and in real-time applications, which is crucial for settings where quick and accurate diagnosis is needed, such as in remote or resource-limited environments. MobileNet provides the capability to perform real-time brain hemorrhage detection, assisting healthcare professionals in making timely decisions and potentially improving patient outcomes.

5.2 ResNet (Residual Networks):

ResNet, or Residual Networks, is a groundbreaking deep learning architecture known for its deep residual learning capabilities. It addresses the vanishing gradient problem by introducing shortcut connections that allow gradients to flow more easily through the network. This enables the training of much deeper networks, which can learn more complex patterns and representations. The typical ResNet architecture includes multiple layers of residual blocks, each containing convolutional layers and identity mappings. In our brain hemorrhage detection project, ResNet is employed for its ability to handle very deep networks effectively. Its deep architecture allows for capturing intricate details and patterns in brain hemorrhage images, leading to high diagnostic accuracy. ResNet's robustness makes it a reliable model for detecting subtle signs of brain hemorrhages, thus enhancing the precision of our automated diagnostic tool. The model's performance is critical in providing accurate assessments, which can aid in timely and appropriate medical interventions.

5.3 VGG16:

VGG16 is a convolutional neural network known for its simplicity and depth, consisting of 16 layers. Developed by the Visual Geometry Group at Oxford, VGG16 uses small 3x3 convolutional filters throughout the network, stacked on top of each other, followed by max-pooling layers. The network culminates in fully connected layers and a softmax classification layer. VGG16's structured and deep architecture enables it to learn complex features and perform exceptionally well in image classification tasks. In our brain hemorrhage detection project, VGG16 is utilized for its proven performance in image classification. Its deep architecture allows for capturing detailed features of brain hemorrhage images, making it a strong candidate for accurate diagnosis. VGG16 serves as a robust baseline model for comparison with other architectures. Its implementation helps in validating the effectiveness of more complex or efficient models like MobileNet and ResNet. By leveraging VGG16, we ensure that our project incorporates a well-established and reliable network, contributing to the overall robustness and accuracy of our brain hemorrhage detection system.

CHAPTER-6

SYSTEM DESIGN

6.1 UML Diagrams:

UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modelling tools include. In its simplest form, a use case can be described as a specific way of using the system from a User's (actor's) perspective. A more detailed description might characterize a use case as:

- a pattern of behaviour the system exhibits
- a sequence of related transactions performed by an actor and the system
- delivering something of value to the actor Use cases provide a means to:
- capture system requirements
- communicate with the end users and domain experts
- Test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways of using the system.

A UML system is represented using five different views that describe the system from a distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

- This view represents the system from the user's perspective.
- The analysis representation describes a usage scenario from the end-user's perspective.

Structural Model view

- In this model the data and functionality are arrived from inside the system.
- This model view models the static structures.

Behavioural Model View

- It represents the dynamic of behaviour as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

- In these the structural and behavioural parts of the system is represented as they are to be built.

Environmental Model View

- In this the structural and behavioural aspect of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- **UML Analysis Modelling**: This focuses on the user model and structural model views of the system.
- **UML Design Modelling**: This which focuses on the behavioural part of the system.

6.1.1 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

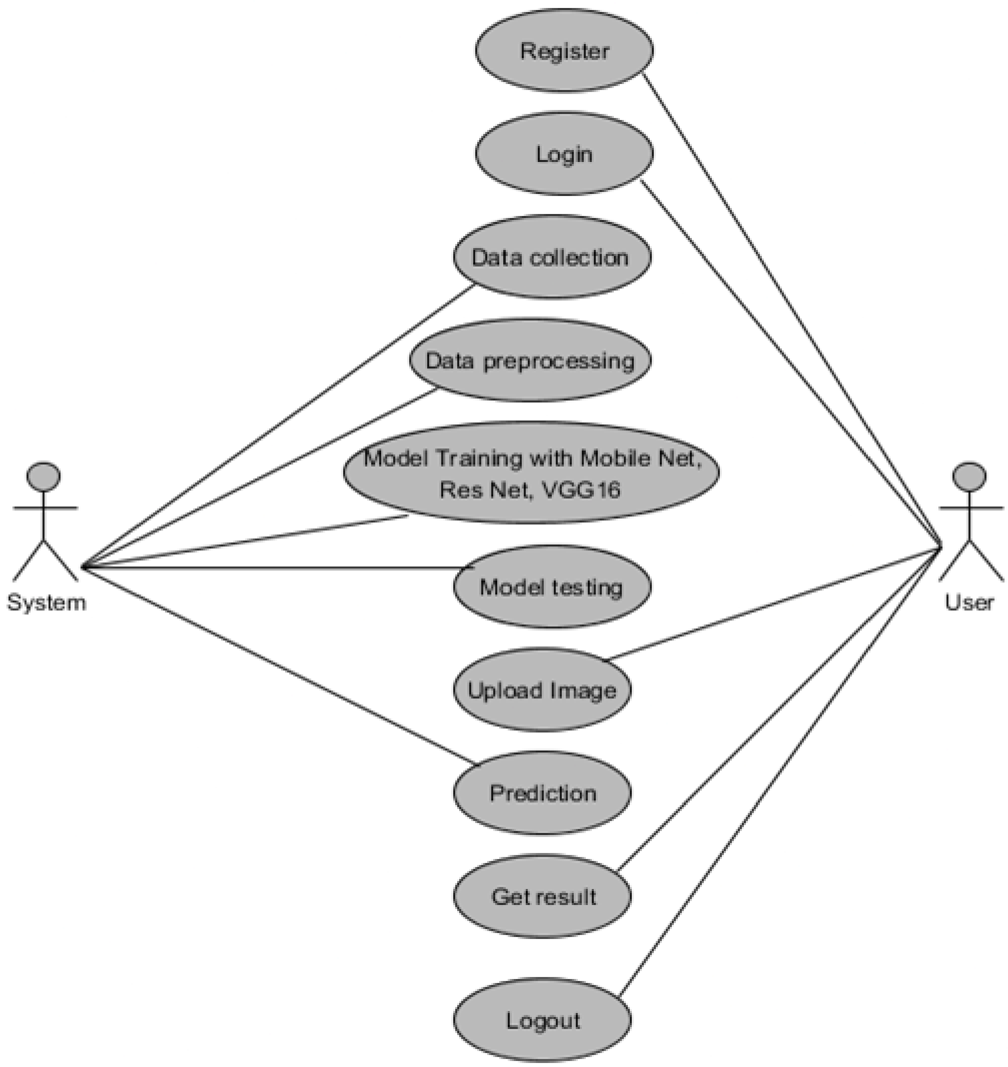

Figure 6.1.1 Use Case Diagram

6.1.2 Class Diagram:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

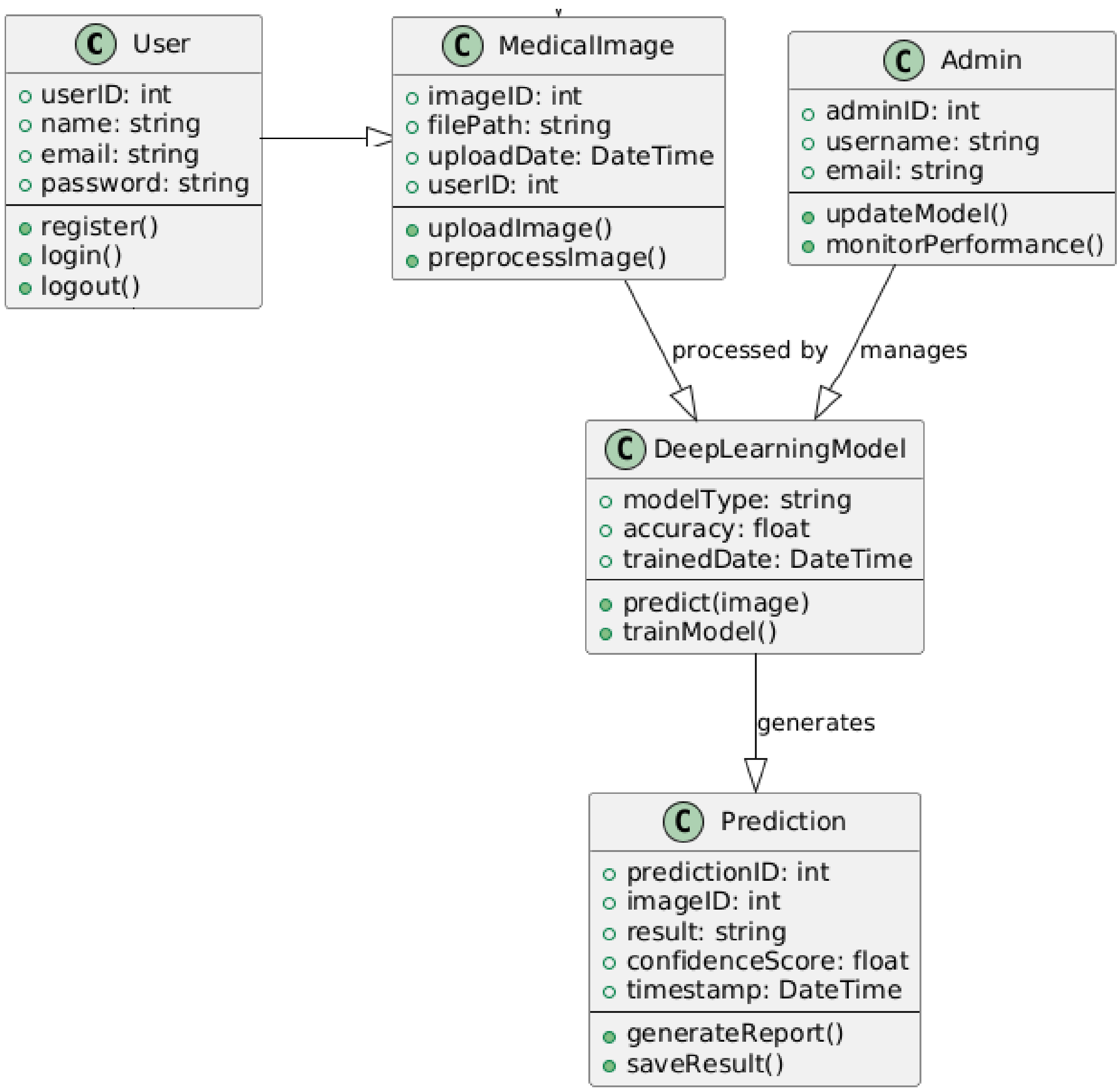

Figure 6.1.2 Class Diagram

6.1.3 Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

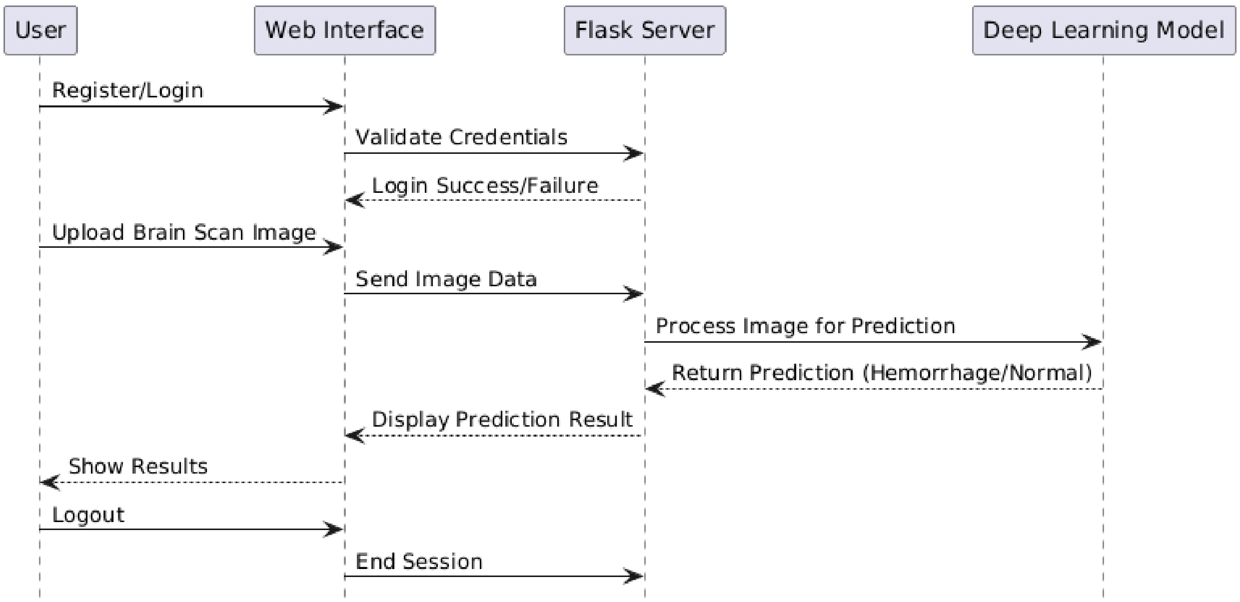

Figure 6.1.3 Sequence Diagram

6.1.4 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

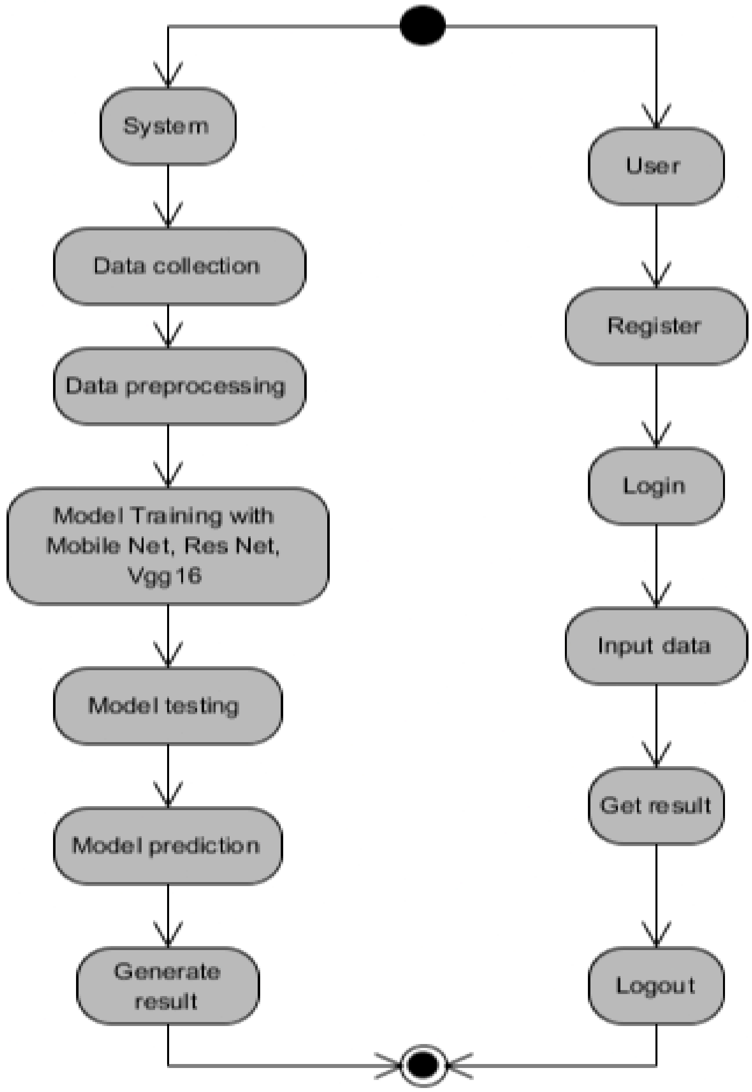

Figure 6.1.4 Activity Diagram

6.1.5 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

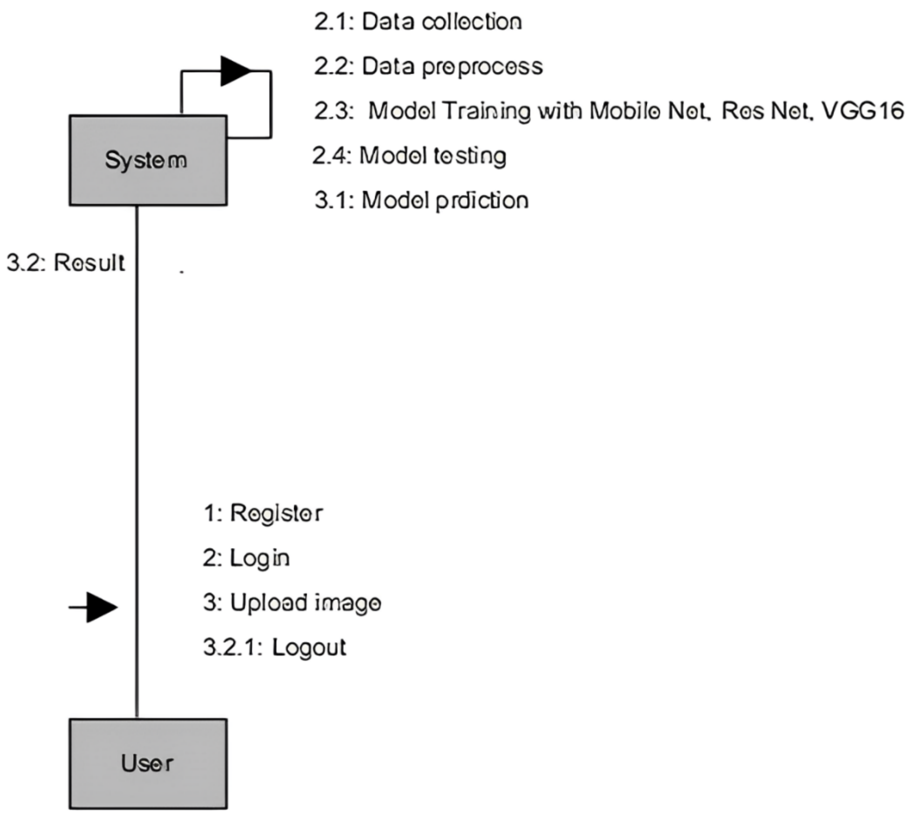

Figure 6.1.5 Collaboration Diagram

6.1.6 Deployment Diagram

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

Figure 6.1.6 Deployment Diagram

6.1.7 Component Diagram:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

Figure 6.1.7 Component Diagram

6.1.8 ER Diagram:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set. An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

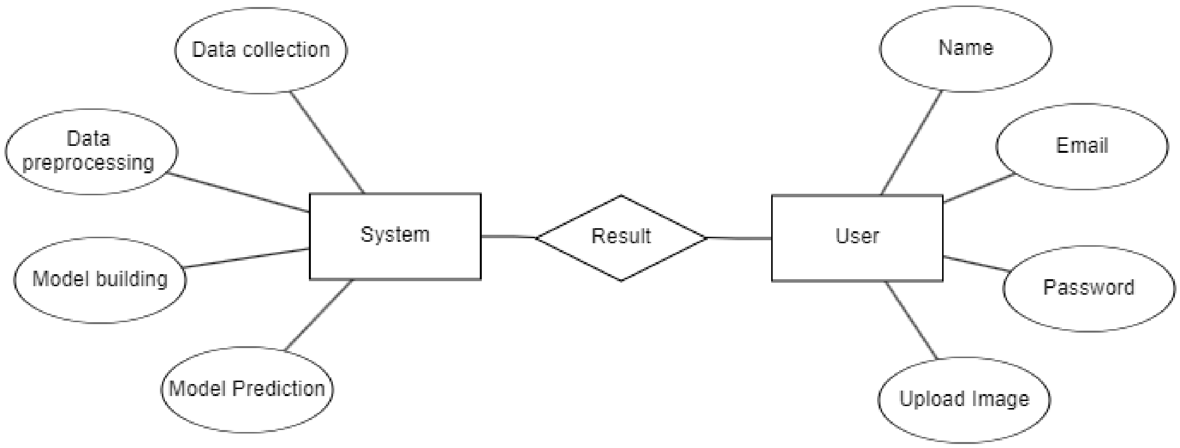

Figure 6.1.8 ER Diagram

6.2 System Architecture:

System architecture for an application refers to the high-level design and structure of the software and hardware components that make up a mobile application. It defines how different parts of the application interact with each other, how data is stored and processed, and how the application communicates with external resources or services. A well-defined system architecture is crucial for ensuring the reliability, scalability, and maintainability of a project.

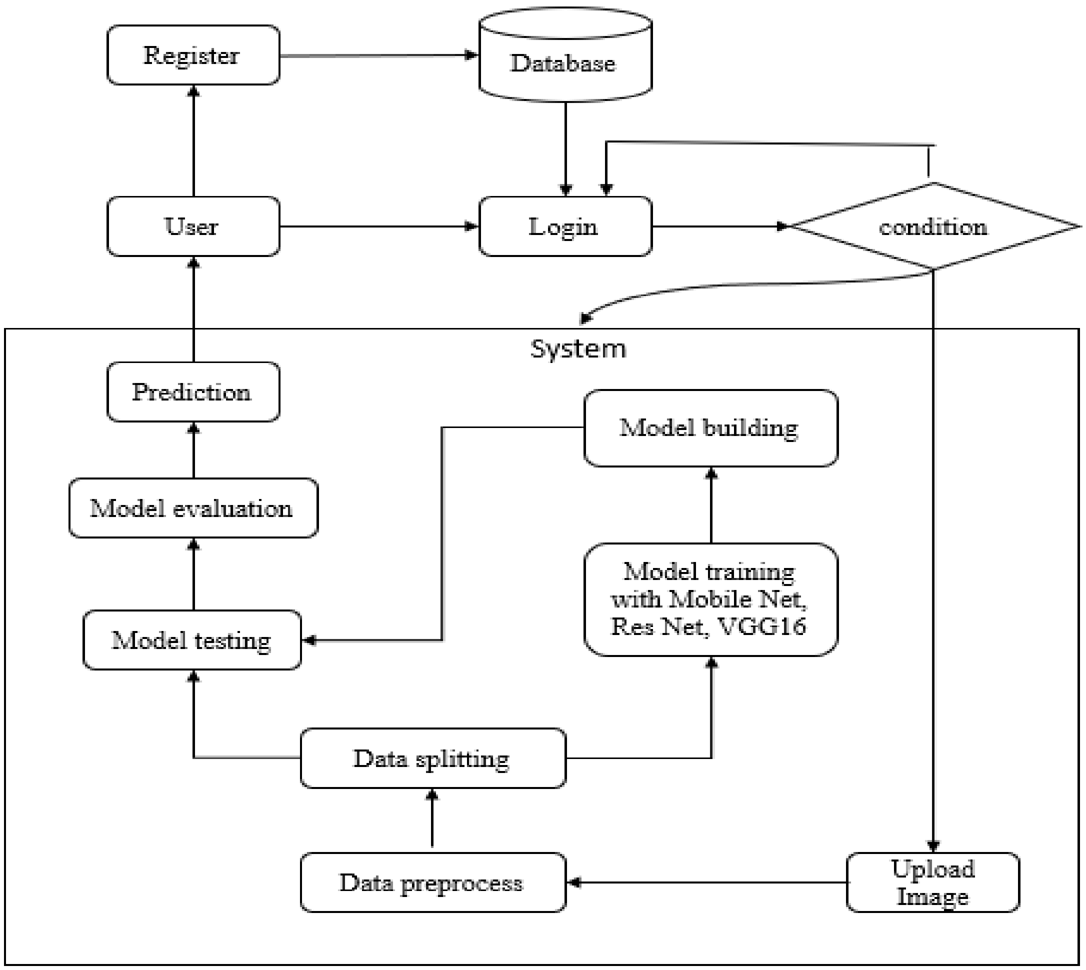

Figure 6.2 Architecture Diagram

CHAPTER-7

IMPLEMENTATION

7.1 System Modules:

1.1 Data Collection: In this module, the dataset containing images for stroke identification is divided into two subsets: the training dataset and the testing dataset. This split is typically done with a test size of 20%. The training dataset is used to teach the model, while the testing dataset is used to evaluate its performance.

1.2 Data Splitting: The pre-processed dataset is split into two subsets – Model training, Model testing.

1.3 Model Training: The training process involves using 80% of the dataset to teach the model. The model parameters are fine-tuned to minimize reconstruction errors through iterative optimization techniques, such as gradient descent.

1.4 Model Testing: The remaining 20% of the dataset is used for testing. The trained model predicts the segmentation of ischemic stroke lesions, and its performance is evaluated to determine the model's accuracy.

1.5 Model Saving: Once trained, the model is saved in a .pt format, preserving its learned weights and biases.

1.6 Model Prediction: Finally, we can input new images into the trained model to predict stroke.

7.2 User Modules:

2.1 Register: Users should first register with their credentials to create an account in the system.

2.2 Login: Users can log in with their registered credentials to access the system.

2.3 Upload Data: Users can upload their images to predict whether it is stroke or normal.

2.4 Viewing Results: That uploaded image will going to the model part to predict and it will give the prediction and user can view the result.

2.5 Logout: Finally, users can log out of the system to secure their session and personal data.

7.3 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It is widely used for data analysis, scientific computing, and machine learning tasks. It runs locally or on a server, providing flexibility in managing dependencies and storage.

Components of Jupyter Notebook:

- **Cells:** Jupyter Notebook organizes content into cells, which can either run code or display text in Markdown format.
- **Kernel:** The computational engine that executes the code within the notebook.
- **Extensions and Libraries:** Allows for seamless integration with Python libraries like Pandas, NumPy, Matplotlib, and Scikit-learn.

Advantages of Jupyter Notebook:

- **Flexibility:** Works offline and supports installation of custom packages.
- **Interactive Visualizations:** Integration with libraries like Matplotlib and Seaborn enables dynamic plotting.
- **Reproducibility:** Saves code and outputs together, ensuring easier sharing and version control.

7.4 PYTHON

Python is a high-level, interpreted programming language known for its readability, simplicity, and versatility. It’s widely used across different platforms, from web development to data science, artificial intelligence, and machine learning. Python’s strength lies in its extensive collection of libraries and packages that extend its core functionality, making it an ideal choice for data science projects.

Key Features of Python:

- **Interpreted:** Execute code line-by-line, visualize results in real-time, and debug easily.
- **Extensive Libraries:** Packages like NumPy, Pandas, and Matplotlib simplify complex data operations and visualizations.
- **Cross-Platform Compatibility:** Python scripts work on multiple operating systems.
- **Advantages of Python:**
- **Readable and Maintainable Code:** Python's syntax promotes readability, which helps in writing clean and understandable code.
- **Extensive Library Support:** Python's ecosystem includes libraries tailored for scientific computing, machine learning, data visualization, and more.
- **Cross-Platform Compatibility:** Python code can run on any operating system with Python installed, making it a versatile choice for various applications.
- **Community Support:** Python has a large, active community that contributes to numerous opensource libraries and tools, offering vast resources for problem-solving.

7.5 LIBRARIES AND TOOLS USED

7.5.1 Machine Learning & Deep Learning Libraries

- **TensorFlow**: Open-source deep learning framework used to build and train convolutional neural networks (CNNs) such as MobileNet, ResNet, and VGG16 for classification tasks.
- **Keras**: High-level API built on top of TensorFlow, utilized for rapid prototyping and model construction.
- **scikit-learn**: Used for model evaluation, preprocessing tasks (e.g., splitting data, calculating metrics like accuracy, precision, recall), and comparison.

7.5.2 Data Processing & Manipulation

- **NumPy**: Provides support for large multi-dimensional arrays and matrices, as well as mathematical operations.
- **Pandas**: Facilitates data manipulation and preprocessing, especially for organizing metadata or managing labels associated with CT images.

7.5.3 Visualization Tools

- **Matplotlib**: Used for plotting training accuracy, loss graphs, and confusion matrices to evaluate model performance.
- **Seaborn**: Employed to visualize correlations and statistical relationships in data through heatmaps and advanced plotting.

7.5.4 Image Handling & Augmentation

- **Pillow (PIL)**: For basic image manipulation operations like resizing and conversion.
- **Keras ImageDataGenerator**: Used for augmenting the dataset through transformations like flipping, rotation, and scaling to improve model generalization.

7.5.5 Web & Backend Integration

- **Flask** : For building a lightweight backend to serve the trained model and create endpoints for image prediction.
- **HTML/CSS/JavaScript**: Frontend technologies for building the user interface to upload images and display results.

CHAPTER-8

SYSTEM STUDY AND TESTING

8.1 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ❖ Economic feasibility
- ❖ Technical feasibility
- ❖ Social feasibility

8.1.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

8.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

8.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

8.2 System Testing

Testing is a pivotal phase within the product development life cycle. In this phase, any remaining errors from previous stages are identified and rectified. Consequently, testing plays a critical role in ensuring quality assurance and the reliability of the software. During testing, the program is executed with a predefined set of test cases, and the program's output for these cases is assessed to determine if it aligns with the expected behaviour. Any errors that are detected are addressed, and these corrections are meticulously documented for future reference. As such, a comprehensive series of tests is conducted on the system before it is deemed ready for implementation.

Testing serves as a process to assess the correctness, completeness, security, and overall quality of developed computer software. It involves a technical investigation carried out on behalf of stakeholders to reveal information related to the product's quality, considering the context in which it is intended to operate. This encompasses executing a program or application with the objective of uncovering errors. Quality, in this context, is not an absolute value but rather a value relative to a particular individual or group.

Therefore, testing cannot guarantee the absolute correctness of arbitrary computer software. Instead, it offers a form of critique or comparison, assessing the state and behaviour of the product against its specifications. It's important to differentiate software testing from the broader discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not limited to testing.

While there are various approaches to software testing, effectively testing complex products primarily involves an investigative process rather than just following routine procedures. Although many of the intellectual processes in testing align with those in reviews or inspections, the term "testing" implies dynamic analysis, actively subjecting the product to a range of scenarios. Common quality attributes under scrutiny include capability, reliability, efficiency, portability, maintainability, compatibility, and usability.

There are various types of tests. Each test type addresses a specific testing requirement.

8.2.1 Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. This system was tested with the set of proper test data for each module and the results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module. This is also known as MODULE TESTING. This testing is carried out during phases, each module is found to be working satisfactory as regards to the expected output from the module.

8.2.2 Integration Testing

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with flow of data across interfaces. The unit-tested modules are grouped together and tested in small segments, which make it easier to isolate and correct errors. This approach is continued until all modules have integrated to form the system as a whole.

8.2.3 Functional Testing

Functional tests provide methodical evidence that the tested functions are available and meet all technical and business requirements as well as those listed in the system documentation and user manuals.

Functional testing is centred on the following items:

Valid Input: Recognized valid input classes need to be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Output: Specific application output classes must be put to use.

Systems or procedures: Interfacing systems or procedures must be invoked. Organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying business process flows, including data fields, predefined processes, and successive processes, must be considered for testing.

8.2.4 System Test

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration testing. System testing is based on process description and flows, emphasizing pre-driver process and integration points.

8.2.5 White Box Testing

This allows the tests to:

- Check whether all independent paths within a module have been exercised at least once
- Exercise all logical decisions on their false sides
- Execute all loops and their boundaries and within their boundaries
- Exercise the internal data structure to ensure their validity
- Ensure whether all possible validity checks and validity lookups have been provided to validate data entry.

8.2.6 Black Box Testing

Black box testing is done to find the following:

- Incorrect or missing functions
- Errors on external database access
- Performance error
- Initialization and termination error

8.2.7 Performance Testing

The performance testing ensures that the output is produced within the time limits and time taken for the system compiling, giving response to the users and requests being sent to the system in order of the results.

8.2.8 Acceptance Testing

This is the final stage of the testing process before the system is accepted for operational use.

The system is tested within the data supplied from the system procurer rather than simulated data.

8.2.9 Validation Testing

This is the final stage of the testing process before the system is accepted for operational use. The system is tested within the data supplied from the system procurer rather than simulated data.

8.2.10 Output Testing

After performing the validation testing, the next step is to test the output of the proposed system, since no system could be useful if it did not produce the required output in the specified format. The outputs produced or displayed by the system under consideration are tested by asking the users about the format they require. As a result, there are two ways to think about the output format: one is on screen, and the other is in printed form

8.3 Test Cases

Test Case 1:

Name of the Test	User Registration
Input	Enter valid username, email, and password
Expected Output	User account is created and redirected to login page
Actual Result	As expected
Remarks	Successful

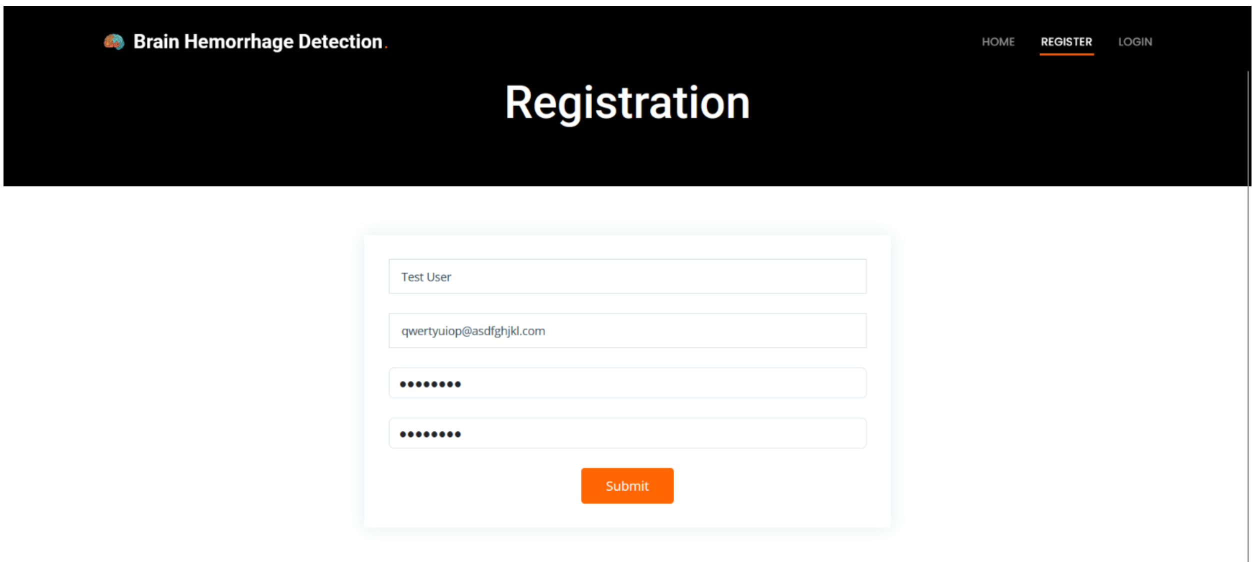

Figure 8.3.1 Test Case 1

Test Case 2:

Name of the Test	Invalid Login
Input	Enter invalid email or password
Expected Output	"ERROR: This email ID does not exist!"
Actual Result	As expected
Remarks	Successful

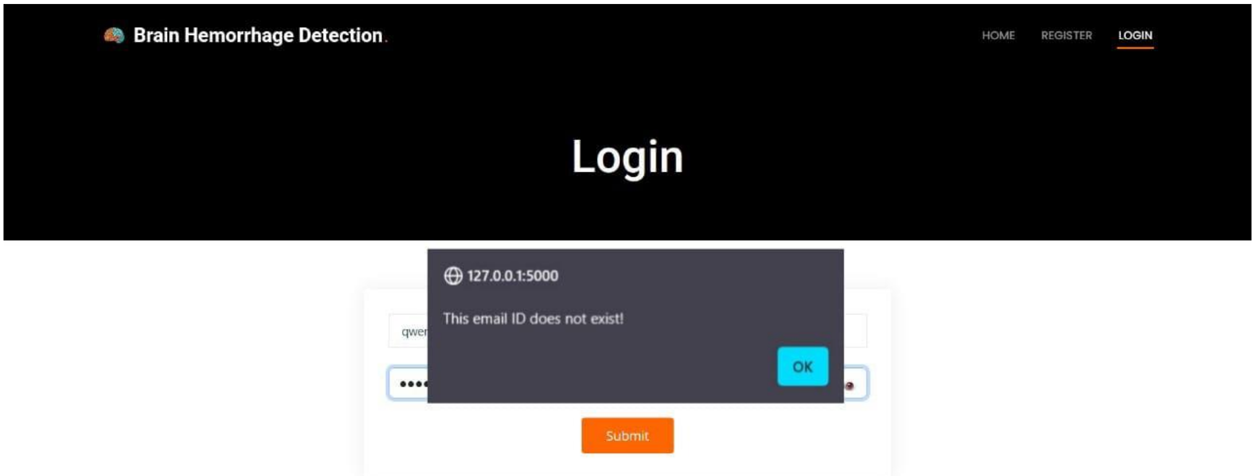

Figure 8.3.2 Test Case 2

Test Case 3

Name of the Test	Upload CT Scan Image
Input	Upload a valid brain CT image
Expected Output	Image uploaded and displayed correctly
Actual Result	As expected
Remarks	Successful

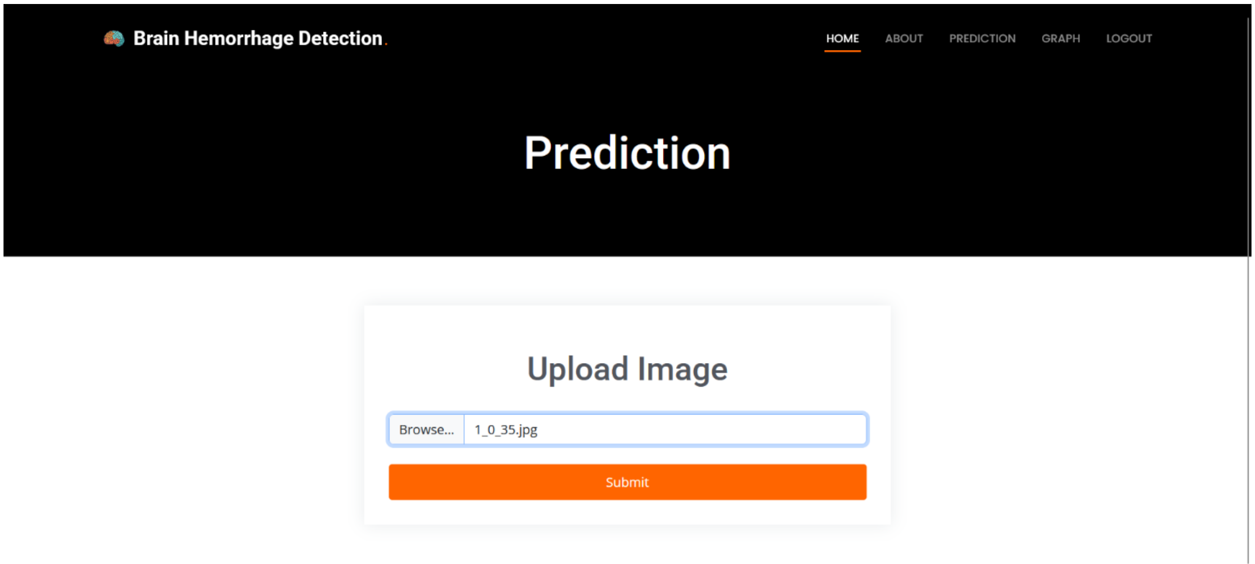

Figure 8.3.3 Test Case 3

Test Case 4

Name of the Test	Positive Prediction
Input	Upload a brain CT scan image showing signs of hemorrhage
Expected Output	"Hemorrhage Detected"
Actual Result	As expected
Remarks	Successful

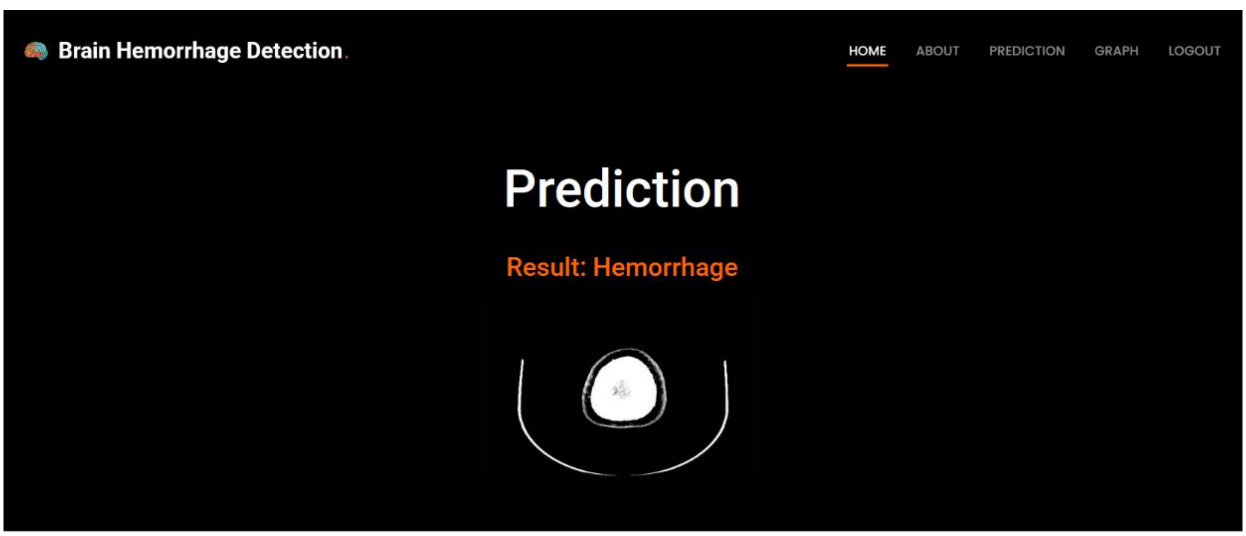

Figure 8.3.4 Test Case 4

Test Case 5

Name of the Test	Negative Prediction
Input	Upload a normal brain CT scan image
Expected Output	“Normal"
Actual Result	As expected
Remarks	Successful

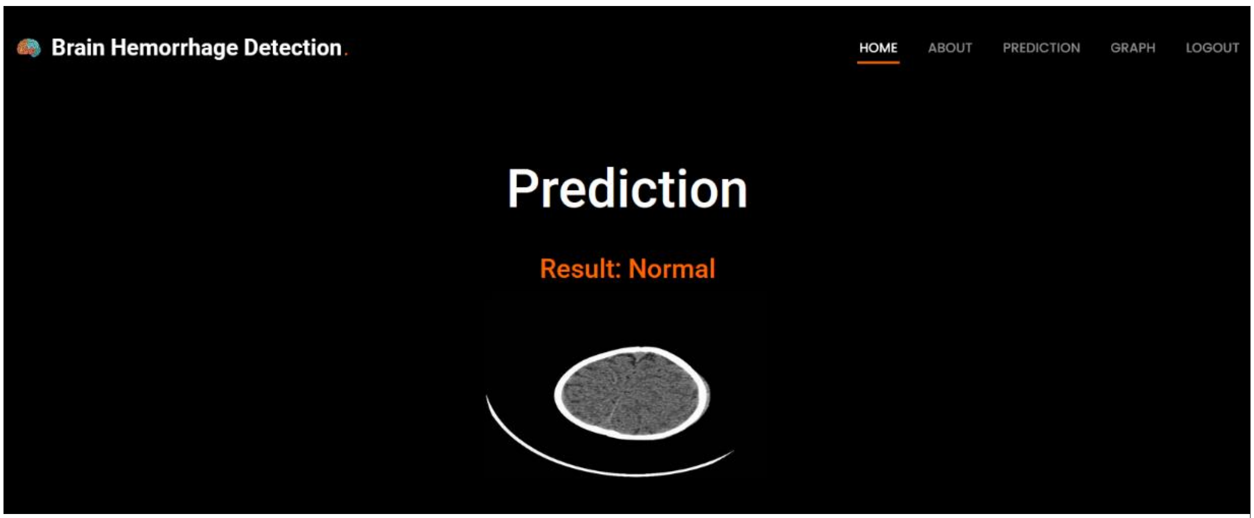

Figure 8.3.5 Test Case 5

CHAPTER-9

Results and Output Screens

9.1 Home page: This is the index Page of our website.

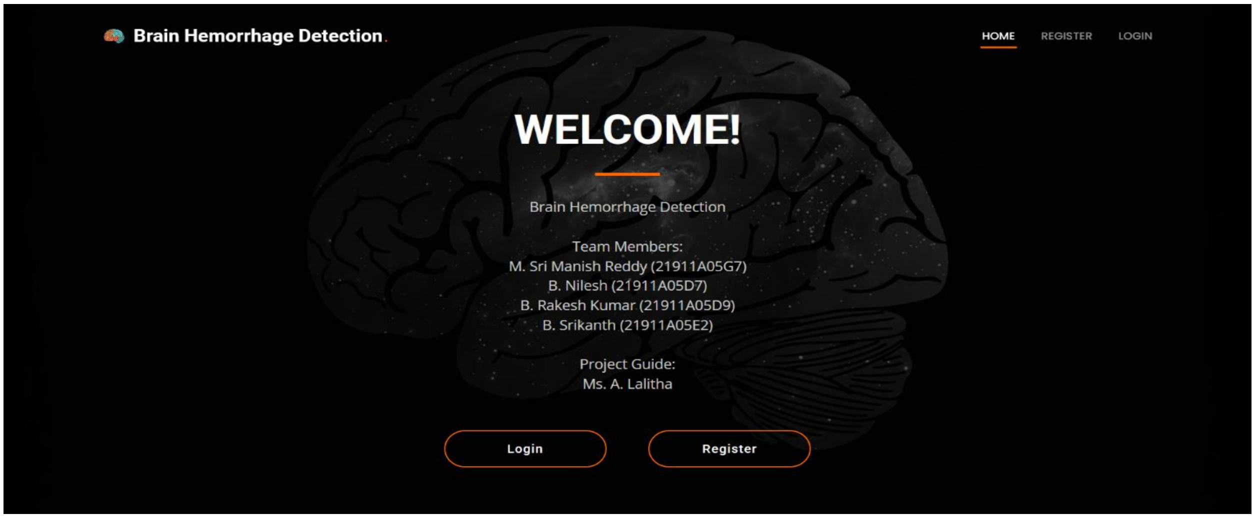

Figure 9.1 Home page

9.2 Registration Page: User can register with their credentials.

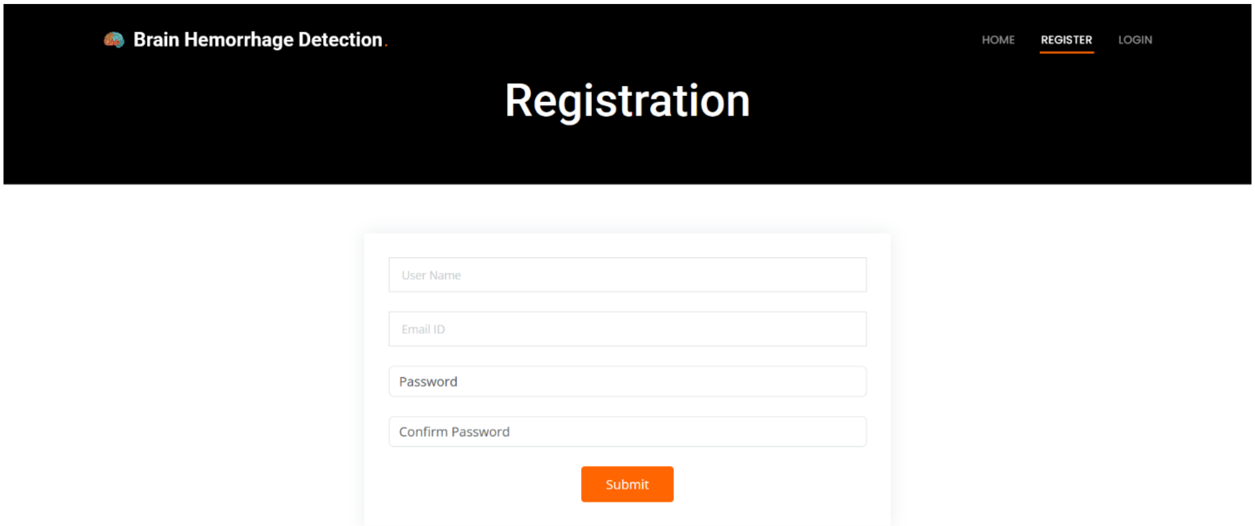

Figure 9.2 Registration Page

9.3 Login Page: User can login with their registered credentials.

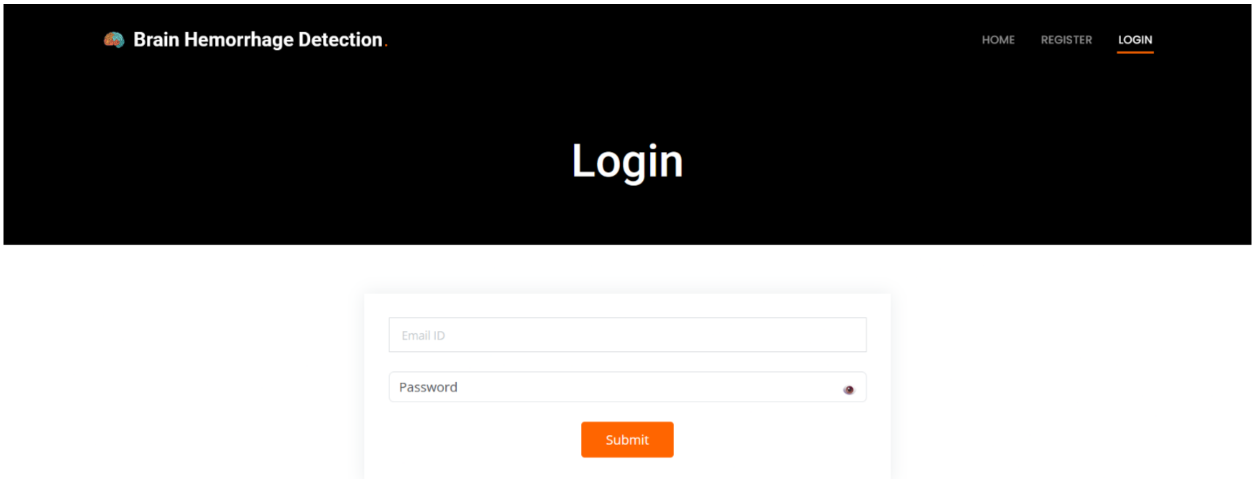

Figure 9.3 Login Page

9.4 About Page: This page contains about our website.

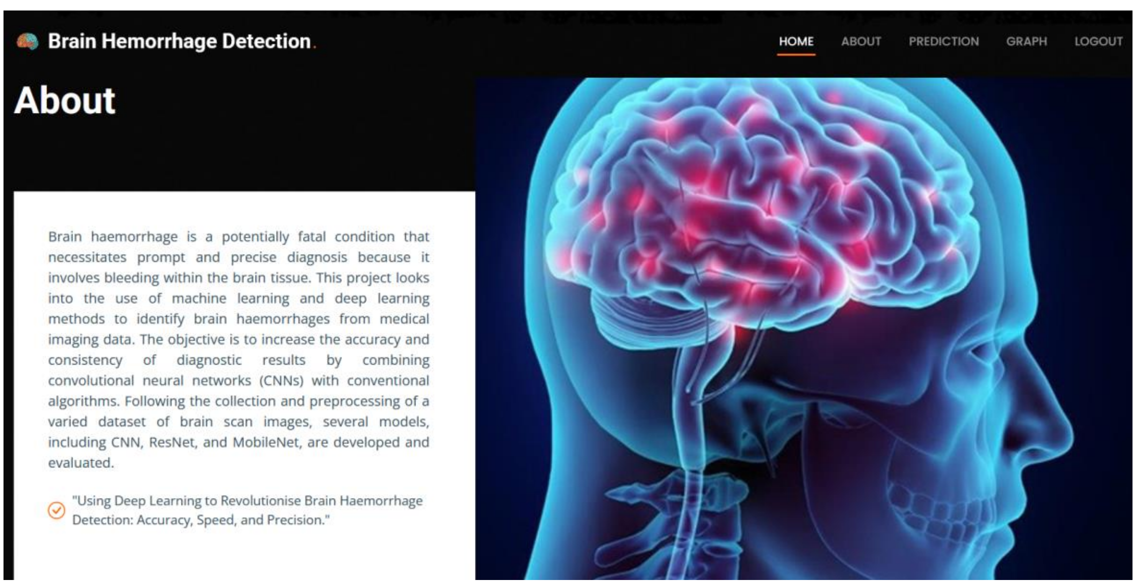

Figure 9.4 About Page

9.5 Prediction Page: This is the prediction page. User can give upload image here and get prediction result in result page.

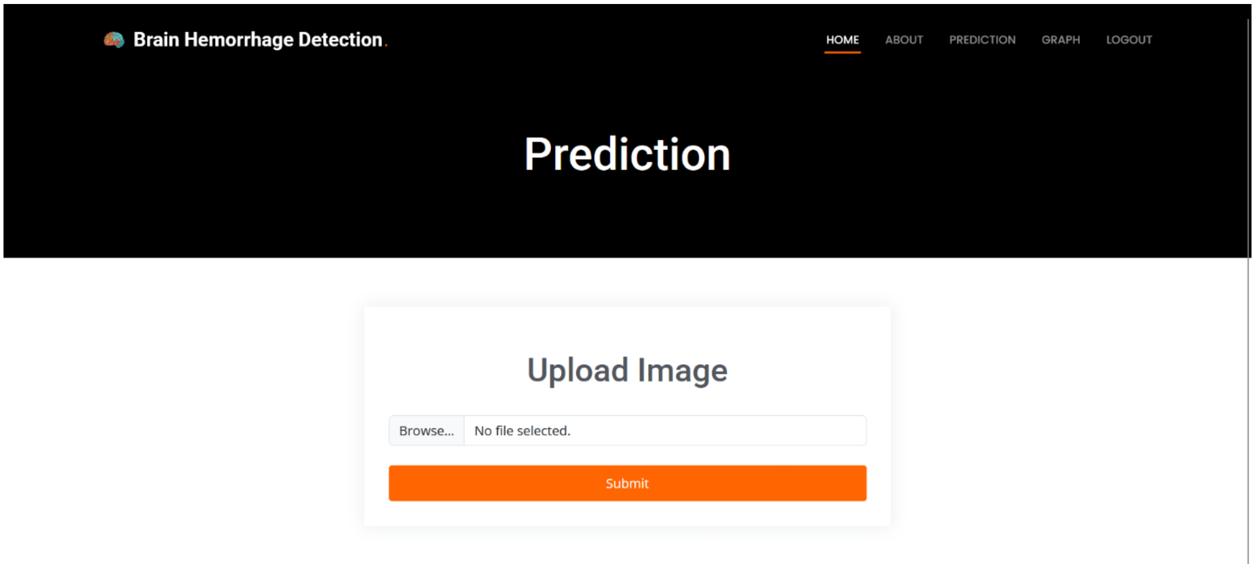

Figure 9.5 Prediction Page

CHAPTER-10

CONCLUSION

The Brain Hemorrhage Detection project successfully showcases the potential of deep learning and machine learning techniques in transforming medical image analysis. By implementing advanced models such as MobileNet, ResNet, and VGG16, the system was able to detect brain hemorrhages with high precision, balancing both accuracy and computational efficiency. Among these, MobileNet stood out as the preferred model due to its lightweight architecture and suitability for real-time, resource-constrained environments. A user-friendly web interface built with Flask enabled medical professionals to easily upload and analyze brain scan images, offering immediate diagnostic feedback. The project also incorporated effective visualization tools, including performance plots and confusion matrices, to present model accuracy in an interpretable manner. The integration of these capabilities suggests that AI-powered systems can significantly accelerate diagnosis, reduce manual error, and improve patient care. This comparative study emphasized that while each model offers distinct advantages, ResNet's deep architecture handling complexity and VGG16's simplicity yielding consistent results—their deployment should align with specific clinical needs and system constraints. The outcomes of this research highlight the transformative role AI can play in healthcare by enabling quicker, automated, and more accurate diagnosis of critical conditions like brain hemorrhage. Moving forward, the system could be enhanced through larger and more diverse datasets, integration with hospital workflows, additional preprocessing techniques, and the development of a mobile application. This work lays a strong foundation for future exploration in AI-driven medical diagnostics and supports the broader goal of leveraging technology to improve patient outcomes and streamline clinical processes.

CHAPTER-11

FUTURE ENHANCEMENT

Future enhancements to the Brain Hemorrhage Detection system can significantly elevate its diagnostic accuracy, clinical utility, and accessibility. Optimizing deep learning models such as MobileNet and ResNet, for both accuracy and inference speed remains a priority, alongside exploring ensemble techniques and incorporating advanced architectures like Vision Transformers for improved feature extraction. Enhancing data processing with advanced augmentation methods, contrast enhancement, and noise reduction will increase model robustness across diverse imaging conditions. Synthetic data generation using GANs and transfer learning from large-scale medical datasets can further improve generalization and reduce training overhead. Clinically, integrating the system into hospital workflows via APIs, DICOM support, and EHR compatibility would streamline adoption. User experience can be enriched through mobile applications, batch image processing, and customizable report generation in formats like PDF and DICOM. Feature-wise, expanding capabilities to include hemorrhage localization, severity grading, and multi-modal analysis (e.g., CT and MRI fusion) can provide deeper clinical insights. Real-time deployment on edge devices such as smartphones and tablets could make the tool accessible in rural and emergency settings. Furthermore, incorporating explainable AI will enhance transparency, allowing clinicians to understand and trust model decisions. Ensuring compliant data handling, secure authentication, and comprehensive audit logging is essential for widespread healthcare deployment. On the research front, expanding datasets with diverse and rare cases, adopting active learning for continuous improvement, and running extensive clinical trials will validate efficacy in real-world scenarios. Finally, global accessibility can be addressed through multi-language support, region-specific customization, and cloud-based deployment. Collectively, these enhancements aim to create a scalable, interpretable, and clinician-friendly AI solution that transforms brain hemorrhage detection and broader medical diagnostics.

REFERENCES

[1] Md. Imdadul Haque Emon, Khondoker Nazia Iqbal, Istinub Azad, Amena Akter Aporna, Nibraj Safwan Amlan, Md. Saiful Islam, & Rafeed Rahman (2023). Intracranial Brain Hemorrhage Diagnosis and Classification: A Hybrid Approach. *2023 6th International Conference on Information and Computer Technologies (ICICT)*, 95-99.

[2] Mia Daugaard Jørgensen, Ronald Antulov, SØren Hess, & S. Lysdahlgaard (2021). Convolutional neural network performance compared to radiologists in detecting intracranial hemorrhage from brain computed tomography: A systematic review and meta-analysis.. *European journal of radiology, 146, 110073.*

[3] Vincy Davis, & Satish R. Devane (2017). Diagnosis & classification of brain hemorrhage. *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 1-6.

[4] Areen K. Al-Bashir, & Batool Alsukhni (2023). ANN Algorithm for Brain Hemorrhage Detection Using CT Images. *Proceedings of 2023 the 13th International Workshop on Computer Science and Engineering*.

[5] Jinwei Li, Cong Liang, Junsun Dang, Yang Zhang, Hongmou Chen, Xianlei Yan, & Quan Liu (2024). Predicting the 90-day prognosis of stereotactic brain hemorrhage patients by multiple machine learning using radiomic features combined with clinical features. *Frontiers in Surgery, 11.*

[6] N R Bhuvaneswari, S. Ramakrishna, Mission Vidyalaya, R Karthikamani, R. Sathish Kumar, & S. Sanjayprabu (2023). Detection of Brain Hemorrhage from CT Scan Images using ROF Filter and Classification. *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, 1623-1628.

[7] Karamala Rooshita, N. V. Vyshnavi, Nara R Sanjana Chowdary, Tripty Singh, & Payel Patra (2024). Deep Learning for Precise Brain Hemorrhage Detection from CT scans: A Web - Based Approach. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1-6.

[8] Maryam Wardah, Muhammad Mateen, Tauqeer Safdar Malik, Mohammad Eid Alzahrani, Adil Fahad, Abdulmohsen Almalawi, & Rizwan Ali Naqvi (2023). Automated Brain Hemorrhage Classification and Volume Analysis. *Computers, Materials & Continua*.

[9] Padma Priya, S. Babu, & Dr. T. Brindha (2024). Deep Learning Fusion for Intracranial Hemorrhage Classification in Brain CT Imaging. *International Journal of Advanced Computer Science and Applications*.

[10] Muhammad Faheem Mushtaq, Mobeen Shahroz, Ali M. Aseere, Habib Shah, Rizwan Majeed, Danish Shehzad, & Ali Samad (2021). BHCNet: Neural Network-Based Brain

Hemorrhage Classification Using Head CT Scan. *IEEE Access, 9, 113901-113916.*

[11] Quoc Tuan Hoang, Xuan Hien Pham, Xuan Thang Trinh, Anh Vu Le, Minh Bui Vu, & Trung Thanh Bui (2024). An Efficient CNN-Based Method for Intracranial Hemorrhage Segmentation from Computerized Tomography Imaging. *Journal of Imaging, 10.*

[12] Kevin Haowen Wu, Kevin Zeng, Mikhail Y. Shalaginov, & TingyingHelen Zeng (2024). Brain Hemorrhage CT Image Detection and Classification using Deep Learning Methods. *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 5322-5326.

[13] Anh-Cang Phan, Thi-My-Nga Nguyen, & Thuong-Cang Phan (2019). Detection and Classification of Brain Hemorrhage Based on Hounsfield Values and Convolution Neural Network Technique. *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, 1-7.

[14] Giyaseddin Bayrak, Muhammed Şakir Toprak, Murat Can Ganiz, Halife Kodaz, & Ural Koç (2022). Deep Learning-Based Brain Hemorrhage Detection in CT Reports. *Studies in health technology and informatics, 294, 866-867* .

[15] Tong Duc Phong, Hieu Ngoc Duong, Hien T. Nguyen, Nguyen Thanh Trong, Vu Bao Hung Nguyen, Tran Van Hoa, & Vaclav Snášel (2017). Brain Hemorrhage Diagnosis by Using Deep Learning. *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*.

[16] Heba M. Afify, Kamel K. Mohammed, & Aboul Ella Hassanien (2024). Leveraging hybrid 1D-CNN and RNN approach for classification of brain cancer gene expression. *Complex & Intelligent Systems*.

[17] Abdulkader Helwan, Georges El Fakhri, Hadi Sasani, & Dilber Uzun Ozsahin (2018). Deep networks in identifying CT brain hemorrhage. *J. Intell. Fuzzy Syst., 35, 2215-2228.*

[18] Omar Munthir Al Okashi, Flath M. Mohammed, & Ahmed J. Aljaaf (2019). An Ensemble Learning Approach for Automatic Brain Hemorrhage Detection from MRIs. *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, 929-932.

[19] Mahmood A. Mahmood, Khalaf Alsalem, Murtada K. Elbashir, Sameh Abd El-Ghany, & A. A. Abd El-Aziz (2025). An Intracranial Brain Hemorrhage's Identification and Classification on CT Imaging using Fuzzy Deep Learning. *Int. J. Comput. Commun. Control, 20.*

[20] Sulaiman Khan, Hazrat Ali, & Zubair Shah (2023). Brain Hemorrhage Detection Using Improved AlexNet with Inception-v4. *2023 IEEE International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, 1-5.

[21] Samia Ahmed, Jannatul Ferdous Esha, Md. Sazzadur Rahman, M. Shamim Kaiser, Ieee A. S. M. Sanwar Hosen Senior Member, Deepak Ghimire, Mi-Jin Park, & Md. Sazzadur Rahman (2024). Exploring Deep Learning and Machine Learning Approaches for Brain Hemorrhage Detection. *IEEE Access, 12, 45060-45093.*

[22] Priyanka Datta, & Rajesh Rohilla (2024). An autonomous and intelligent hybrid CNN-RNN-LSTM based approach for the detection and classification of abnormalities in brain. *Multimedia Tools and Applications*, 1-27.

[23] Sumit Tanwar, Nipun Choudhary, Vaibhav Chaudhary, Anish Dahiya, Priyanka Kaushik, & Rachna Rathore (2024). Advancing Healthcare: CNN-Based Brain Hemorrhage Detection in Intelligent Environments. *2023 International Conference on Smart Devices (ICSD)*, 1-6. prioritized code clone refactoring,'' in Proc. IEEE 11th Int. Work. Conf.