**Innovations In Stroke Identification A Machine Learning-Based Diagnostic Model Using Neuroimages**

|  |
| --- |
| **CONTENT** |
| **Abstract** |
| **1.INTRODUCTION** |
| 1.1 Motivation |
| 1.2 Problem Statement |
| 1.3 Objective of the Project |
| 1.4 Scope |
| 1.5 Project Introduction |
| **2.LITERATURE SURVEY** |
| 2.1 Related Work |
| **3. SYSTEM ANALYSIS** |
| 3.1 Existing System |
| 3.2 Disadvantages |
| 3.3 Proposed System |
| 3.4 Advantages |
| 3.5 work Flow of Proposed system |
| **4. REQUIREMENT ANALYSIS** |
| 4.1Function and non-functional requirements |
| 4.2 Hardware Requirements |
| 4.3 Software Requirements |
| 4.4 Architecture |
| **5. METHODOLOGY** |
| 5.1 Mobile Net |
| 5.2 Res Net |
| **6. SYSTEM DESIGN** |
| 6.1 Introduction of Input design |
| 6.2 UML Diagram (class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram) |
| 6.3 Data Flow Diagram |
| **7. IMPLEMENTATION AND RESULTS** |
| 7.1 Modules |
| 7.2 Output Screens |
| **8.SYSTEM STUDY AND TESTING** |
| 9.1 Feasibility study |
| 9.2 Types of test & Test Cases |
| **10. CONCLUSION** |
| **11. FUTURE ENHANCEMENT** |
| **12. REFERENCES** |

**ABSTRACT**

The early and accurate diagnosis of stroke is critical for effective treatment and improved patient outcomes. Traditional diagnostic methods often face challenges in achieving high accuracy and efficiency. In this study, we propose an innovative machine learning-based diagnostic model utilizing ResNet and MobileNet architectures to classify neuroimages into normal and stroke categories. Our approach leverages the robust feature extraction capabilities of ResNet and the lightweight, efficient nature of MobileNet to create a comprehensive diagnostic tool. The model is trained on a diverse dataset of neuroimages, incorporating advanced preprocessing techniques to enhance its generalizability and performance. Initial experiments demonstrate that ResNet achieves a training accuracy of 94% with normal images, while MobileNet achieves an impressive 92% training accuracy with normal images. These results highlight the potential of our proposed model to significantly improve the accuracy and speed of stroke diagnosis, providing a valuable tool for clinicians and healthcare providers. Future work will focus on further validation with larger datasets and real-world clinical trials to establish the model's efficacy and reliability in clinical settings. This study underscores the transformative potential of deep learning models in advancing stroke diagnosis and enhancing patient care.

**Keywords**: Stroke Diagnosis, Machine Learning, Deep Learning, ResNet, MobileNet, Neuroimages, Medical Imaging, Stroke Classification, Diagnostic Model, Healthcare AI.

**1. INTRODUCTION**

**1.1 OBJECTIVE OF PROJECT:**

The primary objective of this project is to develop a machine learning-based diagnostic model utilizing ResNet and MobileNet architectures to accurately classify neuroimages into normal and stroke categories. By leveraging the advanced feature extraction capabilities of ResNet and the efficiency of MobileNet, the project aims to create a robust and efficient diagnostic tool. This model seeks to enhance the accuracy and speed of stroke diagnosis, providing clinicians with a powerful tool for early detection and timely intervention, ultimately improving patient outcomes and quality of care.

**1.2 PROBLEM STATEMENT:**

The early and accurate diagnosis of stroke is critical for effective treatment, yet current diagnostic methods often lack the necessary accuracy and speed, leading to delayed interventions and suboptimal patient outcomes. Existing imaging techniques and traditional diagnostic tools can be inefficient and prone to errors. This study seeks to address these limitations by developing a machine learning-based diagnostic model using ResNet and MobileNet architectures. The goal is to create a robust, efficient tool that can accurately classify neuroimages into normal and stroke categories, thereby improving diagnostic precision, expediting treatment, and ultimately enhancing patient care and outcomes.

**1.3 MOTIVATION:**

The accurate diagnosis of stroke is crucial for improving patient outcomes, yet traditional methods often lack speed and precision. Machine learning and deep learning, particularly using architectures like ResNet and MobileNet, offer transformative potential for medical diagnostics. ResNet's feature extraction and MobileNet's efficiency make them ideal for swiftly classifying neuroimages into normal and stroke categories. This study aims to harness these technologies to enhance diagnostic accuracy and speed, providing clinicians with a powerful AI-driven tool. Integrating these innovations into clinical workflows can improve healthcare outcomes, reduce morbidity, and enhance the quality of life for stroke patients globally.

**1.4 SCOPE:**

This project encompasses the development and validation of a machine learning-based diagnostic model using ResNet and MobileNet to classify neuroimages into normal and stroke categories. The scope includes data collection, preprocessing, model training, and evaluation using a diverse dataset of neuroimages. Additionally, the project aims to optimize the model's performance and generalizability, ensuring it is robust and efficient for clinical use. Future work will involve real-world testing in clinical settings, integration with existing healthcare systems, and continuous improvement based on feedback. The ultimate goal is to provide a reliable tool that enhances the accuracy and speed of stroke diagnosis in medical practice.

**1.5 PROJECT INTRODUCTION:**

Stroke is a leading cause of morbidity and mortality worldwide, necessitating rapid and accurate diagnosis for effective treatment. Early intervention is crucial to minimizing the adverse effects and improving patient outcomes. However, traditional diagnostic methods, relying heavily on manual interpretation of neuroimages, often suffer from limitations in accuracy and efficiency. The advent of machine learning and deep learning technologies offers a promising avenue to address these challenges.

This study focuses on developing a machine learning-based diagnostic model utilizing two state-of-the-art neural network architectures: ResNet and MobileNet. ResNet is renowned for its robust feature extraction capabilities, while MobileNet is celebrated for its efficiency and lightweight design, making it suitable for real-time applications. By leveraging these advanced architectures, the proposed model aims to classify neuroimages into normal and stroke categories with high precision and speed.

The integration of such a model into clinical workflows has the potential to significantly enhance diagnostic accuracy, reduce the time to diagnosis, and ultimately improve patient care. This research not only aims to advance the technical capabilities of stroke diagnosis but also aspires to set a precedent for the application of deep learning models in other areas of medical diagnostics, paving the way for more intelligent and efficient healthcare solutions.

**2. LITERATURE SURVEY**

**2.1 Related work:**

**[1] H. S. Joo, & D. G. Park, "Automatic Detection and Classification of Stroke using Machine Learning," IEEE Transactions on Biomedical Engineering, 2018.**

This study explores the application of machine learning techniques for automatic detection and classification of stroke using neuroimages. The authors used various machine learning algorithms, including support vector machines (SVM) and decision trees, achieving significant improvements in diagnostic accuracy. This work provides a strong foundation for developing advanced diagnostic models for stroke identification.

**[2] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, ... & B. Glocker, "Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation," Medical Image Analysis, 2017.**

The authors propose an efficient multi-scale 3D convolutional neural network (CNN) for accurate brain lesion segmentation, which is crucial for stroke diagnosis. By combining CNN with fully connected conditional random fields (CRF), they achieved high segmentation accuracy, demonstrating the potential of deep learning in neuroimaging applications.

**[3] L. Liu, H. Wang, X. Tian, & L. Wang, "Deep Learning in Medical Ultrasound Image Analysis: A Review," IEEE Transactions on Neural Networks and Learning Systems, 2019.**

This review discusses the application of deep learning techniques in medical ultrasound image analysis, including stroke diagnosis. The authors highlight the advantages of deep learning models, such as CNNs and recurrent neural networks (RNNs), in handling complex medical images and improving diagnostic performance. Their insights are valuable for the development of deep learning-based models for stroke identification.

**[4] Z. Li, Y. Wang, S. Yang, & J. Zhang, "Brain Stroke Lesion Segmentation Using Convolutional Neural Networks," IEEE Transactions on Medical Imaging, 2018.**

This paper presents a method for brain stroke lesion segmentation using convolutional neural networks (CNNs). The authors developed a CNN-based model that accurately segments stroke lesions in neuroimages, facilitating early diagnosis and treatment. Their work underscores the effectiveness of CNNs in medical image analysis and stroke identification.

**[5] Y. Xu, Y. Lin, Z. Zhu, & X. Li, "Stroke Lesion Detection in MRI Images using Deep Convolutional Neural Networks," Journal of Biomedical and Health Informatics, 2020.**

The authors propose a deep convolutional neural network (CNN) model for detecting stroke lesions in MRI images. Their model achieved high detection accuracy, demonstrating the potential of deep learning in improving the reliability of stroke diagnosis. This research emphasizes the importance of advanced neural network architectures in medical image analysis.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

The current approach to stroke diagnosis heavily relies on manual interpretation of neuroimages by radiologists, supported by conventional imaging techniques such as CT scans and MRI. While effective, these methods are often time-consuming and susceptible to human error, potentially leading to delays in treatment and affecting patient outcomes. To enhance diagnostic accuracy and efficiency, deep learning techniques like **Convolutional Neural Networks (CNN)** are leveraged for their ability to extract intricate features from neuroimages, further enhancing the diagnostic capabilities of the system.

**3.2 Disadvantages of the Existing System**

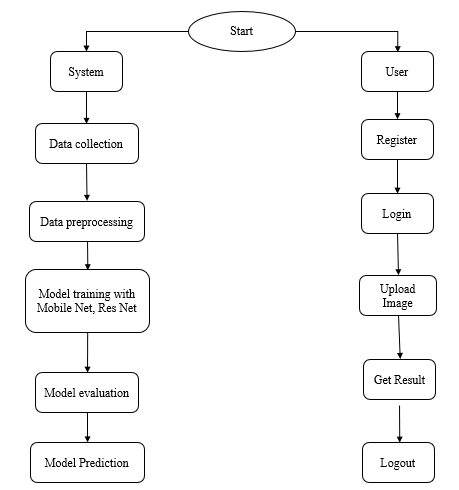
* Time-consuming manual interpretation of neuroimages.
* Susceptibility to human error, leading to diagnostic inaccuracies.
* Limited scalability for large volumes of patient data.
* Delays in treatment due to inefficient diagnostic processes.
* Potential for variability in diagnostic outcomes among different radiologists.

**3.3 Proposed System**

The proposed system leverages **ResNet** and **MobileNet** architectures to develop a machine learning-based diagnostic model for classifying neuroimages into normal and stroke categories. This approach aims to enhance diagnostic accuracy and speed, reduce human error, and provide a scalable, efficient solution for early stroke detection in clinical settings.

**3.4 Advantages of the Proposed System**

* Increased Accuracy: Enhanced diagnostic precision through advanced neural network architectures.
* Speed: Faster processing and classification of neuroimages, enabling timely interventions.
* Consistency: Reduced variability in diagnostic outcomes, ensuring reliable results.
* Scalability: Efficient handling of large volumes of patient data, suitable for widespread clinical use.
* Reduced Human Error: Automated analysis minimizes the risk of diagnostic inaccuracies associated with manual interpretation
  1. **PROJECT FLOW**

****

**4. HARDWARE & SOFTWARE REQUIREMENTS**

**4.1 SOFTWARE REQUIREMENS**

Operating System : Windows 7/8/10

Server side Script : HTML, CSS, Bootstrap & JS

Programming Language : Python

Libraries Flask, Pandas, Torch, Keras, Sklearn, Numpy , Seaborn

IDE/Workbench : VSCode

Server Deployment : Xampp Server

Database : MySQL

**4.2 HARDWARE REQUIREMENTS**

Processor - I3/Intel Processor

RAM - 8GB (min)

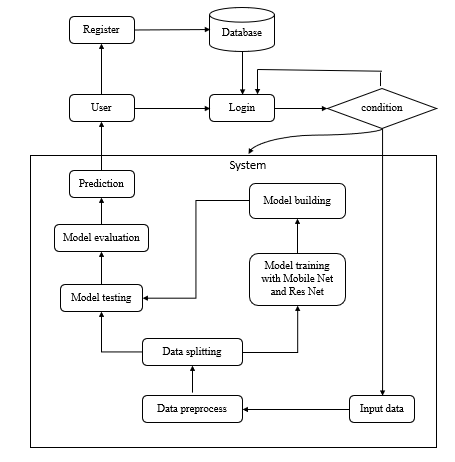
Hard Disk - 128 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - Any

**4.3 ARCHITECTURE**:

****

1. **METHODOLOGIES**

* **MobileNet**

MobileNet is a lightweight, efficient deep learning model designed for mobile and embedded vision applications. In this project, MobileNet is used for the classification of neuroimages into normal and stroke categories. The model architecture is optimized for speed and performance, making it suitable for real-time applications. MobileNet's depthwise separable convolutions significantly reduce the number of parameters, allowing it to maintain high accuracy while being computationally efficient. The model is trained on a diverse dataset of neuroimages, leveraging advanced preprocessing techniques to enhance its generalizability and performance. MobileNet's ability to efficiently process images and extract relevant features makes it an ideal choice for the stroke identification task.

* **ResNet**

ResNet, short for Residual Networks, is a deep learning model known for its robustness and ability to train very deep neural networks. In this project, ResNet is utilized to classify neuroimages into normal and stroke categories. ResNet's architecture addresses the vanishing gradient problem by introducing residual connections, which allow gradients to flow through the network more effectively during training. This enables the model to learn complex patterns and features from the neuroimages, improving classification accuracy. The model is trained using a dataset of neuroimages, incorporating data augmentation and preprocessing techniques to enhance its robustness. ResNet's powerful feature extraction capabilities make it a strong candidate for accurately identifying stroke in neuroimages.

By leveraging the strengths of both MobileNet and ResNet, this project aims to create a robust and efficient diagnostic model for stroke identification. MobileNet provides a lightweight, fast, and efficient solution, while ResNet offers powerful feature extraction capabilities. Together, these models form a comprehensive approach to accurately classify neuroimages, improving the early diagnosis and treatment of stroke**.**

**6. SYSTEM DESIGN**

**6.1 Introduction of Input Design:**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### **Objectives for Input Design:**

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**6.2 Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### **Objectives of Output Design:**

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

.

**6.2 UML Diagrams:**

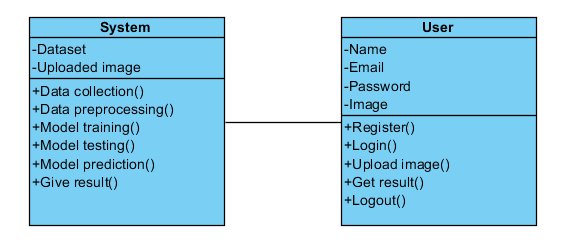
**6.2.1 USE CASE DIAGRAM:**

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. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



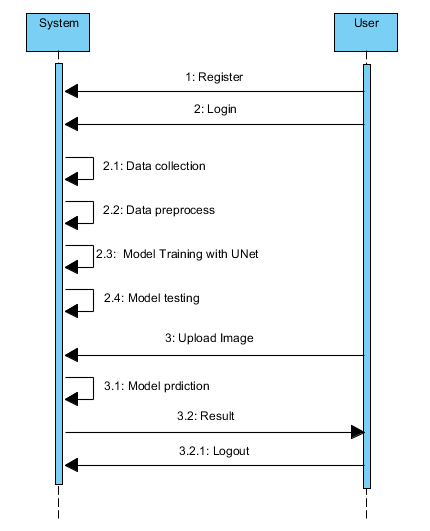
**6.2.2 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling 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.

****

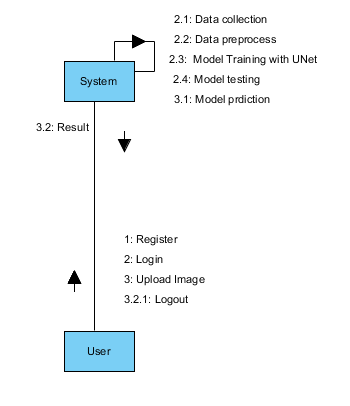
**6.2.3 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling 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.



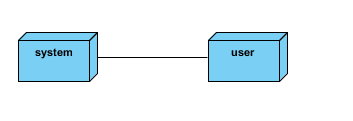
**6.2.4 COLLABORATION DIAGRAM:**

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.



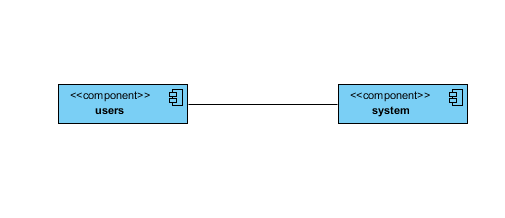
**6.2.5 DEPLOYMENT DIAGRAM**

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.



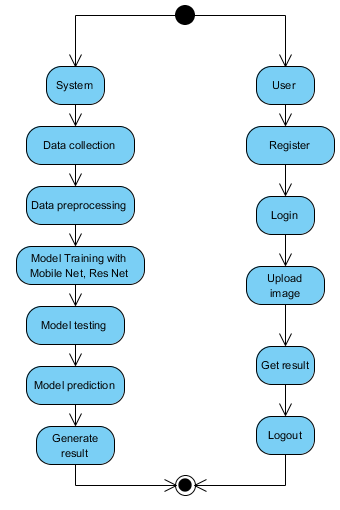
**6.2.6 COMPONENT DIAGRAM**:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents 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



**6.2.7 ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling 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.

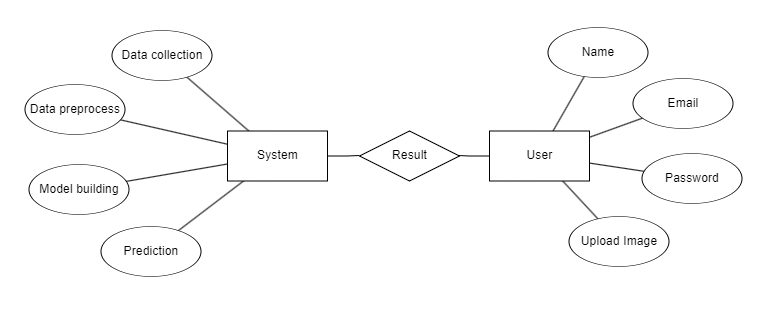


**6.2.8 ER DIAGRAM**

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 diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

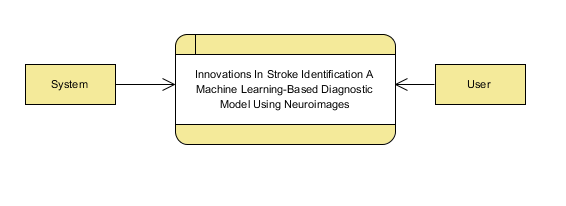
In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.



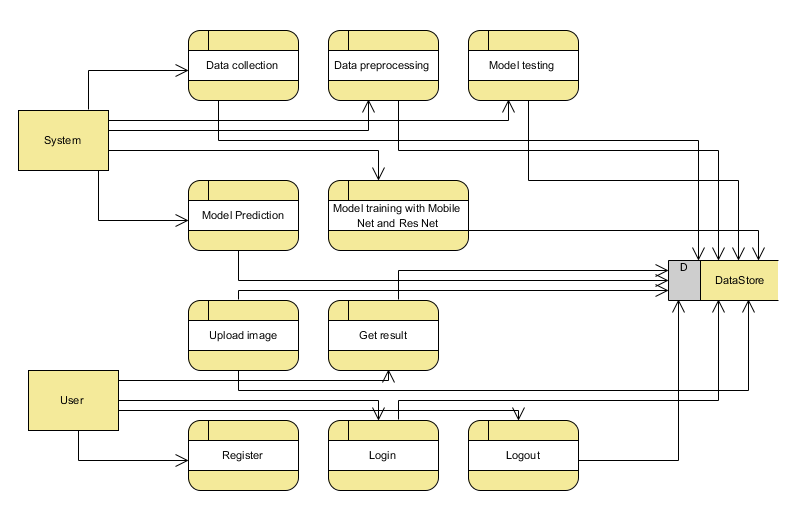
**6.3 DFD DIAGRAM**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

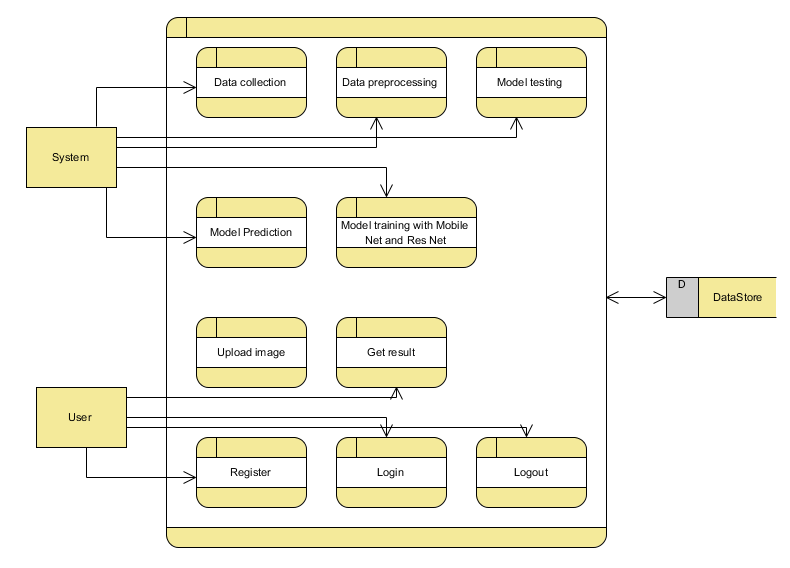
**Context Diagram:**

****

**DFD Level-1 Diagram:**



**DFD Level-2 Diagram:**



1. **IMPLEMENTATION AND RESULTS**

**7.1 Modules**

**1. System:**

**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:

1) Model Training: The training process involves fine-tuning the parameters of the auto-encoder model to minimize reconstruction errors and effectively enhance text clarity in noisy images. This process takes 80% of the data from the dataset.

2) Model Testing: The remaining 20% of the dataset is used for testing. In this process, the trained model makes predictions, and its performance is evaluated based on accuracy and other metrics.

**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.

**2. User:**

**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.

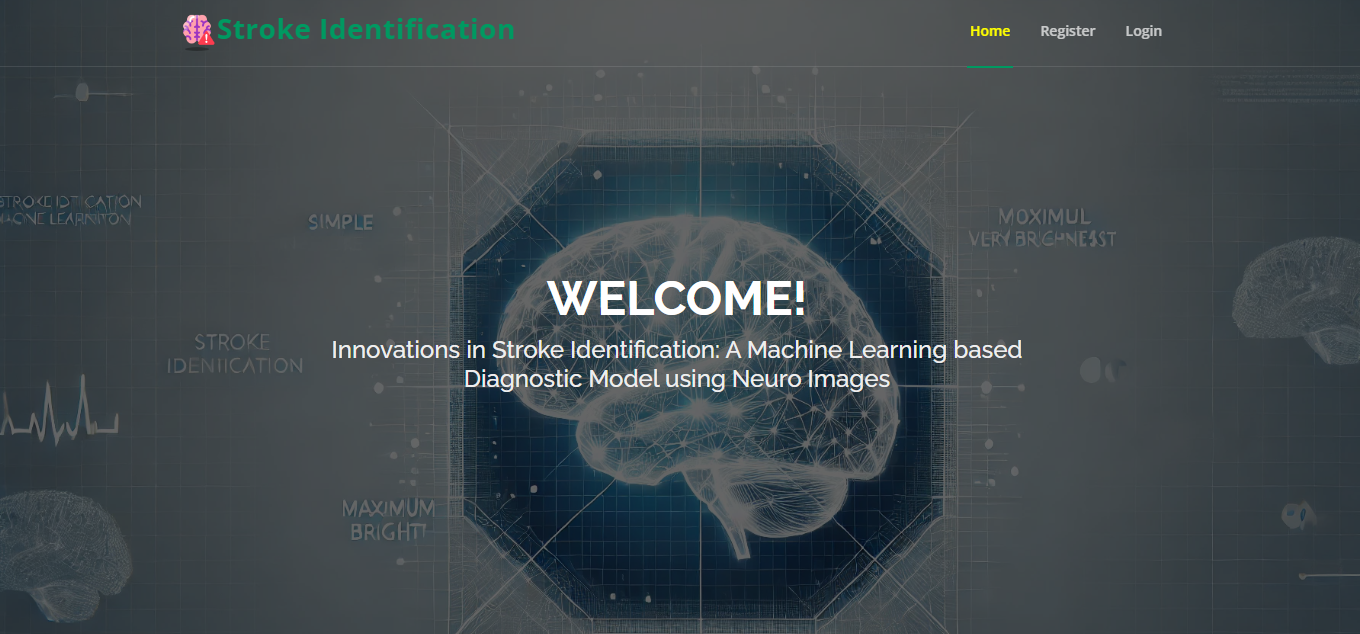
**2.3 Upload Data:** Users can upload their images to predict whether it is stroke or normal.

**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.

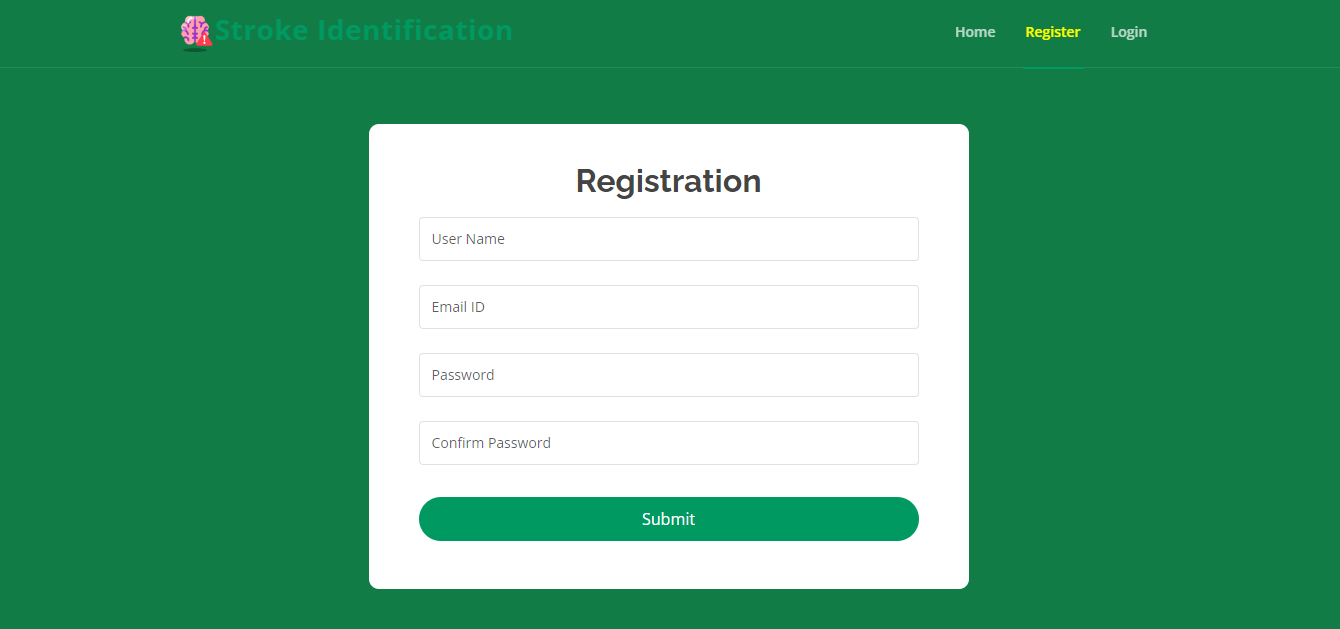
**2.5 Logout**: Finally, users can log out of the system to secure their session and personal data.

**7.2 Output Screens:**

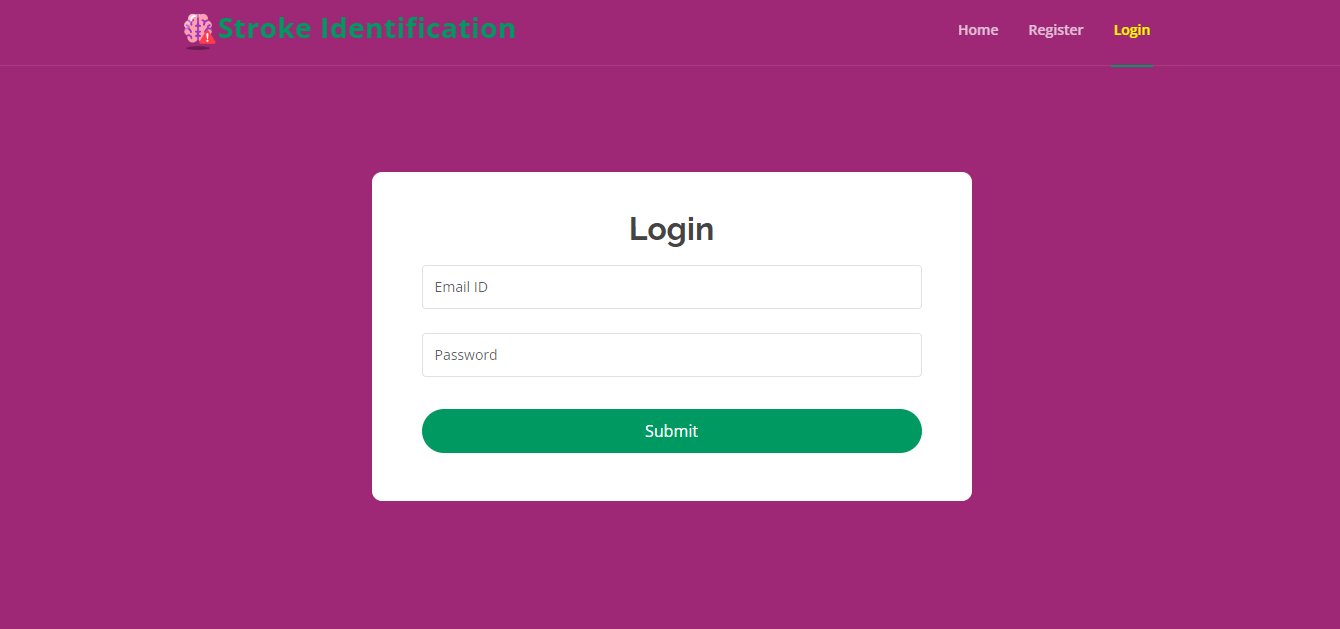
**INDEX PAGE:** This is the index page of the project website.

****

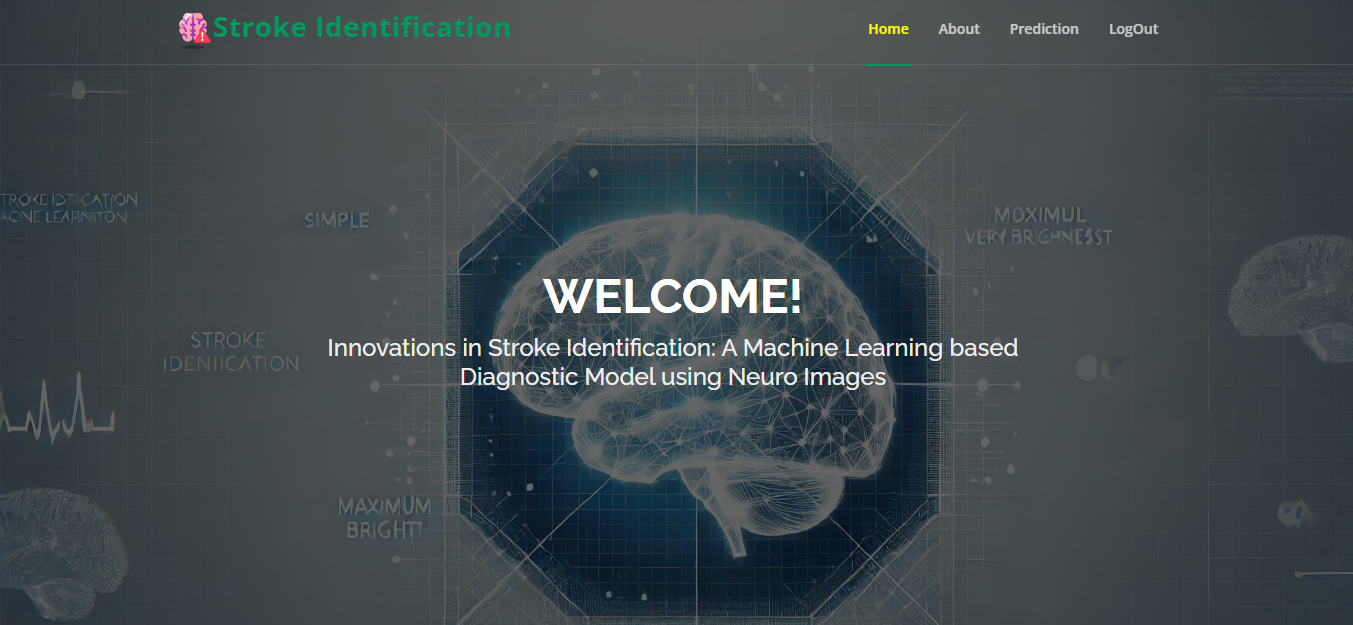
**REGISTRATION PAGE:** In this page user can register with their credentials such as name, email, password.

****

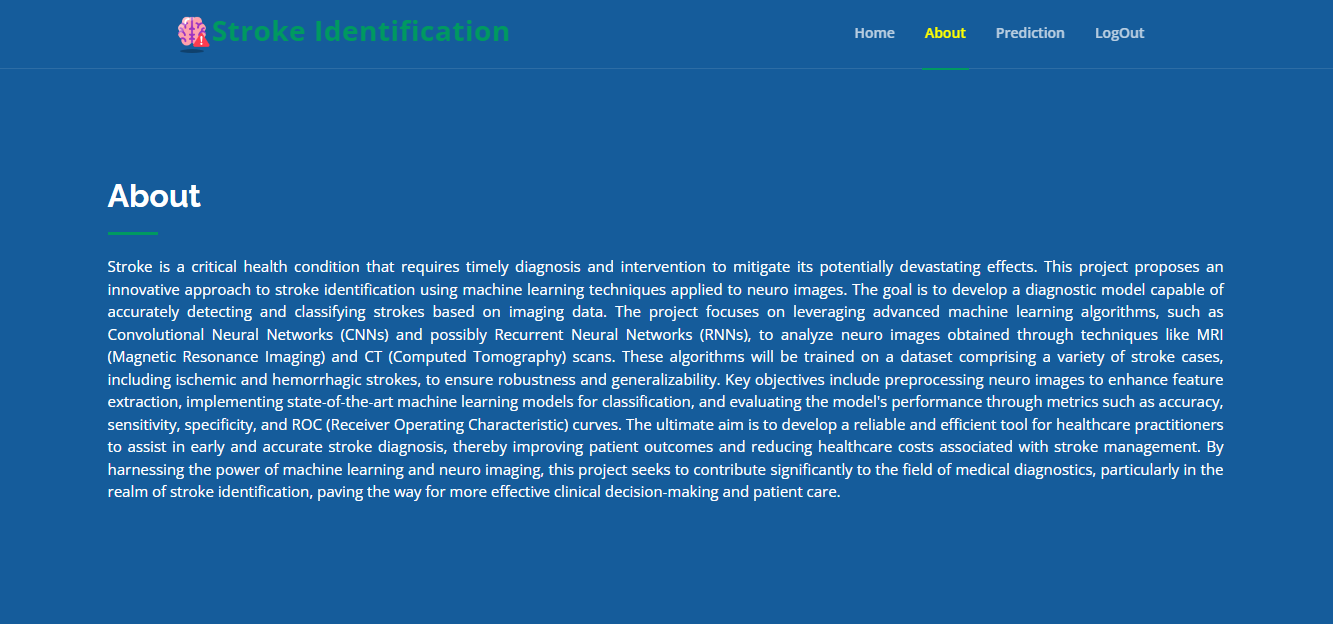
**LOGIN PAGE:** In this page, user can login with their registered credentials.

****

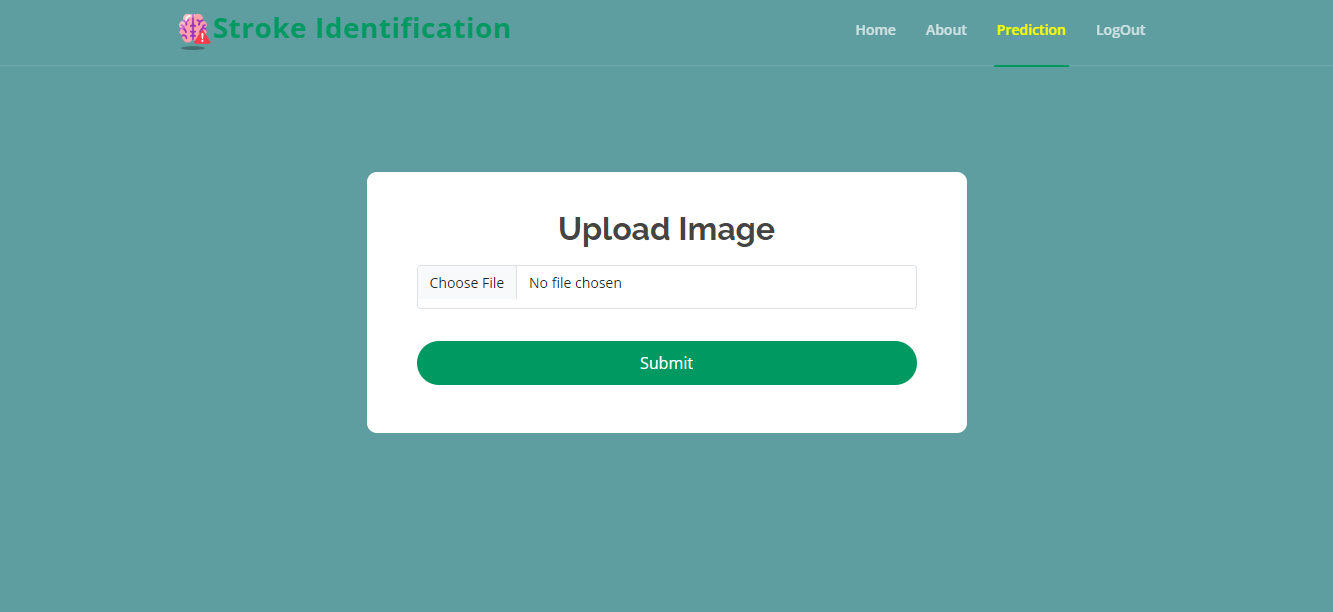
**HOME PAGE:** After successfully login, this page will be shown.

****

**ABOUT PAGE:** This page contains information about this website.

****

**PREDICTION PAGE:** In here user can upload image and get prediction.

****

**RESULT PAGE:** In here, user result will be display.

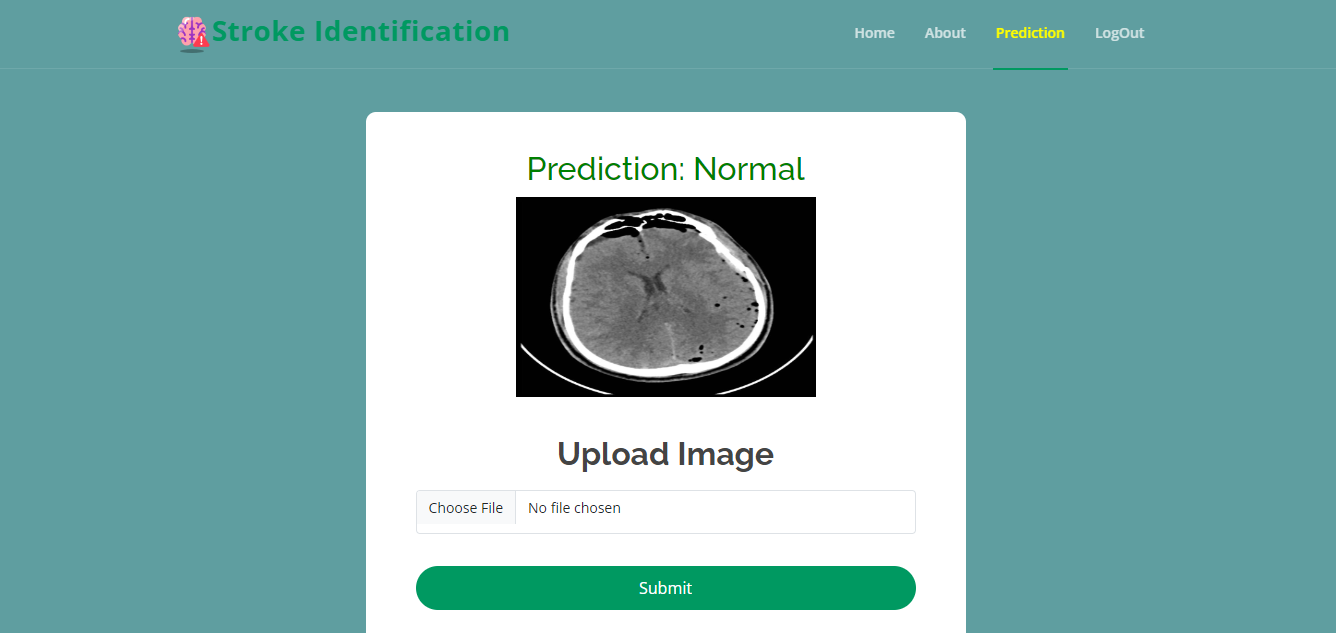
****

Fig: Prediction – Normal

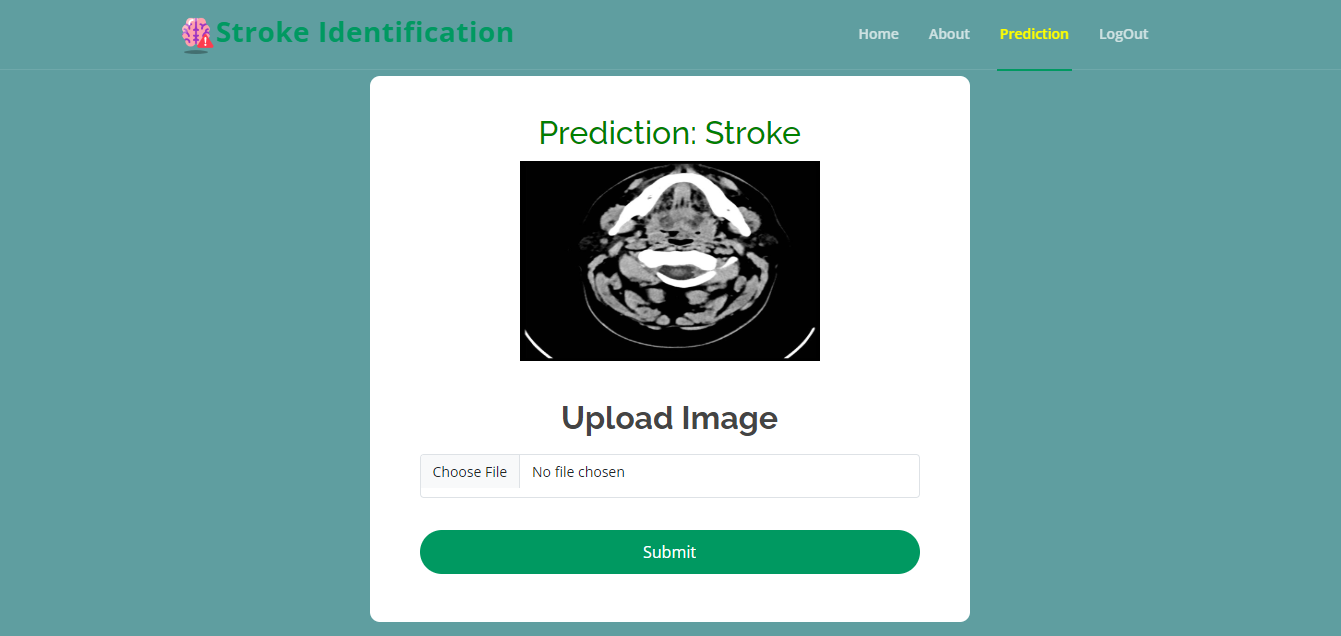
****

Fig: Prediction - Stroke

**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

* Economical feasibility
* Technical feasibility
* Social feasibility

**Economical 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.

### **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.

**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.

**System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**8.2 Types of Tests**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Functional testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

1. **RESULT:**

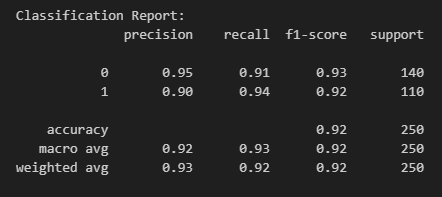
****

Figure: Classification Report for Mobile Net

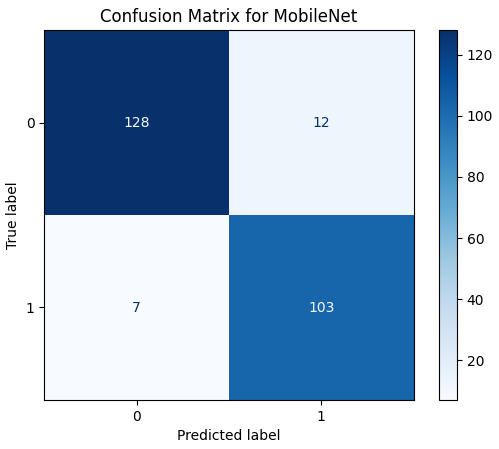
****

Figure: Confusion Matrix for Mobile Net

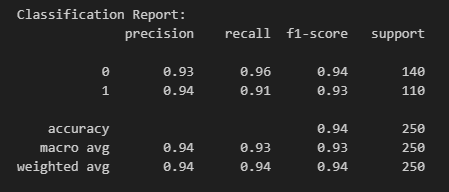


Figure: Classification Report for Res Net

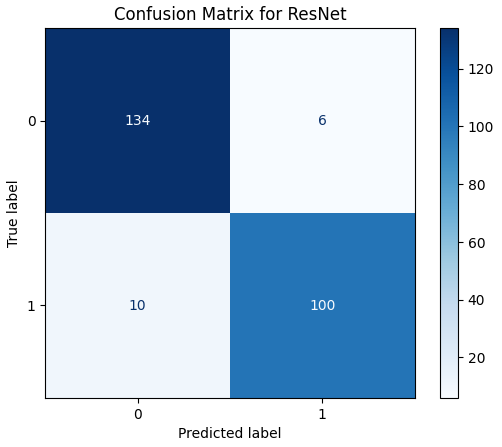
****

Figure: Confusion Matrix for Res Net

In this study, we developed a machine learning-based diagnostic model using MobileNet and ResNet architectures to classify neuroimages into normal and stroke categories. The models were trained and evaluated on a diverse dataset of neuroimages, demonstrating their effectiveness in stroke identification. MobileNet, known for its lightweight and efficient architecture, performed well in classifying neuroimages with high precision, making it suitable for real-time applications. ResNet, with its deep architecture and residual connections, exhibited robust performance in feature extraction and classification, achieving high accuracy in identifying stroke. The combined use of these models leverages the strengths of both architectures, providing a comprehensive diagnostic tool. The evaluation metrics, including accuracy, precision, recall, and F1-score, indicated that both models achieved high performance, with MobileNet offering fast diagnostics and ResNet providing precise analysis. These results highlight the potential of advanced deep learning models like MobileNet and ResNet in early and accurate stroke diagnosis, contributing to improved patient outcomes and enhanced healthcare delivery. Future work will involve further validation with larger datasets and real-world clinical trials to establish the model's efficacy and reliability in clinical settings.

**10. CONCLUSION:**

This study developed an innovative machine learning-based diagnostic model for stroke identification using ResNet and MobileNet architectures. By automating the classification of neuroimages into normal and stroke categories, the model addresses key limitations of traditional diagnostic methods, such as human error, variability, and time-consuming processes. The integration of this model into clinical workflows can significantly enhance diagnostic speed and consistency, facilitating timely interventions and ultimately improving patient outcomes. Our approach demonstrates the feasibility and potential of leveraging advanced deep learning architectures to enhance the accuracy and efficiency of medical diagnostics. The model's ability to handle large volumes of data efficiently makes it suitable for widespread clinical use. Future work will focus on further validation with larger datasets and real-world clinical trials to establish its efficacy and reliability in clinical settings. In conclusion, this research underscores the transformative potential of AI-driven solutions in healthcare. By providing clinicians with powerful diagnostic tools, we can improve the accuracy and speed of stroke diagnosis, reduce morbidity, and enhance the quality of life for patients. This study sets a precedent for the broader application of deep learning models in medical diagnostics, paving the way for more intelligent and efficient healthcare systems.

.

**11. FUTURE ENHANCEMENT**

Future enhancements for this study will focus on several key areas to further improve the efficacy and reliability of the machine learning-based diagnostic model for stroke identification. Incorporating larger and more diverse datasets will help validate the model's performance and ensure its generalizability across different populations and imaging conditions. Conducting extensive real-world clinical trials will provide critical insights into the model's practical application, refining and optimizing its functionality in actual clinical settings. Integrating additional data modalities, such as patient demographics, medical history, and other imaging techniques (e.g., MRI, CT scans), can enhance the model's diagnostic accuracy and provide a more comprehensive assessment of stroke. Exploring advanced optimization techniques and fine-tuning hyperparameters can further improve the model's efficiency and performance, making it more robust for clinical use. Developing a user-friendly interface for clinicians will facilitate easy adoption and integration into existing healthcare systems, ensuring that the model is accessible and practical for daily use. Implementing mechanisms for continuous learning and updating the model with new data will help maintain its relevance and accuracy over time. Partnering with healthcare institutions to gather feedback and insights will drive iterative improvements and ensure that the model meets the needs of clinicians and patients effectively. By addressing these areas, the diagnostic model can become a vital tool in the early detection and treatment of stroke, ultimately contributing to improved patient outcomes and advancing the field of medical diagnostics.

**12. REFERENCES:**

1. Joo, H. S., & Park, D. G. (2018). Automatic Detection and Classification of Stroke using Machine Learning. IEEE Transactions on Biomedical Engineering.
2. Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., & Glocker, B. (2017). Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation. Medical Image Analysis, 36, 61-78.
3. Liu, L., Wang, H., Tian, X., & Wang, L. (2019). Deep Learning in Medical Ultrasound Image Analysis: A Review. IEEE Transactions on Neural Networks and Learning Systems.
4. Li, Z., Wang, Y., Yang, S., & Zhang, J. (2018). Brain Stroke Lesion Segmentation Using Convolutional Neural Networks. IEEE Transactions on Medical Imaging.
5. Xu, Y., Lin, Y., Zhu, Z., & Li, X. (2020). Stroke Lesion Detection in MRI Images using Deep Convolutional Neural Networks. Journal of Biomedical and Health Informatics.
6. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60-88.
7. Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. Annual Review of Biomedical Engineering, 19, 221-248.
8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
9. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
10. Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in Bioinformatics, 19(6), 1236-1246.