International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Automated Brain Stroke Detection Using Image Processing

A Vasavi Sujatha[1], Pinjara Akshita[2], Poreddy Harthika[3], Godavari Manyasree[4]
[1]Assistant Professor; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.
[2,3,4]B.Tech Students; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.
Mail Id; akshitapinjere@gmail.com[2], harthikaporeddy2@gmail.com[3], maaanyasri@gmail.com[4]

Abstract
Stroke remains one of the leading causes of mortality and long-term disability across the globe. It occurs when the blood supply to a region of the brain is interrupted, resulting in neuronal damage and functional impairment. Rapid identification of stroke is critical for minimizing neurological damage and improving patient prognosis. Traditional diagnostic procedures primarily rely on expert radiologists to manually interpret brain Magnetic Resonance Imaging (MRI) scans. This manual evaluation is often time-consuming, expensive, and susceptible to inter-observer variability. Furthermore, in rural and resource-limited healthcare facilities, the shortage of experienced specialists frequently leads to delayed diagnosis and treatment.To address these challenges, this work proposes an Artificial Intelligence (AI)-based framework for automated detection of stroke-affected regions in brain MRI images. The system integrates image processing techniques with machine learning algorithms to analyze medical images and detect abnormal tissue patterns associated with stroke. By automating the identification process, the proposed approach aims to enhance diagnostic speed and accuracy while reducing dependence on manual interpretation.The developed model assists healthcare professionals in early decision-making by providing reliable and consistent detection results. Consequently, the system supports timely medical intervention, reduces diagnostic workload, and improves clinical outcomes. Overall, this AI-driven solution offers a promising tool for efficient stroke screening, particularly in underserved regions, thereby contributing to improved patient recovery and healthcare delivery.

Keywords-*Stroke Detection, Brain MRI, Machine Learning, Image Processing, Artificial Intelligence, Medical Image Analysis, Early Diagnosis, Healthcare Technology, Automated Detection, Deep Learning*

Introduction

Stroke is a life-threatening neurological disorder and a major contributor to mortality and long-term disability worldwide. It occurs when the blood supply to the brain is either obstructed or ruptured, preventing brain cells from receiving adequate oxygen and nutrients, which results in rapid neuronal damage. Stroke is commonly categorized into two primary types: ischemic stroke, which is caused by blockage of blood vessels, and hemorrhagic stroke, which occurs due to bleeding within the brain. Both types demand immediate diagnosis and prompt medical intervention to minimize neurological impairment and improve patient survival ratesMedical imaging modalities such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) scans play a vital role in stroke diagnosis. These imaging techniques provide detailed structural information about brain tissues and enable clinicians to identify abnormal regions associated with stroke. However, conventional diagnosis largely relies on manual interpretation of MRI and CT images by experienced radiologists. This process is often time-intensive, expensive, and subject to inter-observer variability, particularly when abnormalities are subtle. Furthermore, healthcare facilities in rural and resource-constrained areas frequently face shortages of skilled professionals, leading to delays in diagnosis and treatment.Recent advancements in image processing and deep learning offer promising solutions for automated medical image analysis. These techniques can enhance image quality, remove noise, extract meaningful features, and detect abnormal patterns with improved accuracy. This project focuses on developing an intelligent system capable of analyzing brain MRI and CT images to automatically identify stroke-affected regions. By assisting clinicians in rapid and reliable decision-making, the proposed approach aims to reduce diagnostic delays, minimize workload, and improve patient outcomes through early detection and timely intervention.

Existing System

Currently, stroke detection primarily depends on manual analysis of MRI and CT images performed by expert radiologists. This procedure involves examining multiple image slices, comparing tissue structures, and identifying variations in intensity and morphology that indicate stroke. Although widely adopted, this method has several limitations. The process is time-consuming and may delay treatment in emergency situations. Additionally, diagnostic accuracy can vary depending on the radiologist's experience and level of concentration, which may result in missed detections or inconsistent interpretations.Hospitals also handle large volumes of medical images daily, increasing the workload on radiologists and further slowing down the diagnostic process. In rural and under-equipped healthcare centers, the shortage of trained professionals and advanced infrastructure often leads to delayed diagnosis and treatment. These challenges highlight the need for automated systems capable of assisting clinicians in faster and more reliable stroke detection.

Problems in Existing System

• Manual analysis of MRI and CT images is time-consuming, particularly in emergency stroke cases where rapid diagnosisis essential.
• Diagnostic accuracy depends heavily on the radiologist's expertise and concentration, leading to potential human errors.
• Advanced imaging equipment and skilled professionals increase the overall cost of stroke diagnosis.
• Limited availability of trained specialists in rural and remote areas delays timely treatment.
• Large volumes of medical data increase the workload on radiologists, affecting efficiency and consistency.

Proposed System

The proposed system introduces an automated stroke detection framework using image processing and deep learning techniques to analyze both MRI and CT brain images. The workflow begins with data acquisition from standard datasets, followed by preprocessing steps including noise removal, normalization, and contrast enhancement to improve image quality. To overcome limited data availability and enhance model generalization, augmentation techniques such as rotation, flipping, scaling, and zooming are applied.Next, segmentation is performed to isolate regions of interest, particularly stroke-affected areas. Deep learning architectures such as U-Net are utilized for accurate segmentation due to their effectiveness in medical image analysis. Feature extraction is then conducted to capture relevant characteristics including texture, intensity, and structural variations.The extracted features are used for classification through machine learning and deep learning models such as Support Vector Machine (SVM), Convolutional Neural Networks (CNN), and MobileNet. These models classify input images into stroke and non-stroke categories with improved accuracy. Finally, the system produces visual outputs by highlighting detected stroke regions using colored overlays, enabling clinicians to interpret results quickly.

Literature Survey

Stroke detection using medical imaging has attracted significant research attention over the years. Early approaches relied primarily on manual interpretation of MRI and CT scans performed by radiologists. Feature extraction methods have been widely explored to enhance classification accuracy. Texture-based features such as Gray Level Co-occurrence Matrix (GLCM), Local Binary Patterns (LBP), and intensity histograms have proven effective in capturing structural differences between normal and abnormal brain tissues. These features provide meaningful inputs to machine learning algorithms for distinguishing stroke-affected regions. Studies have also shown that combining multiple feature types improves robustness and reliability in clinical applications.Recent literature further emphasizes system scalability and adaptability. As medical datasets grow and imaging technologies evolve, automated systems must support retraining and continuous learning. Incorporating retraining mechanisms helps maintain accuracy across diverse patient populations and imaging conditions. Hybrid deep learning architectures combining CNNs with models such as U-Net, autoencoders, or recurrent networks have also demonstrated improved localization accuracy and reduced false detections. Additionally, cloud-based and real-time diagnostic systems are gaining attention, enabling remote upload of MRI and CT images for rapid analysis. These solutions significantly improve accessibility in resource-limited regions.

Requirement Analysis

Requirement analysis is a crucial stage in system development that defines both functional and non-functional specifications required for implementing the proposed brain stroke detection system. This phase ensures that the system satisfies user expectations while delivering accurate and reliable results. The proposed framework integrates image processing and deep learning techniques to detect stroke-affected regions in brain MRI and CT images. The system includes modules for image acquisition, preprocessing, data augmentation, segmentation, feature extraction, classification, and visualization. Advanced models such as Convolutional Neural Networks, Support Vector Machines, U-Net, and MobileNet are incorporated to enhance detection accuracy and computational efficiency.

Functional Requirements

The functional requirements describe the essential operations performed by the system. Initially, the system allows users to upload MRI or CT images, which are then preprocessed to improve quality through resizing, normalization, and noise reduction. These preprocessing steps ensure consistency across images and enhance analysis accuracy. Following preprocessing, segmentation techniques are applied to divide the image into meaningful regions and identify potential stroke-affected areas. Feature extraction methods are used to capture relevant information such as texture, intensity, and structural patterns.The system then performs model training and classification using

ISSN 2347–3657

Volume 14, Issue 1, 2026

deep learning architectures such as CNN and MobileNet. The trained models classify input images into stroke and non-stroke categories. Once classification is complete, the system displays the prediction result along with a confidence score. To improve interpretability, visualization techniques highlight affected regions using colored overlays or heatmaps. Additionally, the system supports dataset management and model retraining, allowing new data to be incorporated to improve performance over time.

Non-Functional Requirements

Non-functional requirements define quality attributes necessary for efficient system operation. The system must deliver high performance by processing medical images quickly while maintaining accuracy. Scalability is required to handle large datasets and increasing user demand without performance degradation. Security measures must be implemented to protect sensitive medical data and prevent unauthorized access. Reliability is essential to ensure consistent operation and accurate predictions. The system should also provide user-friendly interfaces that simplify image upload and result interpretation. Transparency is another important requirement, enabling users to understand how predictions are generated, thereby improving trust in the system.

Tools and Technology Requirements

The proposed system is developed using Python as the primary programming language, along with deep learning frameworks such as TensorFlow and Keras for model training. Image processing tasks are performed using OpenCV and Pillow, while NumPy and Pandas handle numerical computations and dataset management. MATLAB may also be used for advanced image visualization and analysis. The web interface is implemented using Flask or Django, and databases such as SQLite or MySQL are used for storing images and results.

Deep learning technologies include TensorFlow for model development, Keras for building CNN architectures, and MobileNetV2 as a pretrained model for efficient classification. Backend development is handled using Python and Flask, while SQLite or MySQL manage database operations. Image preprocessing utilizes OpenCV, Pillow, and NumPy libraries. Data analysis and visualization are performed using Pandas and Matplotlib. Data augmentation techniques such as rotation, flipping, and scaling are applied to improve model generalization.

Frontend technologies include HTML, CSS, and JavaScript for designing the user interface. The system is developed using Visual Studio Code as the primary development environment. Model training is integrated with Flask to enable real-time predictions. Data augmentation is incorporated during training to enhance accuracy and reduce overfitting.

Design

System Architecture

The proposed brain stroke detection system is designed using a hybrid deep learning architecture that processes brain MRI and CT images to identify stroke-affected regions. The architecture begins with an input layer that receives medical imaging datasets containing structural and pathological information of the brain. These images undergo preprocessing, which includes resizing to uniform dimensions, noise reduction using filtering techniques, and normalization to ensure consistency across the dataset. During this stage, relevant features such as texture, edges, and abnormal intensity variations are emphasized while redundant information is minimized, improving the quality of data used for further analysis.

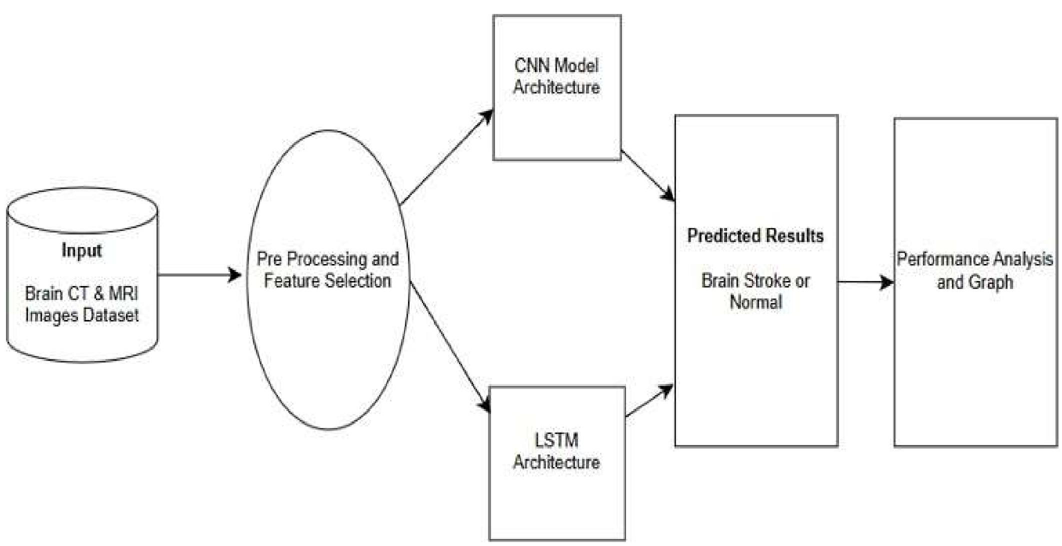


Fig 1 System Architecture

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

Following preprocessing, the system employs a hybrid deep learning framework consisting of parallel Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models. The CNN component focuses on spatial feature extraction by learning patterns related to shapes, textures, and lesion boundaries through convolutional and pooling layers. This allows the model to effectively identify stroke-related abnormalities within individual images. In parallel, the LSTM network is used to capture sequential dependencies when analyzing multiple image slices or time-based variations. This is particularly beneficial when working with volumetric MRI or CT scan sequences where contextual relationships between slices provide additional diagnostic information.

The outputs generated by the CNN and LSTM models are combined using a fusion mechanism to produce a final classification result. The integrated model predicts whether the input image corresponds to a stroke-affected brain or a normal condition. This hybrid approach improves detection performance by leveraging both spatial and sequential learning capabilities. Finally, performance evaluation is conducted using metrics such as accuracy, precision, recall, and loss. These metrics are visualized using graphs to assess model performance and guide further optimization, ensuring reliable stroke detection.

Technical Architecture

The technical architecture of the proposed system follows a three-tier structure consisting of the front-end interface, backend processing module, and database layer. The front-end is developed using HTML, CSS, and JavaScript, which provides an interactive interface for users to upload MRI or CT images and view prediction results. This layer is responsible for managing user input and communicating with the backend server.

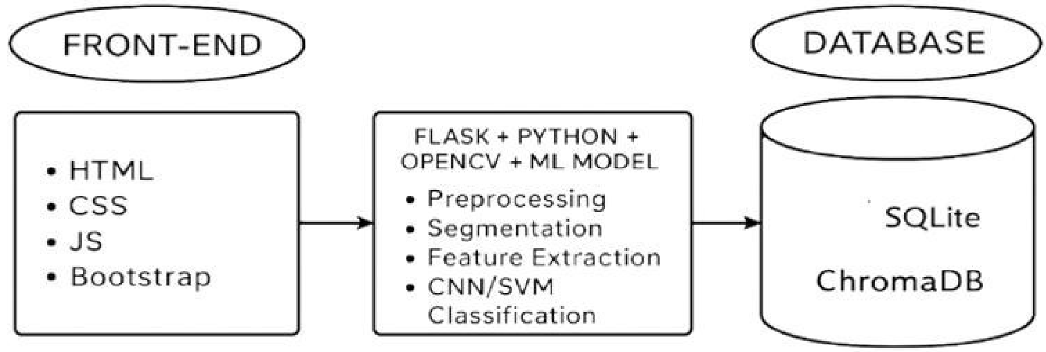


Fig 2 Technical Architecture

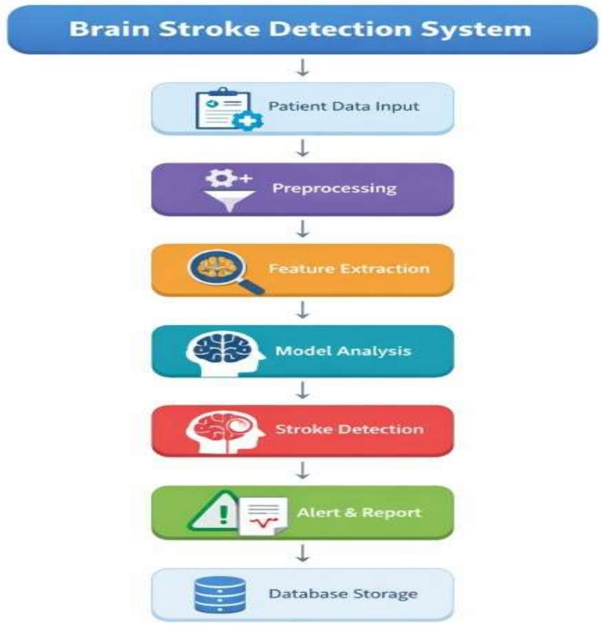
The backend is implemented using Python and Flask, which handle image processing and model inference. Once an image is uploaded, preprocessing steps such as resizing, normalization, and noise removal are applied using OpenCV and related libraries. After preprocessing, segmentation techniques are employed to isolate regions of interest within the brain. Feature extraction methods are then applied to capture important patterns, including texture and structural abnormalities. These features are passed to classification models such as CNN or SVM, which determine whether the input image indicates stroke or normal condition.

The database layer is responsible for storing system data and results. SQLite is used to manage structured data such as uploaded images, prediction outputs, and system logs. Additionally, advanced storage solutions such as ChromaDB may be used to store feature embeddings generated during model processing. This layered architecture enables efficient data handling, improves scalability, and supports future system enhancements. Overall, the technical architecture integrates web technologies with deep learning models to provide an end-to-end automated stroke detection solution.

Workflow Diagram

The workflow of the proposed stroke detection system begins with image acquisition, where MRI or CT scan images are collected from datasets or uploaded by users. These images are passed to the preprocessing stage, where noise removal, normalization, and resizing are performed to enhance image quality. Data augmentation techniques may also be applied to increase dataset diversity and improve model generalization.

After preprocessing, segmentation is carried out to identify the region of interest containing potential stroke-affected areas. The segmented images are then processed for feature extraction, where relevant characteristics such as texture, intensity distribution, and structural patterns are obtained. These features are used as inputs for classification models, including CNN, MobileNet, or hybrid architectures, which analyze the images and classify them as stroke or non-stroke.



Fig; 3 Work Flow Diagram

Following classification, the system generates visual outputs by highlighting detected stroke regions using overlays or heatmaps. The prediction results, along with confidence scores, are displayed to the user through the web interface. Finally, the system stores the results in the database for future reference and supports model retraining using newly added data. This workflow ensures efficient, accurate, and automated stroke detection while assisting healthcare professionals in decision-making.

Methodology

The proposed brain stroke detection system follows a structured artificial intelligence–based development methodology. The process begins with problem identification, where the need for early and accurate stroke detection was analyzed. Conventional diagnostic approaches depend heavily on manual interpretation of MRI and CT images by medical experts, which may lead to delays in treatment. To address this issue, the proposed system utilizes deep learning–based medical image analysis to automate stroke detection.

Brain MRI and CT scan images are collected from standard datasets and undergo preprocessing operations including resizing, normalization, and augmentation .The system is evaluated through functional testing, model performance analysis, and user testing. Performance metrics such as accuracy, precision, recall, and loss are monitored to validate reliability. Optimization techniques including hyperparameter tuning and model refinement are applied to improve detection accuracy. Future enhancements include real-time detection capabilities, cloud-based deployment, and integration with hospital information systems.

Modules

The proposed system is divided into several functional modules, each responsible for a specific task within the overall stroke detection workflow. These modules include image preprocessing, segmentation, feature extraction, classification, result visualization, and administrative management. This modular design improves system organization, scalability, and maintainability.The classification module applies machine learning and deep learning models such as CNN or SVM to categorize images as stroke or normal. The result visualization module presents predictions by highlighting affected regions using overlays and displaying confidence scores. The administrative module supports dataset management, model retraining, and system monitoring. This module allows new data to be incorporated and ensures continuous improvement in model performance.

Implementation

The brain stroke detection system is implemented using Python, which provides flexibility for machine learning and image processing applications. Deep learning frameworks such as TensorFlow and Keras are used to design and train the CNN model. Image processing tasks are performed using OpenCV and Pillow, which handle resizing, normalization, and enhancement of MRI and CT images. The backend is developed using Flask, which manages image uploads, prediction requests, and communication with the trained model. Databases such as MySQL or SQLite are used for storing user data, prediction results, and system logs.Python serves as the primary programming language for model development and backend integration. model inference, and database interaction. The system can be deployed locally or on cloud platforms to enable

ISSN 2347–3657

Volume 14, Issue 1, 2026

real-time stroke detection. Security mechanisms such as authentication and encrypted data storage are incorporated to protect user information. Supporting libraries including NumPy, Pandas, Matplotlib, and OpenCV improve computational efficiency and visualization of training performance.

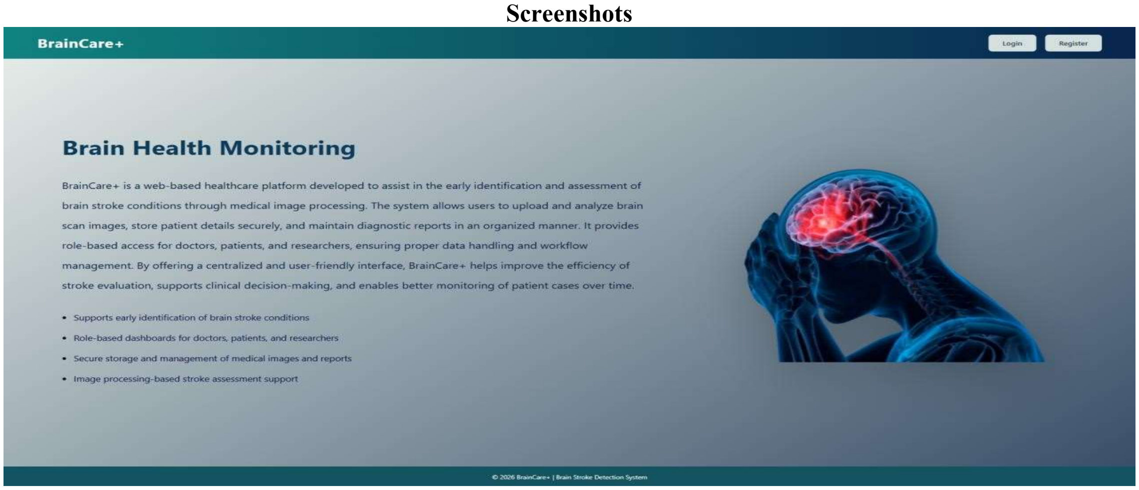
Pseudo Code

The system implementation follows a structured pseudocode workflow that begins with importing required modules for image processing, model loading, database connectivity, and web integration. User management functions handle registration, login, and session control. The preprocessing module loads input images, resizes them to standard dimensions, normalizes pixel values, and removes noise. The segmentation module converts images to grayscale, applies filtering, detects edges, and extracts regions of interest.

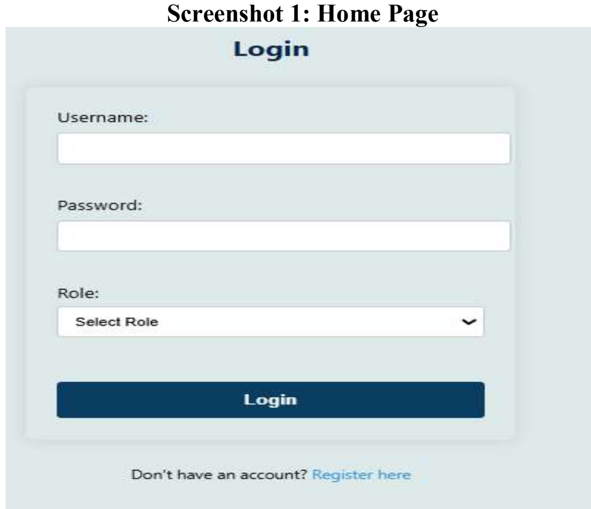
Feature extraction functions compute texture, intensity, and shape-based attributes and combine them into feature vectors. The classification module loads the trained model, preprocesses the input image, and predicts whether stroke is present. If stroke is detected, a highlighting module generates activation maps and overlays them on the original image to visualize affected regions. The upload and prediction module handles file upload, preprocessing, classification, and database storage of results. Visualization functions display prediction outputs along with highlighted images.

Database modules manage storage of patient details, prediction results, and uploaded images. The model training module loads datasets, trains the CNN architecture using training and validation data, and returns the trained model. Finally, the evaluation module computes validation accuracy and loss to assess model performance. This structured pseudocode ensures modular implementation and efficient system operation.

Screenshots



Screenshot 1: Home Page

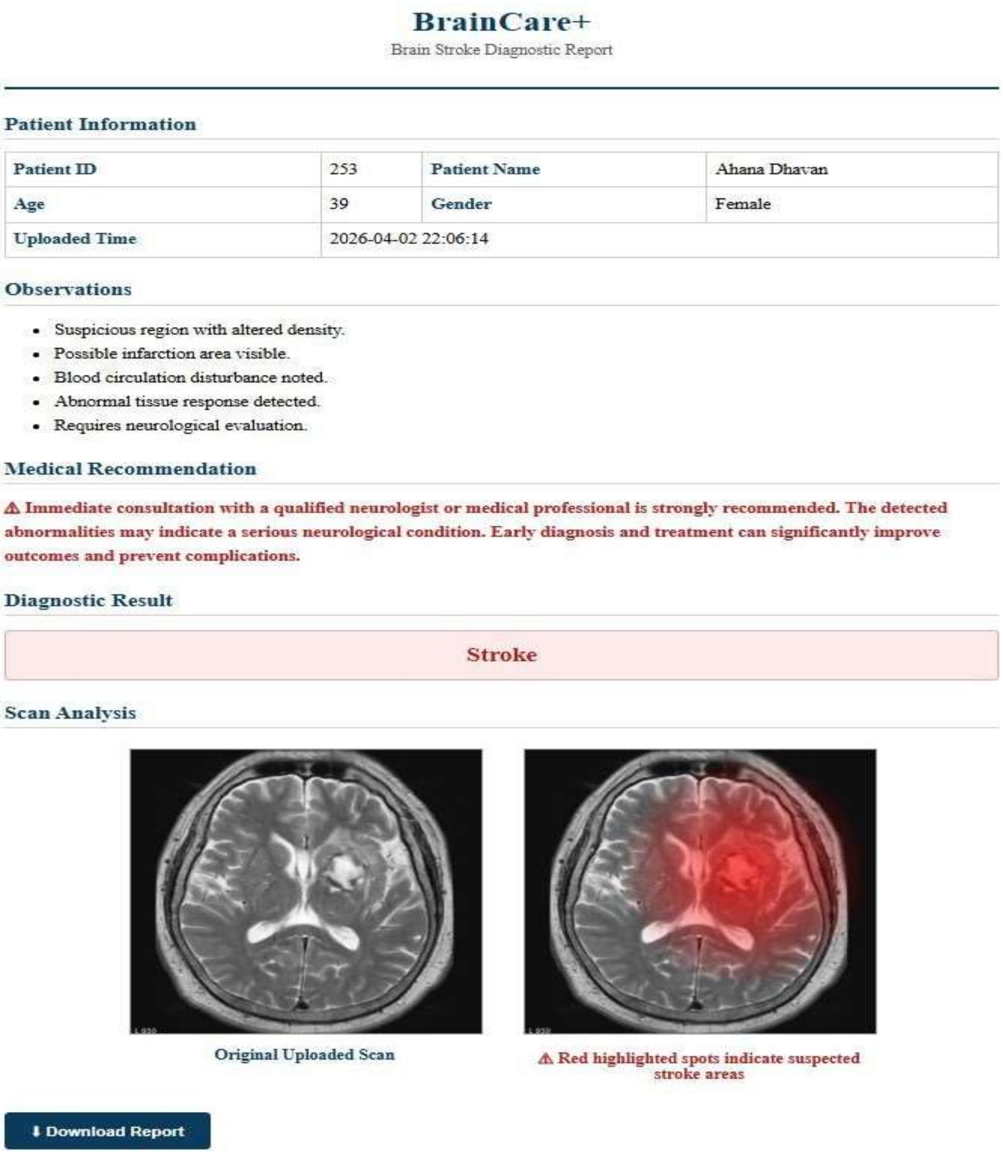


Screenshot 2: Login Page

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026



BrainCare+

Brain Stroke Diagnostic Report

Patient Information

Patient ID	253	Patient Name	Ahana Dhavan
Age	39	Gender	Female
Uploaded Time	2026-04-02 22:06:14		

Observations

- Suspicious region with altered density.
- Possible infarction area visible.
- Blood circulation disturbance noted.
- Abnormal tissue response detected.
- Requires neurological evaluation.

Medical Recommendation

⚠ Immediate consultation with a qualified neurologist or medical professional is strongly recommended. The detected abnormalities may indicate a serious neurological condition. Early diagnosis and treatment can significantly improve outcomes and prevent complications.

Diagnostic Result

Stroke

Scan Analysis

Original Uploaded Scan

⚠ Red highlighted spots indicate suspected stroke areas

⬇ Download Report

Screenshot 3: Report page-Stroke

Researcher – View Datasets

datasets uploaded by doctors and patients

ID	Image	Label	Uploaded By	Time
190		Stroke	maaanyasri@gmail.com	2026-04-02 22:06:14
189		Stroke	maaanyasri@gmail.com	2026-04-02 22:03:08
188		Normal	maaanyasri@gmail.com	2026-03-30 12:41:29
187		Stroke	maaanyasri@gmail.com	2026-03-30 12:39:31
186		Stroke	akshita@gmail.com	2026-03-30 10:32:05
185		Normal	akshita@gmail.com	2026-03-30 10:30:13
184		Normal	yash@gmail.com	2026-03-30 09:58:27

ISSN 2347–3657

Volume 14, Issue 1, 2026

Screenshot 4: View Datasets page

Screenshot 5: Analysis page

Conclusion

The proposed Brain Stroke Detection System presents an efficient and automated approach for early stroke diagnosis using medical imaging data. The developed framework enhances both the speed and reliability of stroke identification by minimizing dependence on manual interpretation. Conventional diagnostic procedures often involve time-intensive analysis and are susceptible to human error; the proposed system addresses these limitations through automation and intelligent analysis.The system follows a structured processing pipeline consisting of image preprocessing, segmentation, feature extraction, and classification using a deep learning model trained on MRI and CT scan datasets. One of the key features of the proposed solution is the stroke localization module, which generates heatmap overlays on affected regions. This visual representation supports clinicians in identifying abnormal areas quickly and improves interpretability of the model's predictions.

The backend implementation using Flask and SQLite ensures organized patient data management, secure storage of scan results, and easy traceability of diagnostic outcomes. Experimental evaluation demonstrated high detection accuracy, consistent highlighting of stroke regions, and fast processing time. The system also showed robustness when tested on diverse imaging inputs, maintaining stable performance across multiple cases.

Future Scope

The Brain Stroke Detection System can be further expanded to function as a comprehensive clinical decision-support platform. One potential enhancement involves replacing the lightweight SQLite database with scalable database management systems such as MySQL or PostgreSQL. This upgrade would allow efficient handling of large patient datasets and support concurrent multi-user access in hospital environments.Future work may also include the integration of advanced deep learning architectures such as ensemble models and attention-based networks. These improvements can increase classification accuracy and minimize false detections, particularly in complex imaging scenarios. Additionally, incorporating multi-modal imaging data, including MRI, CT, and angiography scans, could provide more comprehensive diagnostic insights.Another promising direction is the implementation of temporal scan comparison, enabling clinicians to monitor disease progression and recovery over time. Automated report generation with structured diagnostic summaries and visual heatmaps can further reduce manual documentation workload.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems (NeurIPS), 2012.

[2] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," Proceedings of the International Conference on Machine Learning (ICML), 2019.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[4] PyTorch Official Documentation. Available: https://pytorch.org/
[5] OpenCV Documentation. Available: https://opencv.org/
[6] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," Pattern Recognition, 1996.
[7] F. Chollet, *Deep Learning with Python*, Manning Publications, 2017.
[8] Kaggle Brain MRI Dataset. Available: https://www.kaggle.com/
[9] SQLite Documentation. Available: https://www.sqlite.org/docs.html
[10] Python Software Foundation, Python Documentation. Available: https://docs.python.org/
[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations (ICLR), 2015.
[12] C. Szegedy et al., "Going deeper with convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," CVPR, 2015.
[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," ICLR, 2015.