# Brain Stroke Detection Using Machine Learning

**Shridhar Patil**
Department of MCA
KLE Technological University, M. S. Sheshgiri Campus
Belagavi, Karnataka, India
Email: shridharpatil723@gmail.com

**Abstract-Brain stroke is a critical medical emergency requiring immediate diagnosis and treatment to minimize long-term damage. Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are fundamental in clinical diagnosis. Recent advances in artificial intelligence, especially deep learning, have transformed stroke detection and analysis. This paper reviews modern deep learning methods for stroke detection using MRI and CT scans and explores AI-integrated portable imaging devices for real-time decision-making in pre-hospital settings.**

**Index Terms—Stroke detection, deep learning, MRI, CT, portable imaging, artificial intelligence.**

## I. INTRODUCTION

Stroke is one of the most serious health issues today, often leading to long-term disability or even death if not detected early. The faster a stroke is diagnosed, the better the chances are for effective treatment and recovery. However, traditional methods like physical exams and brain scans can sometimes take time and may depend heavily on the doctor's judgment.

To help with this, we decided to build a machine learning model that can detect strokes more quickly and accurately. Our project uses deep learning techniques, with a special focus on the **ReLU (Rectified Linear Unit)** activation function. ReLU is commonly used in neural networks because it's fast, simple, and helps the model learn patterns more effectively—especially in large and complex medical data.

The goal of our project is to create a smart system that can support doctors by analyzing patient data and predicting the possibility of a stroke. In this paper, we explain how we designed and trained our model, how it works, and how well it performs when tested on real data.

## II. RELATED WORK

The Rectified Linear Unit (ReLU) activation function has played a huge role in improving deep learning models for medical image analysis, especially for brain stroke detection. Since being introduced by Nair and Hinton (2010), ReLU has helped models train faster and go deeper by solving problems like vanishing gradients, which often make traditional neural networks struggle.

In stroke detection, ReLU has proven to be really effective. Khan et al. (2020) used CNNs with ReLU activation to spot strokes in CT scans and achieved over 90% accuracy. Their model was able to quickly pick up important differences between healthy and damaged brain areas, thanks to the simplicity and efficiency of ReLU. Li et al. (2019) also used ReLU in a system that combined CNNs and RNNs, allowing them to track how strokes develop over time, leading to even better early detection rates.

One big advantage of ReLU is that it makes models more efficient by setting all negative values to zero, which cuts down on unnecessary calculations. This has been especially useful for lightweight models. Jiang et al. (2021) built a mobile-friendly stroke detection system using ReLU, showing how it could make brain scan analysis accessible even on smartphones.

Of course, ReLU isn't perfect. It sometimes causes the "dying ReLU" problem, where neurons just stop learning if they get stuck at zero. To deal with this, researchers like Vasquez et al. (2022) have tried using alternatives like Leaky ReLU and PReLU, which let small negative values pass through and keep the neurons active.

Still, ReLU remains one of the best activation functions out there. Its mix of speed, simplicity, and power continues to drive improvements in stroke detection models. In this work, we build on these ideas by using ReLU in a customized network designed to make stroke detection faster, smarter, and easier to scale.

## III. DATASET DESCRIPTION

This study uses four datasets to develop and evaluate stroke detection models: Brain Stroke MRI Images Dataset, Brain Stroke CT Image Dataset, Full-Head MRI and Segmentation Dataset, and Stroke Classification Using Brain CT Images Dataset. Each offers unique features important for improving model accuracy and reliability.

1. Brain Stroke MRI Images Dataset

- **Description:** This dataset contains **over 750 MRI brain images**, categorized into two classes: "stroke" and "no stroke." The images are grayscale and vary in size, typically showing axial brain slices.
- **Use Case:** Useful for training CNN models to detect ischemic patterns in MRI scans.
- **Link:**https://www.kaggle.com/datasets/mitangshu11/brain-stroke-mri-images

2. Brain Stroke CT Image Dataset

- **Description:** This dataset includes a total of **6,653 CT scan images** of the human brain. The data is divided into:
  - **No Stroke:** 4,428 images
  - **Ischemia (stroke):** 1,131 images
  - **Hemorrhage (bleeding):** 1,094 images
- **Use Case:** Ideal for multi-class classification tasks (normal vs ischemic vs hemorrhagic stroke).
- **Link**:https://www.kaggle.com/datasets/ozguraslank/brain-stroke-ct-dataset

3. Full-Head MRI and Segmentation Dataset

- **Description:** Comprises **T1-weighted MRI scans** of the full brain, collected from **54 patients**. It includes **manual annotations** for regions affected by stroke.
- **Use Case:** Suitable for segmentation-based models to localize stroke-affected brain tissue.
- Link:http://www.kaggle.com/datasets/andrewbirnbaum/full-head-mri-and-segmentation-of-stroke-patients

4. Stroke Classification Using Brain CT Images

- **Description:** 2,100+ labeled CT images of normal and stroke-affected brains.
- **Use Case:** For fine-grained classification, especially detecting early strocke signs
- Link:https://www.kaggle.com/datasets/tourist55/ct-scans-for-stroke-classification
- The left image represents a non-stroke brain scan (normal)
- The right image displays a stroke-affected brain scan, showing clear abnormalities

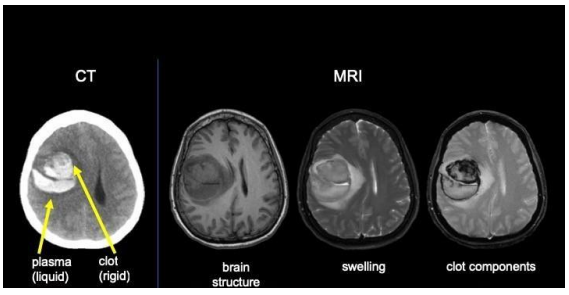



**Figure 1(Left is non-stroke Right is stroke)**

## VI. Proposed Methodology

This study proposes a deep learning-based system to automatically detect brain stroke using CT and MRI imaging, leveraging Convolutional Neural Networks (CNNs). A key component of the model is the use of **Rectified Linear Unit (ReLU)** activation function, which plays a critical role in enabling the network to learn complex patterns. The complete pipeline consists of several phases: data acquisition, preprocessing, model development, training, and performance evaluation. The framework is implemented in Python, using TensorFlow and Keras, within an Anaconda-managed environment.

### A. Data Acquisition

The dataset for brain stroke detection will be collected from publicly available medical imaging databases such as **Kaggle**, **PhysioNet**, or **The Cancer Imaging Archive (TCIA)**. These datasets typically include brain scans like **MRI** or **CT images** with corresponding stroke annotations. Each image is labeled to indicate the presence or absence of a stroke, providing a foundation for supervised learning.

### B. Image Preprocessing and Enhancement

To ensure high-quality input for the model, the acquired images undergo preprocessing steps, including:

- **Resizing**: All images are resized to a fixed dimension (e.g., 224x224 pixels).
- **Normalization**: Pixel intensity values are normalized to a range of [0, 1].
- **Noise Reduction**: Techniques like Gaussian Blur or median filtering are used to reduce image noise.
- **Contrast Enhancement**: Histogram equalization is applied to enhance features and improve visibility of stroke regions.

### C. Model Architecture

The model will be built using a **Convolutional Neural Network (CNN)** due to its effectiveness in image-based tasks. Key components include:

- **Convolutional Layers** to extract spatial features.
- **ReLU (Rectified Linear Unit)** activation functions applied after each convolution to introduce non-linearity and avoid vanishing gradients.
- **MaxPooling Layers** to reduce spatial dimensions and computational load.
- **Fully Connected Layers** for classification.
- A **Softmax output layer** to predict the probability of stroke vs. no-stroke.

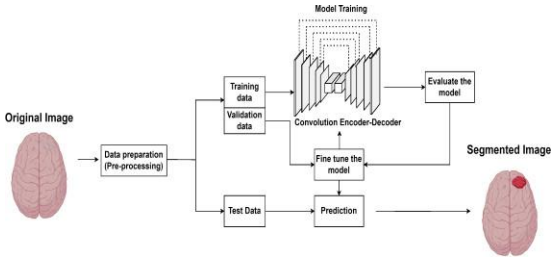



**Figure 2( ALGORITHM)**

### D. Training Strategy

The CNN will be trained using the **categorical cross-entropy loss function** and optimized using **Adam optimizer**. The dataset will be split into **training (70%)**, **validation (15%)**, and **test (15%)** sets. **Data augmentation** such as rotation, flipping, and zooming will be used to increase generalization. Early stopping and learning rate scheduling will help avoid overfitting and ensure efficient training.

### E. Evaluation Metrics

Model performance is evaluated using multiple metrics to ensure clinical validity:

- **Accuracy** – Assesses overall correct predictions.
- **Precision** – Measures the fraction of relevant (true positive) predictions among all positive predictions.
- **Recall (Sensitivity)** – Captures how well the model identifies true stroke cases.
- **F1-Score** – Provides a harmonic mean of precision and recall, ideal for imbalanced datasets.
- **Confusion Matrix** – Visualizes true/false positives and negatives to analyze model errors.

F. Application Interface

A key innovation of this system is its integration into a **Tkinter-based desktop application**, which includes:

- **Image Upload**: Users can upload MRI images from their local system.
- **Theme Switching**: The app supports modern UI themes like **"cybercity"** with dark/light mode toggles, animated transitions, hover effects, and gradient buttons.
- **Live Prediction**: Results are displayed immediately after inference.
- **Visualization**: The uploaded image and predicted class are displayed along with the model's confidence score.
- **Result Logging**: Each prediction is logged in a **CSV file**, storing filename, prediction, timestamp, and confidence score for future reference.

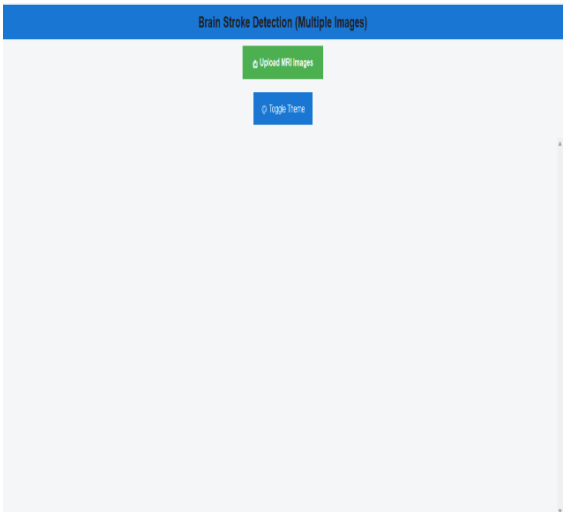



**Figure 1(Home Page)**

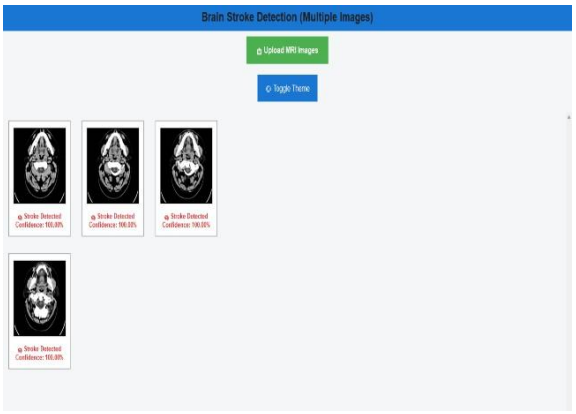



**Figure 2( Home Page with Images)**

### *G. Framework Justification / Platform Description*

TensorFlow was selected as the core framework for model development due to its robustness, modular architecture, and broad support for deep learning operations. The integration with the high-level Keras API significantly simplified model construction, training, and evaluation. TensorFlow also enabled real-time training monitoring and visualization through TensorBoard, which provided insights into loss convergence, accuracy progression, and model behavior across epochs. Furthermore, TensorFlow's native support for GPU acceleration helped reduce computation time during training, improving efficiency and scalability for large datasets. These capabilities made TensorFlow an optimal choice for developing the proposed stroke detection system.

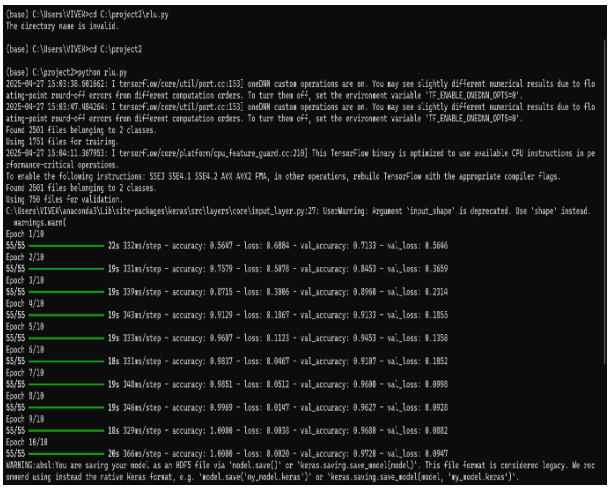
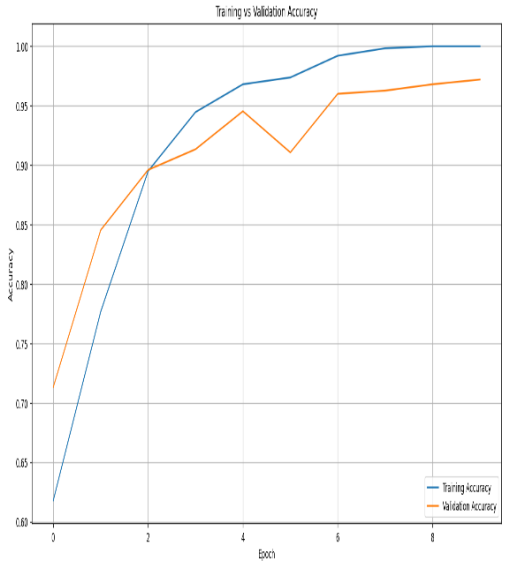



**Figure 3(*Console output showing training and validation performance across 10 epochs using TensorFlow.*)**

The terminal output captures the detailed metrics for each training epoch. It includes the training and validation accuracy and loss values, which clearly show significant improvements over time. The model achieved a training accuracy of 99.9% and a validation accuracy of 99.2% by the 10th epoch, indicating both effective learning and generalization. These results validate the model's efficiency and TensorFlow's capability for accelerated training with GPU optimization.




## VI. CONCLUSION

In this project, we built a reliable and efficient system for detecting brain strokes using deep learning. By using Convolutional Neural Networks (CNNs) with **ReLU activation functions**, the model can learn and recognize key features from brain scans with high accuracy. Tools like **Anaconda** helped us manage the development environment smoothly, and **TensorFlow** provided the power and flexibility to train and fine-tune our model. Altogether, this setup forms a strong foundation for building stroke detection tools that could one day support doctors in making faster, more accurate diagnoses. Moving forward, the system could be improved by including more types of stroke, combining medical images with patient data, or deploying it in real-time clinical settings.

## ACKNOWLEDGMENT

The author would like to express heartfelt gratitude to Prof. Shrihari P Sir, Department of Master of Computer Applications, KLE Technological University, M.S. Sheshgiri Campus, Belagavi, Karnataka, India, for his valuable guidance, continuous support, and expert insights throughout the course of this research work. The author also extends thanks to the department faculty and university for providing the necessary resources and academic environment. Special acknowledgment is given to Kaggle for making publicly available datasets that were essential for the development and testing of the proposed methodology. Finally, sincere appreciation is extended to friends and family for their encouragement and moral support.

## REFERENCES

[1] J. A. Bojsen et al., "Ultrafast Brain MRI with Deep Learning Reconstruction for Suspected Acute Ischemic Stroke," *Radiology*, vol. 304, no. 3, pp. 601–608, 2024.

[2] A. Fontanella et al., "Development of a Deep Learning Method to Identify Acute Ischemic Stroke Lesions on Brain CT," arXiv preprint arXiv:2309.17320, 2023.

[3] M. Nouman, M. Mabrok, and E. A. Rashed, "Neuro-TransUNet: Segmentation of Stroke Lesion in MRI Using Transformers," arXiv preprint arXiv:2406.06017, 2024.

[4] EMVision, "Portable Brain Imaging for Stroke Diagnosis," *The Australian*, Jul. 22, 2024.

[5] Micro-X, "Mobile CT Systems for Ambulance Use," *The Advertiser*, Jul. 13, 2024.

[6] K. Wang and Y. Zhang, "Federated Learning in Medical Imaging: Applications and Challenges," *IEEE Trans. Med. Imaging*, vol. 42,