

A Comprehensive Review of Deep Learning Techniques for Stroke Detection Using CT Brain Neuroimages

Abhay Gaidhani^{1*}, Prem Kanade², Yash Rathod³, Kunal Bhuyal⁴, Chetan Devkar⁵

^{1,2,3,4,5}Department of Computer Engineering, Sandip Institute of Technology and Research Centre
abhay.gaidhani@sitrc.org¹, premkanade27@gmail.com², yr636220@gmail.com³, bhuyalk0249@gmail.com⁴,
chetandevkar.engg@gmail.com⁵

*Corresponding Author: abhay.gaidhani@sitrc.org

Peer Review Information

Type: Article

Received: 25 July 2026

Revised: 22 Aug 2026

Accepted: 28 Sep 2026

Published: 24 Oct 2026

Abstract

Stroke is one of the main causes of death and long-term disability worldwide. How well a patient recovers often depends on how quickly the diagnosis is made. The first test that doctors usually order is a contrast CT scan of the human head. It is fast and available in every hospital. Reading the scan correctly is not so simple. During a shift a radiologist often misses the early signs of stroke, such as faint spots and small changes, in contrast. That's where deep learning, a part of intelligence comes in. It can help automate parts of the reading. In this paper we check some deep learning methods that are used to detect stroke using CT brain images. We have to focus on a convolutional neural network (CNN) and on four important architectures: ResNet, DenseNet, MobileNet and EfficientNet. The CNN is central, to my study. Provides the foundation for all these architectures. We compare how well each model learns, how computing power they need and how useful they are in real medical settings. So we built a framework that separates CT brain images into two groups Stroke and Normal. Each model CNN, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0 is trained individually. Then we compare their performance side by side. The process includes preprocessing the images using data augmentation applying transfer learning evaluating the models and finally deploying them with Streamlit. Finally, we cover the open challenges in this area. There are still some issues and challenges, in this area. We also discuss that how future research can or will make AI-based stroke detection more free from error or mistake.

Keywords: Stroke Detection, Deep Learning, Artificial Intelligence, CT Brain Images, Neuroimages, Convolutional Neural Networks, ResNet50, DenseNet121, MobileNetV2, EfficientNetB0, Transfer Learning, Medical Image Analysis, Computer-Aided Diagnosis.

How to Cite This Article

Gaidhani, A., Kanade, P., Rathod, Y., Bhuyal, K., Devkar, C. (2015) A Comprehensive Review of Deep Learning Techniques for Stroke Detection Using CT Brain Neuroimages. *International Journal on Theoretical and Applied Research in Mechanical Engineering*04(1), 1–12

Introduction

A stroke takes place when the blood flow to part of the human brain is interrupted. If a blocked vessel causes it, the result is an ischemic stroke. If a vessel in the human brain bursts or gets damaged, then the result is a hemorrhagic stroke. Either way, the affected tissue is cut off from oxygen and nutrients, and its cells begin to die. Many people are left with lasting brain damage, and some don't survive. Because every minute matters, a late or wrong diagnosis can shrink the window for treatment [1]–[3].

Doctors usually order a head CT scan first when they suspect a stroke. It takes a few minutes scanners are in most hospitals and bleeding and structural changes appear clearly. At this stage, NCCT quickly shows if there's any involvement of bleeding because of stroke. The hard part is reading the scan. Signs of ischemia in the few hours are weak. The skill of the reader matters a lot and a busy radiology department often gives less consistent reports [3]–[5].

Early computer-aided diagnosis tools used hand-made features, like brightness, texture, shape and edge measurements. How well they worked depended on which features the designers chose. Stroke lesions come in shapes and image quality changes when the scanner or protocol changes so a fixed set of features rarely stayed reliable [6]–[8].

Deep learning has changed everything. A deep network learns representations layer by layer from the pixels removing most of the dependence on hand-built features. CNNs have become the choice for classifying, detecting and segmenting medical images because stacked convolutional layers pick up local spatial patterns as well as higher-level structure [6], [8], [9].

Many CNN families were designed to improve feature extraction, reduce computation, or generalise better. VGGNet gets its depth by stacking many small convolutional layers [10]. ResNet adds skip connections that let very deep networks train more easily [11].

DenseNet links each layer to all the layers after it which encourages reuse of features [12]. MobileNet reduces the computation each layer needs, so models can run on phones or smartphones and other mini devices [13], [14]. EfficientNet scales depth, input resolution together to balance capacity and cost [15]. All of these models are now used for image classification and computer-aided diagnosis.

Because medical datasets are so small, transfer learning lets a model borrow what it learned from larger photo collections. With transfer learning, you can reuse an ImageNet-pretrained network as a fixed feature extractor or fine-tune some of its layers for your task. Transfer learning reduces training effort. Makes the model more stable when annotated cases are scarce [16]–[18].

Current research goes beyond CNNs. Newer stroke analysis tools combine several techniques, including segmentation networks, attention mechanisms, hybrid designs, explainable AI (XAI), also the transformers [19]. Results are promising. The same challenges keep showing up in one paper after another: small datasets, class imbalance, variations in imaging protocols and scanners overfitting, heavy computation, limited external validation and models that are hard to understand. No single architecture can be fairly called the best for every dataset and every clinical setting [5], [20], [21].

This paper reviews learning techniques for stroke detection in CT brain neuroimages and focuses on studies from around 2020 to 2026. This paper reviews six CNN architectures (VGGNet, ResNet, DenseNet, MobileNet, EfficientNet, and the basic CNN), plus transfer learning and some newer approaches. These are compared based on how they extract features how much computation they need the performance reported, their limitations and how suitable they are for real clinical use [4], [5], [20].

This review leads us to a five-model approach for labelling CT brain images as Stroke or Normal, using a basic CNN, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0. Each one is trained separately. None is merged into an ensemble so the behavior and performance of each architecture can be studied on its own. The method includes image preprocessing, data augmentation, transfer learning where appropriate, model training, performance evaluation and a final release, as a Streamlit application. The goal of the comparison is to help choose the model based on accuracy, speed and what the target application needs [11], [12], [14], [15].

Literature Review

AI and DL have become tools for analyzing medical images such as CT stroke images to detecting stroke automatically. When a stroke is suspected, doctors usually start with an NCCT scan because it takes only a few minutes. Many studies have looked into CNNs, transfer learning, segmentation, lightweight models, AI and hybrid methods to identify and understand stroke-related changes. Still differences, in datasets, imaging protocols and validation methods can greatly affect how well these models work when used outside the study [4]–[6], [20].

A. Development of Deep Learning-Based Stroke Detection

The earliest studies showed that deep learning could detect stroke on NCCT in every case. Nishio et al. what he did is, In a two-stage design YOLOv3 was combined with VGG16.

Radiologists did better when they had the AI output to use as a help. This was noted in reference [4]. Attention then shifted toward locating and segmenting lesions, which helps the clinician know where the damage is and how large it is. Fontanella et al. Trained a CNN on CT scans collected during care. The results were promising. Lesion characteristics posed real challenges and it was not clear how the model would perform on data, from other sources [22].

B. CNN-Based Approaches

CNNs are used a lot in imaging because they learn layers of features without being told what to look for. Early layers detect simple features like edges and texture, and deeper layers combine them into larger patterns. They have been used for both stroke

classification and, for lesion segmentation [6]–[8]. Regular CNNs, however, often need large amounts of training data to work well. May behave poorly when imaging conditions change. These problems led researchers to use pretrained networks and transfer learning [16], [17].

C. Transfer Learning and Pretrained CNN Architectures

Transfer learning uses a network that has already been trained and adapts it to a medical imaging task. Reusing a pretrained model saves time, and that matters most when your dataset is small. Over the years researchers have tried kinds of networks on this task. They experimented with structures, sizes and designs. Some worked better, than others. These include VGGNet, ResNet, DenseNet, MobileNet and EfficientNet. Each one has its advantages. ResNet adds skip connections that let very deep networks train without losing accuracy. DenseNet goes further by reusing features in every layer. 3.MobileNet is designed to reduce needs so it runs faster and uses fewer resources. EfficientNet finds a ground between model size and performance by using compound scaling. Choosing the architecture means balancing accuracy, with how much computing power is needed how sensitive the model is and the challenges of putting it into real use.

D. Segmentation and Multi-Stage Deep Learning

Recent work often goes beyond classification. Multi-stage designs first detect a lesion, then segment it, and then classify the case [4]–[6]. U-Net. Its variants have been used to find and outline ischemic lesions on NCCT [23], [24]. Segmentation adds information about location. It needs precise pixel-level or voxel-level annotation. Preparing a dataset, in this way is slow and expensive [6], [23], [24].

E. Lightweight and Efficient Architectures

Medical AI that needs to work on simple computers has made people look at small models like MobileNet [13], [14]. EfficientNet scales depth, width, and image resolution together, which gives good accuracy without a large computing cost [15]. Being efficient does not always mean the outcome. The final result depends a lot, on how big the data set's how the data is prepared how it is changed to make the data also varies in how the classes are distributed and how the model is tuned [5], [16], [20].

F. Explainable AI and Clinical Interpretability

Deep models are hard to understand. In healthcare that is a big problem. Grad-CAM, saliency maps and attention visualization can show which parts of an image affected a prediction [21], [25], [26]. Just because a network highlights a region doesn't mean it has learned something useful for diagnosis. Explainability therefore needs to be supported by real-world testing and expert opinions. It should not take the place of those things [21], [25], [26].

G. Recent Trends: Clinical Validation and Generalization

As study evolved they checked whether deep learning models still perform on patients, hospitals, and scanners they haven't seen before. They also ask if these models work in a hospital setting [5] [20]. The meta-analysis by Gete and Ayele shows this well. In 16 studies they found that the average ability to find ischemic stroke on non-contrast CT scans was 0.91. The average ability to correctly identify those without the condition was 0.90. When tested on outside groups the ability to find the condition dropped to 0.82. This was much lower than the 0.95 found in the groups [5]. Systematic reviews also point to inconsistencies in datasets, validation methods, and testing procedures [5], [20]. Because of this newer research is focusing on better outside testing, real hospital use, clearer explanations and stronger support, for doctors [5], [20], [21], [26].

H. Research Gaps

The study shows that there are issues that need attention. There are problems that are still not solved.

- Small datasets and the chance of overfitting that comes with them [5], [6], [20].
- The data has a lot cases that are not strokes compared to cases that are strokes or the other way around [5] [20].
- Limited external validation across hospitals and different scanners [5], [20].
- Differences in imaging protocols and image quality [5]–[7].
- Predictions that are still hard to understand [21], [25], [26].
- Inconsistent evaluation methods between studies [5], [20].
- Limited use in clinical settings and little long-term testing [5] [20].
- High computational cost, for some models [11] [12] [15].

Because of these issues future work should focus more on being able to repeat results working well in situations being easy to understand working fast and being able to stand up to clinical tests [5], [20], [21].

I. Summary of Recent Studies

Table I collects studies from 2020 to 2026. When I read the table in order the studies show a gradual progression. Early studies relied on CNN-based detectors. Later work turned to segmentation and transfer learning. The studies next streamlined

architectures and explainability. Finally the studies point toward validation. In these studies, progress comes in steps rather than one big leap, with each stage building on the last [4], [5], [20], [22].

Table 1. Summary of recent literature (2020–2026)

Study	Year	Model / Technique	Major Contribution	Limitation
Nishio et al. [4]	2020	YOLOv3 + VGG16	Automated ischemic stroke detection from NCCT	Limited sensitivity
Liang et al. [27]	2021	Attentionbased CNN	Improved infarct localization	Localizationfocused
Chen et al. [28]	2022	DL + ASPECTS	Automated infarct assessment	Limited external validation
Sahoo et al. [29]	2022	Deep learning (custom CNN)	Early ischemic lesion detection	Singleinstitution data
Ostmeier et al. [23]	2023	3D CNN	Automated ischemic lesion segmentation	Segmentation -focused
Lin et al. [30]	2023	DGA3-Net	Parameter efficient ASPECTS assessment	Limited validation
Fontanella et al. [22]	2023	CNN	CT lesion and affected-side detection	Chronic abnormalities affected performance
Marcus et al. [31]	2024	CNN with radiomics	Stroke lesion-age biomarker estimation	Further validation needed
Gete & Ayele [5]	2025	Systematic review / metaanalysis	Evaluated DL for acute ischemic stroke on NCCT	External validation limitations
Inamdar et al. [32]	2026	TrustNet + XAI	Lightweight DL with uncertainty and XAI	Multicenter validation needed
Kunduracioglu et al. [20]	2026	Systematic review	Reviewed DL approaches for stroke imaging	Limited clinical deployment

J. Overall Observation

Across these studies the field has shifted from CNN-based detection to transfer learning, segmentation, lighter architectures, explainable AI and clinical validation [4], [5] [20], [22]. No single architecture performs best in every situation. Results vary with the dataset, preprocessing, imaging setup, and validation method [5], [20]. That's why comparing architectures matters. Comparing the models shows which ones work in real-world settings and what limits each design [11]– [15].

Based on these observations, we test a basic CNN, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0 separately on one task. The task separates CT brain images into Stroke and Normal categories [11], [12] [14], [15].

Comparative Analysis of Deep Learning Models

I notice that deep learning architectures vary in how deep learning architectures pull out useful details how much processing power deep learning architectures need and how suitable deep learning architectures are, for medical image work. In the area of CT-based stroke detection I see that five models receive the attention: CNN, ResNet50 DenseNet121 MobileNetV2 and EfficientNetB0 [6], [11], [12], [14], [15].

A. Conventional CNN

The normal CNN is a method for classifying images. By using convolution and pooling it picks up patterns that begin as lines and colors and become more complicated as it goes deeper. The setup is simple, which makes it easy to build and test. However, it tends to fit to a small set of data and the things it learns might not be as good as those, from more recent designs [6], [8], [9]. B. ResNet50

ResNet50 uses connections. These connections allow information and gradient signals to move through a set of layers. This lets the network learn features more reliably. It is a choice for transfer learning. The cost of adding depth is higher because adding depth needs computation and memory [11] [16].

C. DenseNet121

In DenseNet121 every layer is connected to every layer that comes after it. Because of this features are. Gradients flow with less resistance. In practice DenseNet121 often performs well on medical images especially when data are limited. However the dense connections, in DenseNet121 use memory and require more computation compared with networks [6] [12].

D. MobileNetV2

MobileNetV2 is designed to work with limited computing power. MobileNetV2 cuts its parameter count with depthwise convolutions and inverted residual blocks. This makes the network faster and less expensive to run. That is why MobileNetV2 performs well in real-time applications and, on devices that have limited resources. The simpler design can be a problem. When images are MobileNetV2 may not capture the features well as other models [13] [14].

E. EfficientNetB0

EfficientNetB0 uses a scaling method that balances depth, width and input size all at once. This gives a tradeoff between how much it can learn and how fast it runs. EfficientNetB0 works well when you adapt it to data, such, as medical pictures. However you must tune it carefully. Set up the images right otherwise the results may get worse [15], [16].

F. Comparative Discussion

The five architectures each have strengths. A CNN is easy to understand. ResNet50 targets deep feature learning, DenseNet121 reuses features across layers, and MobileNetV2 prioritizes efficiency. EfficientNetB0 is in the middle with a mix of quality and efficiency [11], [12], [14], [15].

A model should therefore be evaluated on more, than accuracy. Precision, recall, specificity, F1-score and ROC-AUC all are important as are the computational needs and how easy it is to put into use. These differences support the -model approach that is suggested in which CNN, ResNet50 DenseNet121 MobileNetV2 and EfficientNetB0 are all tested on their own for CT-based stroke classification [5], [20].

The table shows clearly that each type of architecture has its benefits and some downsides. The system we suggested does not think that one network is the best, in all situations. It looks at a models that were trained on their own and lets the person who understands the specific use case make the final decision [11] [12] [14] [15].

Table 2. Comparison of deep learning models for stroke detection

Model	Main Advantage	Main Limitation	Computational Requirement	Suitability
CNN [6], [8], [9]	Simple architecture and automatic feature extraction	Higher overfitting risk on small datasets	Low– Moderate	Baseline
ResNet50 [11]	Residual learning and deep feature extraction	Higher computational cost	High	High
DenseNet121 [12]	Feature reuse and improved gradient propagation	Higher memory consumption	High	High
MobileNetV2 [14]	Lightweight architecture and fast inference	Lower feature capacity than deeper models	Low	High for resource constrained systems
EfficientNetB0 [15]	Balanced feature extraction and computational efficiency	Requires appropriate fine-tuning	Low– Moderate	High

Proposed Methodology

A. Aim of the Proposed Methodology

The research shows that a good deep learning system for finding strokes in CT scans of the brain needs image processing making data getting important features classifying and checking how well it works [5] [6] [20]. We want a system that can work well when the steps in the process change from one place to another. The approach is based on the idea that deep models can pick up details directly from CT scans so there is not much need for features that people create by hand [6] [8]. The best choice among a regular CNN, ResNet, DenseNet, MobileNet, and EfficientNet depends on the task and the available hardware [11]-[13], [15]. If the medical data is not big transfer learning can be added [16], [17]. This study uses the system to classify brain CT scans as either Stroke or Normal. The same setup could be used for tasks like telling the difference, between two kinds of stroke finding where the problem is or measuring how bad it is [4], [5], [24].

B. Overall Workflow

The generalised deep learning framework consists of the stages listed below and shown in Fig. 1.

1. CT brain image dataset collection
2. Image preprocessing
3. Data augmentation
4. Deep feature extraction
5. Model. Fine-tuning
6. Stroke classification
7. Performance evaluation
8. Explainability and clinical validation

Each stage contributes to the reliability and clinical usefulness of an automated stroke detection system [5], [6], [20].

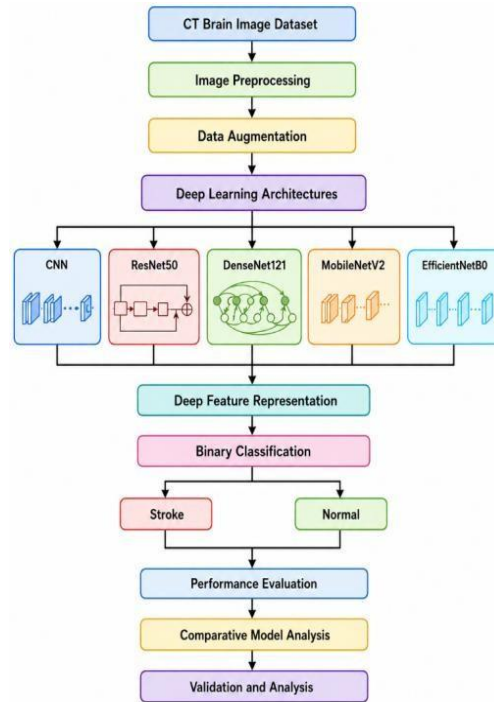


Fig. 1. Generalized deep learning framework for CT-based stroke detection

C. Step-by-Step Methodology

Step 1: Dataset collection. CT brain neuroimages are collected from datasets, hospital archives or sources obtained in a clinical setting. The kind of content included depends on the research goal. A dataset might contain cases ischemic stroke cases, hemorrhagic stroke cases or lesion regions that have already been labeled [4], [5], [22]. After collecting the images they are divided into training sets, validation sets and testing sets. Each set has its purpose. Split the data by patient whenever possible, so no patient appears in more than one subset. This helps prevent data leakage and avoids problems [5], [20].

Step 2: Image preprocessing. CT images can have sizes, brightness levels scanning rules and overall quality. Preprocessing. Standardizes the input before any modeling starts [6] [7]. The operations often reported are:

- image resizing;
- intensity normalization;
- noise reduction;
- contrast enhancement;
- Removal of the skull and other irrelevant regions, where appropriate;
- conversion, into a suitable input format.
- Consistent preprocessing helps a network learn representations and removes variation that has nothing to do with the pathology of interest [6] [7]

Step 3: Data augmentation. Medical imaging datasets are often small. Data augmentation widens the variety of the training data. Data augmentation tends to reduce overfitting because the model sees extra views of the data without any new cases being collected [6], [7]. Data augmentation also helps the model learn from viewpoints. Typical techniques are:

- rotation;
- flipping, where clinically appropriate;
- scaling and zooming;

- translation;
- cropping;
- brightness and contrast variation.

The parameters require care. If a transformation is unrealistic, it can distort meaningful information and erase the signal the model is supposed to learn [6], [7].

Step 4: feature extraction. After preprocessing, the CT images are fed into a deep learning model. A CNN learns visual patterns in stages, and each layer builds on the one before it [6], [8], [9]. Different models approach this in ways. Standard CNNs simply stack layers one after another. ResNet uses connections to help train deeper networks. DenseNet focuses on reusing features across layers. MobileNet is designed to be efficient. MobileNet is efficient when it comes to computation. EfficientNet uses a method called compound scaling to adjust the network's depth, width, and resolution together [11]–[15]. Small datasets are a common reason to use transfer learning. This means beginning with a model that was already trained on a dataset. Then a few layers are fine-tuned using CT brain images. This approach is different, from training the network from scratch [16]–[18].

Step 5: Model training and fine-tuning. The selected network is trained using labeled CT brain images. In every round it makes guesses the classification loss is calculated the mistake is sent back and a tool like Adam or stochastic gradient descent changes the values [33], [34]. Settings such as learning rate, batch size, number of epochs and regularization are picked to match the dataset and the structure. The validation data show how good the model is, at handling information and so help avoid overfitting [16], [34].

Step 6: Stroke classification. In stroke detection the trained model assigns each image to one of two target groups: Normal. The choice of activation function in the layer depends on how the model architecture and classification head are designed. It may use a sigmoid activation to produce probabilities for two classes. It may switch to SoftMax if the problem involves more, than two classes. I think the same framework can be adapted to distinguish between stroke, hemorrhagic stroke and normal cases. This means that the same system can be used to separate a stroke from a bleeding stroke and also from other situations. The framework helps to distinguish stroke, from hemorrhagic stroke. The framework also helps to separate stroke from cases.

Step 7: Performance evaluation. Accuracy alone can mislead therefore deep learning systems must be assessed on more than accuracy.

Many papers report a set of measures [5], [6], [20]:

- accuracy;
- precision;
- sensitivity / recall;
- specificity;
- F1-score;
- ROC-AUC;
- confusion matrix.

Segmentation studies usually measure overlap with the Dice coefficient and Intersection over Union (IoU) [23], [24]. In stroke detection the use of metrics carries particular weight because a false negative can have serious clinical consequences. A patient may go unrecognised even when the model looks good on a single metric [5], [20].

Step 8: Clinical validation. In a clinic setting every prediction must have a basis Grad-CAM, saliency maps, and attention visualisation show which image regions influenced a model's decision [21], [25], [26]. However interpretability alone does not guarantee that a system is clinically reliable. External validation is also required, meaning that the system should be tested on data from hospitals, other scanner types and other patient groups. Prospective clinical trials should be carried out along, with a comparison and proper interpretation before the tool is introduced into everyday care [5] [20] [26].

D. Generalized Model Selection Strategy

The literature shows that different model architectures end up at points when balancing how well they predict and how much computing power they need. So picking a model should start with what you plan to use it for not just because it has the score on one test [5], [11], [15], [20]. A regular CNN can be a starting point to compare against. ResNet and DenseNet tend to learn detailed features than a basic model. ResNet and DenseNet give results especially with complex images. These models are designed to capture patterns in the data. They use connections or dense connections to help the network learn deeper representations. This leads to performance on complex tasks. If you need to save on computing power MobileNet is a choice. EfficientNet usually finds a balance, between how strong the model is and how much it costs to run. [11]–[15]. Testing models side by side instead of just picking the one, with the best reported accuracy gives a clearer and more useful view [5], [20].

E. Summary of the Proposed Framework

The system brings together the techniques used in research and joins them into one simple process: CT Image Acquisition → Preprocessing → Augmentation → Deep Feature Extraction → Model, Models are trained or fine-tuned, used to classify scans, evaluated on performance, checked with explainability methods, and finally validated on external or clinical data.

This isn't based on a single experiment. It's an approach meant to help researchers get started with deep learning for computeraided stroke detection. It also targets the weaknesses found in the reviewed work [5], [6], [20], [21].

It gives a step-, by-step guide. You can use it as a base. Build from there. The goal is to make the process clearer and more consistent so people can compare results. Improve the models over time.

F. Training Pipeline

Fig. 2 Shows the training stage in detail. I first take the image from a CT brain image. Then I clean it and add some variations so the data looks more diverse. After that I feed this CT brain image into one of the chosen selected model – for example CNN, ResNet50, DenseNet121, MobileNetV2 or EfficientNetB0. The selected model extracts features from the scan, and a final classification layer outputs the probability of Stroke or Normal. I compute the loss. Use backpropagation to adjust the weights. Next I run a validation check to see how well the selected model is doing. If performance can still get better I start another training cycle. If the performance is good enough I stop training keep the model. Evaluate the final models accuracy precision recall F1 score and ROC AUC.

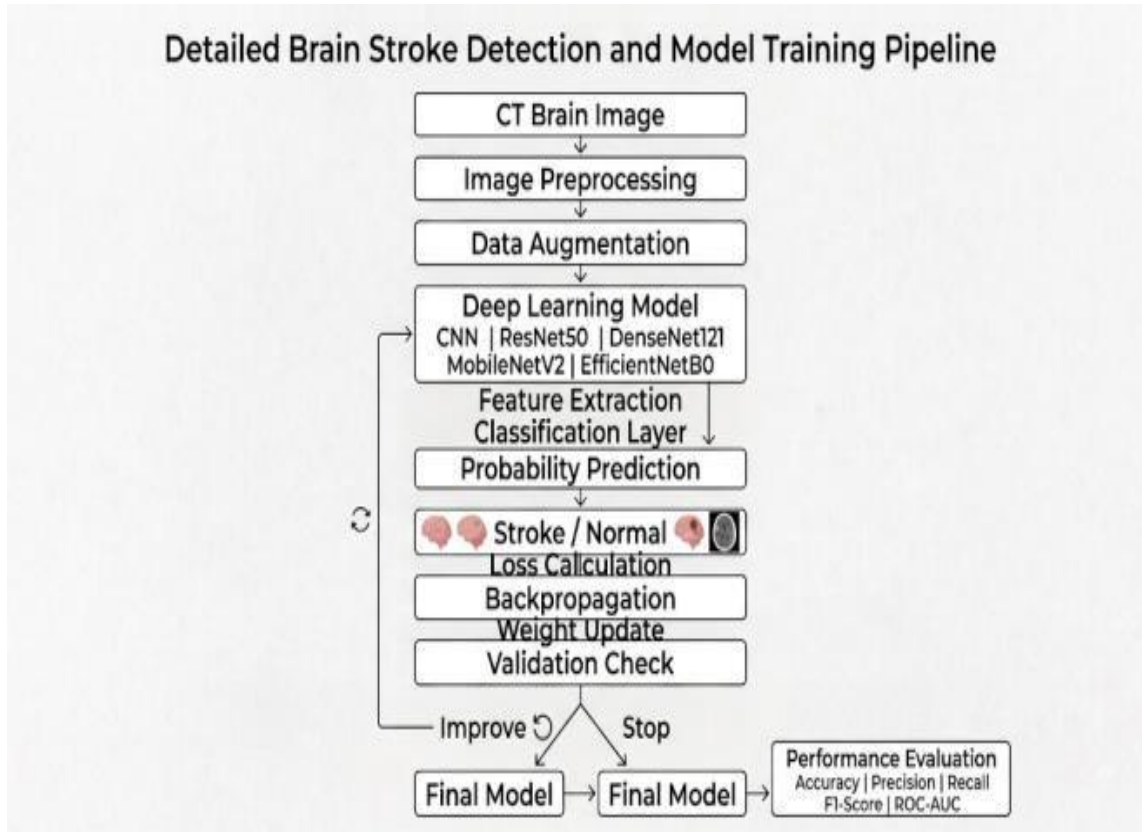


Fig. 2. Detailed brain stroke detection and model training pipeline

V. Algorithm and Mathematical Model

A. Algorithm

Algorithm 1: Generalized deep learning-based stroke detection

1. Load the CT brain neuroimage dataset.
2. Resize the images. Normalize the pixel values. Use enhancement techniques. These techniques include contrast adjustment. References are [6]. [7].
3. Use data augmentation methods on the training images to improve model robustness [6], [7].
4. Choose a model architecture—options include CNN, ResNet50 DenseNet121 MobileNetV2 or EfficientNetB0 [11], [12], [14], [15].
5. If transfer learning is being used, initialize the model with pretrained weights to leverage knowledge from datasets [16]– [18].
6. Extract features from the CT brain images using the selected model [6] [8] [9].
7. Train the classification layers using an optimizer like Adam to update the model's weights [33].
8. Predict the class, for each image either Stroke or Normal.
9. Evaluate the model's performance using metrics such, as accuracy, precision, recall, specificity, F1-score, ROC-AUC and the confusion matrix [5], [6], [20].
10. Include explainability techniques and external validation when the model is intended for world clinical use [5], [20], [21], [25].

B. Mathematical Model

1) Convolution operation. The convolution layer pulls out the features from the input image. The convolution operation is essential for CNNs. The convolution operation allows the convolution network to learn spatial patterns, from CT brain images [8] [9].

$$M-1 \ N-1$$

$$Y(i, j) = \sum \sum X(i + m, j + n) K(m, n)$$

$$m=0 \ n=0$$

Here, X is the input CT image, K is the convolution kernel, and Y is the output feature map.

2) **SoftMax function.** SoftMax changes the outputs of the model into class probabilities. SoftMax turns scores into class probabilities [34].

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}}$$

Here $P(y_i)$ is the probability of class i , z_i is the output score (logit) for class i n is the total number of classes and e is Euler's number (2.71828). The class that has the probability becomes the prediction.

3) **Cross-entropy loss.** Training minimises the cross-entropy loss, the gap between the true class labels and the predicted class probabilities [34].

$$L = - \sum_{i=1}^n y_i \log(\hat{y}_i)$$

In this expression the actual class label is y_i . The analyzed prediction probability is $\hat{y}_i$. A lower loss means that the model is better and efficient.

C. Methodology Diagram

Fig. 3 Shows the analysis pipeline. It starts with CT neuroimages. Goes through comparative analysis, performance measures and finally highlights research gaps and future directions.

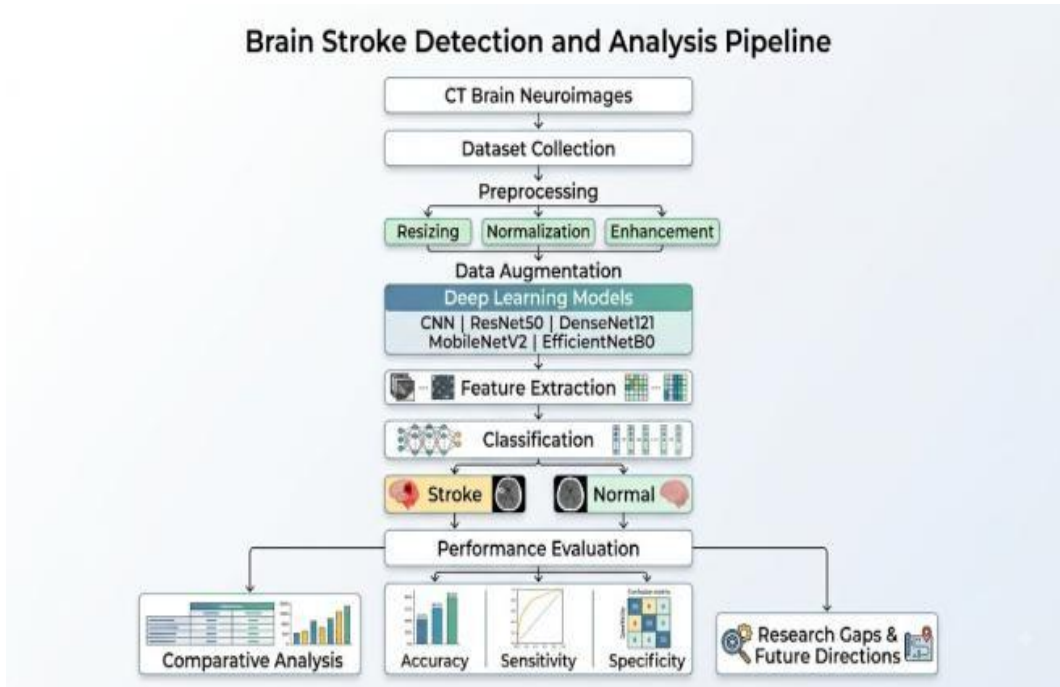


Fig. 3. Brain stroke detection and analysis pipeline

Performance Evaluation

Deep Learning systems for stroke detection are usually tested with a confusion matrix and a few common classification metrics. These metrics show how well each learning system sorts CT brain images into Stroke and Normal groups. I notice that no single metric explains the diagnostic picture so several metrics are normally looked, at together [5], [6], [20].

A. Confusion Matrix

Table 3. Confusion matrix for binary stroke classification

	Predicted Stroke	Predicted Normal
Actual Stroke	TP	FN
Actual Normal	FP	TN

The four entries are defined like this:

- TP (positive): a stroke image that is correctly identified as stroke.
- TN (negative): a normal image that is correctly identified as normal.

•FP (positive): an image that is very normal but classified as a stroke.

•FN (negative): an image that's a stroke but classified as normal..

These metrics measure how well the system separates stroke images from normal images. It gets some cases right and misses others. TP and TN are the calls. FP and FN are the mistakes. It's useful to track them all to see where the system needs improvement.

B. Evaluation Metrics

1) **Accuracy.** Accuracy measures how often the model's prediction matches the true label. Accuracy is essential to understand how well the model performs during predictions It looks that how frequently the model gets the answer.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) **Precision.** Precision measures how many of the images the model said are stroke cases actually are stroke cases [6], [7]. A higher precision means fewer wrong positive predictions, so a positive result is more likely to be correct.

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) **Recall (sensitivity).** The ability of the model to find stroke cases is called recall. It tells how many of the stroke cases are found [5] [20].

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) **Specificity.** Specificity refers to how the model can recognize normal cases. It means that images which are actually healthy are not accidentally put in the group as stroke cases [6] [7]. When specificity is high there are false alarms. This makes the system more reliable, in identifying -stroke cases as healthy.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5) **F1-score.** The In my view the F1-score is the mean of precision and recall [6], [7].

F1-score gives a balanced view than either figure alone especially on skewed datasets such as those, in stroke detection, where false positives and false negatives both carry weight.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

6) **ROC-AUC.** The Receiver Operating Characteristic (ROC) curve shows how the Positive Rate (recall) compares with the False Positive Rate [5] [20].

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

where TPR is the true positive rate (sensitivity or recall) and FPR is the false positive rate.

C. Comparative Evaluation

The research I have read shows that deep learning systems can work in ways and deep learning systems can adapt to many tasks. The reason for this is the dataset but the preprocessing method, the training approach the way classes are distributed and the validation process. CNN, ResNet50 DenseNet121 MobileNetV2 and EfficientNetB0 should therefore be evaluated using different metrics because accuracy by itself can hide problems [5], [11], [12], [14], [15], [20].

EfficientNet-based models get attention because of their compound scaling, which aims to keep power and efficiency in balance [15]. It is hard to support the claim that one model's better, for all stroke detection datasets because each model behaves differently on each dataset. The results need to be looked at along with the dataset, the experiment setup, the evaluation measures and the validation method from each study.

If a model is meant for clinical use, accuracy alone is not enough. They also need to have validation that works well across scanners and hospitals. Validation should be easy to understand require a reasonable amount of computing power and allow a quick time to make a decision. Models must also fit into clinical workflows without slowing doctors down [5], [20], [21], [26].

Conclusion

Deep I have seen how deep learning has shown promise for automated stroke detection on CT brain neuroimages but progress has been uneven. This review followed the evolution of CNN-based methods to ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0. The studies suggest that transfer learning, data augmentation, segmentation and explainable AI can make learning-based stroke analysis steadier and more useful in practice and not only on paper [4] [5], [16], [21], [22]. I notice that when the models are compared the lighter ones, MobileNetV2 and EfficientNetB0 have an advantage in computation. The heavier ResNet50 and DenseNet121 learn representations and may capture subtle patterns better [11], [12], [14], [15]. Because studies use

different datasets, imaging protocols, metrics, and validation methods, comparing them directly is hard [5], [20]. I see several hurdles remain: dataset diversity, class imbalance the need for external validation problems of interpretability and the step into real clinical deployment. Main thing is future work should look into datasets that have validation routines. Future work should also use AI methods increase computational efficiency and integrate more closely with clinical workflows. Then will stroke detection systems be dependable and of genuine benefit to patients instead of merely impressive, on a benchmark [5], [20], [21], [26].

References

1. World Health Organization, "Stroke, cerebrovascular accident," World Health Organization, 2024.
2. G. J. Hankey, "Stroke," *The Lancet*, vol. 389, no. 10069, pp. 641–654, 2017.
3. W. J. Powers et al., "Guidelines for the early management of patients with acute ischemic stroke: 2019 update to the 2018 guidelines," *Stroke*, vol. 50, no. 12, pp. e344–e418, 2019.
4. M. Nishio, S. Koyasu, S. Noguchi, T. Kiguchi, K. Nakatsu, T. Akasaka, H. Yamada, and K. Itoh, "Automatic detection of acute ischemic stroke using non-contrast computed tomography and two-stage deep learning model," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105711, 2020.
5. K. Y. Gete and A. A. Ayele, "Deep-learning-based non-contrast CT for detecting acute ischemic stroke: a systematic review and HSROC meta-analysis of patient-level diagnostic accuracy," *BMC Neurology*, vol. 25, art. 507, 2025, doi: 10.1186/s12883025-04528-3.
6. G. Litjens et al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.
7. S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: A review," *Journal of Medical Systems*, vol. 42, 2018.
8. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
9. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
10. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations*, 2015.
11. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
12. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.
13. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
14. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
15. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning*, 2019, pp. 6105–6114.
16. N. Tajbakhsh et al., "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.
17. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, vol. 27, 2014.
18. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.
19. A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
20. I. Kunduracioglu, S. Ince, B. Bayram, S. Kilicarslan, E. Veziroglu, and I. Pacal, "Deep learning in acute ischemic stroke imaging: a systematic review of CT- and MRI-based segmentation, triage, and prognostic modeling," *Neuroradiology*, 2026, doi: 10.1007/s00234-026-04095-5.
21. R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE International Conference on Computer Vision*, 2017, pp. 618–626.
22. A. Fontanella, W. Li, G. Mair, A. Antoniou, E. Platt, P. Armitage, E. Trucco, J. Wardlaw, and A. Storkey, "Development of a deep learning method to identify acute ischemic stroke lesions on brain CT," arXiv preprint arXiv:2309.17320, 2023.
23. S. Ostmeier, B. Axelrod, B. F. J. Verhaaren, S. Christensen, A. Mahammedi, Y. Liu, B. Pulli, L.-J. Li, G. Zaharchuk, and J. J. Heit, "Non-inferiority of deep learning ischemic stroke segmentation on non-contrast CT within 16-hours compared to expert neuroradiologists," *Scientific Reports*, vol. 13, art. 16153, 2023, doi: 10.1038/s41598-023-42961-x.
24. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.
25. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

27. A. Holzinger, "From machine learning to explainable AI," in *Proc. World Symposium on Digital Intelligence for Systems and Machines*, 2018, pp. 55–66.
28. K. Liang et al., "Symmetry-enhanced attention network for acute ischemic infarct segmentation with non-contrast CT images," in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2021, doi: 10.1007/978-3-030-872342_41.
29. W. Chen et al., "Improving the diagnosis of acute ischemic stroke on non-contrast CT using deep learning: a multicenter study," *Insights into Imaging*, vol. 13, art. 184, 2022, doi: 10.1186/s13244-022-01331-3.
30. P. K. Sahoo, S. Mohapatra, C.-Y. Wu, K.-L. Huang, T.-Y. Chang, and T.-H. Lee, "Automatic identification of early ischemic lesions on non-contrast CT with deep learning approach," *Scientific Reports*, vol. 12, art. 18054, 2022, doi: 10.1038/s41598-022-22939-x.
31. W. Lin et al., "DGA3-Net: A parameter-efficient deep learning model for ASPECTS assessment for acute ischemic stroke using non-contrast computed tomography," *NeuroImage: Clinical*, 2023.
32. A. Marcus, G. Mair, L. Chen, C. Hallett, C. Ghezzou CuervasMons, D. Roi, D. Rueckert, and P. Bentley, "Deep learning biomarker of chronometric and biological ischemic stroke lesion age from unenhanced CT," *npj Digital Medicine*, vol. 7, art. 338, 2024, doi: 10.1038/s41746-024-01325-z.
33. Inamdar et al., "TrustNet: a lightweight network with integrated uncertainty quantification and quantitative explainable AI for ischemic stroke detection in CT images," *Scientific Reports*, 2026, doi: 10.1038/s41598-026-37169-8.
34. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations*, 2015.
35. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.