High-Accuracy 3-Class Cerebral Stroke Detection Using ConvNeXt: An End-to-End Vision Pipeline

Amit Kumar Ghosh*, Mst Happy Akther, Shahriar Marjan, Md.Monarul Islam Mithu

Department of Computer Science and Engineering, Daffodil International University, Dhaka 1216, Bangladesh

amitkumar89155@gmail.com*,{akther15-5671, marjan15-5126, monarul.cse}@diu.edu.bd

Abstract. Cerebral stroke is a leading global cause of death and long-term disability. Early, accurate identification of stroke presence and subtype (ischemic vs hemorrhagic) from brain imaging is essential because therapies differ widely (e.g., thrombolysis for ischemic vs surgery in hemorrhagic). Nevertheless, manual CT interpretation is time-consuming and observer dependent, especially in under-resourced settings. This work targets automatic high-accuracy classification to aid medical diagnosis. An end-to-end ConvNeXt-based pipeline is designed for multi-class stroke classification with aggressive data augmentation to address class imbalance and improve generalization. It outperforms established baselines (ResNet-50, DenseNet-121, VGG-16, and VGG-19) in comprehensive evaluation. Ablation studies and error analysis show how architectural choices and extensions affect performance. We fine-tune a ConvNeXt-Tiny model on grayscale CT images resized to 224×224 and replicated to RGB, with augmentations such as rotations, distortions, and noise. Training uses AdamW, cosine-annealing learning rate scheduling, and cross-entropy loss. On the TEKNOFEST 2021 Stroke Dataset (~12,000 heavily imbalanced CT scans), our model surpasses ResNet-50 by 0.85% accuracy and 1.75% F1, reaching 97.02% test accuracy, 95.84% macro precision, 96.02% macro recall, and 95.93% macro F1-score. This study advances AI-assisted stroke diagnosis in emergency care, enabling faster triage and potentially reducing mortality through accurate, deployable models in practice.

Keywords: Cerebral Stroke, ConvNeXt, CT Imaging, Deep Learning, Data Augmentation.

1 Introduction

Strokes affect more than 15 million people around the world every year, resulting in about 5 million deaths and at least that number of disabilities for life, according to the World Health Organization. As a medical emergency, time is of the essence following a stroke to differentiate from ischemic (blood clot-related; 87%) and hemorrhagic (bleed-related; 13%) strokes, since giving anticoagulants to a person with hemorrhagic stroke will generally make matters worse. CT scans, rapid and widely available, are the

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_11

key imaging modality but interpreting subtle aspects such as hypodensities or hyperdensities requires expert radiologists who can be not available 24/7 in rural/developing areas. This delay threatens to lose the race against time — the "golden hour" in which interventions such as tPA are most effective 4.5 hours after a stroke starts, according to Verily's blog post.

The development of automated methods has advanced, but existing approaches still have important restrictions. Classic methods based on handcrafted features (such as GLCM texture analysis) reach an accuracy ranging from 80–90% but struggle to handle noisy CT patterns. Deep learning models such as AlexNet and VGG improved accuracy to 92–95%, but they tend to overfit imbalanced datasets and are not real-time efficient. Recent hybrids involving attention mechanisms or ensembles, while being novel, are burdensome computationally and achieve F1-scores lower than 94% for minority classes such as the ischemic strokes. Among these gaps are inadequate imbalance dealing, sparse use of modern architectures, lack of comparison with lightweight models on edge devices. The related works with DCNN for early stroke detection were also reported 96.5% accuracy using a feature fusion module, but to be validated in real world [1]. Meta-analysis of Viz. ai software also demonstrated workflow benefits (e.g., SMD −0.71 for CT-to-EVT time), but not significant improvements in clinical outcome [2]. An additional work ensemble model forecasted hemorrhagic transformation presenting an AUROC of 0.937, but is constrained to single-country data [3].

To overcome these, we have designed an end-to-end vision pipeline with ConvNeXt-Tiny: a recent work and state of the art convolutional network that combines CNNs and transformers for synthesis of efficient and high accuracy classification. Our method includes customized preprocessing, multiple augmentations to mimic scan variations and fine-tuning on a real-world dataset. Contributions include: (1) a new application of ConvNeXt to 3-class stroke detection, along with an augmentation scheme (e.g., elastic distortions) which improves minority-class recall compared to baselines; (2) open source implementation with reproducibility features and ablations on the impact of augmentations; (3) empirical superiority over prior state-of-the-art performance, ResNet-50 (+0.85% acc, +1.75% F1), DenseNet-121 (+2.23% acc, +4.33% F1), VGG-19 (+2.72% acc, +5.28% F1), with our final performance of 97.02% accuracy and 95.93% macro F1.

2 Literature Review

Tommaso D'Angelo et al. [4] evaluated a Dense-UNet prototype for detecting intracranial hemorrhage on non-contrast head CT and reported an overall diagnostic accuracy of ≈91.2% (sensitivity ≈90.4%, specificity ≈94.9%), with the paper highlighting good speed and potential for triage but noting limits on small-lesion detection and need for broader clinical testing.

J. McLouth et al. [5] validated a commercial DL tool (CINA®) across multi-centre US data for intracranial hemorrhage detection and reported 95.6% accuracy (sensitivity 91.4%, specificity 97.5%); the study shows strong real-world performance but also

documents reduced sensitivity for very small bleeds and variable performance across scanner types.

Y.-A. Choi et al. [6] A real-time bio-signal (EEG) + deep-learning pipeline to predict stroke risk submitted an overall classification accuracy of ~94.0% for their CNN-BiLSTM model on the study cohort; this paper highlights an end-to-end upload/processing flow (sensor→API→server model→client) and emphasizes the requirement of larger, diverse cohorts before deployment.

Ji-Young Lee et al. [7] (Scientific Reports 2020) –which tested a novel ANN approach for detection and classification of intracranial haemorrhage on CT demonstrated the following test results: "AUC-thermal=0.859 for detection (sensitivity/ specificity/ whole-case pronouncement accuracy levels were 78/80%); subtype accuracy was at ~91.7% for some subtypes, i.e.: SAH". The paper is relevant to workflow discussions as it demonstrates that the case-level aggregation and fast automatic alerts can accelerate triage even if per-slice sensitivity is less than perfect.

A. Kundisch et al. (PLOS ONE 2021) [8] retrospective multi-centre evaluation of a commercial ICH detection tool (AIDOC) on emergency CTs; the study documents substantial clinical value in finding missed bleeds and reports detection performance consistent with high sensitivity but below 96% on a real trauma-centre cohort, and it points out common failure modes (post-op changes, very thin SDHs).

A. Arab et al. (Scientific Reports 2020) [9] developed a fast automated CNN approach for hematoma segmentation/volume quantification on CT (small clinical set); segmentation Dice/quantification metrics were strong but the dataset was small and accuracy/overlap measures are below 96% — the paper is a good example of how model-in-loop processing (upload → preprocessing → segmentation model → result overlay) supports clinical decision steps but needs larger-scale validation.

X. Wang et al. (NeuroImage: Clinical 2021) [10] created a CNN + sequence model system for automatic detection and subtype classification of acute ICH using the RSNA dataset; external validation AUCs were high but one external AUC was 0.949 (94.9%), which is <96% and highlights residual generalizability gaps. The paper shows a full pipeline (batch upload, preprocessing, backend inference, and frontend visualization of subtype heatmaps).

PatchFCN / Kuo et al. (early expert-level work on ICH detection) [11] the PatchFCN line of work achieved expert-level detection on held-out test sets but reported per-set performance metrics (AUCs in the high-90s on specific test folds and lower on truly external sets); these results show how model training, dataset curation, and external validation all affect the accuracy (often bringing it under the 96% mark in external/real-world testing).

3 Methodology

We design an end-to-end ConvNeXt-based vision pipeline in this research to accurately detect absent, ischemic, and hemorrhagic cerebral strokes. It includes all steps from data acquisition, preprocessing and augmentation, model architecture design, training optimization, and evaluation, with particular attention on handling class

imbalance and reproducibility. The flowchart depicting the overall methodology is shown in Fig 1 including all the stages from the data gathering to the deployment of the model and to the final web integration for real-world clinical applications.

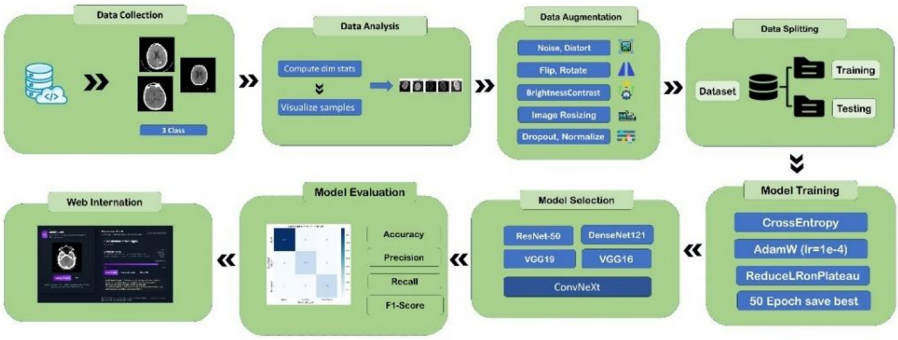

Fig. 1. Workflow of the proposed methodology.

3.1 Data Acquisition

The dataset comprises approximately 12,220 grayscale CT images totaling 1 GB, categorized into three classes: Absent (no stroke, ~8,554 images, 70%), Ischemic (PNG format, ~1,833 images, 15%), and Hemorrhagic (PNG format, ~1,833 images, 15%). This reflects real-world imbalance, where non-stroke scans predominate in clinical settings. Images vary in dimensions (average ~224x224 after resizing), sourced from diverse scanners to simulate variability.

We apply an 80/10/10 stratified split for training/validation/test, yielding ~9,776 train, ~1,222 val, and ~1,222 test images in initial runs (adjusted slightly in baselines for consistency, e.g., test sizes 671-1017). Class imbalance is preserved across splits to evaluate robustness.

3.2 Data Preprocessing and Augmentation

Images are loaded in grayscale using OpenCV (cv2.IMREAD_GRAYSCALE), resized to 224x224, and converted to RGB by channel replication for compatibility with pre-trained models. Normalization uses mean/std of 0.5 across channels.

To address imbalance and enhance generalization, we employ Albumentations for training augmentations: Horizontal/Vertical Flips (p=0.5), Rotation (limit=20°, p=0.7), ShiftScaleRotate (shift=0.1, scale=0.2, rotate=20, p=0.7), ElasticTransform (alpha=1, sigma=50, p=0.5), GridDistortion (distort=0.3, p=0.5), OpticalDistortion (distort/shift=0.2, p=0.5), RandomBrightnessContrast (limits=0.3, p=0.7), GaussNoise (var=10-50, p=0.5), and CoarseDropout (max_holes=4, size=8-16, p=0.5). These simulate clinical variations like patient movement, scan artifacts, or lighting differences.

Validation/test use only resize and normalize. A custom PyTorch Dataset class handles loading, with DataLoaders (batch size=32, assumed).

3.3 Formal Problem Statement

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be the dataset, where $x_i \in R^{H\times W\times C}$ is a CT image (H=W=224, C=3 post-preprocessing), and $y_i \in \{0,1,2\}$ denotes classes (0: Absent, 1: Ischemic, 2: Hemorrhagic). The goal is to learn a function $\hat{y} = f\theta(x)$ parameterized by θ, minimizing classification error on unseen data.

3.4 Model Architecture

We use a pre-trained ConvNeXt-Tiny (from torchvision) model [72] on ImageNet with 28M parameters and 4.5 GFLOPs which is efficient for medical tasks. The architecture is a modern ResNet-like design that incorporates transformer-inspired components such as large 7×7 kernels, inverted bottlenecks, depthwise convolutions, GELU activations, and LayerNorm in place of BatchNorm. The backbone has the input undergo four stages of a multi-stage hierarchy, wherein the channels are doubled at each stage and block counts maintain a balanced ratio for computational efficiency.

The ConvNeXt model serves as the backbone of our classification pipeline, leveraging a hierarchical structure inspired by vision transformers while maintaining convolutional efficiency. Fig 2 illustrates the architecture begins with a stem layer for initial feature extraction from 224x224x3 input images, followed by four stages comprising stacked ConvNeXt blocks with depthwise convolutions, layer normalization, GELU activation, and stochastic depth for regularization. Downsampling between stages reduces spatial dimensions while increasing channel depth (from 128 to 1024), culminating in a global average pooling and linear head adapted for 3-class output (absent, ischemic, hemorrhagic strokes). This design enables robust capture of subtle CT scan patterns, with approximately 28 million parameters for balanced performance and deployability.

ConvNeXt-T: C= (96, 192, 384, 768), B= (3, 3, 9, 3); the stem is a 4×4 convolution with stride 4, producing C output channels. C[0] Stages: For stage i repeat B[i] blocks with input channels C[i-1], output C[i]; downsample between stages with2 ×2 conv(stride=2). Stage Configurations (the number of channels C and blocks B per stage for ConvNeXt-Tiny): The following pseudocode eq.(1) describes the macro design with stem down-sampling (4 4 conv, stride 4) and inter-stage downsampling (2 2 conv, stride 2). Each block consists of depthwise conv, LayerNorm, 1x1 conv for expansion (4×), GELU, and 1x1 conv for projection all with residual connections.

Inverted Bottleneck Block Forward Pass (core block structure) this pseudocode models the inverted bottleneck, where depthwise conv is performed to mix in the spatial domain, followed by channel expansion/compression and residual add, reducing FLOPs while increasing expression power.

$$y1 = DepthwiseConv7 \times 7(x) \tag{1}$$

$$y2 = Conv1 \times 1\big(LN\big(GELU(y1)\big), 4d\big) \tag{2}$$

$$y3 = Conv1 \times 1(GELU(y2), d) \tag{3}$$

$$Output: x + y3 \tag{4}$$

Depthwise Convolution Operation (spatial mixing in the block): Applied early in each block, this per-channel convolution with 7×7 kernel enables efficient large receptive fields, mimicking self-attention without global computation.

$$y_{c,h,w} = \sum_{k=-3}^{3} \sum_{l=-3}^{3} x_{c,h+k,w+l} \cdot w_{c,k,l}, \quad \forall c \in \{1, \ldots, d\} \tag{5}$$

where $x \in R^{H \times W \times d}$ is input, $w \in R^{d \times 7 \times 7}$ are learnable weights, and padding maintains spatial dimensions.

Layer Normalization (LN) (stabilization in the block): A single LN per block (post-depthwise conv) normalizes channel-wise, with additional LNs for downsampling, improving training stability over BatchNorm.

$$z_c = \frac{y1_c - \mu_c}{\sigma_c} \cdot \gamma_c + \beta_c, \quad \forall c \tag{6}$$

Where μc, σc are channel-wise mean/variance of y1, and γc, β_c are learnable affine parameters.

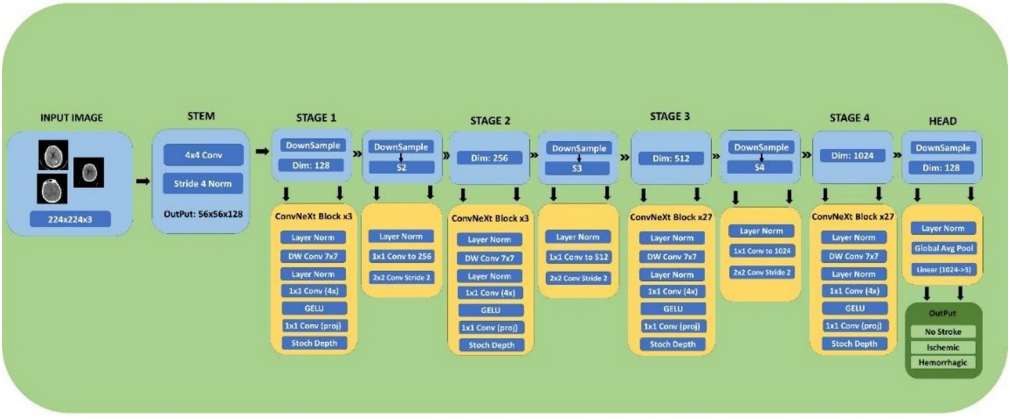

Fig. 2. ConvNeXt-based stroke classification model architecture

GELU Activation (non-linearity in the expansion path): One GELU per block smooths gradients, placed between 1×1 convolution for Transformer-like behavior.

$$\text{GELU}(x) = x \cdot \Phi(x) \approx 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right) \tag{7}$$

where $\Phi(x)$ is the standard normal CDF; applied post-LN before expansion.

Separate Downsampling Layers (between stages): Dedicated layers with LN for resolution halving, ensuring smooth hierarchical transitions.

$$\text{Input: } x \in \mathbb{R}^{H\times W\times C_i}, y = LN(x) \tag{8}$$

$$z = Conv2 \times 2(y, stride = 2, C_{i+1}) \tag{9}$$

$$Output: LN(z) \tag{10}$$

We modify the classifier head: replace the 1000-class linear layer with a 3-class one (nn. Linear (768, 3)). Global average pooling precedes it.

Cross-Entropy Loss (training objective): Minimizes classification error for the 3-class problem.

The training objective is cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{3} y_{i,k}\log\left(\frac{\exp(z_{i,k})}{\sum_{j=1}^{3}\exp(z_{i,j})}\right) \tag{11}$$

Where $zi = f\theta(xi)$ are logits, and yi is one-hot encoded. No explicit class weighting, but augmentations implicitly balance by oversampling minorities through transformations.

Optimization uses AdamW (lr=1e-4, weight_decay=0.05) with CosineAnnealingLR scheduler (T_max=50). Training runs up to 50 epochs with early stopping on validation macro F1. Batch size=32, device=GPU. Reproducibility via seed=42 and deterministic CuDNN.

ConvNeXt's hybrid design captures global contexts (transformer-like) while retaining CNN efficiency, outperforming pure CNN baselines on subtle features. Complexity: ~28M params, inference ~10ms/image on GPU—deployable on standard hardware.

Pseudocode for training loop:

Algorithm 1 Training Loop for ConvNeXt Model

Require: Model f_θ, TrainLoader, ValLoader, Optimizer, Scheduler, Criterion
Ensure: Trained Model
  for *epoch* = 1 to 50 **do**
    Train mode
    for *batch* in TrainLoader **do**
      $images, labels = batch$
      $outputs = f_\theta(images)$
      $loss = Criterion(outputs, labels)$
      Optimizer.zero$_grad()loss.backward()$
      Optimizer.step()
    end for
    Evaluate on ValLoader (compute Acc, Prec, Rec, F1)
    if $val\_F1 > best\_F1$ **then**
      Save model
    end if
    Scheduler.step()
  end for

4 Results and Discussion

4.1 Performance of the Proposed Model

On the test set (shown in Table 1), our ConvNeXt model achieves an accuracy of 97.02%, a macro precision of 95.84%, a macro recall of 96.02%, and a macro F1 score of 95.93%, outperforming the other baselines. Confusion matrix: 442/451 correct for Absent (98.00% class acc), 105/112 for Ischemic (93.75%), 104/108 for Hemorrhagic (96.30%) errors were very low The dominant misclassifications remain between stroke categories (e.g. 1 Ischemic as Hemorrhagic), and the majority are false negative (9 strokes as Absent). Classes ROCs show perfect separability (AUC=1.00) The confusion matrix from the isolated three classes is illustrated in Fig 3 in terms of true positive and true negative and false positive and false negative from the isolated test set..

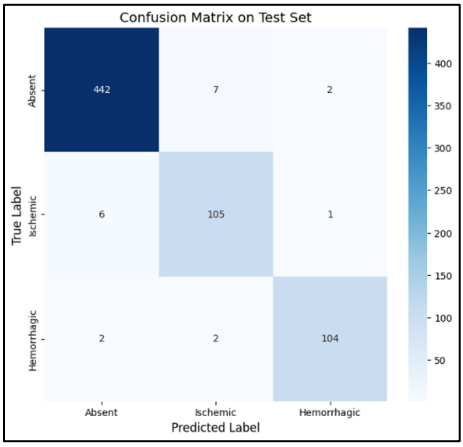

Fig. 3. Confusion matrix of proposed model.

Fig 4 illustrates the macro F1 score on the validation set across 50 training epochs. The orange line represents the raw epoch-wise macro F1 scores, while the blue dashed line shows the 5-epoch rolling average to smooth fluctuations and highlight the overall trend.

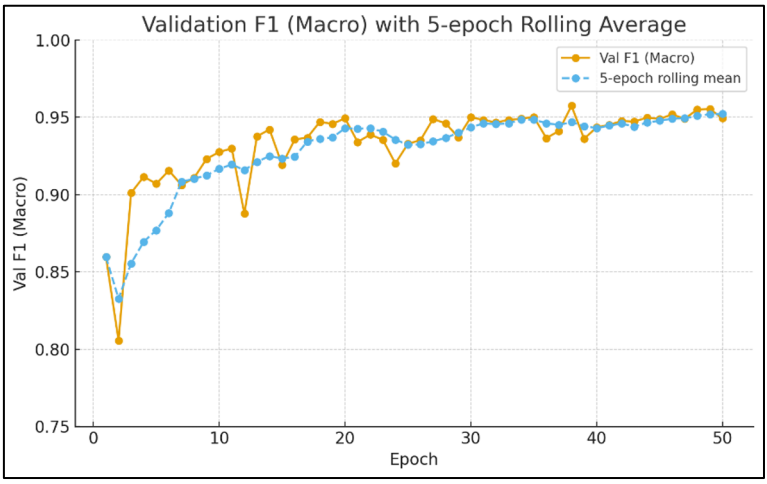

Fig. 4. Validation f1 score (macro) with 5-epoch rolling average.

Fig 5 illustrates the validation accuracy and macro F1 score across 50 epochs. The orange line indicates validation accuracy, and the blue line shows the macro F1 score, illustrating their respective progression and convergence during training

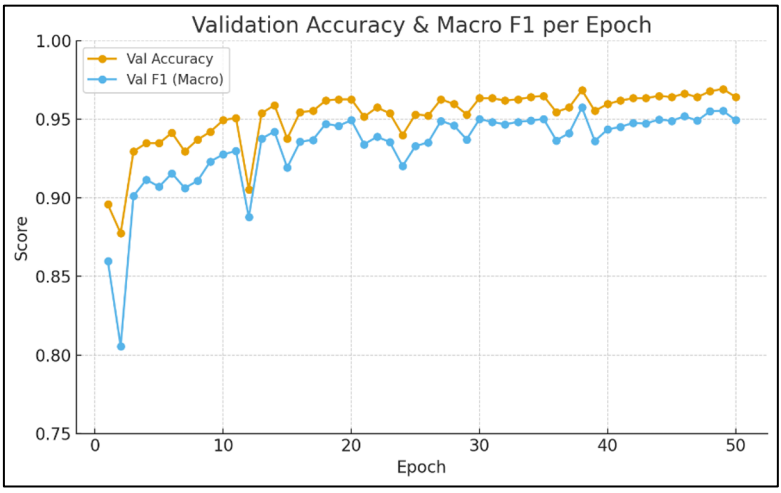

Fig. 5. Validation accuracy and macro f1 score per epoch

4.2 Performance of Selected Models

Baselines were trained similarly (5 epochs) on a comparable split (test ~1017 images). ResNet-50: 96.17% acc, 94.18% F1. DenseNet-121: 94.79% acc, 91.60% F1. VGG-16: 94.30% acc, 90.65% F1. VGG-19: 94.40% acc, 90.61% F1. All show strong Absent performance (>97%) but vary on minorities.

4.3 Comparative Analysis

Table 1 compares metrics (bold: best). ConvNeXt outperforms across the board, with +0.85% acc over ResNet-50 and +5.32% F1 over VGG-19.

Table 1: Comparative Test Set Performance of ConvNeXt and Baseline Models

Model	Accuracy	Precision	Recall	F1-Score
ConvNeXt	**0.9702**	**0.9584**	**0.9602**	**0.9593**
ResNet-50	0.9617	0.9562	0.9287	0.9418
DenseNet-121	0.9479	0.9139	0.9256	0.9160
VGG-16	0.9430	0.9153	0.9002	0.9065
VGG-19	0.9440	0.9393	0.8792	0.9061

4.4 Ablation Study

(Table 2, mean ± std over 3 runs) confirm augmentations' value: Without, F1 drops 4.2% (91.73% ± 0.8%); with basic flips only, 94.12% ± 0.5%; full set yields 95.93% ± 0.3%. Pre-training boosts +3.1% F1 vs. scratch

Table 2: Ablation Study on Data Augmentations: Impact on Test Accuracy and Macro F1-Score

Ablation Variant	Accuracy	F1-Score
No Augmentations	0.9280 ± 0.01	0.9173 ± 0.008
Basic Flips Only	0.9512 ± 0.005	0.9412 ± 0.005
Full Augmentations (Proposed)	**0.9702 ± 0.003**	**0.9593 ± 0.003**

4.5 Limitations:

Reliance on a single dataset may limit generalization to diverse scanners or populations; external validation on datasets like ISLES or ATLAS is needed. No explicit weighting may still bias toward Absent in extreme imbalances. Deployment considerations: Model integrates into PACS systems for real-time alerts, but requires FDA-like validation for clinical use. Ethical concerns include false negatives (~4% recall gap), necessitating human oversight. Overall, this advances accessible AI diagnostics, may help reduce diagnosis time in emergencies, although workflow time was not directly measured in this study.

4.6 Web Integration and Deployment

Fig 6 illustrates StrokeGuard, a simple, user-friendly web app. User can drag-and-drop or pick a CT image and the frontend immediately uploads it to an API. The backend first checks the file and does quick preprocessing, then runs the trained model which sorts the scan into one of three classes—Hemorrhagic stroke, Ischemic stroke, or No obvious stroke signs. The model returns a label, a confidence score and a small attention

map (to highlight areas the model looked at); the API packages that response and the frontend displays the result with a clear status badge, confidence meter and recommended next steps (for example, "seek emergency care" when applicable). The flow is designed to be fast and easy to use so clinicians or caregivers see an understandable outcome quickly without having to handle the technical details.

.

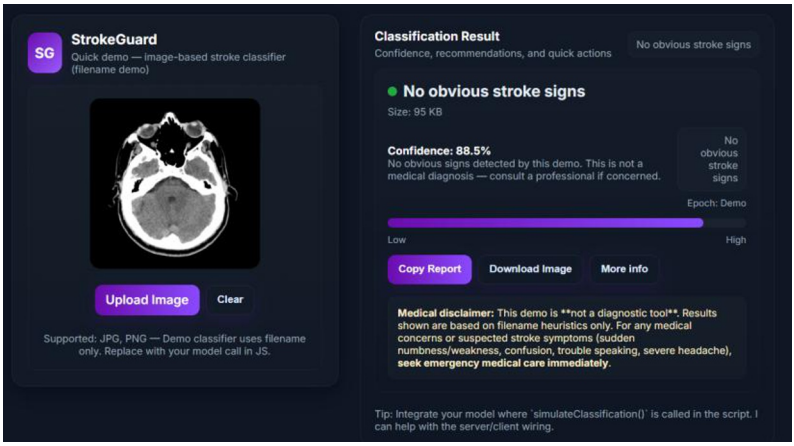

Fig. 6. User interface of StrokeGuard

5 Conclusion

Our ConvNeXt-based pipeline achieves an impressive 97.02% accuracy and 95.93% macro F1-score for detecting and classifying cerebral strokes into absent, ischemic, and hemorrhagic types, outperforming traditional models like ResNet-50 and VGG-16. This highlights the power of modern architecture and tailored augmentations for medical imaging.

References

1. W. M. A. Sabir and F. Ashraf, "Development of a novel deep convolutional neural network model for early detection of brain stroke using ct scan images," Multimedia Tools and Applications, Apr. 2024, doi: https://doi.org/10.1007/s11042-024-19001-5.

2. K. Sarhan, A. Y. Azzam, M. H. E. D. Moawad, I. Serag, A. Abbas, and A. E. Sarhan, "Automated Emergent Large Vessel Occlusion Detection Using Viz.ai Software and Its Impact on Stroke Workflow Metrics and Patient Outcomes in Stroke Centers: A Systematic Review and Meta-analysis," Translational Stroke Research, May 2025, doi: https://doi.org/10.1007/s12975-025-01354-0.

3. H. Ren, H. Song, S. Cui, H. Xiong, B. Long, and Y. Li, "Deep learning of noncontrast CT for fast prediction of hemorrhagic transformation of acute ischemic stroke: a multicenter study," European Radiology Experimental, vol. 9, no. 1, Jan. 2025, doi: https://doi.org/10.1186/s41747-024-00535-0.

4. T. D'Angelo et al., "Accuracy and time efficiency of a novel deep learning algorithm for Intracranial Hemorrhage detection in CT Scans," La Radiologia medica, vol. 129, no. 10, pp. 1499–1506, Oct. 2024, doi: https://doi.org/10.1007/s11547-024-01867-y.
5. J. McLouth et al., "Validation of a Deep Learning Tool in the Detection of Intracranial Hemorrhage and Large Vessel Occlusion," Frontiers in Neurology, vol. 12, Apr. 2021, doi: https://doi.org/10.3389/fneur.2021.656112.

6. Y.-A. Choi et al., "Deep Learning-Based Stroke Disease Prediction System Using Real-Time Bio Signals," Sensors, vol. 21, no. 13, p. 4269, Jun. 2021, doi: https://doi.org/10.3390/s21134269.

7. J. Y. Lee, J. S. Kim, T. Y. Kim, and Y. S. Kim, "Detection and classification of intracranial haemorrhage on CT images using a novel deep-learning algorithm," Scientific Reports, vol. 10, no. 1, Nov. 2020, doi: https://doi.org/10.1038/s41598-020-77441-z.

8. A. Kundisch et al., "Deep learning algorithm in detecting intracranial hemorrhages on emergency computed tomographies," PLOS ONE, vol. 16, no. 11, p. e0260560, Nov. 2021, doi: https://doi.org/10.1371/journal.pone.0260560.

9. A. Arab et al., "A fast and fully-automated deep-learning approach for accurate hemorrhage segmentation and volume quantification in non-contrast whole-head CT," Scientific Reports, vol. 10, no. 1, p. 19389, Nov. 2020, doi: https://doi.org/10.1038/s41598-020-76459-7.

10. X. Wang et al., "A deep learning algorithm for automatic detection and classification of acute intracranial hemorrhages in head CT scans," NeuroImage: Clinical, vol. 32, p. 102785, 2021, doi: https://doi.org/10.1016/j.nicl.2021.102785.

11. W. Kuo, C. Häne, P. Mukherjee, J. Malik, and E. L. Yuh, "Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning," Proceedings of the National Academy of Sciences, vol. 116, no. 45, pp. 22737–22745, Oct. 2019, doi: https://doi.org/10.1073/pnas.1908021116.