JCTA Journal of Computing Theories and Applications ISSN:3024-9104

*Research Article*

# Adversarial Convolutional Neural Network for Predicting Blood Clot Ischemic Stroke

**Moshood A. Hambali * and Paul A. Agwu**

Department of Computer Science, Faculty of Computing and Information System, Federal University Wukari, Nigeria; e-mail : hambali@fuwukari.edu.ng; pauljtechcom@gmail.com
* Corresponding Author : Moshood A. Hambali

**Abstract:** Digital Pathology Image Analysis (DPIA) is one of the areas where deep learning (DL) techniques offer modern, cutting-edge functionality. Convolutional Neural Network (CNN) technology outperforms the competition in classification, segmentation, and detection tasks while being just one of numerous DL techniques. Classification, segmentation, and detection methods can often be used to address DPIA concerns. Some difficulties can also be resolved using pre- and post-processing techniques. However, other CNN models have been investigated for use in addressing DPIA-related issues. Furthermore, the research seeks to explore how susceptible the model is to adversarial attacks and suggest strategies to counteract them. To predict ischemic strokes caused by blood clots, the authors of this study developed CNN with a pixel brightness transformation (PBT) technique for image enhancement and developed several approaches of image augmentation techniques to increase and provide the learning model with more diverse features. Also, adversarial training was integrated into CNN models to train the model with perturbed data in order to assess the impact of adversarial noise at different stages of training. Several metrics, including precision, F1-score, accuracy, and recall, are utilized to assess the experiments' effectiveness. The research findings indicate that employing transfer learning with a deep learning model achieved an accuracy of up to 97% using the ReLU activation function. Also, data augmentation helps improve the accuracy of the model.

**Keywords:** CNN; Data augmentation; Adversarial training; Deep learning; Pixel brightness transformation.

**Received:** April, 19th 2024
**Revised:** May, 18th 2024
**Accepted:** May, 29th 2024
**Published:** June, 1st 2024



## 1. Introduction

Stroke is the second leading cause of global mortality and has consistently contributed significantly to the decline in human well-being and vitality[1]. In the United States, ischemic strokes occur at a rate of approximately 795,000 annually[2]. It stands as the second most prevalent cause of disability after dementia, impacting 15 million individuals each year, leading to the deaths of about 6 million and causing permanent disability in another 5 million[3]. This condition is marked by heightened levels of morbidity, disability, mortality, and recurrence, imposing a substantial burden on society and families. Notably, there has been a significant increase in stroke prevalence among low-income individuals and younger age groups[4]. Magnetic Resonance Imaging (MRI) has become crucial in diagnosing and treating stroke, particularly in distinguishing ischemic stroke from hemorrhagic stroke[5].

A stroke, characterized by sudden blood vessel rupture or blockage in the brain, results in acute cerebral vascular disease[6]. It can manifest as ischemic or hemorrhagic stroke, both causing brain tissue damage. Ischemic stroke occurs due to inadequate blood flow to a specific brain area, leading to impaired neural processes and enduring impairments[7]. MRI scans offer rich data for assessing ischemic stroke severity, yet analyzing results is challenging due to the complexity of identifying subtle image alterations indicative of stroke severity[8]. Manual image analysis requires significant effort and can lead to cognitive fatigue errors and diminishing diagnostic quality[7]. Moreover, inter- and intra-observer variability in human judgment-based categorization methods adds to the challenges. Educating individuals to achieve expertise is time-consuming and costly, making routine stroke risk estimation economically unfeasible.

However, researchers have increasingly focused on diagnosing ischemic stroke using artificial intelligence (AI) methods in MRI scans. Machine learning (ML) and deep learning (DL) algorithms, including K-Nearest Neighbors (KNN), support vector machines (SVM), decision trees (DT), random forests (RF), and neural networks, have shown promise in ischemic stroke detection [9]. DL, particularly convolutional neural networks (CNNs), closely aligns with AI's core objectives, replicating the hierarchical organization observed in the human brain [10]. It autonomously extracts features, exhibits high learning proficiency, and has advanced significantly due to high-performance GPU servers and extensive datasets[8].

These advancements have led to exploring DL technology for ischemic stroke prediction using CNNs. This study proposes the Adaptive CNN with pixel brightness transformation technique for image enhancement and adversarial training to predict Blood Clot Ischemic Stroke. It aims to classify, detect, and segment strokes using MRI datasets from Kaggle repositories. It also seeks to explore how susceptible the model is to adversarial attacks and suggest strategies to mitigate against them. The paper's subsequent sections are, section 2 reviews related literature; section 3 outlines methodology and describes performance metrics; section 4 presents results and discussions, and section 5 concludes the paper.

## 2. Related Works

Takahashi et al. [11] developed Return on Investment (ROIs) around the Sylvian fissure region and identified MCA dots using morphologic top-hat transformation, employing SVM with four features. SVM classification on 297 CT images from seven patients with MCA dot signs achieved a maximum sensitivity of 97.5% at a false positive rate of 1.28 per image and 0.5 per hemisphere. Forkert et al.[12] utilized 12 SVM models to predict 30-day mRS scores of ischemic stroke patients, integrating lesion overlap, stroke laterality, and optional features like infarct volume and NIHSS. Integrating optional features and stroke location information improved mRS prediction, reaching 56% for multi-value prediction accuracy and 85% for dichotomized mRS prediction.

Ho et al. [13] compared ML methods to classify transcription start site (TSS) using magnetic resonance (MR) imaging features, proposing a DL model to extract hidden representations from MR perfusion-weighted images, enhancing classification. Cross-validation showed the best classifier achieving an AUC of 0.68, outperforming current clinical methods. Yu et al. [14] developed a method to detect hemorrhagic transformation in stroke using Percussion-weighted imaging (PWI) and diffusion-weighted imaging (DWI), trained CNN with 3-fold cross-validation, achieving an accuracy of 83.7 ± 2.6% with Kernel spectral regression. Subudhi et al.[15]proposed a watershed-based lesion segmentation algorithm (WLSA) for lesion segmentation in DWI MR images, achieving high accuracy in delineating lesions with a 96% dice similarity index (DSI) using an RF classifier.

Scalzo et al.[16] studied hemorrhagic transformation prediction using PWI in MRI, achieving >85% accuracy with nonlinear predictive models. Nielsen et al.[17] employed CNNdeep to calculate lesion volume in Intravenous tissue-type Plasminogen Activator (IV tPA) - treated patients, achieving 88% accuracy in predicting final infarct volume. Monteiro et al.[18] used ML to predict functional outcomes in ischemic stroke patients, observing improved AUC (>0.90) with additional features over time. Subudhi et al.[19] integrated Delaunay triangulation (DT) and fractional order Darwinian particle swarm optimization (DT-FODPSO) for ischemic lesion segmentation, achieving high sensitivity (0.93) and accuracy (0.95) using the RF classifier. Karthik et al.[20] employed a Fully Convolutional Network (FCN) for ischemic lesion segmentation, achieving a mean accuracy of 0.70 on the Ischemic Stroke Lesion Segmentation (ISLES) 2015 dataset. Govindarajan et al.[21] utilized text-mining and ML algorithms for stroke classification, achieving a classification accuracy of 95% with ANN. Yu et al.[22] used a U-net neural network for infarct lesion prediction, achieving an AUC of 0.92 and a Dice Score Coefficient (DSC) of 0.53, aiding clinical decision-making. Vupputuri et al.[23] proposed superpixel-based hierarchical clustering (SSHC) for lesion detection, outperforming state-of-the-art methods with a Dice score of 0.704.

Liu et al.[24]developed Res-CNN for ischemic lesion segmentation, achieving good performance compared to other networks. Badriyah et al.[25] optimized CT image quality and employed RF for stroke classification, achieving an accuracy of 95.97%. Tazin et al.[26] used

LR, DT, RF, and Voting Classifier to predict stroke occurrence, achieving 96% accuracy with RF. Shoily et al. [27] employed Naive Bayes, J48, k-NN, and RF for stroke classification, with Naive Bayes achieving 85.6% accuracy. Fang et al. [28] utilized DL and ML for ischemic stroke subtype classification, finding Resnet and RF effective in subtyping with neurological deficits. Tarkanyi et al.[29] employed ML methods for predicting LVO in AIS patients, achieving AUC values up to 0.775 with LR. Kuang et al.[30] automated ASPECTS scoring using RF, achieving an ICC of 0.76 and high sensitivity and specificity.

## 3. Proposed Method

The methodological approach proposed by this study considers four phases. The first step encompasses reading the stroke images dataset. The second phase encompasses fixing the dataset; this consists of correctly sorting the slices, removing the existing dumps of features, and enhancing the images with the Pixel brightness transformation (PBT) approach. The idea is to create a dictionary where each key represents a patient ID, while the value is the list of correctly sorted images. Creating such a dictionary was quite demanding since it required a visual analysis of the entire dataset and determining the correct sequence for each patient. The next stage involves augmenting the images and conducting adversarial training on the filtered images to a convolutional neural network. Lastly, a performance evaluation analysis from the output of the convolutional neural network is conducted. The stepwise approach for implementing the proposed stroke classification model is shown in Figure 1.

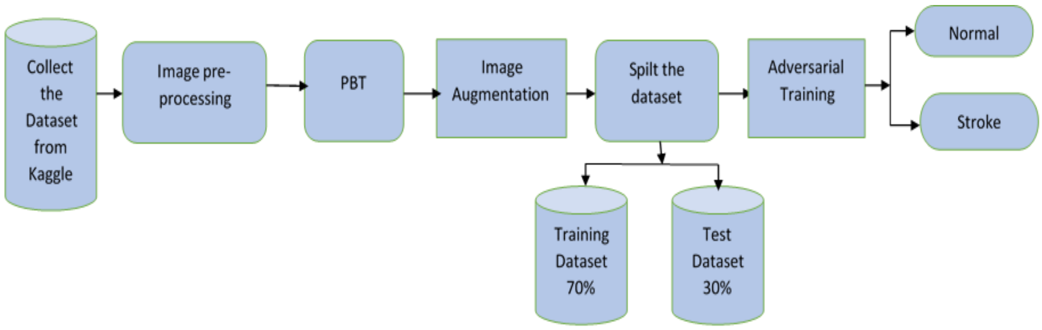


**Figure 1.** Proposed methodology framework.

### 3.1. Dataset Descriptions

The dataset used for analyzing brain strokes is sourced from Kaggle and consists of Computed Tomography (CT) images. The dataset can be found at www.kaggle.com/afridirahman/brain-stroke-ct-image-dataset. The dataset comprises 2,501 high-resolution whole-slide digital pathology images with a total of 1551 images labeled as normal and 950 images labeled as strokes. Each image in the dataset has dimensions of 224 pixels by 224 pixels. The sample images can be visualized in Figure 2. The "Brain Stroke CT Image Dataset" available on Kaggle stands out due to its focus on CT images dedicated to brain strokes. CT images are commonly employed in medical imaging to visualize internal body structures, including the brain. This dataset exclusively concentrates on CT images associated with brain stroke cases, a medically critical condition. Additionally, the datasets encompass images medical practitioners and researchers utilize to diagnose and plan treatments for brain strokes. As a valuable resource, this dataset caters to researchers, clinicians, and data scientists interested in advancing medical imaging technology and enhancing brain stroke diagnosis and treatment.

### 3.2 Data Preprocessing

Data cleansing is a crucial step before building a model in machine learning. Image preprocessing aims to improve the image data by suppressing undesired distortions or enhancing some of the image features relevant to the analysis task. The concept of image preprocessing doesn’t increase image formation content but decreases it if the entropy is an information measure. In essence, the idea behind image preprocessing is to process image data at its lowest level of abstraction. The approach to image preprocessing adopted by this study is the PBT, data fixing and data scaling.

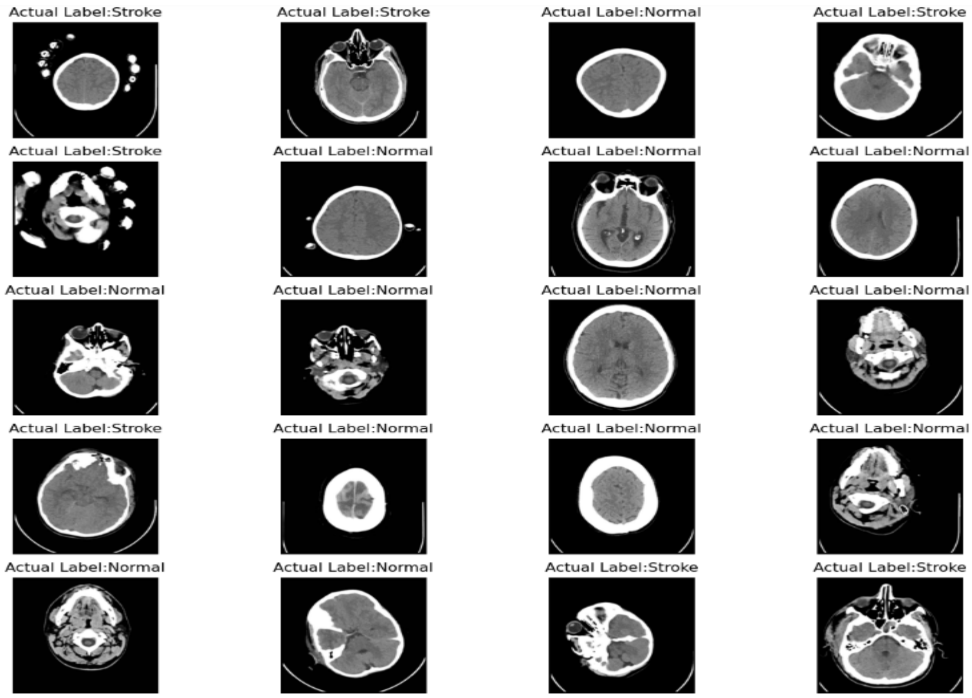


**Figure 2.** Dataset Samples

### *3.2.1 Pixel Brightness Transformation (PBT)*

PBT is also referred to as brightness corrections or intensity transformation. Brightness transformations modify pixel brightness, and the transformation depends on the properties of an image pixel. This implies that the output pixel's value depends only on the corresponding input value. The choice of using PBT for this study is because it offers several advantages over other image enhancement techniques, which include Simplicity, Real-time processing, Non-destructive of original image data, Global and Local Enhancement, Enhancement control through the choice of transformation functions and parameters, Adaptability, Compatibility and Low computational cost.

The approach to BPT techniques adopted by this study is the grayscale transformation, where an image is transformed into a grey scale to extract objects in images. The operations proposed for the grayscale transformation are gamma correction or power law transformation, which uses two common operations, namely multiplication and addition with a constant, as mathematically represented in Equation (1).

$$g(x) = \alpha f(x) + \beta \tag{1}$$

The parameters $\alpha > 0$ and $\beta$ are called the gain and bias parameters, and sometimes these parameters are said to control contrast and brightness, respectively. Hence, the image brightness and contrast vary for alpha and beta values. The pseudocode for PBT is depicted in Algorithm 1

**Algorithm 1.** Pixel Brightness Transformation

INPUT: Raw_CTImages, Parameters
OUTPUT: Tranformed_Images

```
# Define a function for pixel brightness transformation
def brightness_transform(pixel_value):
    # Define transformation logic here
    # Increase brightness by a factor of 1.2
    transformed_value = pixel_value * 1.2
    # Pixel values must be within valid range (e.g., 0-255 for grayscale)
    transformed_value = min(255, max(0, transformed_value))
    return transformed_value
# Apply transformation to each pixel in the image
```

**Algorithm 1.** Pixel Brightness Transformation

```
def apply_brightness_transformation(input_image):
    # Create an empty matrix for the transformed image
    transformed_image = empty_matrix_like(input_image)
    # Iterate through each pixel in the input image
    for row in range(height(input_image)):
        for col in range(width(input_image)):
            # Get the brightness value of the current pixel
            pixel_value = input_image[row][col]
            # Transform the brightness value
            transformed_value = brightness_transform(pixel_value)
```

#### *3.2.2 Dataset Fixing*

It is necessary to try to fix the dataset. Otherwise, it will be quite challenging to expect good results. The fixing consists of correctly sorting the slices and removing the existing holes by identifying the current order of the slices in the dataset and determining the proper sorting criteria, such as numerical order or chronological order, examine the dataset for any gaps or missing elements. If any holes exist, we consider filling them in with the appropriate data or removing them as needed. The idea is to create a dictionary where each key represents a patient ID, while the value is the list of correctly sorted images. Creating such a dictionary was quite demanding since it required a visual analysis of the entire dataset to determine the correct sequence for each patient.

#### *3.2.3 Data Scaling*

Image scaling involves adjusting the pixel values of the dataset images to a standardized range. In this study, the images are scaled by dividing their pixel values by 255. This process normalizes the pixel intensities to a range between 0 and 1. Standardizing the pixel values ensures that the input data fed into the CNN model is consistent and falls within a uniform range, facilitating the training process. Scaling the images helps prevent issues related to varying pixel intensities across different images in the dataset, ensuring that the model learns from meaningful features rather than being influenced by differences in pixel values.

### 3.3 Image Augmentation

Data augmentation is a method to substantially increase the number of data instances within a dataset to facilitate model training. In the context of image datasets, this technique involves employing basic image processing operations like flipping, rotating, cropping, or padding[31]. These operations transform existing images in the dataset, thereby expanding its size for training neural networks. This study utilized data augmentation to address the challenge of having a small dataset size, which negatively impacted the performance of the proposed CNN. Augmenting the dataset increased its size, providing the learning model with more diverse features. Specifically, five image processing operations were implemented: flipping (at the horizontal axis), shifting (the width and height range for shifting was set to 0.2), zooming at 0.2, shear range at 0.2, and rotation image at 20 degrees augmentation.

### 3.4 Convolutional Neural Network

The proposed CNN is a bio-inspired neural network that integrates convolutional layers, pooling, and a fully connected network akin to a Multi-layer Perceptron[32]. Serving as a complement to the traditional feed-forward network (FFN) in image processing, it embodies three layers: input, hidden, and output, with units representing the neurons in each layer.

The proposed CNN architecture comprises a convolutional 1D layer, a pooling 1D layer, and a fully connected layer. The convolutional layer extracts features from the 1D sequence data through sliding filters, generating feature maps. Hyperparameter tuning selects the number and length of filters, employing non-linear activation functions such as Rectified Linear Unit (RELU) for input and hidden layers and sigmoid for the output layer. RELU addresses vanishing gradient and error issues, accelerating learning. Mathematically, RELU is denoted as Equation (2).

$$f(x) = \max(0, x) \tag{2}$$

Where $x$ denotes the input. A maximum 1D pooling method was suggested to diminish feature dimensions. The proposed CNN includes a fully connected layer for stroke classification, ensuring comprehensive connectivity between neurons. This arrangement guarantees that cell state information is stored and transmitted across layers. The pseudocode for CNN is depicted in Algorithm 2.

**Algorithm 2.** Convolutional Neural Network

```
INPUT: Images, parameters
OUTPUT: Evaluation metrics
Count = -1, train_image [ ], train_labels =[ ]
for the dataset:
        Count=count +1
        Img=resize (file, 60,60)
        train images append (img)
        Grayscale(img)
        train_labels append(count)
mu=mean (train _image)
sigma =seddev[train_image]
for image in train_image
        img=(img-mu) sigma
Validation_image =train_image [1,500]
Data-gen=imageDataGenerator(rotation_range=20
        Horizontal_flip=true,
               vertical_flip=true
         width_shift + 0, 1
         Height_shift=0,1)
Apply data-gen to CNN
Validate CNN-Model
Perform evaluation
```

This study proposed a vertical filter for convolving images in a convolutional model, leveraging spatial locality and pointwise multiplication of functions to generate activation maps. A pooling layer was applied to reduce feature map size, retain vital features, and control overfitting. Max pooling was chosen for its efficacy in identifying sharp features. Batch normalization normalized input, expediting learning, while dropout with a 20% rate mitigated overfitting. Flattening layers consolidated pixel data into a one-dimensional vector for input into dense layers, connecting each neuron to every other neuron. These dense layers classified images into specific labels based on associated probabilities, culminating in the classification decision by the fully connected network.

### *3.4.1 Adversarial Training*

This study employs multiple pre-trained models for transfer learning. Various factors, such as optimization function, activation function, learning rate, group size, dropout percentage, and other hyperparameters, play a significant role in shaping the network's learning process. The most effective settings for each model are determined by conducting extensive experimentation and evaluating performance metrics. To prevent overfitting, a dropout rate of 20% was implemented, and training was carried out for 150 epochs.

During training, adversarial perturbations are introduced to the input data using Neural Structured Learning (NSL). Following this, the models are trained on the perturbed data to assess the impact of adversarial noise at different stages of training, as depicted in Figure 3.

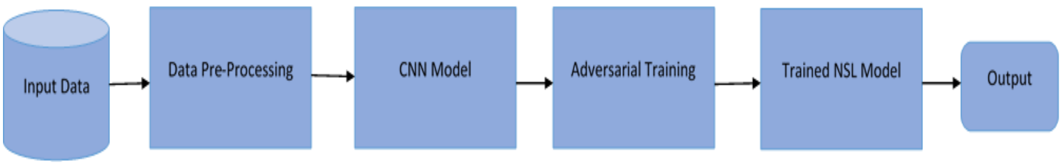


**Figure 3.** Adversarial Training Model

NSL layers were integrated into the CNN architecture. These layers add a regularization term to the loss function, encouraging the model to learn structured representations. For this study, "AdversarialRegularization" was employed and the following hyperparameters were considered: learning rate = 0.0001, batch size = 0.2, NSL regularization strength = 0.2, and optimizer choice ='Adam'.

### 3.5 Evaluation Metric

In assessing the classifier's effectiveness, this research introduced several conventional evaluation measures, including accuracy, precision, recall, and F1-score. These metrics are derived from the concepts of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP denotes the count of correctly identified stroke samples, TN signifies the count of correctly identified non-stroke samples, FP represents the count of non-stroke samples erroneously classified as strokes, and FN indicates the count of stroke samples inaccurately classified as non-strokes.

Accuracy: calculates the ratio of inputs in the test set correctly labeled by the classifier. Mathematically, accuracy can be denoted as Equation (3).

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Precision: defines the percentage of the number of correctly predicted positive outcomes divided by the total number of predicted positive outcomes. Thus, precision can be mathematically denoted as Equation (4).

$$\text{precision} = \frac{TP}{TP + FP} \tag{4}$$

Recall: measures the classifier's completeness. It is the percentage of correctly predicted positive output to the actual number of positive outcomes from the dataset. Recall can be mathematically denoted as Equation (5).

$$\text{recall} = \frac{TP}{TP + FN} \tag{5}$$

Area Under the Curve (AUC) is a metric used to encapsulate the overall effectiveness of a model or to measure data distribution across different fields. It yields a single numeric value that captures performance or distribution within a specific context, making it a broadly utilized and significant metric in numerous applications.

## 4. Results and Discussion

This section presents the results of the experiments carried out. The stroke detection model was meticulously crafted within Anaconda's robust computational environment. Its intricacies are seamlessly integrated into Python, executed on a Windows OS, harnessing the power of a dual-core Intel Core i5 processor and 4GB RAM. Leveraging the TensorFlow API, it meets the intricate demands of deep neural networks in machine learning. Alongside TensorFlow, pivotal Python libraries like NumPy handle numerical operations, while pandas aid in dataset parsing. Matplotlib, a versatile visualization tool, is judiciously employed to illustrate model behavior with precision and clarity.

After the data preprocessing stage, Figures 4 and 5 depict the enhanced images that serve as input images for the model. Figure 4 shows the clean image after executing all the necessary preprocessing stages. Meanwhile, Figure 5 shows the outcome of image augmentation.

The data augmentation process implemented in this study focused on the strategy involved rotating, zooming, shifting, brightening, and contrasting mirrored versions, enriching the dataset with diverse anatomical structures and orientations. This targeted augmentation approach, emphasizing structure changes and lightening, contributed to increased dataset diversity. Consequently, it facilitated improved generalization and robustness of the CNN model during training. The original dataset contains 2,501 image data. After the augmentation, the images increased to 6,126, the distribution depicted in Figure 6. Both the original dataset and augmented data were mixed and used for the training and testing of the model's performance.


**Figure 4.** Enhanced image

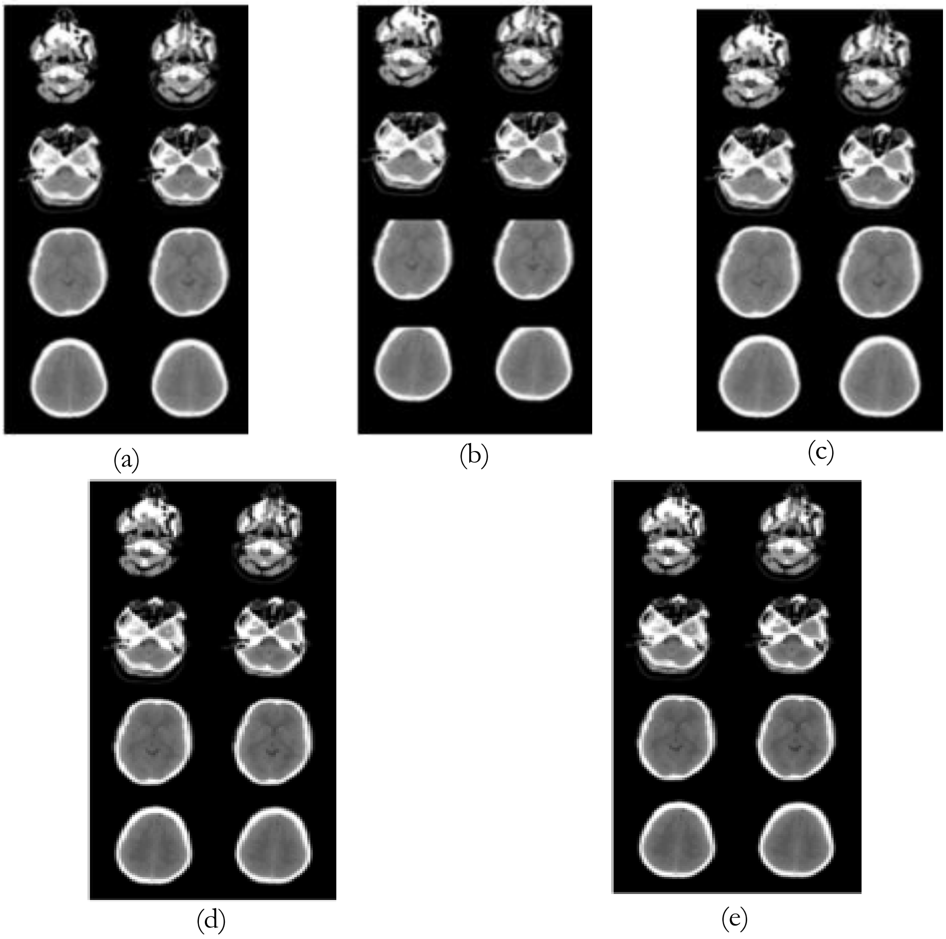


**Figure 5.** Images Augmentation Outcome (**a**) Rotated CT scan; (**b**) Shifted CT scan; (**c**) Zoomed CT scan; (**d**) Brightened CT scan; (**e**) Contrasted Image.

### 4.1 Parameter Setting

Table 1 outlines the parameter configurations for the proposed CNN model. It's crucial to emphasize that these parameters are pivotal for training and optimizing the ischemic stroke detection model using the CNN algorithm. The selection of the loss function, specified as Binary-Crossentropy, indicates the model's setup for binary classification, suitable for tasks with outputs of either positive or negative outcomes, as seen in this context likely representing the presence or absence of ischemic stroke. The maximum number of training epochs is defined as 150, determining how many times the model iterates over the complete training dataset. The Adam optimizer, known for its effectiveness and adaptability, is chosen to adjust the model weights during training. A learning rate of 0.0001 is employed to control the step size in weight updates, influencing the speed and stability of the optimization process. Activation functions utilized include Rectified Linear Unit (ReLU) and Sigmoid, introducing non-linearity to enhance the model's learning capabilities. Additionally, a decay rate of 0.96 is set, potentially indicating a diminishing factor applied to the learning rate over time, which aids in fine-tuning the model's performance throughout training.

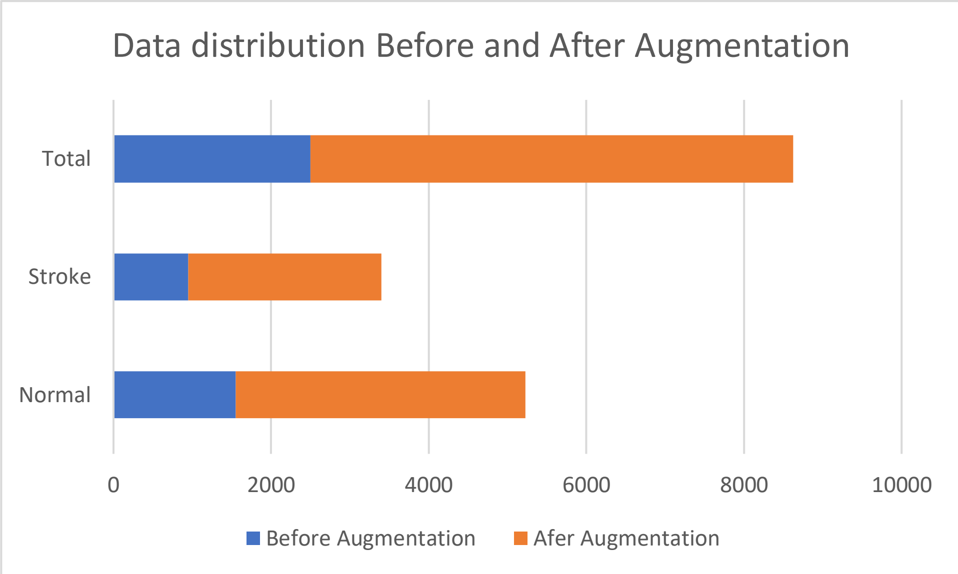


**Figure 6.** Data Distribution

**Table 1.** Ischemic stroke model parameter setting.

| Parameter | Values |
|---|---|
| Loss | Binary-Crossentropy |
| Max epochs | 150 |
| Optimizer | Adam |
| Learning Rate | 0.0001 |
| Activation | ReLU/Sigmoid |
| Decay Rate | 0.96 |
| Loss | Binary-Crossentropy |

These parameter configurations collectively establish the training and optimization attributes of the ischemic stroke detection model, influencing its capacity to learn from and generalize across the input data. To ensure a thorough assessment, 30% of the data records from each dataset were reserved for testing purposes, while the remaining 70% were employed for the extensive training of the ischemic stroke detection model.

### 4.2. Model Architecture

Figure 7 shows the CNN model architecture designed for ischemic stroke detection. The model architecture begins with a Reshape layer, named "reshape," which transforms input data into a 5D tensor with dimensions (128, 128, 64, 1), essential for preparing three-dimensional volumetric data. Four sets of convolutional layers with max pooling, denoted as "conv3d" and "maxpooling3d," are responsible for learning hierarchical features and reducing spatial dimensions. Batch Normalization layers, named "batch_normalization," enhance convergence and training speed. Subsequently, a Global Average Pooling 3D layer condenses feature maps, and two Dense layers, one with 512 units and dropout regularization, precede the final prediction by the second dense layer, "dense_1."

### 4.3 Models Result

The outcomes from the ischemic stroke detection model, as displayed in Table 2, demonstrate noteworthy performance indicators for the CNN model trained over 150 epochs without data augmentation. The 94% precision signifies the model's capacity to correctly identify actual positive cases within the projected positive instances, showcasing a strong level of dependability in categorizing ischemic stroke occurrences. The recall rate of 81% represents the percentage of true positive instances accurately detected by the algorithm, highlighting its sensitivity in identifying cases of ischemic stroke. The model's accuracy of 91% demonstrates its efficacy in accurately predicting outcomes in both positive and negative circumstances, offering a full assessment of its performance. The results indicate that the CNN model demonstrates a favorable equilibrium between precision, recall, and accuracy, emphasizing its potential usefulness in detecting ischemic stroke. The excellent precision signifies a reduction in false positives, while the commendable recall showcases the model's capacity to identify a significant proportion of true positives accurately. The attained precision highlights the model's competence in accurately classifying images, further emphasizing its potential as a valuable tool in diagnosing ischemic stroke in medical image analysis.

```
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 128, 128, 64, 1)   0

 conv3d (Conv3D)             (None, 126, 126, 62, 64   1792
                             )

 max_pooling3d (MaxPooling3  (None, 63, 63, 31, 64)    0
 D)

 batch normalization (Batch  (None, 63, 63, 31, 64)    256
 Normalization)

 conv3d_1 (Conv3D)           (None, 61, 61, 29, 64)    110656

 max_pooling3d_1 (MaxPoolin  (None, 30, 30, 14, 64)    0
 g3D)

 batch normalization 1 (Bat  (None, 30, 30, 14, 64)    256
 chNormalization)

 conv3d_2 (Conv3D)           (None, 28, 28, 12, 128)   221312

 max_pooling3d_2 (MaxPoolin  (None, 14, 14, 6, 128)    0
 g3D)

 batch normalization 2 (Bat  (None, 14, 14, 6, 128)    512
 chNormalization)

 conv3d 3 (Conv3D)           (None, 12, 12, 4, 256)    884992

 max_pooling3d_3 (MaxPoolin  (None, 6, 6, 2, 256)      0
 g3D)

 batch normalization 3 (Bat  (None, 6, 6, 2, 256)      1024
 chNormalization)

 global average pooling3d (  (None, 256)               0
 GlobalAveragePooling3D)

 dense (Dense)               (None, 512)               131584

 dropout (Dropout)           (None, 512)               0

 dense 1 (Dense)             (None, 1)                 513

=================================================================
Total params: 1352897 (5.16 MB)
Trainable params: 1351873 (5.16 MB)
Non-trainable params: 1024 (4.00 KB)
_________________________________________________________________
```

**Figure 7.** Proposed CNN model architecture

We also performed an ablation analysis to remove the preprocessing step with PBT. The purpose was to demonstrate the impact of PBT on the Model's performance. The average of ten experiments carried out were detailed in Table 2. The findings indicate that there was a significant improvement in the model's performance when PBT was used for image enhancement.

**Table 2.** Ischemic stroke CNN model result.

| Mode | Epoch | Precision (%) | Recall (%) | Accuracy (%) | AUC (%) |
|---|---|---|---|---|---|
| Without Augmentation | 150 | 0.94 | 0.81 | 0.91 | 0.98 |
| With PBT | 150 | 0.94 | 0.81 | 0.91 | 0.98 |
| Without PBT | 150 | 0.80 | 0.67 | 0.76 | 0.83 |

The Convolutional Neural Network (CNN) model results for detecting ischemic stroke, as shown in Table 3, exhibit encouraging performance with data augmentation under 150 epochs. The precision of 1.0 signifies that all instances classified as positive (ischemic stroke) were correctly identified as true positives. This indicates a significant degree of precision in detecting cases of ischemic stroke without producing incorrect positive results. The recall, measured at 0.91, indicates that the model accurately detected a significant part (91%) of the true positive instances, demonstrating its efficacy in correctly identifying cases of ischemic stroke. The model's high overall accuracy of 0.97 demonstrates its ability to anticipate outcomes, whether good or negative, accurately. These results highlight the strength and dependability of the CNN model, demonstrating its potential for accurately detecting ischemic strokes. The model's excellent precision and recall values and its commendable overall accuracy indicate its strong performance in recognizing cases of ischemic stroke. This makes it a significant tool in medical image analysis and diagnosis.

**Table 3.** Ischemic stroke CNN model result with augmentation.

| Epoch | Precision (%) | Recall (%) | Accuracy (%) | AUC (%) |
|---|---|---|---|---|
| 15 | 0.6522 | 0.6818 | 0.7414 | 0.8472 |
| 30 | 0.8571 | 0.8182 | 0.8793 | 0.9470 |
| 45 | 0.9091 | 0.9091 | 0.9310 | 0.9870 |
| 60 | 1.0000 | 0.8636 | 0.9483 | 0.9962 |
| 75 | 0.8363 | 0.8363 | 0.8966 | 0.9703 |
| 90 | 0.9524 | 0.9091 | 0.9483 | 0.9924 |
| 105 | 1.0000 | 0.9545 | 0.9828 | 1.0000 |
| 120 | 1.0000 | 0.9091 | 0.9655 | 1.0000 |
| 135 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 150 | 1.0000 | 0.9191 | 0.9789 | 0.9886 |

The results from Table 3 showed that: 1. As the number of epochs increases, precision generally increases, indicating the model makes fewer false positive predictions. 2. Recall fluctuates across epochs but tends to improve over time, meaning the model is better at correctly identifying positive instances. 3. Accuracy shows an increasing trend, signifying that the overall correct predictions by the model are improving. 4. AUC values are consistently high, indicating that the model has a strong ability to distinguish between different classes.

Overall, the table suggests that as the training progresses through epochs, the model's precision, recall, accuracy, and AUC performance generally improves, with some fluctuations at certain epochs.

The graphical presentation of results from two CNN models is shown in Figure 8, one with data augmentations (CNN-A) and the other without. The bar chart displays precision, recall, and accuracy performance metrics on the x-axis. CNN-A incorporates data augmentations during training, such as rotation and scaling, to enhance model generalization. Precision, recall, and accuracy are common evaluation measures plotted on the x-axis. Comparing CNN-A and CNN results on the chart offers insights into the impact of data augmentation on model performance, aiding interpretation of their strengths and weaknesses. Therefore, the results showed there was an improvement in the performance metrics of the model when the augmentation images were used to train the model despite the adversarial situation (noise added to the images) used for the model.

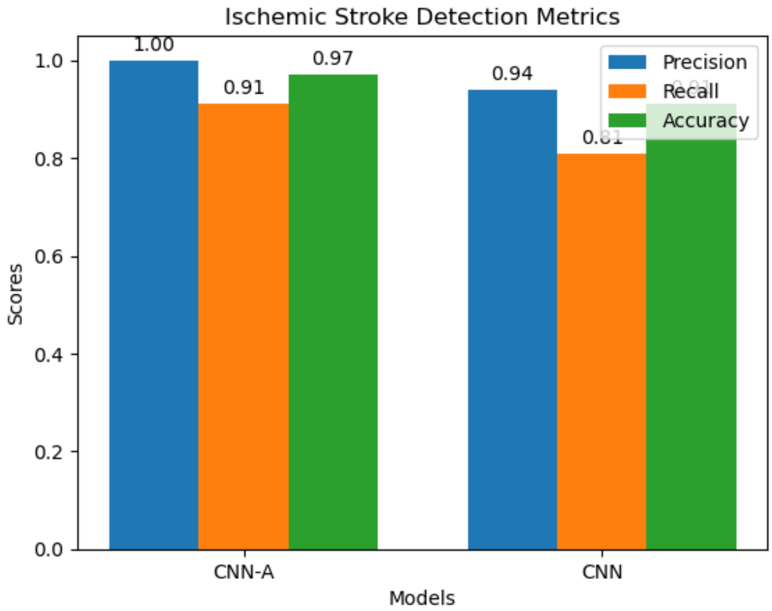


**Figure 8.** Ischemic CNN model graphical result with and without augmentation.

## 5. Comparison

Table 4 compares state-of-the-art algorithms comprehensively, contrasting the utilized algorithms with those employed by three other authors in various research studies. The table encompasses five distinct features as columns for the state-of-the-art comparison. These columns delineate the respective authors, the datasets utilized, the algorithms implemented, the optimal algorithm chosen by each author, and the corresponding best algorithm scores recorded in the final column. This comparative analysis offers a detailed insight into the performance and efficacy of the algorithms applied by different researchers, aiding in identifying superior approaches within the given context.

**Table 4.** State-of-the-Art comparison.

| Authors | Dataset Used | Algorithm Used | Best Algorithm | Accuracy |
|---|---|---|---|---|
| Alom et al., (2019)[33] | Digital Pathology Image | DCNN | DCNN | 96% |
| Badriyah et al., (2020)[25] | Ischemic Stroke & Stroke Haemorrhage | Eight machine learning algorithms | Random Forest | 95.97% |
| Tarkanyi et al., (2022)[29] | (LVO) strokes | four various machine learning methods | All proved to perform well | 73-77% |
| Bacchi et al., (2020)[34] | Ischemic Stroke | CNN & ANN | Both at the ratio | 74:71% |
| **Current Study** | Ischemic Stroke | CNN | CNN | 97% |

Table 4 represents the state-of-the-art comparison of the accuracy performance against the various models of other relevant studies used to classify and predict ischemic stroke etiology. The current study's performance accuracy was rated above that of other models. There is a significant improvement from the work of Bacchi et al.[34] which yielded 74% accuracy compared to the current study, which gave 97%. This results from the PBT used for image enhancement, the augmentation approach to generate more images to train the model and adversarial training integrated into CNN model training.

## 6. Conclusions

Deep learning models computer-aided techniques, though, will never replace doctors and radiological experts. But it helps automate image processing and analysis. Stroke medicine has evolved rapidly in the past 30 years, epitomized by mechanical thrombectomy's considerable difference in patients' lives. Also, computer-aided techniques for analyzing medical images have grown significantly recently, contributing to medical research and clinical applications. DL has shown continuous optimization in the segmentation process of stroke lesion regions from the brain stroke. This research aimed to observe the improvements and the growth of DL architectures adopting the CNN model with augmentation and adversarial training approach of stroke lesions so that the model mitigates against the attack or compromising of medical images. Over the past few years. Despite these advancements, there are still some limitations and, thus, more improvements are needed. This pattern of gradual enhancements in stroke lesion region augmentation can potentially become a scientific revolution if medical doctors and radiological experts also play a part in conceiving and building the framework for deep learning models. Though deep learning has yielded significant results in the medical domain, remarkable research prospects exist for exploiting the above-discussed methods and deep architectures to solve complex image augmentation problems. The results showed a significant improvement in classification accuracy from 91% to 97%. This study recommends using this model for doctors and radiological experts in their diagnosis, and further studies should be focused on modifying the CNN algorithm based on augmentation with more datasets. The future direction for this study is to employ an autoencoder approach for feature extraction and consider other CNN architecture.

**Author Contributions:** Conceptualization: Moshood A. Hambali and Paul A. Agwu; methodology, Paul A. Agwu; validation: Moshood A. Hambali.; formal analysis: Moshood A. Hambali.; investigation: Paul A. Agwu.; resources: Moshood A. Hambali; data curation: Paul A. Agwu.; writing—original draft preparation: Paul A. Agwu; writing—review and editing: Moshood A. Hambali; visu-alization: Moshood A. Hambali; supervision: Moshood A. Hambali; project administration: Paul A. Agwu.

**Funding:** This research received no external funding.

**Data Availability Statement:** We used publicly available data and it can be found at www.kaggle.com/afridirahman/brain-stroke-ct-image-dataset.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

[1] S. Zhang, S. Xu, L. Tan, H. Wang, and J. Meng, "Stroke lesion detection and analysis in MRI images based on deep learning," *J. Healthc. Eng.*, vol. 2021, pp. 1–9, 2021.
[2] W. L. Baker *et al.*, "Neurothrombectomy devices for the treatment of acute ischemic stroke: state of the evidence," *Ann. Intern. Med.*, vol. 154, no. 4, pp. 243–252, 2011.
[3] E. J. Benjamin *et al.*, "Heart Disease and Stroke Statistics—2017 Update: A Report From the American Heart Association," *Circulation*, vol. 135, no. 10, Mar. 2017, doi: 10.1161/CIR.0000000000000485.
[4] M. Lopez-Espejo, R. Poblete, and G. Bastias, "Social and health determinants related to adverse short-term outcomes after a first-ever stroke in adults younger than 65 years," *J. Stroke Cerebrovasc. Dis.*, vol. 32, no. 8, p. 107153, 2023.
[5] F. Arba, C. Rinaldi, D. Caimano, F. Vit, G. Busto, and E. Fainardi, "Blood–brain barrier disruption and hemorrhagic transformation in acute ischemic stroke: systematic review and meta-analysis," *Front. Neurol.*, vol. 11, p. 594613, 2021.
[6] J. Li *et al.*, "Tree-Based Risk Factor Identification and Stroke Level Prediction in Stroke Cohort Study," *Biomed Res. Int.*, vol. 2023, 2023.
[7] S. A. Sheth, L. Giancardo, M. Colasurdo, V. M. Srinivasan, A. Niktabe, and P. Kan, "Machine learning and acute stroke imaging," *J. Neurointerv. Surg.*, vol. 15, no. 2, pp. 195–199, 2023.
[8] U. R. Acharya *et al.*, "Automatic detection of ischemic stroke using higher order spectra features in brain MRI images," *Cogn. Syst. Res.*, vol. 58, pp. 134–142, 2019.
[9] L. Cui *et al.*, "Deep learning in ischemic stroke imaging analysis: A comprehensive review," *Biomed Res. Int.*, vol. 2022, 2022.
[10] F. Yousaf, S. Iqbal, N. Fatima, T. Kousar, and M. S. M. Rahim, "Multi-class disease detection using deep learning and human brain medical imaging," *Biomed. Signal Process. Control*, vol. 85, p. 104875, 2023.
[11] N. Takahashi, Y. Lee, D.-Y. Tsai, E. Matsuyama, T. Kinoshita, and K. Ishii, "An automated detection method for the MCA dot sign of acute stroke in unenhanced CT," *Radiol. Phys. Technol.*, vol. 7, pp. 79–88, 2014.
[12] N. D. Forkert, T. Verleger, B. Cheng, G. Thomalla, C. C. Hilgetag, and J. Fiehler, "Multiclass support vector machine-based lesion mapping predicts functional outcome in ischemic stroke patients," *PLoS One*, vol. 10, no. 6, p. e0129569, 2015.
[13] K. C. Ho, W. Speier, H. Zhang, F. Scalzo, S. El-Saden, and C. W. Arnold, "A Machine Learning Approach for Classifying Ischemic Stroke Onset Time From Imaging," *IEEE Trans. Med. Imaging*, vol. 38, no. 7, pp. 1666–1676, Jul. 2019, doi: 10.1109/TMI.2019.2901445.

[14] Y. Yu, D. Guo, M. Lou, D. Liebeskind, and F. Scalzo, "Prediction of hemorrhagic transformation severity in acute stroke from source perfusion MRI," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 2058–2065, 2017.
[15] A. Subudhi, S. Jena, and S. Sabut, "Delineation of the ischemic stroke lesion based on watershed and relative fuzzy connectedness in brain MRI," *Med. Biol. Eng. Comput.*, vol. 56, pp. 795–807, 2018.
[16] F. Scalzo *et al.*, "Multi-center prediction of hemorrhagic transformation in acute ischemic stroke using permeability imaging features," *Magn. Reson. Imaging*, vol. 31, no. 6, pp. 961–969, 2013.
[17] A. Nielsen, M. B. Hansen, A. Tietze, and K. Mouridsen, "Prediction of tissue outcome and assessment of treatment effect in acute ischemic stroke using deep learning," *Stroke*, vol. 49, no. 6, pp. 1394–1401, 2018.
[18] M. Monteiro *et al.*, "Using machine learning to improve the prediction of functional outcome in ischemic stroke patients," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 15, no. 6, pp. 1953–1959, 2018.
[19] A. Subudhi, U. R. Acharya, M. Dash, S. Jena, and S. Sabut, "Automated approach for detection of ischemic stroke using Delaunay Triangulation in brain MRI images," *Comput. Biol. Med.*, vol. 103, pp. 116–129, 2018.
[20] R. Karthik, U. Gupta, A. Jha, R. Rajalakshmi, and R. Menaka, "A deep supervised approach for ischemic lesion segmentation from multimodal MRI using Fully Convolutional Network," *Appl. Soft Comput.*, vol. 84, p. 105685, 2019.
[21] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, "Classification of stroke disease using machine learning algorithms," *Neural Comput. Appl.*, vol. 32, no. 3, pp. 817–828, 2020.
[22] Y. Yu *et al.*, "Use of deep learning to predict final ischemic stroke lesions from initial magnetic resonance imaging," *JAMA Netw. open*, vol. 3, no. 3, pp. e200772–e200772, 2020.
[23] A. Vupputuri, S. Ashwal, B. Tsao, and N. Ghosh, "Ischemic stroke segmentation in multi-sequence MRI by symmetry determined superpixel based hierarchical clustering," *Comput. Biol. Med.*, vol. 116, p. 103536, 2020.
[24] L. Liu, S. Chen, F. Zhang, F.-X. Wu, Y. Pan, and J. Wang, "Deep convolutional neural network for automatically segmenting acute ischemic stroke lesion in multi-modality MRI," *Neural Comput. Appl.*, vol. 32, pp. 6545–6558, 2020.
[25] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, "Machine learning algorithm for stroke disease classification," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2020, pp. 1–5.
[26] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis, and M. M. Khan, "Stroke disease detection and prediction using robust learning approaches," *J. Healthc. Eng.*, vol. 2021, 2021.
[27] T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, "Detection of stroke disease using machine learning algorithms," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019, pp. 1–6.
[28] G. Fang, Z. Huang, and Z. Wang, "Predicting ischemic stroke outcome using deep learning approaches," *Front. Genet.*, vol. 12, p. 827522, 2022.
[29] G. Tarkanyi, A. Tenyi, R. Hollos, P. J. Kalmar, and L. Szapary, "Optimization of large vessel occlusion detection in acute ischemic stroke using machine learning methods," *Life*, vol. 12, no. 2, p. 230, 2022.
[30] H. Kuang *et al.*, "Automated ASPECTS on noncontrast CT scans in patients with acute ischemic stroke using machine learning," *Am. J. Neuroradiol.*, vol. 40, no. 1, pp. 33–38, 2019.
[31] Q. Zheng, M. Yang, X. Tian, N. Jiang, and D. Wang, "A full stage data augmentation method in deep convolutional neural network for natural image classification," *Discret. Dyn. Nat. Soc.*, vol. 2020, 2020.
[32] F. S. Gomiasti, E. Kartikadarma, J. Gondohanindijo, and D. R. I. M. Setiadi, "Enhancing Lung Cancer Classification Effectiveness Through Hyperparameter-Tuned Support Vector Machine," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 396–406, 2024, doi: 10.62411/jcta.10106.
[33] M. Z. Alom *et al.*, "Advanced deep convolutional neural network approaches for digital pathology image analysis: A comprehensive evaluation with different use cases," *arXiv Prepr. arXiv1904.09075*, 2019.
[34] S. Bacchi, T. Zerner, L. Oakden-Rayner, T. Kleinig, S. Patel, and J. Jannes, "Deep learning in the prediction of ischaemic stroke thrombolysis functional outcomes: a pilot study," *Acad. Radiol.*, vol. 27, no. 2, pp. e19–e23, 2020.