

Enhancing Corn Leaf Disease Classification Through Data Augmentation and Deep Neural Networks

Irma Fitriani¹, Mustakim^{1,2,*}, Ma. Angelica M. Qin³, Laurice Tanap⁴

¹ Department of Information System, Faculty of Science and Technology, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

² Department of Computer Science, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

³ Department of Medical Laboratory Science, Dr. Carlos S. Lanting College, Manila, Philippines

⁴ Department of Nursing, Institute of Health Sciences and Nursing, Far Eastern University, Manila, Philippines

Email: ¹12250325541@students.uin-suska.ac.id, ^{2,*}mustakim@uin-suska.ac.id, ³msxynbae@gmail.com,

⁴2023059971@feu.edu.ph

*) Correspondence Author Email

ARTICLE INFORMATION

ARTICLE HISTORY:

Submitted : May 06, 2026

Revised : May 29, 2026

Accept : May 31, 2026

Publish : May 31, 2026

KEYWORD

Corn Leaf Disease;
Data Augmentation;
Deep Learning;
Deep Neural Network;
ResNet50;
DenseNet169;
Flask

CORRESPONDENCE AUTHOR

Email: mustakim@uin-suska.ac.id

ABSTRACT

Manual diagnosis of corn leaf diseases from digital images is time-consuming and prone to misinterpretation, creating the need for reliable automated classification. Although deep neural networks (DNNs) have shown promising performance, previous studies have mainly compared model architectures without comprehensively evaluating the combined effects of architecture, optimizer, learning rate, train-test split, and data augmentation on model generalization. This study addresses this gap by evaluating ResNet50 and DenseNet169 using five optimizers, five learning rates, two train-test split schemes (80:20 and 90:10), and data augmentation across 200 experimental scenarios. The dataset consists of 5,000 corn leaf images equally distributed into four classes and complemented with field images for external validation. ResNet50 with the RMSprop optimizer, a learning rate of 0.001, a 90:10 split, and data augmentation achieved the best performance, reaching 99.21% accuracy, 98.39% validation accuracy, and 0.0845 validation loss. This configuration performed better because ResNet50's residual connections improved feature learning, while RMSprop provided stable adaptive optimization during training. However, field testing produced substantially lower accuracy than laboratory experiments, indicating limited generalization due to variations in illumination, background, and image quality. These findings demonstrate that hyperparameter optimization and data augmentation improve laboratory performance, but enhancing robustness under real-world conditions remains a major challenge. The best model was implemented in a Flask-based web application for early corn leaf disease detection.

1. INTRODUCTION

Corn is one of the main agricultural commodities that plays a strategic role in supporting global food security [1]. In addition to being a food source, corn is also used as raw material for animal feed and the bioenergy industry, making its productivity an important indicator of economic sustainability for farmers, especially in agrarian countries [2], [3]. However, corn productivity is highly susceptible to leaf disease, which can cause yield losses of up to 35% globally [4]. In the United States, leaf diseases such as Northern Leaf Blight and Gray Leaf Spot are reported to cause economic losses of up to approximately US\$1.11 billion per year [5], [6]. In Indonesia, corn ranks as the second staple food after rice and is an important commodity for national food security [7]. The three main diseases that often attack corn plants in Indonesia are leaf blight, common rust, and leaf spot [8]. Leaf spots can reduce crop yields by 70–100% [9], Leaf rust can affect up to 81% of plants in humid conditions [10], while leaf blight remains a major threat in various regions [11]. These three diseases attack photosynthetic tissue and can potentially cause total crop loss if not treated properly [12].

The world population is expected to exceed 9 billion people, so corn production needs are projected to increase by 70% by 2050 [13], [14]. Corn leaf disease not only causes economic losses but also has a social impact on small farmers who depend on their harvest as their main source of income. Therefore, early identification and treatment of corn leaf diseases is very important to minimize losses [15], [16]. However, manual disease identification is often inaccurate because it depends on the farmer's experience, which risks misdiagnosis and delayed treatment [17]. Although leaf disease diagnosis can be performed through laboratory analysis, this method still has limitations in terms of time, sample capacity, and the need for experts. In conditions where disease spreads rapidly, delayed diagnosis can increase the risk of yield loss. Therefore, an early-diagnosis approach based on deep learning is needed that can classify leaf images quickly and efficiently as an initial stage of detection, without replacing the role of advanced laboratory diagnosis.

Deep learning approaches, particularly Deep Neural Networks (DNN), have proven effective in detecting plant diseases through digital images [18]. The ResNet50 architecture utilizes skip connections to maintain training stability, while DenseNet169 uses dense inter-layer connectivity to improve feature propagation and classification accuracy [19],



[20]. Various studies have applied deep learning approaches to leaf disease image classification. Research conducted by Mustakim et al. (2024) achieved the highest accuracy of 94.36% using the ResNet50 architecture with the Nadam optimizer, a learning rate of 0.0001, and a 90:10 data split scheme with data augmentation [21]. Didit Iswantoro and Dewi Handayani (2022) achieved 94% testing accuracy in the classification of two classes of corn leaf diseases using CNN [22]. Research by Aanis Ahmad et al. (2023) compared several DNN architectures and found that DenseNet169 achieved the best performance, with a generalization accuracy of 81.60% across laboratory and field conditions [12]. Research by Md. Manowarul Islam et al. (2023) reported that a DNN model with the Adam optimizer and a learning rate of 0.001 achieved 96.96% accuracy in classifying corn leaf diseases [23]. Research by Patayon and Crisostomo (2022) shows that the DenseNet169 architecture with the RMSProp optimizer achieves 98% accuracy and precision in the classification of peanut leaf diseases [24]. Research by Artika and Wikky Fawwaz Al Maki (2025) shows that the ResNet50 architecture with the RMSprop optimizer achieves 95% accuracy in classifying bitter melon leaf disease [25].

Despite the promising performance reported in previous studies, several important limitations remain. Most studies evaluated only a single DNN architecture or a limited set of hyperparameters, making it difficult to determine whether the reported performance was primarily influenced by the network architecture or by the optimization strategy. Furthermore, the interaction among optimizer selection, learning rate, train-test split, and data augmentation has not been systematically investigated. As a result, the optimal configuration for improving model stability and generalization across different experimental settings remains unclear. Since the optimization algorithm controls parameter updates during training and the learning rate determines convergence behavior, inappropriate hyperparameter selection may lead to unstable optimization, overfitting, or poor generalization, particularly when models are applied to unseen field images. Therefore, a comprehensive evaluation of these interacting factors is required to identify robust configurations that consistently improve classification performance.

To address these limitations, this study systematically evaluates two DNN architectures, ResNet50 and DenseNet169, using five optimizers, five learning rates, two train-test split schemes, and data augmentation across 200 experimental scenarios. Unlike previous studies that focused primarily on comparing architectures or reporting the highest accuracy, this research analyzes the interaction between architectural design and hyperparameter optimization to identify configurations that produce stable learning and better generalization under both laboratory and field conditions. The findings provide practical guidelines for selecting effective DNN configurations for corn leaf disease classification and contribute to the development of more reliable image-based plant disease detection systems..

2. RESEARCH METHODOLOGY

2.1 Research Stages

This study applied an experimental approach with a structured methodological flow as shown in Figure 1. Each stage was designed to support systematic data processing and model development, thereby improving the performance of corn leaf disease image classification.

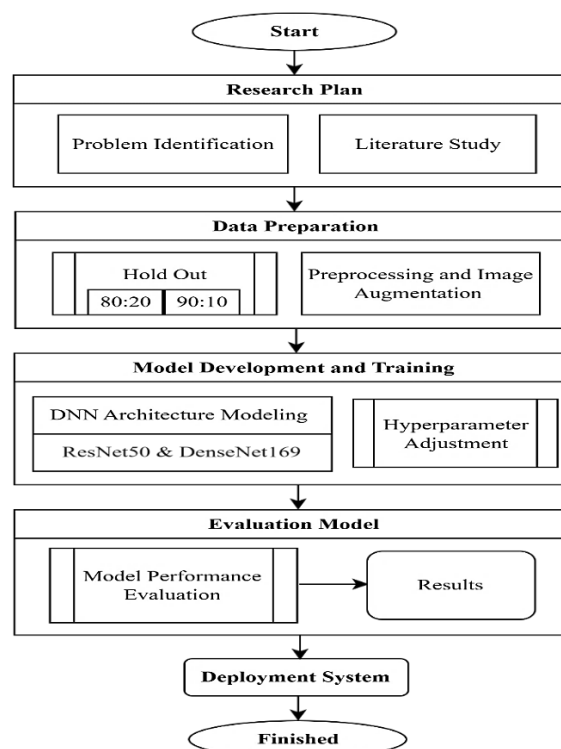


Figure 1. Research Methodology

2.2 Research Plan

This research focuses on identifying problems in accurately detecting corn leaf diseases through visual observation, which can lead to reduced crop yields. This study aims to develop a DNN-based corn leaf disease image classification model using the ResNet50 and DenseNet169 architectures, achieving improved performance through data augmentation and hyperparameter optimization. The scope of the research is limited to the classification of four main classes, namely leaf spots, leaf blight, leaf rust, and healthy leaves, using validated leaf images. This planning stage is supported by a literature study that discusses the basic concepts of DNN, the characteristics of the ResNet50 and DenseNet169 architectures, and the role of data augmentation and optimization techniques in improving the accuracy of classification models.

To provide a systematic evaluation, five optimizers (SGD, Adam, RMSprop, Adamax, and Nadam) and five learning rates (0.1, 0.01, 0.001, 0.0001, and 0.00001) were selected because they represent different optimization strategies and convergence behaviors widely adopted in deep learning. Their evaluation is particularly relevant, as ResNet50 relies on residual learning to stabilize gradient propagation, whereas DenseNet169 employs dense feature reuse, which may respond differently to optimization dynamics. Rather than relying on a single configuration, multiple train-test split ratios were investigated to assess performance consistency under varying data availability. Although hold-out validation was adopted to reflect practical deployment scenarios and ensure independent testing, all experiments were conducted under identical conditions, enabling fair and reproducible comparisons across architectures and hyperparameter configurations.

2.3 Data Preparation

Data preparation includes collecting corn leaf disease image datasets from public sources and validated field data. To maintain class balance, 1,250 images per class were used, for a total of 5,000 images. The data was cleaned of unusable images, divided using the hold-out method (80:20 and 90:10), and processed through preprocessing and augmentation of 224×224-pixel images to increase data variation and model generalization.

2.4 Model Development and Training

Model development and training were carried out by building two DNN architectures, namely ResNet50 and DenseNet169, using pre-trained ImageNet models as the basis for feature extraction. In both architectures, classification layers were added using Global Average Pooling, dropout, and dense layers with L2 regularization and ReLU activation, culminating in a softmax output layer for four classes. Model training was conducted with and without data augmentation, and various combinations of optimizers and learning rates were tested to obtain model configurations with the best performance and generalization. The parameters used are shown in Table 1.

Table 1. Hyperparameters Used

Training Parameters	Configuration
Model Architecture	ResNet-50 dan DenseNet169
Optimizer	Adam, RMSprop, SGD, Nadam, Adamax
Learning Rate	0.1, 0.01, 0.001, 0.0001, 0.00001
Batch Size	32
Epoch	35
Data Distribution	Hold-Out
Method	
Data Ratio	80:20 dan 90:10

2.5 Corn Leaf Disease

Corn leaf disease is a disorder caused by pathogens such as fungi, bacteria, or viruses, characterized by visual changes in the leaves and directly impacting growth and yield [26]. This study focuses on three major diseases, namely Leaf Spot, Leaf Blight, and Leaf Rust. Leaf Spot causes necrosis in leaf tissue, thereby inhibiting photosynthesis. Leaf Blight causes systemic infection that can lead to drying and death of leaves, while Leaf Rust causes pustules on the leaf surface, which reduce photosynthetic activity and slow plant growth [27]. The visualization of the corn leaf disease classes used in this study is shown in Figure 2.

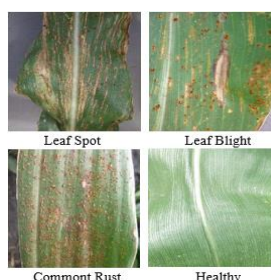


Figure 2. Visualization of Corn Leaf Disease Classes

2.6 Deep Neural Network (DNN)

DNN is a layered neural network inspired by biological neural systems and designed to extract hierarchical features from complex data [28]. Each layer of a DNN learns patterns incrementally, starting from simple features and progressing to more abstract representations. This capability makes DNNs highly effective for image classification tasks because they support end-to-end learning without requiring manual feature extraction [18]. DNN is capable of capturing nonlinear relationships and visual variations, such as differences in size, color, and lighting in plant images [29]. The structure of the DNN algorithm is shown in Figure 3.

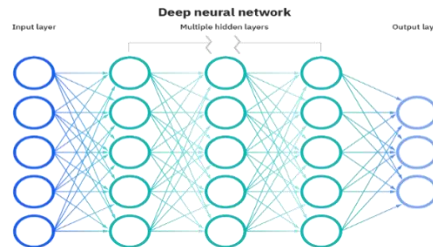


Figure 3. Structure Deep Neural Network (DNN)

2.7 ResNet-50 Architecture

ResNet50 is a convolutional neural network architecture in the Residual Network (ResNet) family, designed to address the issues of vanishing gradients and degradation in very deep networks [12]. This architecture consists of 50 layers, including 48 convolution layers, one max pooling layer, and one average pooling layer, with the main feature being skip connections or residual connections that allow gradient flow to pass through several layers without degradation [21], [30]. Figure 4 shows the structure of the ResNet50 architecture with skip connections [14], [31].

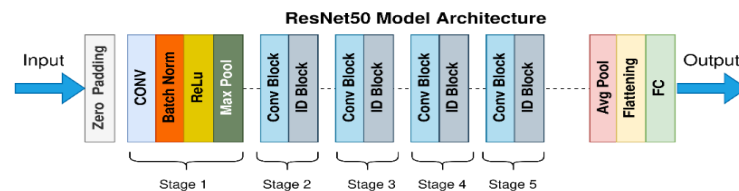


Figure 4. ResNet-50 Architecture Layers

2.8 DenseNet169 Architecture

DenseNet169 is a DNN architecture with a dense connectivity concept, where each layer is directly connected to all previous layers to improve information flow and gradients [19]. This structure enables more efficient feature utilization, reduces vanishing gradient issues, and suppresses the number of parameters through concatenation-based feature merging [32]. With the support of ImageNet pre-trained weights, DenseNet169 is effectively used in image classification, including plant diseases, because it is capable of recognizing complex visual patterns accurately and efficiently [14], [32]. A visualization of the DenseNet169 architecture is shown in Figure 5. [32].

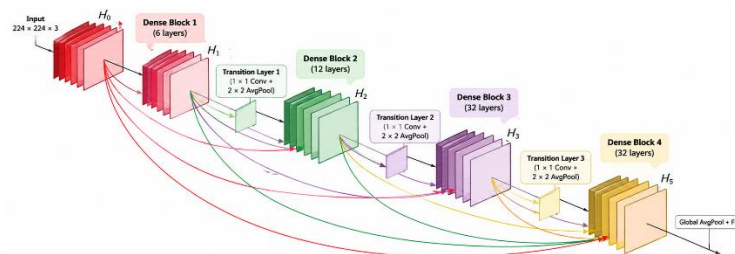


Figure 5. DenseNet169 Architecture

2.9 Optimizers

An optimizer is a method for updating neural network weights based on gradients of the loss function to improve accuracy and reduce model error [21]. Choosing the right optimizer directly affects convergence speed and model performance and is related to determining hyperparameters to ensure neural networks achieve optimal performance in classification and prediction [33]. Here are some of the optimizers used in this study.

2.9.1 Adam Optimizer

In principle, Adam combines two previously popular optimization approaches, namely the Adaptive Gradient Algorithm (AdaGrad), which is effective for handling sparse gradients, and Root Mean Square Propagation (RMSProp), which is

good at overcoming non-stationary problems in data [26], [34]. Adam is widely used because it achieves high accuracy with stable convergence, making it effective across a wide range of deep learning models. Adam updates parameters with two decay factors, β_1 for momentum and β_2 for variance, so that convergence is more balanced and oscillations are reduced during training [35].

$$m_0 = 0, \quad v_0 = 0 \quad (1)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3)$$

2.9.2 Nadam Optimizer

Nadam (Nesterov-accelerated Adaptive Moment Estimation) is an extension of Adam that combines the advantages of Adaptive Moment Estimation with Nesterov Accelerated Gradient (NAG) [36]. Nadam uses Nesterov's gradient correction, which makes optimization more adaptive, accelerates convergence, and reduces the risk of getting stuck in local minima, resulting in more stable performance on large and complex datasets [37], [38].

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (4)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (6)$$

2.9.3 SGD Optimizer

Stochastic Gradient Descent (SGD) is an optimization algorithm commonly used in neural network training because it is simple, efficient, and capable of handling large amounts of data [39], [40]. Unlike batch gradient descent, which computes gradients using the entire dataset, SGD updates the parameters based on a small subset of data (mini-batches), making it faster and more computationally efficient [41], [42]. The main advantage of SGD is its stochastic nature, which makes the convergence process faster and helps escape local minima [43].

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t; x^{(i)}, y^{(1)}) \quad (7)$$

2.9.4 RMSprop Optimizer

Root Mean Square Propagation (RMSProp) is an adaptive learning rate-based optimization algorithm developed to overcome the weaknesses of Adagrad, particularly the problem of learning rates decreasing too quickly [44]. RMSProp uses the exponential moving average of the gradient squared to adjust the learning rate for each parameter, resulting in more stable updates on non-stationary data, such as neural network training [45]. RMSProp excels at accelerating convergence, maintaining update stability, and effectively handling gradient scale variations [46].

2.9.5 Adamax Optimizer

Adamax is a variant of the Adam algorithm developed by Kingma and Ba as part of an adaptive optimization method for deep neural networks [47]. This optimizer is designed using the infinity norm to stabilize weight updates in the training process of models that have large gradients or data with varying value distributions [48]. The basic principle of Adamax addresses Adam's weakness in handling extreme gradients by using the ratio of the first moment to the exponential maximum of the second moment, thereby making parameter updates smoother and more stable during convergence.

2.10 Augmentation Data

Augmentation Data is a pre-processing technique to expand and enrich training datasets by transforming existing data, without the need to manually add new data [49]. Data augmentation methods, such as rotation, flipping, cropping, scaling, adding noise, and utilizing Generative Adversarial Networks (GAN), are used to enrich data variation so that models are better able to recognize patterns [50]. Data augmentation is important to address the imbalance in the amount of data between classes, which can cause bias in model training [51].

2.11 Confusion Matrix

A confusion matrix is a widely used evaluation metric in machine learning to assess the performance of a classification model [52]. In the Confusion Matrix, there are four main components, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP indicates the number of positive data correctly predicted, while TN indicates the number of negative data correctly classified. Conversely, FP indicates the amount of negative data incorrectly predicted as positive, and FN describes positive data incorrectly predicted as negative. Mathematically, the Confusion Matrix can be expressed in equation 8 [53].

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (8)$$

From the confusion matrix, key evaluation metrics can be calculated, namely Accuracy, Precision, Recall, and F1-Score. Accuracy measures the overall correct prediction rate, Precision assesses the accuracy of positive class predictions, Recall

indicates the ability to detect positive data, and F1-Score represents the balance between Precision and Recall.

2.12 Deployment System

The deployment stage involved implementing the best DNN model into a web-based corn leaf disease image classification system using Flask. The system receives leaf images, performs preprocessing, and generates predictions of four disease classes as output to support image-based early detection.

3. RESULT AND DISCUSSION

3.1 Data Collection

This study uses two types of datasets: public datasets for training and field datasets for testing to evaluate the model's generalization ability. The public dataset consists of 5,000 corn leaf images divided evenly into four classes, namely Leaf Spot, Northern Corn Leaf Blight, Common Rust, and Healthy, with 1,250 images in each class. This balanced distribution aims to prevent class bias during model training. Meanwhile, the field dataset is used as test data and consists of 40 images collected in the field, evenly distributed across 10 classes. This combination of public and field datasets enables analysis of model performance both on controlled data and in real-world conditions. Details of the initial data distribution for each class are shown in Table 2.

Table 2. Data Distribution for Each Class

No.	Class	Amount of Data
1	Leaf Spot	1250
2	Leaf Blight	1250
3	Common Rust	1250
4	Healthy	1250
	Total	5000

3.2 Preprocessing Data

The data cleaning stage was carried out by selecting 5,000 balanced corn leaf images from four classes, namely Leaf Spot, Leaf Blight, Leaf Rust, and Healthy Leaves, and removing damaged, blurred, or duplicate images to maintain the quality of the dataset. All images were in JPG format with an initial resolution of 256×256 pixels, then processed through a preprocessing stage with size adjustments to 224×224 pixels and pixel value normalization. To increase data variety and model generalization, augmentation was applied to the training data using geometric transformations, including rotations of up to $\pm 60^\circ$, 25% shear, 30% zoom, 10% horizontal and vertical shifts, and a horizontal flip. Next, the data was divided using the hold-out method (80:20 and 90:10) into training, validation, and test subsets to support objective model evaluation, with details of the data distribution shown in Tables 3 and 4.

Table 3. Distribution of Data Holdout 80:10:10

Class	Subset			Amount of Data
	Train (80%)	Validation (10%)	Test (10%)	
Leaf Spot	1000	125	125	1250
Leaf Blight	1000	125	125	1250
Common Rust	1000	125	125	1250
Healthy	1000	125	125	1250
Total 4000		500	500	5000

Tables 3 and 4 show the dataset split using the hold-out method at ratios of 80:20 and 90:10, respectively. Both tables display the distribution of data across the training, validation, and test sets used to objectively evaluate model performance.

Table 4. Distribution of Data Holdout 90:5:5

Class	Subset			Amount of Data
	Train (90%)	Validation (5%)	Test (5%)	
Leaf Spot	1125	62	63	1250
Leaf Blight	1125	62	63	1250
Common Rust	1125	62	63	1250
Healthy	1125	62	63	1250
Total 4500		248	252	5000

3.3 Model Implementation and Performance Evaluation

DNN experiments were conducted to determine the best configuration for classifying corn leaf disease images. 224×224-pixel input images, with a batch size of 32 and 35 epochs, were used with two transfer learning architectures, ResNet50 and DenseNet169, equipped with Global Average Pooling, dropout, a dense layer of 256 neurons, and a softmax output for four disease classes. Each architecture was tested with five optimizers and five learning rates, resulting in 200 experiments. The evaluation used accuracy, F1-score, confusion matrix, and ROC-AUC to assess the model's ability to recognize image patterns. The results were presented in four data-partitioning and augmentation scenarios, which served as the basis for selecting the optimal configuration for implementing a web-based classification system. From each scenario, the five models with the highest accuracy were selected, yielding a total of 20 models as the basis for classification performance analysis. Table 5 below presents a summary of the precision, recall, and F1-score of each of the best models.

Table 5. Recap of the Best Models

No	Split Data	Type Data	Precision	Recall	F-1 Score	Accuracy
1	80:20:00	Non Augmentation	97,65%	97,60%	97,60%	97,60%
2	80:20:00	Non Augmentation	97,84%	97,80%	97,80%	97,80%
3	80:20:00	Non Augmentation	98,99%	97,78%	99,00%	98,60%
4	80:20:00	Non Augmentation	97,29%	97,20%	97,18%	97,20%
5	80:20:00	Non Augmentation	97,85%	97,80%	97,79%	97,80%
6	80:20:00	Augmentation	97,41%	97,40%	97,40%	97,40%
7	80:20:00	Augmentation	98,81%	98,80%	97,80%	98,80%
8	80:20:00	Augmentation	98,60%	98,60%	98,60%	98,60%
9	80:20:00	Augmentation	98,19%	98,20%	98,19%	98,20%
10	80:20:00	Augmentation	98,41%	98,40%	98,40%	98,40%
11	90:10:00	Non Augmentation	98,04%	98,02%	98,01%	98,02%
12	90:10:00	Non Augmentation	98,02%	98,02%	98,02%	98,04%
13	90:10:00	Non Augmentation	98,07%	98,02%	98,01%	98,02%
14	90:10:00	Non Augmentation	98,82%	96,81%	98,81%	98,81%
15	90:10:00	Non Augmentation	96,44%	98,41%	98,42%	98,41%
16	90:10:00	Augmentation	97,02%	98,02%	98,04%	98,02%
17	90:10:00	Augmentation	97,32%	97,22%	97,21%	97,22%
18	90:10:00	Augmentation	98,21%	99,21%	99,21%	99,21%
19	90:10:00	Augmentation	97,21%	97,22%	97,21%	97,42%
20	90:10:00	Augmentation	98,01%	98,02%	98,01%	98,02%

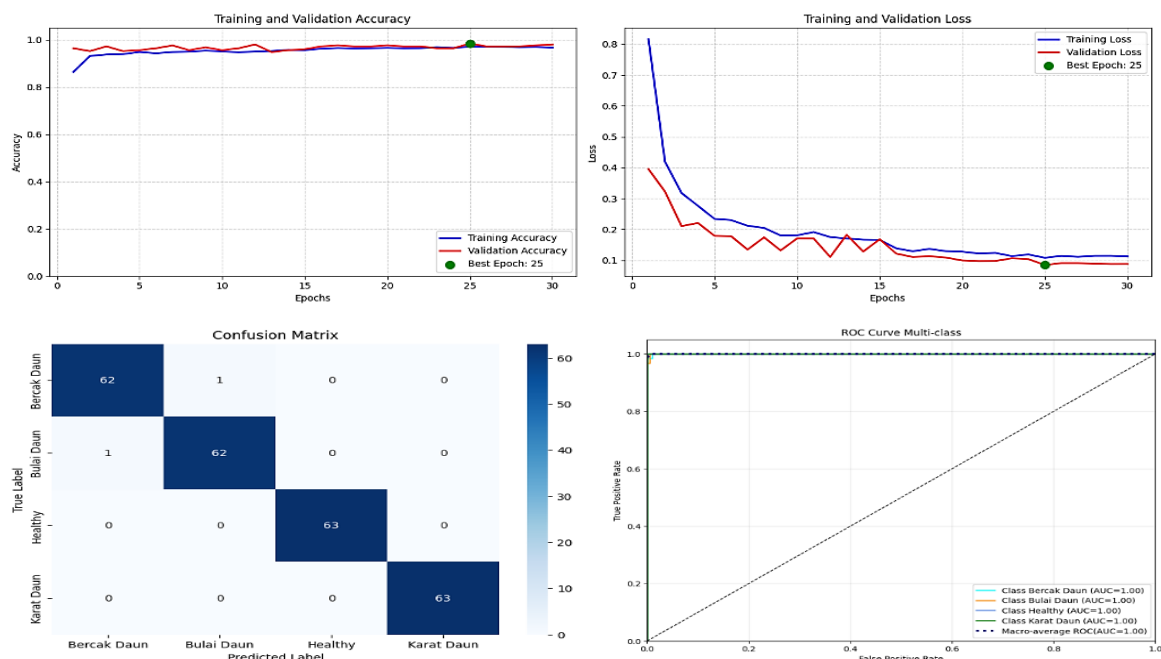


Figure 6. Loss and Accuracy Model Graph, Confusion Matrix, ROC Curve experiment number 18 Hold Out 90:10 with Augmentation Data

Based on Table 5, which shows the evaluation results for the best models in each scenario, it can be concluded that the DenseNet169 architecture shows more stable and superior performance in the Non Augmentation data scenario, both in the 80:20 and 90:10 data splits. Meanwhile, ResNet50 achieves better performance with data augmentation, demonstrating its ability to leverage data variation to improve model generalization.

The best model evaluation results in experiment 18, with the ResNet50 architecture, RMSprop optimizer, and a learning rate of 0.001, in a 90:10 data split scenario with data augmentation, showed the best and most stable performance. The training and validation accuracy and loss graphs, the confusion matrix, and the ROC curve in Figure 6 are presented as a representation of the model's evaluation results in that experiment.

Figure 6 shows that the loss and accuracy graphs for the data augmentation application yield a stable training process, with a validation accuracy of 0.9839 and a validation loss of 0.0845, indicating improved generalization on the ResNet50 architecture. The confusion matrix shows that data augmentation and the ResNet50 architecture produce consistent classification with minimal errors, while the ROC curve shows optimal class separation. In general, DenseNet169 outperforms non-augmentation, while ResNet50 performs better with data augmentation. The comparison graph of the average accuracy of the architecture and optimizer from all experiments conducted is shown in Figures 7 and 8.

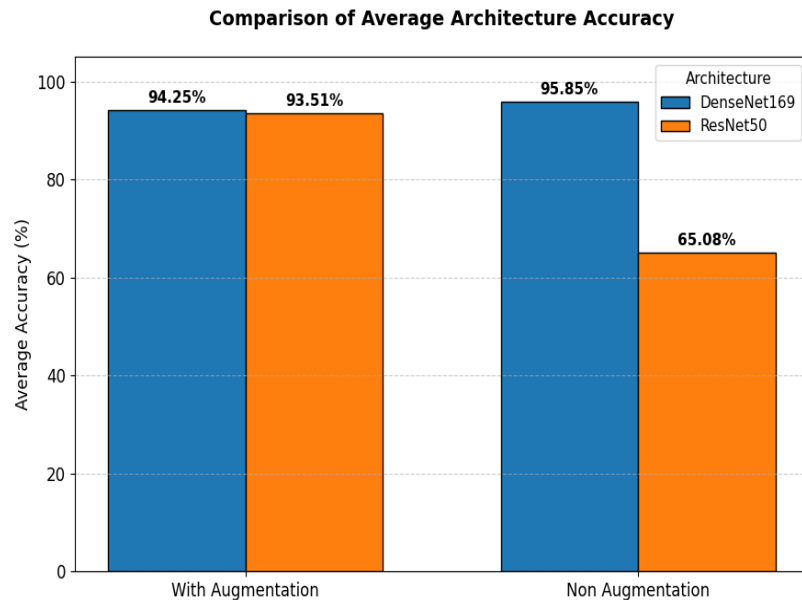


Figure 7. Comparison of Average Architecture Accuracy

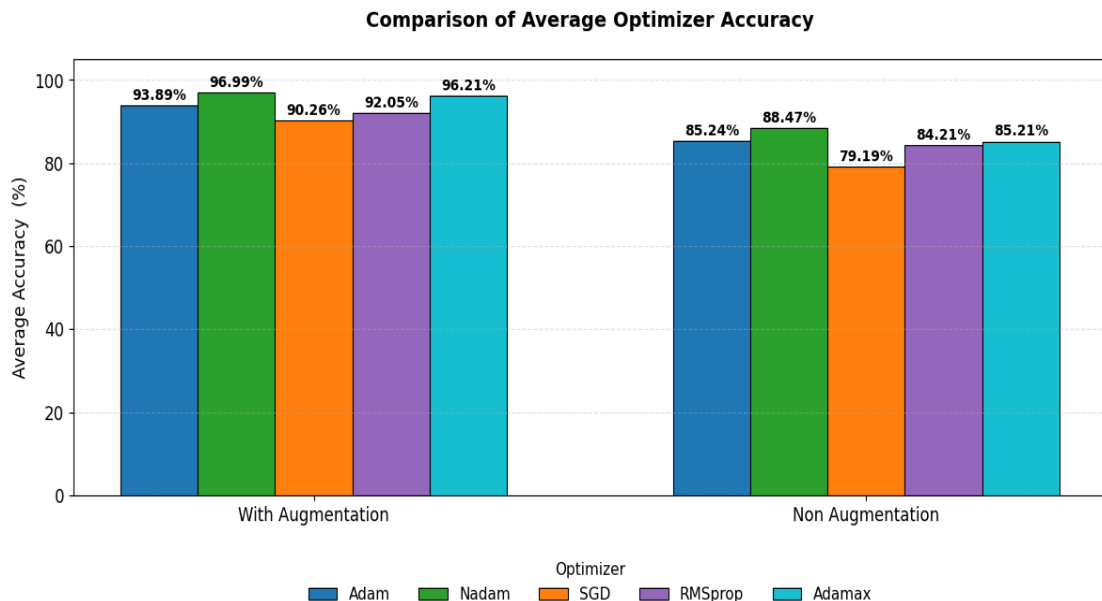


Figure 8. Comparison of Average Accuracy Optimizer

Figure 7 and Figure 8 compare architecture and optimizer performance under conditions with and without data augmentation. DenseNet169 appears superior and more stable than ResNet50 under unaugmented conditions, achieving an accuracy of around 95.85%, while ResNet50 declines to around 65.08%. After augmentation is applied, ResNet50's performance improves, and the gap with DenseNet169 becomes insignificant, indicating that ResNet50 is greatly aided by augmented data, while DenseNet169 is more robust on limited data. In terms of optimizers, Nadam and Adamax achieved the best performance in both conditions, with accuracies of approximately 96.99% and 96.21% on Augmentation

data, respectively, and remained superior on Non-Augmentation data. Conversely, SGD performed worst, making it suboptimal for the corn leaf disease classification scenario in this study.

3.4 Model Testing

Testing was conducted to evaluate the performance of the corn leaf disease classification system using field-collected image data as test data. Each image was labeled according to its class, then tested using the system to assess the consistency between the model's predictions and the original class. The evaluation was based on the level of prediction accuracy and the accuracy value produced. The results of testing the corn leaf disease classification system using 10 test images obtained from the field are presented in Table 6.

Table 6. System Testing Results Using Primary Data

No	Dataset Testing	Original Class	DNN Prediction Class	DNN Accuracy Class	Count
1	Leaf Image 1	Leaf Spot	Leaf Spot	99,91%	1
2	Leaf Image 2	Leaf Spot	Leaf Spot	99,86%	1
3	Leaf Image 3	Leaf Spot	Leaf Spot	92,29%	1
4	Leaf Image 4	Leaf Blight	Leaf Blight	96,01%	1
5	Leaf Image 5	Leaf Blight	Leaf Blight	99,20%	1
6	Leaf Image 6	Common Rust	Leaf Spot	80,74%	0
7	Leaf Image 7	Common Rust	Leaf Blight	78,51%	0
8	Leaf Image 8	Healthy	Healthy	99,73%	1
9	Leaf Image 9	Healthy	Healthy	99,60%	1
10	Leaf Image 10	Common Rust	Leaf Spot	94,06%	0
		True Value			7
		Model Test Accuracy		93,99%	70%

Table 6 shows the test results using 10 field image data. The corn leaf disease classification system correctly predicted 7 data points and incorrectly predicted 3, resulting in a prediction accuracy of 70%. The system's average prediction accuracy was 93.99%, indicating that the model deployed at the stage was able to classify corn leaf images quite well. The model tended to predict images as Leaf Spot class due to the unbalanced distribution of training data and the more dominant visual patterns of Leaf Spot, which were easier for the model to learn. In addition, differences in conditions between the training and test data during deployment, such as lighting, shooting angle, and image quality, led to a domain shift that reduced the model's generalization ability. This resulted in some images from other classes in the test table being classified as Leaf Spot.

The field evaluation demonstrates that the model achieved only 70% classification accuracy despite obtaining more than 99% accuracy during laboratory testing. This performance gap indicates that high accuracy on controlled datasets does not necessarily translate into robust real-world performance. Misclassification mainly occurred in the Common Rust class, which was frequently predicted as Leaf Spot or Leaf Blight. This finding suggests that variations in illumination, background complexity, viewing angle, and symptom similarity significantly affect feature extraction and model discrimination. Although data augmentation and hyperparameter optimization improved convergence and laboratory performance, they were insufficient to fully overcome domain shift between laboratory and field environments. These results are consistent with previous studies reporting reduced generalization under real-world conditions and highlight the need for more diverse field datasets and domain adaptation strategies to improve deployment reliability.

3.5 Discussion

The results of this study show that the best configuration was achieved with the ResNet50 architecture, the RMSprop optimizer, and a learning rate of 0.001, in a 90:10 data split with augmentation, yielding the highest accuracy of 99.21%. This achievement demonstrates that combining data Augmentation, a larger training data ratio, and the selection of the right optimizer can significantly improve the model's generalization ability. These findings confirm that ResNet50 is highly responsive to data augmentation, resulting in consistent performance improvements across various testing scenarios.

Compared to the study by Mustakim et al. (2024), which used the ResNet50 architecture and reported the best accuracy of 94.36% in a 90:10 scenario with Augmentation and the Nadam optimizer, the results of this study show a significant improvement in performance [21]. This difference indicates that selecting the RMSprop optimizer with an appropriate learning rate, along with exploring a wider range of training configurations, improves model stability and accuracy. Thus, this study reinforces previous findings and shows that ResNet50 still has the potential for higher performance when configured optimally.

Compared to the research by Didit Iswantoro and Handayani (2022), which used a conventional CNN for two classes of corn leaf diseases and obtained a testing accuracy of 94%, this research shows superiority in terms of model complexity and classification scope [22]. This study involved four classes of corn leaf diseases, used a transfer-learning-based deep learning architecture, and evaluated various combinations of optimizers and data partitioning scenarios. This

shows that a deeper, more systematic DNN approach can improve accuracy and generalization in plant disease image classification.

The results of this study are also relevant when compared to the study by Ahmad et al. (2023), which evaluated several deep learning architectures, including DenseNet169 and ResNet50, under laboratory and field conditions, with the best generalization accuracy of 81.60% [12]. In this study, DenseNet169 proved superior and more stable in non-augmentation conditions, while ResNet50 achieved the best performance with augmentation. These differences in results show that data characteristics, preprocessing strategies, and the application of Augmentation have a significant impact on the effectiveness of an architecture, even more so than the choice of architecture alone.

In line with the research by Islam et al. (2023), which reported an accuracy of 96.96% using DNN with data augmentation and the Adam optimizer, and the research by Patayon and Crisostomo (2022), which showed the superiority of DenseNet169 in plant disease classification with an accuracy of up to 98%, this research succeeded in surpassing these achievements with an accuracy of over 99% [23], [24]. Furthermore, compared to Artika and Maki's (2025) study, which showed the highest accuracy of 95% on ResNet50, the results of this study confirm that thorough exploration of optimizers, learning rates, and data partitioning schemes plays an important role in optimizing model performance [25]. Overall, this study shows that a comprehensive evaluation of two DNN architectures, optimizer variations, learning rates, and data augmentation is important for improving classification performance. The results of this study not only reinforce previous findings but also demonstrate that combining ResNet50 with data augmentation and appropriate training settings can achieve high, stable accuracy. In addition, applying the model to a web-based system using Flask demonstrates that the results of this study can be practically used to support the identification of corn leaf diseases in field conditions.

Overall, the findings demonstrate that the study's objective to identify an effective DNN configuration for corn leaf disease classification through systematic hyperparameter optimization has been successfully achieved. By evaluating two representative architectures, five optimizers, five learning rates, two train-test split schemes, and data augmentation across 200 experimental scenarios, this study shows that classification performance is determined not by network architecture alone but by the interaction between architectural characteristics and training configuration. The superior performance of ResNet50 with RMSprop indicates that residual learning, combined with adaptive optimization, leads to more stable convergence and more discriminative feature representations under controlled conditions. Nevertheless, the decline in accuracy during field testing highlights that optimizing training configurations alone is insufficient to eliminate domain shift, emphasizing the need for more diverse field datasets and domain adaptation techniques to improve model robustness and practical deployment.

4. CONCLUSION

This study systematically evaluated the effects of DNN architecture, optimizer, learning rate, train-test split ratio, and data augmentation on corn leaf disease classification through 200 experimental scenarios. The results demonstrate that classification performance is influenced not only by network architecture but also by the interaction between architectural characteristics and training configuration. Among the evaluated models, ResNet50 combined with the RMSprop optimizer, a learning rate of 0.001, a 90:10 train-test split, and data augmentation achieved the best laboratory performance, with an accuracy of 99.21%, validation accuracy of 98.39%, and validation loss of 0.0845. These findings confirm that appropriate hyperparameter optimization and data augmentation can substantially improve training stability and classification accuracy under controlled conditions. However, the field evaluation revealed that the model achieved only 70% accuracy, indicating that the excellent laboratory performance did not fully generalize to real-world environments. This limitation suggests that variations in illumination, background complexity, image quality, and symptom appearance remain significant challenges for practical deployment. Therefore, although the proposed model provides a promising baseline for automated corn leaf disease classification, further research is required to improve its robustness under field conditions. Future work should investigate domain adaptation techniques, more diverse field datasets, advanced augmentation strategies, and cross-validation-based evaluation protocols to reduce the performance gap between laboratory and real-world applications.

REFERENCES

- [1] S. Mishra, R. Sachan, and D. Rajpal, "Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition," *Procedia Comput. Sci.*, vol. 167, pp. 2003–2010, 2020, doi: 10.1016/j.procs.2020.03.236.
- [2] M. M. Malik et al., "A novel deep CNN model with entropy coded sine cosine for corn disease classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 7, p. 102126, 2024, doi: 10.1016/j.jksuci.2024.102126.
- [3] B. Mazumder, M. S. I. Khan, and K. M. Mohi Uddin, "Biorthogonal wavelet based entropy feature extraction for identification of maize leaf diseases," *J. Agric. Food Res.*, vol. 14, no. August, p. 100756, 2023, doi: 10.1016/j.jafr.2023.100756.
- [4] A. H. Ali, A. Youssef, M. Abdelal, and M. A. Raja, "An ensemble of deep learning architectures for accurate plant disease classification," *Ecol. Inform.*, vol. 81, no. September 2023, p. 102618, 2024, doi: 10.1016/j.ecoinf.2024.102618.
- [5] D. Jiang, S. Chen, J. Useya, L. Cao, and T. Lu, "Crop Mapping Using the Historical Crop Data Layer and Deep Neural Networks: A Case Study in Jilin Province, China," *Sensors*, vol. 22, no. 15, 2022, doi: 10.3390/s22155853.
- [6] D. S. Mueller et al., "Corn yield loss estimates due to diseases in the United States and Ontario, Canada, from 2016 to 2019," *Plant Heal. Prog.*, vol. 21, no. 4, pp. 238–247, 2020, doi: 10.1094/PHP-05-20-0038-RS.
- [7] I. P. Putra, R. Rusbandi, and D. Alamsyah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural



- Network,” *J. Algoritme.*, vol. 2, no. 2, pp. 102–112, 2022, doi: 10.35957/algoritme.v2i2.2360.
- [8] S. Hobi, “Penyakit Tanaman Jagung yang Menghambat Pertumbuhan,” Kumparan.com. Accessed: May 06, 2025. [Online]. Available: <https://kumparan.com/seputar-hobi/penyakit-tanaman-jagung-yang-menghambat-pertumbuhan-22nQ6Ae8pAO/full>
 - [9] N. Chitriningrum *et al.*, “Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8,” *J. Eng. Technol. Sci.*, vol. 56, no. 1, pp. 61–70, 2024, doi: 10.5614/j.eng.technol.sci.2024.56.1.5.
 - [10] Sopialena, Suyadi, and S. A. Noor, “Ecosystem Monitoring on Leaves of Leaf Rust Disease of Maize (*Zea mays* L.),” *Caraka Tani J. Sustain. Agric.*, vol. 37, no. 1, pp. 89–99, 2022, doi: 10.20961/carakatani.v37i1.34920.
 - [11] Ulfah Nur Oktaviana, Ricky Hendrawan, Alfian Dwi Khoirul Annas, and Galih Wasis Wicaksono, “Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih Resnet101,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1216–1222, 2021, doi: 10.29207/resti.v5i6.3607.
 - [12] A. Ahmad, A. El Gamal, and D. Saraswat, “Toward Generalization of Deep Learning-Based Plant Disease Identification Under Controlled and Field Conditions,” *IEEE Access*, vol. 11, no. December 2022, pp. 9042–9057, 2023, doi: 10.1109/ACCESS.2023.3240100.
 - [13] N. Sachin Patil and E. Kannan, “An efficient corn leaf disease prediction using Adaptive Color Edge Segmentation with Resnext101 model,” *J. Saudi Soc. Agric. Sci.*, no. July, 2024, doi: 10.1016/j.jssas.2024.09.002.
 - [14] A. Chakrabarty, S. T. Ahmed, M. F. U. Islam, S. M. Aziz, and S. S. Maidin, “An interpretable fusion model integrating lightweight CNN and transformer architectures for rice leaf disease identification,” *Ecol. Inform.*, vol. 82, no. July, p. 102718, 2024, doi: 10.1016/j.ecoinf.2024.102718.
 - [15] K. Manjula, S. Spoorthi, R. Yashaswini, and D. Sharma, “Plant Disease Detection Using Deep Learning,” *Lect. Notes Electr. Eng.*, vol. 783, no. 1, pp. 1389–1396, 2022, doi: 10.1007/978-981-16-3690-5_133.
 - [16] N. Nigar, H. Muhammad Faisal, M. Umer, O. Oki, and J. Manappattukunnel Lukose, “Improving Plant Disease Classification with Deep-Learning-Based Prediction Model Using Explainable Artificial Intelligence,” *IEEE Access*, vol. 12, no. July, pp. 100005–100014, 2024, doi: 10.1109/ACCESS.2024.3428553.
 - [17] M. M. Rahaman, S. Umer, M. Azharuddin, and A. Hassan, “Image-based blight disease detection in crops using ensemble deep neural networks for agricultural applications,” *J. Nat. Pestic. Res.*, vol. 12, no. December 2024, p. 100130, 2025, doi: 10.1016/j.napere.2025.100130.
 - [18] P. Sajitha, A. Diana Andrushia, N. Anand, M. Z. Naser, and E. Lubloy, “A deep learning approach to detect diseases in pomegranate fruits via hybrid optimal attention capsule network,” *Ecol. Inform.*, vol. 84, no. February, p. 102859, 2024, doi: 10.1016/j.ecoinf.2024.102859.
 - [19] L. A. Abraham, G. Palanisamy, and G. Veerapu, “Transparent brain tumor detection using DenseNet169 and LIME,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–21, 2025, doi: 10.1038/s41598-025-13233-7.
 - [20] M. A. R. Nishad, M. A. Mitu, and N. Jahan, “Predicting and Classifying Potato Leaf Disease using K-means Segmentation Techniques and Deep Learning Networks,” *Procedia Comput. Sci.*, vol. 212, no. C, pp. 220–229, 2022, doi: 10.1016/j.procs.2022.11.006.
 - [21] M. Mustakim, A. R. Pratama, I. Ahmad, T. Arifianto, K. Sussolaikah, and S. Sepriano, “Image Classification of Corn Leaf Diseases Using CNN Architecture ResNet-50 and Data Augmentation,” *2024 Int. Conf. Decis. Aid Sci. Appl. DASA 2024*, pp. 1–6, 2024, doi: 10.1109/DASA63652.2024.10836354.
 - [22] D. Iswantoro and D. Handayani UN, “Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN),” *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 2, p. 900, 2022, doi: 10.33087/jiubj.v22i2.2065.
 - [23] M. M. Islam *et al.*, “DeepCrop: Deep learning-based crop disease prediction with web application,” *J. Agric. Food Res.*, vol. 14, no. March, p. 100764, 2023, doi: 10.1016/j.jafr.2023.100764.
 - [24] A. Ahmad *et al.*, “Peanut leaf spot disease identification using pre-trained deep convolutional neural network,” *J. Agric. Food Res.*, vol. 7, no. 1, pp. 3005–3012, 2022, doi: 10.1016/j.jafr.2023.100764.
 - [25] Artika and W. F. Al Maki, “Classification of bitter melon leaf disease using deep learning architecture: ResNet50,” *Int. J. Adv. Intell. Informatics*, vol. 11, no. 2, pp. 324–335, 2025, doi: 10.26555/ijain.v11i2.1925.
 - [26] D. A. N. Gookyi *et al.*, “TinyML for smart agriculture: Comparative analysis of TinyML platforms and practical deployment for maize leaf disease identification,” *Smart Agric. Technol.*, vol. 8, no. June, p. 100490, 2024, doi: 10.1016/j.atech.2024.100490.
 - [27] H. Yusuf, H. Fitriyah, and S. R. Akbar, “Deteksi Penyakit pada Daun Jagung berdasarkan Hue menggunakan Metode K-Nearest Neighbor berbasis Raspberry Pi,” vol. 6, no. 12, pp. 5608–5614, 2022.
 - [28] Z. Sun, L. Di, H. Fang, and A. Burgess, “Deep learning classification for crop types in north dakota,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 2200–2213, 2020, doi: 10.1109/JSTARS.2020.2990104.
 - [29] M. I. Nuhash, M. Sohag, S. S. Ramit, and R. T. H. Tusher, “A deep ensemble learning and explainable AI framework for accurate bottle gourd disease diagnosis and deployment,” *Smart Agric. Technol.*, vol. 12, no. October, p. 101541, 2025, doi: 10.1016/j.atech.2025.101541.
 - [30] C. V. Kwatra, H. Kaur, M. Mangla, A. Singh, S. N. Tambe, and S. Potharaju, “Early detection of gynecological malignancies using ensemble deep learning models: ResNet50 and inception V3,” *Informatics Med. Unlocked*, vol. 53, no. January, p. 101620, 2025, doi: 10.1016/j.imu.2025.101620.
 - [31] P. Bachhal *et al.*, “Maize leaf disease recognition using PRF-SVM integration: a breakthrough technique,” *Sci. Rep.*, vol. 14, no. 1, pp. 1–20, 2024, doi: 10.1038/s41598-024-60506-8.
 - [32] D. Science, T. Nadu, C. Science, T. Nadu, D. Science, and T. Nadu, “Pneumonia The Application Of Densenet169 And Transfer Learning To The Detection Of Pneumonia In Chest X-Ray Images,” vol. 13, no. 6, pp. 1–9, 2025, doi: <https://dx.doi.org/10.21037/jmai-24-356>.
 - [33] K. Joshi *et al.*, “Precision diagnosis of tomato diseases for sustainable agriculture through deep learning approach with hybrid data augmentation,” *Curr. Plant Biol.*, vol. 41, no. December 2024, p. 100437, 2025, doi: 10.1016/j.cpb.2025.100437.
 - [34] A. K. Shahade, K. H. Walse, V. M. Thakare, and M. Atique, “Multi-lingual opinion mining for social media discourses: an approach using deep learning based hybrid fine-tuned smith algorithm with adam optimizer,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 2, p. 100182, 2023, doi: 10.1016/j.jjime.2023.100182.
 - [35] K. Ahn, Z. Zhang, Y. Kook, and Y. Dai, “Understanding Adam optimizer via Online Learning of Updates: Adam is FTRL in Disguise,” *Proc. Mach. Learn. Res.*, vol. 235, pp. 619–640, 2024, doi: <https://doi.org/10.48550/arXiv.2402.01567>.



- [36] S. Bamohabbat Chafjiri, P. Legg, M. A. Tsompanas, and J. Hong, “Leveraging activation and optimisation layers as dynamic strategies in the multi-task fuzzing scheme,” *Comput. Stand. Interfaces*, vol. 94, no. March, p. 104011, 2025, doi: 10.1016/j.csi.2025.104011.
- [37] Q. Zhang *et al.*, “Boosting Adversarial Attacks with Nadam Optimizer,” *Electron.*, vol. 12, no. 6, pp. 1–16, 2023, doi: 10.3390/electronics12061464.
- [38] A. Saber, T. Emara, S. Elbedwehy, and E. Hassan, “Open A novel approach for breast cancer detection using a Nesterov accelerated adam optimizer with an attention mechanism,” pp. 1–14, 2025, doi: 10.1038/s41598-025-12070-y.
- [39] R. Sweke *et al.*, “Stochastic gradient descent for hybrid quantum-classical optimization,” *Quantum*, vol. 4, 2020, doi: 10.22331/q-2020-08-31-314.
- [40] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2017, doi: <https://doi.org/10.48550/arXiv.1609.04747>.
- [41] H. Chen, A. Kuzina, B. Esmaeili, and J. M. Tomczak, “Variational Stochastic Gradient Descent for Deep Neural Networks,” *Trans. Mach. Learn. Res.*, vol. 2025-April, pp. 1–22, 2025, doi: Variational Stochastic Gradient Descent for Deep Neural Networks.
- [42] C. Fjellström and K. Nyström, “Deep learning, stochastic gradient descent and diffusion maps,” *J. Comput. Math. Data Sci.*, vol. 4, no. June, p. 100054, 2022, doi: 10.1016/j.jcmds.2022.100054.
- [43] Elbert, M. Wulandari, and J. Fat, “Optimizer comparison in convolutional neural,” vol. 7, no. 1, pp. 15–24, 2025, doi: 10.21512/emacsjournal.v6.
- [44] R. Elshamy, O. Abu-Elnasr, M. Elhoseny, and S. Elmougy, “Improving the efficiency of RMSProp optimizer by utilizing Nesterov in deep learning,” *Sci. Rep.*, vol. 13, no. 1, pp. 1–16, 2023, doi: 10.1038/s41598-023-35663-x.
- [45] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, “On Empirical Comparisons of Optimizers for Deep Learning,” no. 1, 2020, [Online]. Available: <http://arxiv.org/abs/1910.05446>
- [46] A. El Hajoui, O. Y. Alaoui, O. El Kharki, M. Wahbi, H. Boulaassal, and M. Maatouk, “A Student Performance Prediction Using RNNs Models with variety of optimization Techniques in Deep Learning,” *Int. J. Inf. Educ. Technol.*, vol. 15, no. 6, pp. 1289–1301, 2025, doi: 10.18178/ijiet.2025.15.6.2331.
- [47] M. Uppal *et al.*, “Enhancing accuracy in brain stroke detection: Multi-layer perceptron with Adadelta, RMSProp and AdaMax optimizers,” *Front. Bioeng. Biotechnol.*, vol. 11, no. September, pp. 1–15, 2023, doi: 10.3389/fbioe.2023.1257591.
- [48] M. Nutakki and S. Mandava, “Optimizing home energy management: Robust and efficient solutions powered by attention networks,” *Heliyon*, vol. 10, no. 4, p. e26397, 2024, doi: 10.1016/j.heliyon.2024.e26397.
- [49] Y. Qin *et al.*, “Nearest Neighboring Self-Supervised Learning for Hyperspectral Image Classification,” *Remote Sens.*, vol. 15, no. 6, pp. 1–27, 2023, doi: 10.3390/rs15061713.
- [50] K. Taji, A. Sohail, T. Shahzad, B. S. Khan, M. A. Khan, and K. Ouahada, “An Ensemble Hybrid Framework: A Comparative Analysis of Metaheuristic Algorithms for Ensemble Hybrid CNN Features for Plants Disease Classification,” *IEEE Access*, vol. 12, no. May, pp. 61886–61906, 2024, doi: 10.1109/ACCESS.2024.3389648.
- [51] M. Tasfe, A. Nivrito, F. Al MacHot, M. Ullah, and H. Ullah, “Deep Learning Based Models for Paddy Disease Identification and Classification: A Systematic Survey,” *IEEE Access*, vol. 12, no. July, pp. 100862–100891, 2024, doi: 10.1109/ACCESS.2024.3419708.
- [52] K. Ahadian *et al.*, “Maize disease classification using transfer learning and convolutional neural network with weighted loss,” *Heliyon*, vol. 10, no. 21, p. e39569, 2024, doi: 10.1016/j.heliyon.2024.e39569.
- [53] M. M. Mijwil and M. Aljanabi, “A Comparative Analysis of Machine Learning Algorithms for Classification of Diabetes Utilizing Confusion Matrix Analysis,” *Baghdad Sci. J.*, vol. 21, no. 5, pp. 1712–1728, 2024, doi: 10.21123/BSJ.2023.9010.