Institute of Research and Publication Indonesia (IRPI)
Public Research Journal of Engineering, Data Technology and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/predatecs
Vol. 3 Iss. 2 January 2026, pp: 113-125
ISSN(P): 3024-921X | ISSN(E): 3024-8043

Classification of Corn Leaf Disease Images Using Convolutional Neural Network Algorithm

Irma Fitriani[1*], Rahma Devi[2], Ariandra Fokker Chaya Sajjana[3], Muhammad Irfan[4]

[1,2]Department of Information Systems, Faculty of Science and Technology,
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia
[3]Department of International Relations, Faculty of Political Science,
Social Sciences University of Ankara, Turkey
[4]Department of Islamic Sharia Studies, Faculty of Sharia and Law, Al Azhar University, Egypt

E-Mail: [1]12250325541@students.uin-suska.ac.id, [2]12250323758@students.uin-suska.ac.id,
[3]ariandrafokkerchayasajjana@student.asbu.edu.tr, [4]irfanadelard07@gmail.com

Received Jun 12th 2025; Revised Jan 18th 2026; Accepted Jan 25th 2026; Available Online Jan 31th 2026
Corresponding Author: Irma Fitriani

Abstract

Corn leaf diseases can reduce crop yields and cause financial losses, thus requiring accurate and objective classification methods. This study aims to classify four corn leaf conditions, namely Blight, Common Rust, Gray Leaf Spot, and healthy leaves, using a Convolutional Neural Network (CNN) approach based on image processing. A systematic comparative evaluation was conducted on three CNN architectures, namely MobileNetV2, ResNet50V2, and DenseNet201, by examining the effect of architecture-optimizer pairs using Adam and RMSprop to determine the optimal model configuration. The results showed that the proposed approach was effective at classifying corn leaf diseases, with the highest accuracy of 93% achieved by combining DenseNet201 and the Adam optimizer. This study contributes by providing a structured comparative analysis of the performance of CNN architectures and optimizers, serving as a reference for the development of more accurate and efficient early-detection systems for plant diseases.

Keywords: Convolutional Neural Network, Corn Leaf Disease, Deep Learning, Plant Disease Detection

1. INTRODUCTION

Agriculture is one of the strategic sectors that support global food security. Corn (Zea mays) is one of the primary commodities that significantly contribute to the economies and food security of many countries. However, the yield of maize crops is extremely vulnerable to leaf diseases such as leaf blight, common rust, and gray leaf spot. These diseases not only reduce yields but also cause significant economic losses. In the United States, total losses due to maize diseases are estimated to be around 40 billion dollars annually [1], [2]. With climate change and increasing global food demand, early detection of maize leaf diseases is critical to supporting agricultural productivity and resilience [3], [4].

Corn productivity is highly susceptible to leaf diseases that can reduce photosynthetic capacity and cause significant crop yield losses [5]. In addition, the similarity of visual characteristics between diseases in the early stages of infection often leads to misdiagnosis when relying solely on manual observation, especially given the limited number of experts and diagnostic facilities in the field. Therefore, an accurate approach to classifying corn leaf diseases is needed to support early detection [6].

As artificial intelligence and image processing technologies have advanced, deep learning techniques have emerged as a key tool for visual pattern identification, including the categorization of photos of diseased plant leaves. The Convolutional Neural Network (CNN), which can extract features straight from raw data without the need for intricate manual processing, is one of the most successful methods. CNN have demonstrated superiority in a range of image classification tasks, including identifying diseases in rice and maize leaves [7]. The CNN approach was chosen because it is capable of automatically extracting spatial and textural features from images of corn leaf diseases that have complex visual patterns, and has been empirically proven to be effective and efficient through various architectures such as MobileNetV2, ResNet, and DenseNet in plant leaf disease classification [5], [8]. Despite having less data, training can be done more effectively and accurately by using transfer learning techniques with pre-trained architectures like ResNet, DenseNet, and MobileNetV2. To preserve objectivity in the test data, hold-out validation techniques are typically used for model evaluation [9], [10].

Various studies have shown the effectiveness of CNN in corn leaf disease classification. Imran Khan et al. (2024) utilized ResNet50 and showed good sensitivity despite the class similarities, achieving validation accuracy of 87.51%, precision of 90.33%, and recall of 99.80% [3]. Research by Putra et al. (2022) used CNN ResNet-50 for corn leaf disease image classification, showing that the Adam optimizer produced the highest testing accuracy of 98.4% [11]. Research by Al-Gaashani et al. (2025) applied a MobileNetV2 CNN with an RMSprop optimizer and data augmentation techniques for leaf disease image classification, achieving a testing accuracy of 92.50% [12]. Research by Gumelar et al. (2025) used a CNN based on the MobileNetV2 architecture for corn leaf disease image classification and achieved a testing accuracy of 96.20% on five leaf condition classes [5]. Research by Mengesha and Mengistie (2025) used a DenseNet201 CNN and Adam optimizer, resulting in a testing accuracy of 99.17% [8]. Research by Sharma et al. (2025) applied CNN based on ResNet50 and MobileNetV2 architectures in an ensemble scheme with Adam optimizer for tomato leaf disease image classification, achieving a testing accuracy of 99.23% [13].

Although previous studies have shown that CNN are capable of producing good performance in corn leaf disease classification, most of these studies still test a single model configuration or optimizer separately. Comparisons of performance between different model configurations and optimizers within a consistent testing framework are still limited. This condition indicates the need for research that systematically evaluates combinations of models and optimization strategies to obtain more comprehensive results.

The performance of three CNN architectures MobileNetV2, ResNet50V2, and DenseNet201 along with two optimizers Adam and RMSprop is investigated in this study. Accuracy, precision, recall, and F1-score were used for evaluation, along with an 80:20 hold-out validation split. In contrast to earlier research, this strategy seeks to determine the optimal model and optimizer combination for the categorization of maize leaf diseases while broadening the range of tests. The results are expected to support the development of a more accurate and applicable CNN-based plant disease classification system in modern agriculture [1], [14].

2. MATERIAL AND METHOD

This study applied an experimental approach with methodological stages that were systematically designed to support the data processing, model development, and performance evaluation of corn leaf disease classification. In general, the research methodology flow is presented in Figure 1.

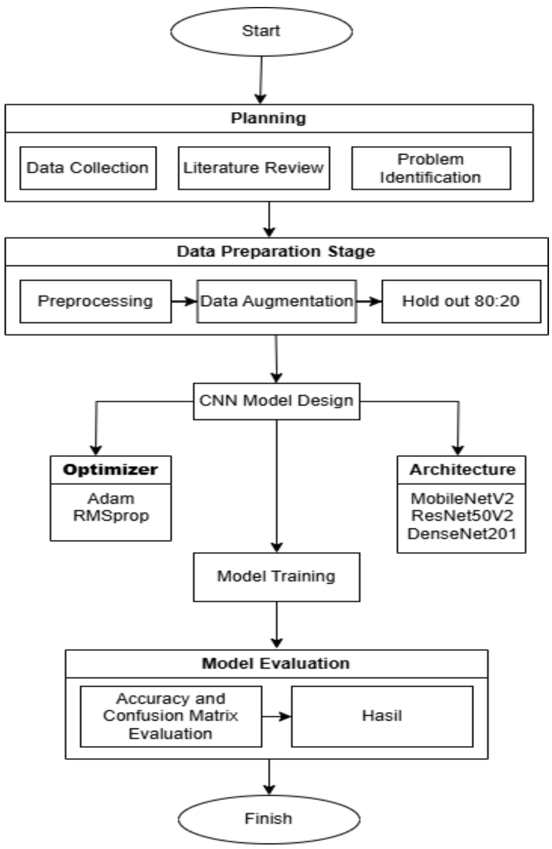

Figure 1. Research Methodology

2.1. Data Collection and Data Preparation

The planning includes collecting image data on corn leaf diseases from Kaggle, reviewing relevant literature, and identifying research issues in corn leaf disease classification using a CNN approach. At this stage, image preprocessing and data splitting are performed using an 80:20 hold-out split. 80% of the data is used for training and validation, and 20% for testing. Data augmentation is applied to the training data to increase data diversity and reduce overfitting.

2.2. Training and Evaluation Model

The model was trained on training and validation data with uniform parameters 20 epochs, a categorical cross-entropy loss function, and a batch size of 32 for each architecture. Next, an evaluation was performed using test data using confusion matrix analysis and classification reports to compare model performance.

2.3. Corn Leaf Disease

Maize leaf diseases are disorders of leaf tissue due to infection with pathogens such as fungi and bacteria, which thrive in conditions of moisture, high temperature, and minimal air circulation [15], [16]. The disease is generally caused by pathogenic infections such as fungi and bacteria, which thrive in humid environmental conditions, high temperatures, and poor air circulation [5], [17]. There are four classifications of maize leaf diseases: Blight (grayish-brown oblong spots, caused by Exserohilum turcicum), Common Rust (brownish-red pustules, by Puccinia sorghi), Gray Leaf Spot (narrow dark gray spots parallel to the leaf bone, by Cercospora zeae-maydis), and Corn Health (healthy green leaves without symptoms).

2.4. Data Mining

The systematic process of discovering patterns, relationships, and hidden knowledge from large and complex data sets is referred to as Data Mining [18],[19]. Data mining is the process of discovering patterns or useful information from big data using mathematical, statistical, and artificial intelligence techniques to support data-driven decision-making [20]. Structurally, data mining consists of several main functions, namely classification, regression, clustering, association, and summarization [21].

2.5. Deep Learning

Layered artificial neural networks are used in the machine learning subfield of "deep learning" to automatically extract intricate features from large amounts of data [22],[23]. It is excellent at detecting plant diseases by processing complex images and performing multi-class classification automatically [10],[24]. This method does not require manual feature engineering, mimicking the way the human brain recognizes patterns and learns from raw data. In plant disease classification, Khan et al. (2024) found that deep learning improved the generalization and effectiveness of leaf image processing [3]. This technology can identify subtle disease patterns that are difficult to detect manually or through conventional machine-learning approaches [25]. Activation function (ReLU) as formula 1.

$$f(x) = \max(0, x) \tag{1}$$

The ReLU activation function is f(x). function for an input x, and max(0,x) is the maximum value between two values, namely 0 and x. That is, if x is greater than 0, then the result is x; but if x is less than or equal to 0, then the result is 0.

2.6. Convolutional Neural Network (CNN)

One type of deep learning system is the Convolutional Neural Network (CNN) method, which is designed to process grid-like data such as images [26]. CNN does not require human feature design to automatically extract features from photos [10],[27]. A convolutional layer, ReLU, a pooling layer, a fully connected layer, and softmax for classification make up the architecture [28]. CNN simulates how neurons in the visual cortex of the brain react to a little region known as the receptive field [29]. The feature extraction process is performed using a small kernel (e.g., 3×3) through a convolution operation formulated 2

$$S(i, j) = (X * K)(i, j) = \sum_m \sum_n X(i + m, j + n). K(m, n) \tag{2}$$

S(i, j) is the result of the convolution operation at the i-th row and jth column positions of the output. X input matrix, K: convolution kernel or filter. (i, j) pixel coordinates at the output of the convolution result. X(i+m,j+n): the input image's pixel value at the filter-affected location. K(m,n): the kernel's element value at location (m,n).

2.7. Data Augmentation

Data augmentation is an important pre-processing technique in image classification with CNNs, such as corn leaf disease detection, which increases the diversity of training data without adding new data to help the model generalize better and reduce overfitting [30],[31]. Augmentation is done with image transformations such as Rotation Mirroring (flipping) Cropping Contrast, luminance, and color saturation adjustments [32], Mathematically, augmentation is modeled as a non-linear transformation of the image: x^'=T (X) where X is the alteration of the original image, T, and x' is the augmented image. An example of a rotation of θ degrees on a 2D image is formulated as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos 0 - \sin 0 \\ \sin 0 \quad \cos 0 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

Cos θ sin θ: the trigonometric component that determines the rotation of the point in the 2D plane. x, y: the coordinates of the point in the original image before augmentation. x', y: point coordinates after the image undergoes a rotation transformation.

2.8. MobileNetV2, ResNet50V2 and DenseNet201 Architecture

MobileNetV2 is an advanced version of MobileNet designed for high-efficiency image classification. With bottleneck and linear bottleneck techniques, this architecture improves inference speed and minimizes the number of parameters [14]. MobileNetV2 uses Convolutions with depth-wise separability to achieve high accuracy with a small model size, making it effective in image recognition tasks such as plant disease identification on IoT devices [33].

ResNet50V2 is a ResNet design change meant to solve the network's deteriorating accuracy [34],[35]. ResNet50V2 has 50 layers and applies residual learning to overcome vanishing gradients. Improvements through pre-activation (Batch Normalization and ReLU before convolution) increase training stability and accuracy, making it effective for complex image classification such as plant leaf disease detection [36].

To improve the efficiency of feature extraction, the CNN-based deep learning architecture DenseNet201 connects each layer to the previous layer in a single block [35]. DenseNet201, pre-trained on ImageNet, was modified by applying 2D Global Average Pooling, Dense layers, Dropout, and a Softmax output layer while removing the default fully connected layer (top=False). This architecture is efficient, resistant to vanishing gradients, and suitable for leaf image classification with high visual similarity [9].

2.9. Adam and RMSProp Optimizer

One well-liked optimization technique is Adam (Adaptive Moment Estimation) for CNNs which adjusts the learning rate of each parameter adaptively based on the estimated mean (first moment) and variance (second moment) of the gradient and combines the advantages of momentum [34]. Adam Optimizer's main advantage lies in its ability to adaptively update each parameter, making it very stable and efficient in the face of varying and sparse gradients [2].

RMSProp is an optimization algorithm that overcomes the drawbacks of SGD on non-stationary data, such as disease patterns or lighting variations, by adjusting the learning rate based on the most recent gradient mean square [23]. RMSProp maintains training stability and prevents loss oscillations by calculating the exponential average of gradient square and then adjusting the learning rate adaptively for each parameter [3].

2.10. Confusion Matrix

A popular evaluation technique for evaluating the effectiveness of classification algorithms is a confusion matrix, particularly in multi-class scenarios like the categorization of [37]. A Confusion Matrix helps evaluate model performance in detail by showing TP, FP, FN, and TN values, making it easier to identify errors and improve accuracy in the future [38]. This matrix displays the number of accurate and inaccurate classifications for each category by comparing the model's anticipated outcomes with the actual labels [39]. The basic structure of a Confusion Matrix for two (binary) classes is (see Table 1).

Table 1. Confusion Matrix

	Predicstion of Positive	Prediction of Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

3. RESULTS AND DISCUSSION

This section presents the results and analysis of deep learning-based corn leaf disease classification. Starting from data collection, preprocessing, and data sharing, to training CNN models with various architectures. Each stage is explained to demonstrate the method's effectiveness in automatically detecting and classifying diseases.

3.1. Data Collection

The corn leaf disease dataset comes from Kaggle with a total of 4,188 images labeled with four disease classes: Blight, Common Rust, Corn Gray Spot, and Corn Health [6]. However, the test only used half of them, which is 2,094 images. Data distribution per class can be seen in Table 2.

Table 2. Classification of Corn Leaf Diseases

No.	Class	Amount of Data	Data Used
1.	Blight	1.146 images	573 images
2.	Common Rust	1.306 images	653 images
3.	Corn Gray Spot	574 images	287 images
4.	Corn Gray Spot	1.162 images	581 images
	Total	4.188 images	2.094 images

Fig.2 shows the 4 classes of corn leaf diseases that will be used for modeling.

Figure 2. Corn Leaf Disease Class

3.2. Preprocessing Data

Data preprocessing is done through three main stages, namely resizing, normalization, and augmentation. All corn leaf images were resized to 224x224 pixels to make them uniform and suitable for the CNN model input. After that, each pixel is normalized on a scale of 0 to 1 using a division of RGB values by 255, to speed up the training process and stabilize model learning. The augmentation techniques used include random rotation of up to 30 degrees, zoom of 15%, horizontal and vertical shift of 20%, shear of 15%, and horizontal and vertical reversal of the image. The data augmentation configuration in this study appears in the Table 3.

Table 3. Data Augmentation Splitting

Architecture	Optimizer	Epoch	Batch Size	Image Size	Range	Flip	Zoom Range	Learning Rate
DenseNet201	Adam	20	32	224×224	45°	Horizontal	0.2	0.0001
DenseNet201	Adam	20	32	224×224	45°	Horizontal	0.2	0.0001
MobileNetV2	Adam	20	32	224×224	45°	Horizontal	0.2	0.0001
MobileNetV2	RMSprop	20	32	224×224	45°	Horizontal	0.2	0.0001
ResNet50V2	RMSprop	20	32	224×224	45°	Horizontal	0.2	0.0001
ResNet50V2	RMSprop	20	32	224×224	45°	Horizontal	0.2	0.0001

The composition of the augmented data can be seen in Figure 3

Figure 3 shows the original image data and the augmented image data. The augmented image data is created by applying various transformations and augmentation techniques to the original data. This provides additional information that varies and enriches the dataset, assisting the model in coping with variations in real-world conditions.

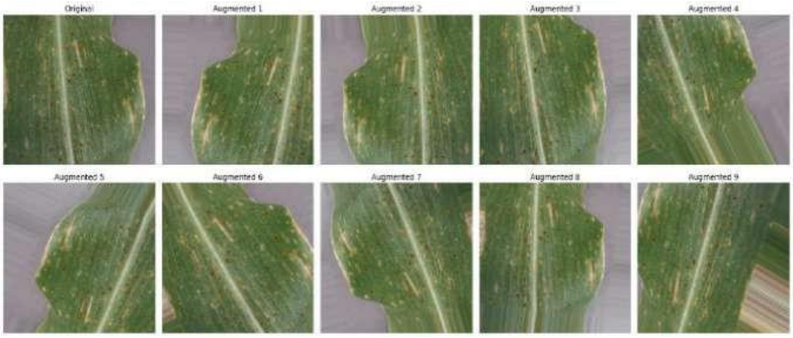

Figure 3. Data Augmentation Result

3.3. Splitting Data

After data augmentation, data sharing with deep learning consists of training, validation, and testing data. Training data is used to train the model to recognize corn leaf diseases, validation data to evaluate the model's performance at the end of each epoch, and testing data to evaluate the model's final performance. The holdout technique is used with an initial ratio of 80:20, where 80% is for training and the remaining 20% is divided equally into 10% validation and 10% testing. This distribution ratio balances the availability of sufficient data for training and sufficient independent data for evaluating the model's performance and generalization. With this scheme, the potential for overfitting can be reduced and the model evaluation results become more objective and reliable. The distribution of data sharing with the Holdout Technique can be seen in Table 4.

Table 4. Results of 80:20 Hold Out Data Splitting

Hold Out		80:20	
Class	Data Train	Data Validation	Data Testing
Blight	460	57	58
Common Rust	522	65	66
Corn Gray Leaf Spot	229	29	29
Corn Healthy	464	58	59

Table 4 shows the results of data division using the holdout technique with a ratio of 80:20 for testing the classification of corn leaf diseases using the CNN algorithm. Data from each class, namely Blight, Common Rust, Corn Gray Leaf Spot, and Corn Healthy, were divided into three parts: training data, validation data, and testing data. Most of the data was used for training (about 80%), while the rest was divided equally for validation and testing (about 10% each).

3.4. Data Training Using Deep Learning

This model classifies 224x224 pixel corn leaf images into four disease classes. Each model does not include the top classification layer, and GlobalAveragePooling2D, Dense 512 units (ReLU), 20% dropout, and 4 units of dense output (softmax) layers are added. The softmax activation function is used because the classification is multi-class. Adam's optimizer and RMSprop were used to train the model, which had a loss and a learning rate of 0.0001. function of categorical_crossentropy. Training was performed for 20 epochs with the augmented data. EarlyStopping and ModelCheckpoint callbacks are used to stop training if there is no improvement in validation loss and save the best weights. The training process of the deep learning model of DenseNet201 architecture using Adam and RMSprop optimizer can be seen in Figures 4 and 5, The deep learning model's training procedure of ResNet0V2 architecture Figures 6 and 7 show the training process of the deep learning model of the MobileNetV2 architecture utilizing Adam and RMSprop optimizer, while Figures 8 and 9 show the same procedure.

Based on Figure 4, training the DenseNet201 model with data augmentation and the Adam optimizer results in consistent increases in training and validation accuracy above 93%, accompanied by a stable decrease in loss, indicating good generalization capabilities. Figure 5 shows the performance of DenseNet201 with the RMSprop optimizer, which is also stable, with training and validation accuracy in the range of 92–93% and a balanced decrease in loss without any indication of overfitting. In Figure 6, the graph shows an increase in training and validation accuracy to around 93% with a continuing downward trend in loss, indicating an effective and stable training process. Figure 7 shows the training results of ResNet50V2 with the RMSprop optimizer, where the validation accuracy is relatively stable at around 90% while the training accuracy reaches around 93%, with a fairly consistent decrease in loss. Furthermore, Figure 8 shows the performance of MobileNetV2 with the Adam optimizer, achieving an accuracy of around 91% on both training and validation data and a stable decrease in loss, indicating good generalization. Figure 9 shows the training results of MobileNetV2 with the RMSprop optimizer, which produce consistent training and validation accuracies of 88–90% and a decreasing loss trend, indicating that the training process runs well without significant overfitting. The results and training durations of the deep learning models with DenseNet201, ResNet50V2, and MobileNetV2 architectures are shown in Table 5.

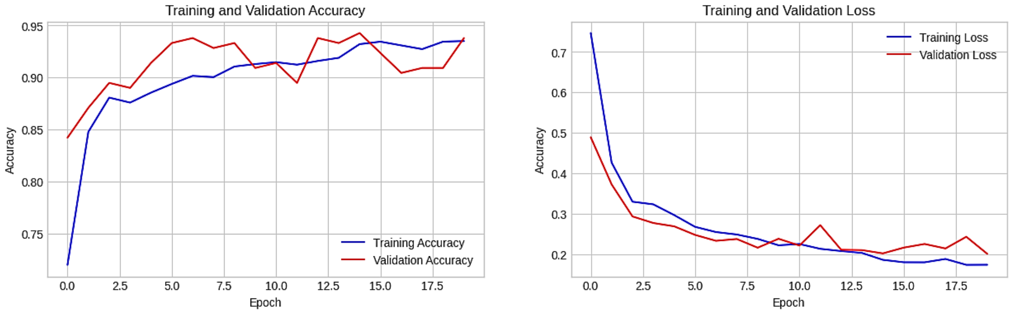

Figure 4. Deep Learning DenseNet201 Optimizer Adam Model Curve

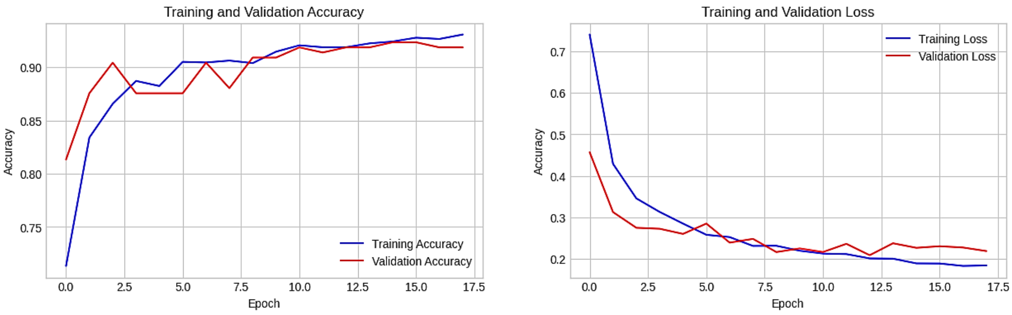

Figure 5. Deep Learning DenseNet201 Curve Model RMSprop Optimizer

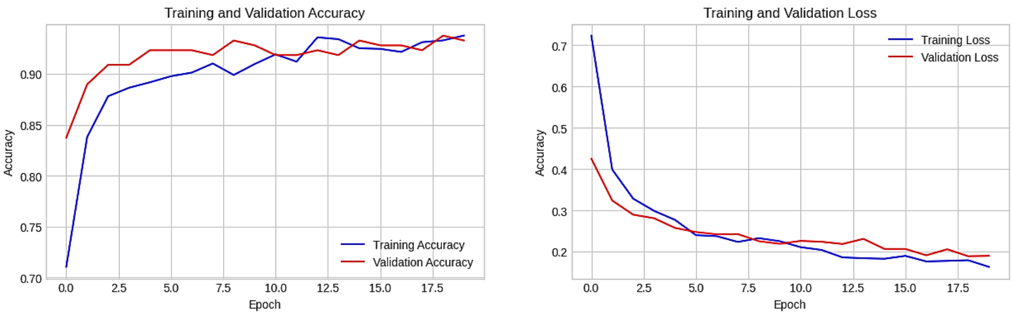

Figure 6. Deep Learning ResNet50V2 Optimizer Adam Model Curve

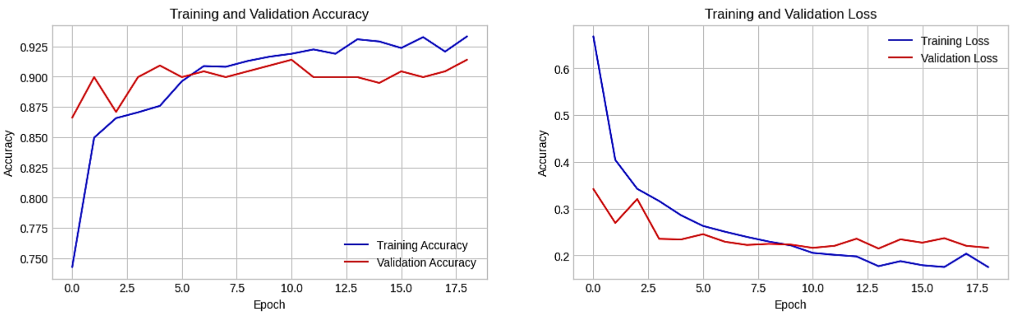

Figure 7. Deep Learning Model Curve ResNet50V2 Optimizer RMSprop

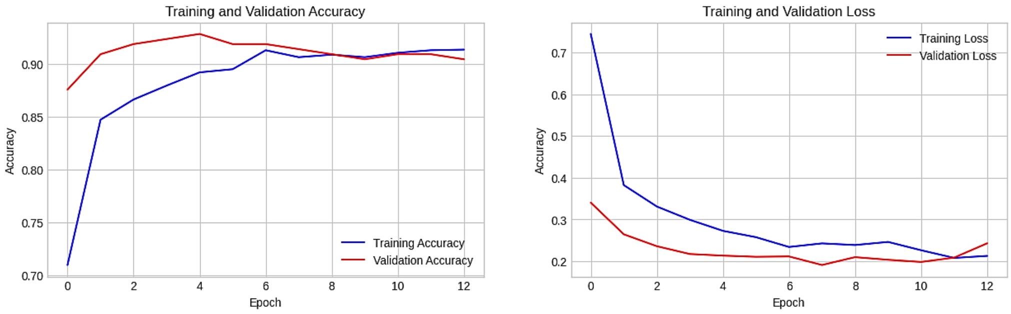

Figure 8. Deep Learning MobileNetV2 Optimizer Adam Model Curve

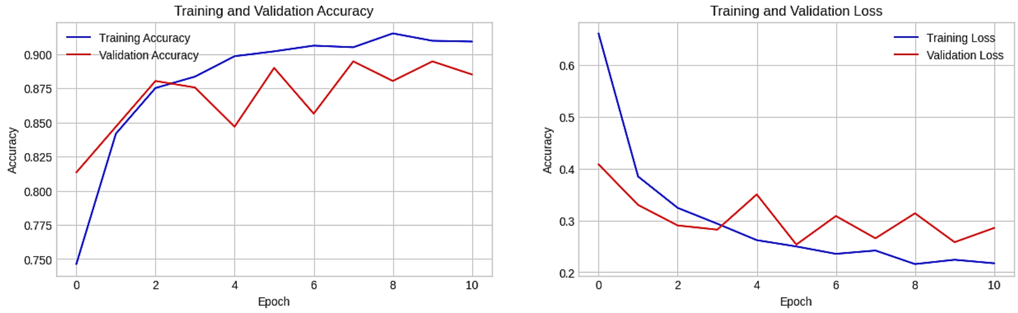

Figure 9. Deep Learning Model Curve MobileNetV2 Optimizer RMSprop

Table 5. Model Training Accuracy Results

Architecture	Optimizer	Train Accuracy	Valid Accuracy	Testing Accuracy	Training Loss	Valid Loss	Testing Loss
DenseNet201	Adam	94.50%	93.77%	92.92%	0.157119	0.200669	0.180396
	RMSprop	93.97%	91.86%	91.50%	0.177038	0.208606	0.224577
MobileNetV2	Adam	91.88%	91.38%	90.09%	0.201050	0.190720	0.231950
	RMSprop	92.17%	88.99%	87.73%	0.208946	0.253396	0.287457
ResNet50V2	Adam	93.55%	93.77%	91.50%	0.164651	0.189147	0.209229
	RMSprop	93.25%	89.95%	91.50%	0.171962	0.214695	0.251979

3.5. Confusion Matrix Evaluation

Subsequently, the models were evaluated for their performance in data classification using a Confusion Matrix. The findings from the matrix of misunderstanding of the DenseNet201 architecture using the Adam optimizer and RMprop shows in Fig. 10 and 11. The results of the confusion matrix of the ResNet50V2 architecture using the Adam optimizer and RMSprop can be seen in Fig.12 and 13. The results of the confusion matrix of the MobileNetV2 architecture using the Adam optimizer and RMSprop can be seen in Fig. 14 and Fig.15.

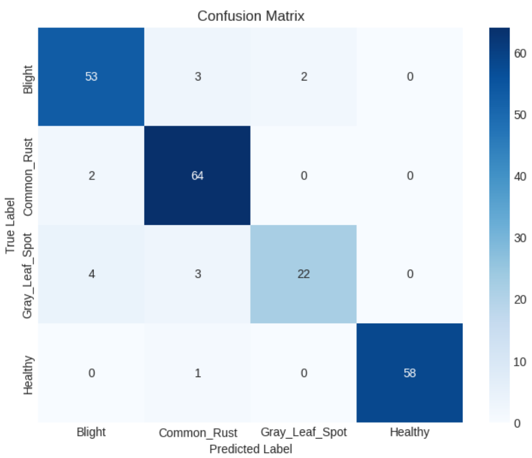

Figure 10. Confusion Matrix Architecture DenseNet201 Optimizer Adam

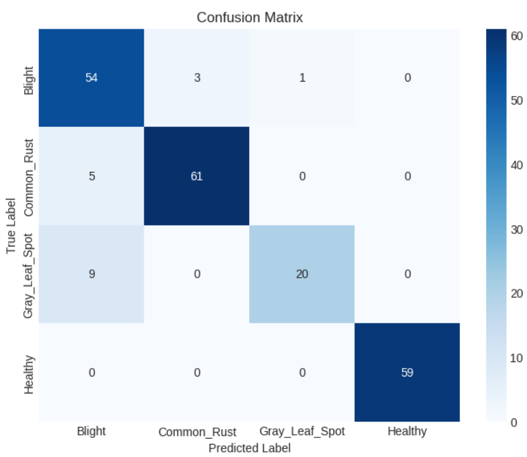

Figure 11. Confusion Matrix Architecture DenseNet201 Optimizer RMSprop

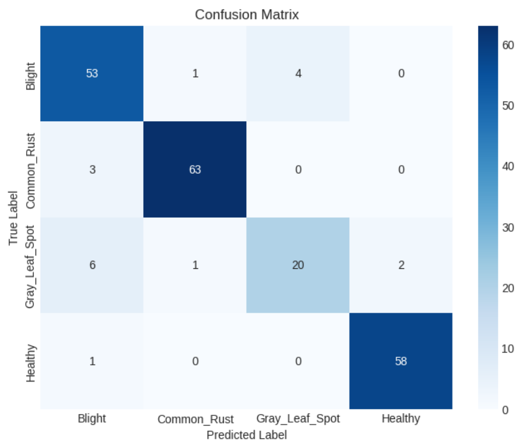

Figure 12. Confusion Matrix Architecture ResNet50V2 Optimizer Adam

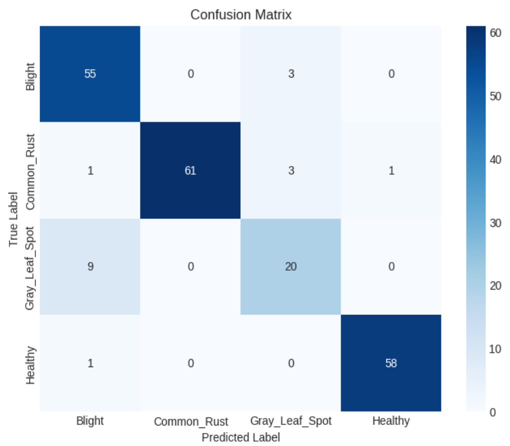

Figure 13. Confusion Matrix Architecture ResNet50V2 Optimizer RMSprop

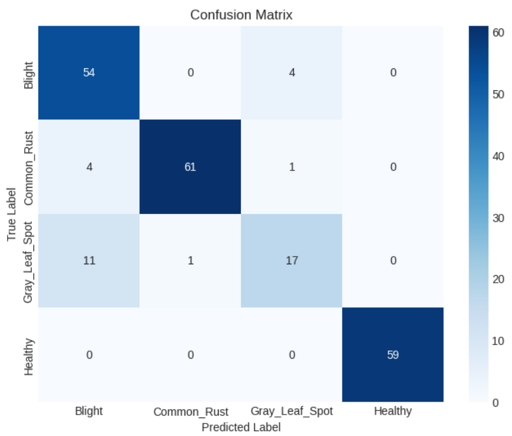

Figure 14. Confusion Matrix of MobileNetV2 Optimizer Adam Architecture

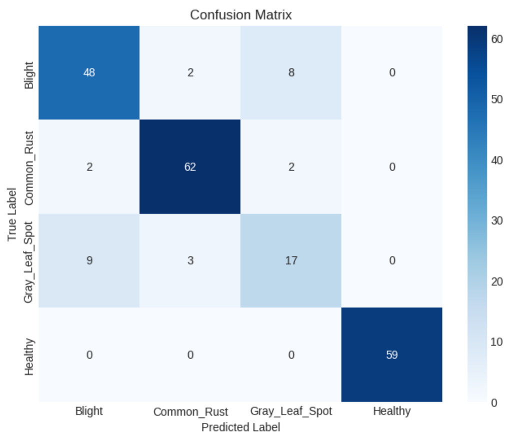

Figure 15. Confusion Matrix Architecture MobileNetV2 Optimizer RMSprop

Based on Figure 10, the DenseNet201 model with the Adam optimizer shows good performance, where the Common Rust and Healthy classes have the highest classification rates, Blight is identified quite well, while Gray Leaf Spot has the lowest accuracy and is often confused with Blight and Common Rust. Figure 11 shows that DenseNet201 with the RMSprop optimizer and data augmentation achieves high accuracy on the Healthy and Common Rust classes and stable performance on Blight, but Gray Leaf Spot remains the most difficult class to recognize. In Figure 12, the ResNet50V2 model with the Adam optimizer classifies the Blight, Common Rust, and Healthy classes well, but still experiences significant errors in the Gray Leaf Spot class. Figure 13 shows similar results for ResNet50V2 with the RMSprop optimizer, where Blight and Common Rust are classified accurately, while Gray Leaf Spot still shows a higher error rate. Furthermore, Figure 14 shows that MobileNetV2 with the Adam optimizer and data augmentation achieves excellent classification for the Healthy, Common Rust, and Blight classes, but still struggles to distinguish Gray Leaf Spot from Blight. Figure 15 shows that MobileNetV2 with the RMSprop optimizer achieves high accuracy in classifying Common Rust and Healthy, but still shows limitations in distinguishing Blight and Gray Leaf Spot. The results of the Confusion Matrix evaluation of the deep learning model using measurement metrics, namely Accuracy, Precision, Recall, and F1-score, can be seen in Table 6.

Table 6. Deep Learning Model Evaluation Results

Architecture	Optimizer	Class	Accuracy	Precision	Recall	F1 Score	Support
	Adam	Blight		0.90	0.91	0.91	58
		Common_Rust		0.90	0.97	0.93	66
		Gray_leaf_spot	0.93%	0.92	0.76	0.83	29
DenseNet201		Healthy		1.00	0.98	0.99	59
	RMSprop	Blight		0.79	0.93	0.86	58
		Common_Rust	0.92%	0.95	0.92	0.94	66

Architecture	Optimizer	Class	Accuracy	Precision	Recall	F1 Score	Support
		Gray_leaf_spot		0.95	0.69	0.80	29
		Healthy		1.00	1.00	1.00	59
	Adam	Blight	0.90%	0.78	0.93	0.85	58
		Common_Rust		0.98	0.92	0.95	66
		Gray_leaf_spot		0.77	0.59	0.67	29
MobileNetV2		Healthy		1.00	1.00	1.00	59
	RMSprop	Blight		0.81	0.83	0.82	58
		Common_Rust	0.88%	0.93	0.94	0.93	66
		Gray_leaf_spot		0.63	0.59	0.61	29
		Healthy		1.00	1.00	1.00	59
	Adam	Blight	0.92%	0.84	0.91	0.88	58
		Common_Rust		0.97	0.95	0.96	66
		Gray_leaf_spot		0.83	0.69	0.75	29
ResNet50V2		Healthy		0.97	0.98	0.97	59
	RMSprop	Blight	0.92%	0.83	0.95	0.89	58
		Common_Rust		1.00	0.92	0.96	66
		Gray_leaf_spot		0.77	0.69	0.73	29
		Healthy		0.98	0.98	0.98	59

Evaluation of deep learning models with DenseNet201, MobileNetV2, and ResNet50V2 architectures and Adam and RMSprop optimizers shows that DenseNet201 with Adam has the best performance. For the Gray leaf spot and Healthy classes, this model produced the best accuracy of 0.93 with Precision, Recall, and F1-Score values over 0.98. Although they are still behind DenseNet201, MobileNetV2, and ResNet50V2 also did well, particularly with RMSprop. Therefore, the best setup for this investigation is DenseNet201 and Adam together.

3.6. Comparison of Evaluation Result Accuracy

The combination of DenseNet201 with the Adam optimizer achieved the highest accuracy of 93%, surpassing MobileNetV2 (90%) and ResNet50V2 (92%). Adam's optimizer also generally gives higher results than RMSprop on the same architecture. Thus, DenseNet201 and Adam are the best configurations for corn leaf disease detection based on accuracy. Comparison of CNN evaluation accuracy is shown in Figure 16.

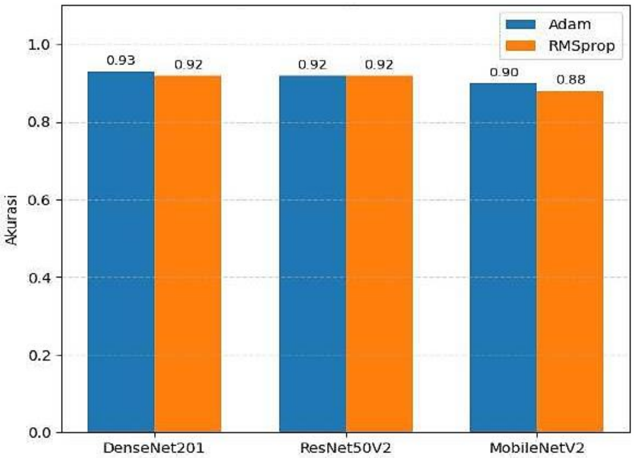

Figure 16. Comparison of CNN Evaluation Accuracy

3.7. Discussion

The results of the study show that the combination of CNN architecture and optimizer has a significant effect on the performance of corn leaf disease classification. Of all the configurations tested, DenseNet201 with the Adam optimizer achieved the best performance, with training accuracy of 94.50%, validation accuracy of 93.77%, and a test accuracy of 92.92%, along with low, stable loss values. These findings indicate that this configuration has the best generalization performance among the models in this study. Compared with previous studies, the performance of DenseNet201–Adam in this study is consistent with the findings of Mengesha and Mengistie (2025), who reported a test accuracy of 99.17% using the same architecture [8]. Although the accuracy in this study is lower, this difference may be influenced by variations in the dataset, the number of test samples, the number of classes, and the validation scheme used. However, the results of this study reinforce the evidence that DenseNet201 is a very effective architecture for handling leaf disease classification with high visual pattern similarity.

In the ResNet50V2 architecture, the use of the Adam optimizer resulted in a validation accuracy of 93.77% and a testing accuracy of 91.50%, which is higher than the results of Imran Khan et al. (2024), who reported a validation accuracy of 87.51% using ResNet50 [3]. This improvement indicates that the application of ResNet version V2 combined with data augmentation and uniform training parameter settings can enhance training stability and model generalization capabilities.

Meanwhile, MobileNetV2 showed relatively lower performance compared to DenseNet201 and ResNet50V2, but still produced competitive performance with the highest testing accuracy of 90.09% using the Adam optimizer. These results align with the research by Al-Gaashani et al. (2025) and Gumelar et al. (2025), which confirms that MobileNetV2 is suitable for leaf disease classification with high computational efficiency, although its accuracy tends to be lower compared to deeper CNN architectures [5], [12].

Confusion matrix analysis shows that the Common Rust and Healthy classes are classified with high accuracy, while Gray Leaf Spot remains the most challenging class due to its visual similarity to other diseases. Unlike previous studies, this study systematically compares three CNN architectures with two optimizers in a consistent testing framework and shows that DenseNet201–Adam is the most optimal configuration for corn leaf disease classification.

4. CONCLUSION

This study evaluates the performance of three Convolutional Neural Network (CNN) architectures, namely DenseNet201, ResNet50V2, and MobileNetV2, with two optimizers (Adam and RMSprop) for image-based corn leaf disease classification. The test results show that DenseNet201 with the Adam optimizer provides the best performance, with the highest accuracy of 93%, as well as consistent precision, recall, and F1-score values across all classes. ResNet50V2 and MobileNetV2 also showed competitive performance with accuracies of approximately 92% and 90%, respectively, confirming that transfer learning-based CNNs are effective for supporting early detection of corn leaf disease.

This study has several limitations, including the use of a partial dataset (2,094 images), which limits the model's ability to generalize to variations in field conditions, and data imbalance between classes, particularly in the Gray Leaf Spot class, which affects classification accuracy due to the similarity of visual features between diseases. Furthermore, this study has not explored advanced hyperparameter variations or ensemble approaches that have the potential to improve model performance. As a further development, it is recommended to use the entire dataset, apply class balancing techniques, and explore ensemble models or lightweight CNN architectures to improve accuracy, generalization, and computational efficiency, so that the resulting classification system is more robust and applicable to agriculture.

REFERENCES

[1] 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.

[2] J. T. Elekterika et al., “Klasifikasi Penyakit Tanaman Jagung Melalui Citra Daun Dengan Menggunakan Metode Deep Learning,” vol. x, no. x.

[3] I. Khan, S. S. Sohail, D. Ø. Madsen, and B. K. Khare, “Deep transfer learning for fine-grained maize leaf disease classification,” J. Agric. Food Res., vol. 16, no. September 2023, 2024, doi: 10.1016/j.jafr.2024.101148.

[4] 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.

[5] G. Gumelar et al., “Jurnal Computer Science and Information Technology (CoSciTech) Implentation of CNN for Corn Leaf Disease Identification,” vol. 6, no. 2, pp. 175–180, 2025.

[6] A. D. Nurcahyati, R. M. Akbar, and S. Zahara, “Klasifikasi Citra Penyakit pada Daun Jagung Menggunakan Deep Learning dengan Metode Convolution Neural Network (CNN),” SUBMIT J. Ilm. Teknol. Infomasi dan Sains, vol. 2, no. 2, pp. 43–51, 2022, doi: 10.36815/submit.v2i2.1877.

[7] S. Babu, M. Maravarman, and R. Pitchai, “Detection of Rice Plant Disease Using Deep Learning Techniques,” J. Mob. Multimed., vol. 18, no. 3, pp. 757–770, 2022, doi: 10.13052/jmm1550-4646.18314.

[8] A. T. Mengesha and M. A. Mengistie, “Applying transfer learning in CNN model architectures for detecting tomato leaf disease with explainable artificial intelligence,” Smart Agric. Technol., vol. 11, no. May, p. 101034, 2025, doi: 10.1016/j.atech.2025.101034.

[9] 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.
[10] A. B. Prakosa, Hendry, and R. Tanone, "Implementasi Model Deep Learning Convolutional Neural Network (CNN) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman," J. Pendidik. Teknol. Inf., vol. 6, no. 1, pp. 107–116, 2023.
[11] I. P. Putra, R. Rusbandi, and D. Alamsyah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network," J. Algoritm., vol. 2, no. 2, pp. 102–112, 2022, doi: 10.35957/algoritme.v2i2.2360.
[12] M. S. A. M. Al-Gaashani, R. Alkanhel, M. A. S. Ali, M. S. A. Muthanna, A. Aziz, and A. Muthanna, "MSCPNet: A Multi-Scale Convolutional Pooling Network for Maize Disease Classification," IEEE Access, vol. 13, no. December 2024, pp. 11423–11446, 2025, doi: 10.1109/ACCESS.2024.3524729.
[13] J. Sharma et al., "Deep learning based ensemble model for accurate tomato leaf disease classification by leveraging ResNet50 and MobileNetV2 architectures," Sci. Rep., vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-98015-x.
[14] R. K. Megalingam, G. Gopakumar Menon, S. Binoj, D. Asandi Sai, A. R. Kunnambath, and S. K. Manoharan, "Cowpea leaf disease identification using deep learning," Smart Agric. Technol., vol. 9, no. October, p. 100662, 2024, doi: 10.1016/j.atech.2024.100662.
[15] 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.
[16] R. Kusumastuti, T. Dwi Putra, and Z. Zulfahmi Yudam, "Klasifikasi Citra Penyakit Daun Jagung Menggunakan Algoritma Cnn Effcientnet," Multitek Indones., vol. 17, no. 2, pp. 143–153, 2024, doi: 10.24269/mtkind.v17i2.10085.
[17] 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.
[18] E. E. Abdallah, E. Abu, and A. E. Abdallah, "ScienceDirect A Survey of Data Mining Techniques for Digital Forensic Analysis," Procedia Comput. Sci., vol. 257, pp. 731–736, 2025, doi: 10.1016/j.procs.2025.03.094.
[19] J. Kpodo and A. P. Nejadhashemi, "Navigating challenges/opportunities in developing smart agricultural extension platforms: Multi-media data mining techniques," Artif. Intell. Agric., vol. 15, no. 3, pp. 426–448, 2025, doi: 10.1016/J.AIIA.2025.04.001.
[20] Y. Mardi, "Data Mining : Klasifikasi Menggunakan Algoritma C4.5," Edik Inform., vol. 2, no. 2, pp. 213–219, 2017, doi: 10.22202/ei.2016.v2i2.1465.
[21] 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.
[22] J. Andrew, J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, "Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications," Agronomy, vol. 12, no. 10, pp. 1–19, 2022, doi: 10.3390/agronomy12102395.
[23] M. Rijal, A. M. Yani, and A. Rahman, "Deteksi Citra Daun untuk Klasifikasi Penyakit Padi menggunakan Pendekatan Deep Learning dengan Model CNN," J. Teknol. Terpadu, vol. 10, no. 1, pp. 56–62, 2024, doi: 10.54914/jtt.v10i1.1224.
[24] 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. November, p. 102859, 2024, doi: 10.1016/j.ecoinf.2024.102859.
[25] M. S. Iqbal, M. Adnan, S. E. G. Mohamed, and M. Tariq, "A hybrid deep learning framework for short-term load forecasting with improved data cleansing and preprocessing techniques," Results Eng., vol. 24, no. November, p. 103560, 2024, doi: 10.1016/j.rineng.2024.103560.
[26] Y. Fu, L. Guo, and F. Huang, "A lightweight CNN model for pepper leaf disease recognition in a human palm background," Heliyon, vol. 10, no. 12, p. e33447, 2024, doi: 10.1016/j.heliyon.2024.e33447.
[27] L. Falaschetti, L. Manoni, D. Di Leo, D. Pau, V. Tomaselli, and C. Turchetti, "A CNN-based image detector for plant leaf diseases classification," HardwareX, vol. 12, p. e00363, 2022, doi: 10.1016/j.ohx.2022.e00363.
[28] A. Maysela and N. Rohma, "Diagnosa Penyakit Tanaman Tomat pada Citra Daun Menggunakan Metode Convolutional Neural Network (CNN)," JIMU J. Ilm. Multi Disiplin, vol. 02, no. 03, pp. 555–567, 2024, [Online]. Available: https://ojs.smkmerahputih.com/index.php/jimu/article/view/407
[29] V. Singh, A. Chug, and A. P. Singh, "Classification of Beans Leaf Diseases using Fine Tuned CNN Model," Procedia Comput. Sci., vol. 218, no. 2022, pp. 348–356, 2022, doi: 10.1016/j.procs.2023.01.017.
[30] F. Sulistiyana and S. Anardani, "Aplikasi Deteksi Penyakit Tanaman Jagung Dengan Convolutional Neural Network dan Support Vector Machine," Semin. Nas. Teknol. Inf. dan Komun., vol. 6, no. 1,

pp. 423–432, 2023.
[31] A. Tan, G. Zhou, and M. He, “Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion,” IEEE Access, vol. 8, pp. 124722–124733, 2020, doi: 10.1109/ACCESS.2020.3007745.
[32] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
[33] V. Balafas, E. Karantoumanis, M. Louta, and N. Ploskas, “Machine Learning and Deep Learning for Plant Disease Classification and Detection,” IEEE Access, vol. 11, no. September, pp. 114352–114377, 2023, doi: 10.1109/ACCESS.2023.3324722.
[34] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, and R. Terebes, “Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture,” IEEE Access, vol. 9, pp. 160085–160103, 2021, doi: 10.1109/ACCESS.2021.3131002.
[35] C. P. Lee, K. M. Lim, Y. X. Song, and A. Alqahtani, “Plant-CNN-ViT: Plant Classification with Ensemble of Convolutional Neural Networks and Vision Transformer,” Plants, vol. 12, no. 14, pp. 1–21, 2023, doi: 10.3390/plants12142642.
[36] 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.
[37] 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.
[38] J. Kusuma, Rubianto, R. Rosnelly, Hartono, and B. H. Hayadi, “Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron,” J. Appl. Comput. Sci. Technol., vol. 4, no. 1, pp. 1–6, 2023, doi: 10.52158/jacost.v4i1.484.
[39] A. N. Akmal, N. Maelasari, T. Ilmu, and P. Islam, “Pemahaman Deep Learning dalam Pendidikan : Analisis Literatur melalui Metode Systematic Literature Review (SLR),” vol. 8, 2025, doi: https://doi.org/10.54371/jiip.v8i3.7442.