IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3757~3770
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3757-3770

MobileChiliNet: convolutional neural network for chili leaves classification

Sayuti Rahman[1], Marischa Elveny[2], Marwan Ramli[3], Dionikxon Manurung[1]
[1]Department of Informatics, Faculty of Engineering, Universitas Medan Area, Medan, Indonesia
[2]Department of Data Science and Artifficial Intelligence, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan. Indonesia
[3]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Banda Aceh, Indonesia

Article Info

Article history:

Received Nov 7, 2024
Revised 14 Jul, 2025
Accepted Aug 6, 2025

Keywords:

Chili leaves
Chili leaves classification
Classification
Convolutional neural networks
MobileChiliNet

ABSTRACT

Chili pepper (*Capsicum annuum*) is an important crop in many countries, including Indonesia, which plays an essential role in the local economy and food production. To meet the high demand, effective agricultural management, especially the diagnosis and treatment of plant diseases, is essential. This study aims to improve the accuracy of chili leaf disease classification while reducing the computational cost so that it can be applied to low-cost smart farming systems. Through the development of the MobileChiliNet architecture, which is the result of pruning and fine-tuning of MobileNetV2, this model achieves the best accuracy, better than other convolutional neural networks (CNNs) such as residual network (ResNet50) and visual geometry group (VGG)16. Testing with various optimizers and learning rate schedulers shows that AdamW with PolynomialDecay provides the best performance by increasing the validation accuracy to 96.48%. The reduced model complexity directly translates into faster inference times and lower hardware requirements, allowing the model to run on edge devices such as Raspberry Pi or smartphones. This makes MobileChiliNet highly practical for smallholder farmers and rural agricultural settings, where computational resources are limited. By balancing high classification performance with minimal computational demands, MobileChiliNet supports scalable, affordable, and real-time disease monitoring for precision agriculture.

Corresponding Author:

Marischa Elveny
Department of Data Science and Artifficial Intelligence
Faculty of Computer Science and Information Technology, Universitas Sumatera Utara
Medan, Indonesia
Email: marischaelveny@usu.ac.id

1. INTRODUCTION

All standard paper components have been specified for the sustainable development goals (SDGs) emphasize the importance of consumption and production in improving the quality of life in society, particularly in the agricultural sector, which plays a key role in ensuring food security and reducing poverty. Agriculture also plays an important role in nation-building [1]. One area of focus is the cultivation and maintenance of chili plants, specifically red chili (*Capsicum annuum*), which is an important crop in many countries, including Indonesia [2], [3]. Red chili not only contributes to the local economy but also to food production, such as sauce production [4], medicine [5], and chili powder [6]. The demand for chili is very high, which requires effective agricultural management, especially in the diagnosis and treatment of plant diseases. This is crucial to support sustainable agricultural practices as targeted by the SDGs [7]. Early detection of diseases in chili leaves can help reduce the risk of crop yield loss and improve production quality.

If chili leaf diseases are not detected early, the impact can be highly significant on production and crop quality. Diseased plants will experience stunted growth, which will ultimately affect the quantity and quality of the fruit produced. Some of the impacts caused by chili leaf diseases include stunted growth, flower drop, undersized fruits, and even rot [8]. Additionally, undetected disease spread can lead to widespread infection throughout the entire farming area, drastically increasing the cost of treatment and disease control, such as the more frequent and intensive use of pesticides. This not only causes financial losses for farmers but also has the potential to harm the environment.

Failure to detect chili leaf diseases early can also extend the recovery time of the plants, thus affecting the next planting cycle. As a result, lower crop yields may affect the supply of chili in the market, driving up prices, and undermining the economic stability of farmers [9]. Therefore, an efficient and accurate approach to chili leaf disease classification is crucial to maintain agricultural productivity and ensure the sustainability of chili farming. Accurate classification of chili leaf diseases is essential for early diagnosis, enabling farmers to take timely action in disease management. Traditional methods of disease detection are often time-consuming and prone to errors. Therefore, automating the process of chili leaf disease classification using image-based machine learning methods is highly beneficial. Through classification techniques, farmers can obtain accurate and reliable information about the condition of their plants, which will help them make better decisions. Similar approaches have been applied to other crops, such as the diagnosis of Alternaria disease and Leafminer pest on tomato leaves using image processing techniques, demonstrating the effectiveness of automated visual analysis in precision agriculture [10].

Deep learning, particularly convolutional neural networks (CNN), has gained a lot of attention due to its better performance in image classification tasks [11], including plant disease detection [12]. CNN has been widely used in various agricultural applications because of its ability to automatically extract important features from images without the need for manual feature engineering. This makes CNN an ideal choice for chili leaf disease classification, as it can handle complex visual patterns and variations in leaf images, resulting in more accurate predictions [13].

Recent studies show that CNN model performance can be enhanced through techniques such as transfer learning [14], fine-tuning [15], and pruning [16]. Transfer learning allows for leveraging pre-trained models to improve classification performance on new datasets with minimal training time, while fine-tuning further optimizes the model by adjusting specific layers. On the other hand, pruning helps reduce model size by eliminating unnecessary parameters, resulting in more efficient computation without sacrificing accuracy.

In addition, recent advancements in lightweight CNN architectures have demonstrated strong potential for agricultural applications on edge devices. A recent study introduced an ultra-lightweight network with a low number of parameters, yet capable of achieving competitive accuracy in plant disease and pest detection while maintaining minimal computational complexity [17]. Another study implemented a MobileNetV3Large-based model for real-time grape leaf disease classification on an edge device (Jetson Nano), achieving over 99% accuracy along with explainability features using Grad-CAM [18]. These developments reinforce the practicality of lightweight CNNs in real-world agricultural environments, particularly in rural or low-resource settings with limited computing power.

In this study, we propose MobileChiliNet, a lightweight and accurate model developed by combining pruning and fine-tuning techniques on MobileNetV2. The objective is to create a compact model with fewer parameters and high classification accuracy, suitable for deployment on low-power hardware such as smartphones or Raspberry Pi. Table 1 summarizes related research on leaf disease classification using various methods and their accuracies.

The methods listed in Table 1 demonstrate various techniques for chili leaf classification. Fine-tuning models like ShuffleNet results in high accuracy, but only for two classes. For a larger number of classes, such as five classes in models like support vector machine+recurrent neural network (SVM+RNN), extreme inception (Xception), and EfficientLeafNetB4, the accuracy remains relatively low, with a maximum accuracy of 92.10%. This indicates that there is still a need for improved accuracy in chili leaf disease classification to ensure its applicability in smart agricultural systems.

This research aims to develop a model that not only improves accuracy but also reduces computational complexity. By utilizing pruning techniques combined with fine-tuning, we target a compact and efficient architecture that can achieve higher accuracy with lower computational costs. The ultimate goal of this research is to produce a model that can be used in real agricultural applications, improving disease detection in chili plants while minimizing computational costs. The major contributions of this study are summarized as: i) proposes MobileChiliNet, a lightweight deep learning model optimized from MobileNetV2 through pruning and fine-tuning for chili leaf disease classification; ii) achieves a validation accuracy of

96.48%, outperforming several state-of-the-art CNNs across five disease classes; iii) evaluates multiple optimization algorithms and learning rate schedulers, identifying AdamW with PolynomialDecay as the most stable and accurate combination; iv) ensures real-time deployment feasibility on low-power edge devices such as Raspberry Pi and smartphones, promoting accessibility in resource-limited agricultural environments; and v) contributes to precision farming and SDGs by enabling early disease detection, reducing crop loss, and supporting smallholder farmers.

The remainder of this paper is organized as: section 2 describes the research methodology, including dataset preparation, model pruning, and fine-tuning processes. Section 3 presents the results and discussions, including comparisons with existing CNNs, evaluation of various optimizers, and the effect of different learning rate schedulers. Section 4 provides the conclusions and implications of the proposed MobileChiliNet model for smart agricultural systems.

Table 1. Research related to leaves classification

Methods	Number of classes	Accuracy (%)
VGGNet [19]	3	97.00
SVM+RNN [20]	5	92.10
GLCM+KNN [21]	2	94.00
Fine tuning ShuffleNet [22]	2	99.30
Inception V3 [23]	4	93.00
Xception [24]	5	79.56
EfficientLeafNetB4 [25]	5	92.00
EfficentNet [26]	4	91.00

2. METHOD

This research aims to develop an optimal CNN model with better accuracy and faster classification performance. The research is divided into four main steps: dataset preparation, testing existing CNN models, pruning the best-performing CNN, and fine-tuning the hyperparameters of the pruned CNN model. The methodology used in this study is illustrated in Figure 1.

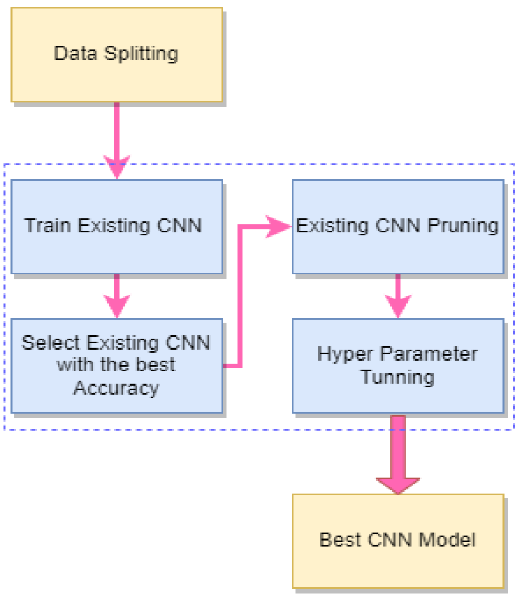


Figure 1. Research methodology

As shown in Figure 1, the dataset is split into 80% for training and 20% for validation. This data is used to test existing CNN models such as MobileNet [27], ResNet [28], VGGNet [29], Alexnet [30], and ShuffleNet [31], which have proven effective in classifying leaf diseases. The existing CNN model with the best accuracy is selected for pruning to create a more compact and faster classification model. The pruned model is then fine-tuned to further improve accuracy, resulting in a faster and more accurate model for classifying chili leaf diseases.

2.1. Datasets

The dataset utilized in this research is comprised of images depicting various diseases that affect red chili leaves, obtained from Mendeley data. These images are categorized into five distinct disease classes [24]. The dataset has been processed into two variants: augmented and non-augmented. Initially, the dataset contained 531 images; however, after applying data augmentation techniques, the dataset expanded significantly to 2,128 images. Table 2 provides a detailed breakdown of the augmented dataset.

Table 2. Augmented dataset

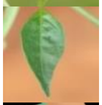
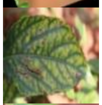
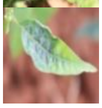
Class name	Image	Training	Validation	Total
Powdery mildew		486	122	608
Healthy leaf		221	55	276
Murda complex (mites, thrips)		342	86	428
Leaf spot (Cercospora)		326	82	408
Nutrient deficiency		327	81	408
Total		1,702	426	2,128

Table 2 outlines the distribution of the augmented dataset used in this study to classify diseases affecting red chili leaves. The dataset encompasses five distinct disease categories: powdery mildew, healthy leaf, murda complex (mites and thrips), leaf spot (*Cercospora*), and nutrient deficiency. Each disease class is represented by a set of images, which have been further divided into training and validation subsets. After augmentation, the dataset totals 2,128 images, with 1,702 designated for training and 426 for validation. The original images, which have undergone augmentation, are depicted in Figure 2.

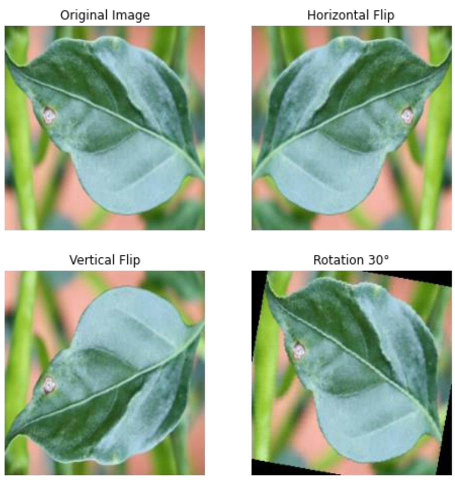


Figure 2. Data augmentation

2.2. MobileNetV2 architecture

The MobileNetV2 architecture utilizes depthwise separable convolutions and inverted residual blocks with a linear bottleneck, designed to improve computational efficiency and performance on devices with limited resources. In the initial stage, a standard convolutional layer with 32 filters and a stride of 2 is used to extract initial features from the input image. The network then progresses through a series of bottleneck blocks, starting with an expansion factor of 1, resulting in 16 filters without changing the output size. After that, a bottleneck block with an expansion factor of 6 is repeatedly applied, where the feature size is temporarily expanded before being compressed again. This process results in changes in output dimensions from 112×112 to 56×56, 28×28, 14×14, and finally 7×7, with the number of filters gradually increasing from 24, 32, 64, 96, 160, to 320. After passing through the final convolutional layer with 1,280 filters, average pooling is applied to reduce the feature dimensions to 1×1. The final layer is a fully connected layer with 5 neurons, which is used to classify the 5 classes of chili leaf diseases. The detailed architecture of MobileNetV2 for chili leaf disease classification is shown in Table 3.

Table 3. MobileNetV2 architecture

Layer type	t	C	n	s	Input size	Output size
Conv2D	-	32	1	2	224×224×3	112×112×32
Bottleneck	1	16	1	1	112×112×32	112×112×16
Bottleneck	6	24	2	2	112×112×16	56×56×24
Bottleneck	6	32	3	2	56×56×24	28×28×32
Bottleneck	6	64	4	2	28×28×32	14×14×64
Bottleneck	6	96	3	1	14×14×64	14×14×96
Bottleneck	6	160	3	2	14×14×96	7×7×160
Bottleneck	6	320	1	1	7×7×160	7×7×320
Conv2D	-	1280	1	1	7×7×320	7×7×1280
Avg pooling	-	-	1	-	7×7×1280	1×1×1280
Fully connected (FC)	-	5	1	-	1×1×1280	1×1×5

In Table 3, several important parameters explain the configuration of each layer: t (expansion factor) indicates how much the number of channels will be expanded before performing the depthwise convolution; c (output channels) represents the number of output channels from each layer or block; n (number of repeats) shows how many times the bottleneck block is repeated to increase complexity and feature extraction capability; and s (stride) indicates the shift step of the kernel during the convolution, which affects the spatial size of the output. A stride value greater than 1 (e.g., s=2) will result in a reduction of the resolution (downsampling) in the output, whereas a stride of 1 keeps the output size the same as the input. These parameter combinations help to understand the structure and functionality of each layer in the MobileNetV2 architecture. The bottleneck in MobileNetV2 is illustrated as shown in Figure 3.

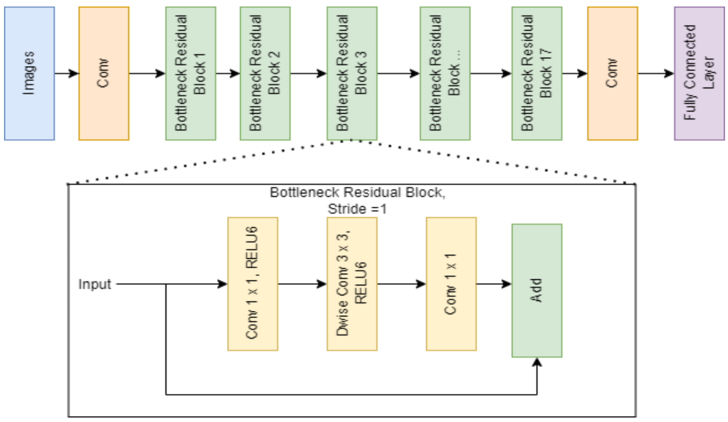


Figure 3. Illustration of the bottleneck in MobileNetV2

The bottleneck residual block in MobileNetV2 has different characteristics depending on the stride value. At stride=1, as shown in Figure 3, there is a branching process where the first branch simply passes the original input without any changes, while the second branch performs several operations. The second branch consists of a 1×1 convolution with a ReLU6 activation function, followed by a 3×3 depthwise convolution with a ReLU6 activation function, and then another 1×1 convolution without any activation function. These two branches are then summed to produce the final output of the block. On the other hand, at stride=2, the process is slightly different as it aims to downsample the features. At this stride value, the initial convolution layer in the second branch remains the same, but the 3×3 depthwise convolution with stride 2 is applied to reduce the spatial size of the output features. Additionally, the first branch does not merely pass the input but also undergoes adjustments to match the dimensions of the second branch. The outputs of both branches are not directly summed but are combined at the final stage to produce more compact and dense features.

2.3. Proposed convolutional neural network architecture

The proposed CNN architecture is the result of pruning the MobileNetV2 model, which previously showed the best accuracy in classifying chili leaf diseases. The pruning process was carried out by reducing the number of bottleneck layers in MobileNetV2, from seven layers to only three bottleneck layers with different output channels. This step aims to decrease computational costs without sacrificing model performance. After the layer reduction, the parameters t, c, n, and s were reconfigured to determine the optimal architecture setup. An illustration of the MobileNetV2 pruning process is shown in Figure 4.

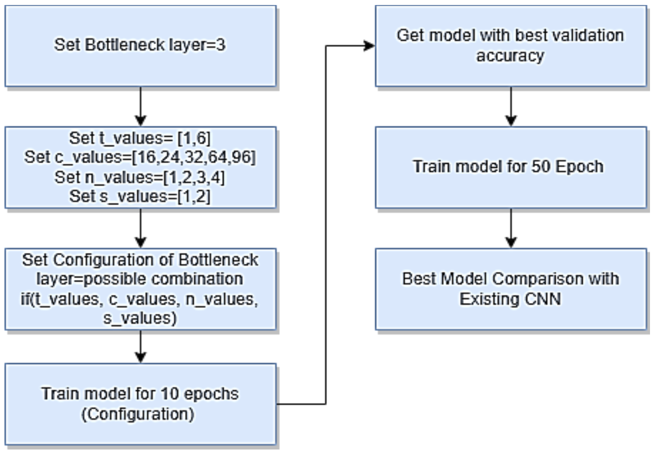


Figure 4. Illustration of the MobileNetV2 pruning process

As shown in Figure 4, each parameter combination was tested and trained over 10 epochs. The combination with the best validation accuracy was then retrained for up to 50 epochs to evaluate the final performance of the resulting CNN model. Based on these experiments, we successfully identified the best combination, which produced an optimal CNN architecture for chili leaf disease classification, which we named MobileChiliNet. The complete structure of the MobileChiliNet architecture is presented in Table 4.

Table 4. MobileChiliNet architecture

Layer type	t	C	n	s	Input size	Output size
Conv2D	-	32	1	2	3×128×128	32×64×64
Bottleneck	1	24	1	1	32×64×64	24×64×64
Bottleneck	6	32	3	2	24×64×64	32×32×32
Bottleneck	6	96	4	2	32×32×32	96×16×16
Conv2D	-	1,280	1	1	96×16×16	1280×16×16
AdaptiveAvgPool2D	-	-	1	-	1280×16×16	1280×1×1
Fully connected (linear)	-	5	1	-	1280	5

The MobileChiliNet architecture, shown in Table 4, consists of several key layers designed to extract important features from chili leaf images. The first layer is a Conv2D layer with 32 output channels and a stride of 2, serving as the initial layer to reduce the input image size from 128×128 to 64×64 and extract basic features. Next, bottleneck layers are used to optimize the number of parameters by utilizing the configuration of the expansion factor (t), the number of output filters (C), the number of repeats (n), and the stride (s).

Initially, a bottleneck with an expansion factor of 1 and 24 filters is applied without changing the output size. Then, a bottleneck with an expansion factor of 6 and 32 filters is used three times with a stride of 2, reducing the feature size to 32×32. A similar process is applied in the next layer with 96 filters and 4 repeats, further reducing the spatial size to 16×16.

Afterward, a Conv2D layer with 1,280 output channels is applied to enrich the feature representation, followed by average pooling using AdaptiveAvgPool2D, reducing the feature size to 1×1. Finally, a fully connected layer with 5 neurons is used as the classification layer to determine the chili leaf disease classes. The number of neurons in this layer corresponds to the number of classes to be classified, which is 5 types of chili leaf diseases.

With this combination of layers, MobileChiliNet achieves optimal performance in chili leaf disease classification while maintaining a lower number of parameters compared to the standard MobileNetV2 architecture, making it more efficient in computation and memory usage. Several reasons why a model with fewer layers can be more accurate include: i) reduced overfitting: CNNs with deeper layers are prone to overfitting as they tend to learn too many patterns from the training data, including noise. A simpler model can capture more general and important patterns, often leading to better accuracy [32]; and ii) more optimal parameter tuning: this allows for more efficient parameter optimization, as the smaller parameter space is easier for optimization algorithms to explore [33].

3. RESULTS AND DISCUSSION

This section presents the results from testing the MobileChiliNet model, which has undergone several stages of optimization. The research began by testing various existing CNN architectures, followed by the pruning process on MobileNetV2, and concluded with fine-tuning the parameters. The accuracy comparison results of each optimizer used during the training of MobileChiliNet, as well as the implementation of various learning rate schedulers, are outlined in the relevant tables. The discussion focuses on the impact of each method on improving accuracy and the stability of the model in chili leaf disease classification.

3.1. Comparison of existing convolutional neural networks

The first step in this research was utilizing existing CNN architectures to classify chili leaf diseases. The chili leaf disease dataset, which consists of five classes, was used as training and validation data for the existing CNN models. All existing CNNs used in this study were retrained on the dataset without using weights from transfer learning. This was done to identify the best architecture that could be further developed. Training was conducted using the SGD optimization function with a momentum of 0.9, a learning rate of 0.01, and a batch size of 64. The training results of the existing CNN models are shown in Table 5.

Table 5. Accuracy comparison of existing CNNs

Methods	Train accuracy (%)	Validation accuracy (%)
MobileNetV2	85.01	89.43
ResNet50	84.13	86.61
VGG16	74.32	76.29
AlexNet	67.21	68.30
ShuffleNet	77.02	77.46

Table 5 shows that the MobileNetV2 architecture achieved the highest accuracy among the evaluated models, with 85.01% training accuracy and 89.43% validation accuracy. Beyond its better performance, MobileNetV2 was selected as the base architecture due to its lightweight design and computational efficiency, which are critical for deployment in resource-limited environments. The model utilizes depthwise separable convolutions and inverted residual blocks with linear bottlenecks, significantly reducing the number of parameters and computational cost without compromising representational capacity. These characteristics make it well-suited for real-time applications on low-power devices such as Raspberry Pi or smartphones, aligning with the objective of this research. The pruning process, combined with hyperparameter tuning, led to the development of a more compact and efficient architecture named MobileChiliNet. The comparison between MobileChiliNet and existing CNNs is presented in Table 6.

Table 6. Accuracy comparison of MobileChiliNet and existing CNNs

Methods	Train accuracy (%)	Validation accuracy (%)
MobileNetV2	85.01	89.43
ResNet50	84.13	86.61
VGG16	74.32	76.29
AlexNet	67.21	68.30
ShuffleNet	77.02	77.46
MobileChiliNet	95.35	94.13

Table 6 shows that after the pruning and fine-tuning process on MobileNetV2, the resulting architecture, MobileChiliNet, achieved a training accuracy of 95.35% and a validation accuracy of 94.13%. This is significantly better than the other CNN architectures tested, such as ResNet50, VGG16, AlexNet, and ShuffleNet. MobileChiliNet demonstrated a substantial performance improvement, particularly in terms of validation accuracy, which indicates better model generalization for classifying chili leaf diseases.

3.2. Results of optimizer tuning on MobileChiliNet

One approach to improving CNN accuracy is tuning the optimizer function [33], [34]. In this research, we tested various optimization functions to enhance the performance of MobileChiliNet. Several optimizers tested include AdaGrad, Adam, AdamW, SDGM, and RMSprop, all of which are commonly used in CNN optimization. Each optimizer has a unique mechanism for updating weights, aiming to find the optimal solution during training. The graph in Figure 5 shows the performance of AdaGrad on MobileChiliNet, providing a visual overview of its impact on model accuracy.

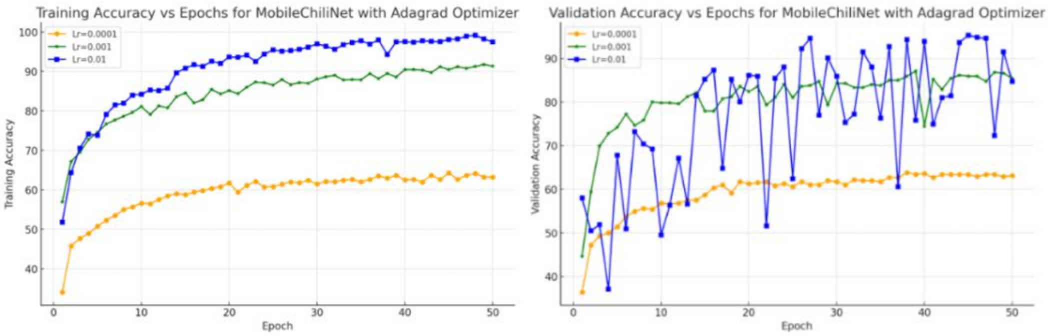


Figure 5. Training and validation accuracy of MobileChiliNet with AdaGrad optimizer

As shown in Figure 5, the use of the AdaGrad optimizer successfully improved the accuracy of MobileChiliNet, both in training and validation data. With a learning rate of 0.01, the training accuracy reached 99.18%, while the validation accuracy reached 95.31%. However, it can be seen that the validation accuracy experienced fluctuations, indicating performance instability. This suggests that high training accuracy does not always guarantee optimal validation accuracy. Next, an experiment was conducted using the Adam optimizer to compare performance, as shown in Figure 6.

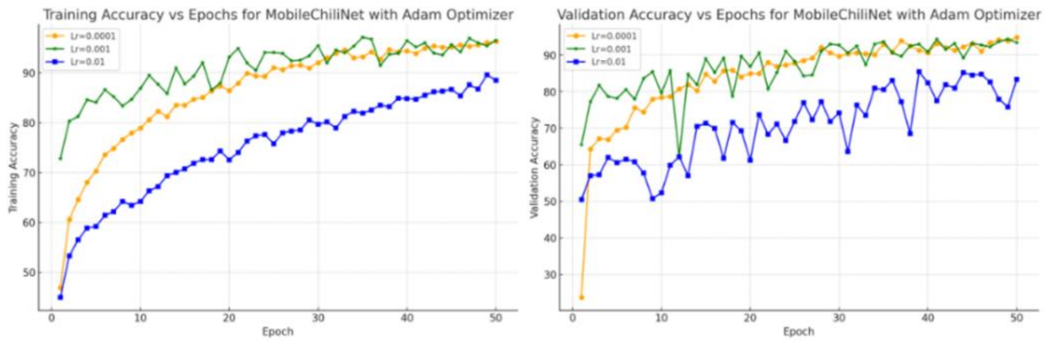


Figure 6. Training and validation accuracy of MobileChiliNet with Adam optimizer

The Adam optimizer also successfully improved the accuracy of MobileChiliNet. As shown in Figure 6, with a learning rate of 0.0001, Adam achieved a highest training accuracy of 96.30% and a highest validation accuracy of 94.84%. Although the graph shows slight fluctuations, the accuracy consistently increased with each epoch. Overall, Adam demonstrated stable performance during training. Next, an experiment was conducted using the SGD optimizer with Momentum 0.9 to compare performance, as shown in Figure 7.

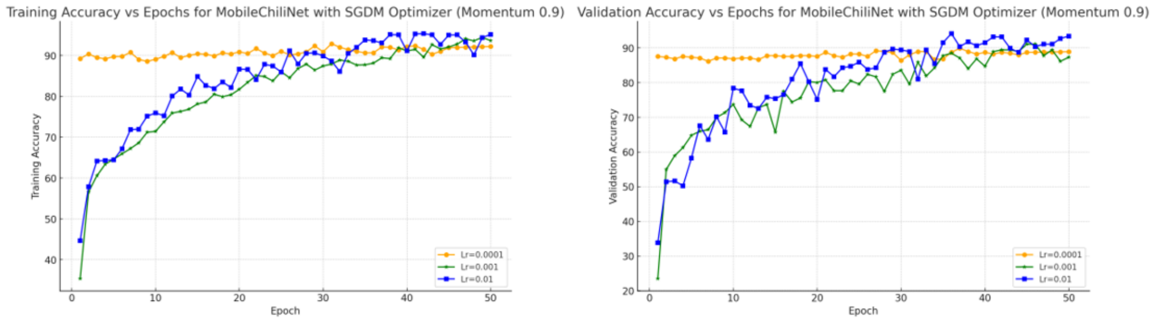


Figure 7. Training and validation accuracy of MobileChiliNet with SGD momentum 0.9

When MobileChiliNet used the SGD optimizer with momentum 0.9, the accuracy results were lower compared to using AdaGrad and Adam. As shown in Figure 7, with a learning rate of 0.01, SGD achieved a highest training accuracy of 95.36% and a highest validation accuracy of 94.13%. Although its performance was good, the accuracy remained lower than the previous optimizers. After changing the momentum to 0.99, as shown in Figure 8, the accuracy fluctuated with each epoch, but the results did not show improvement and even experienced a decline compared to the initial momentum.

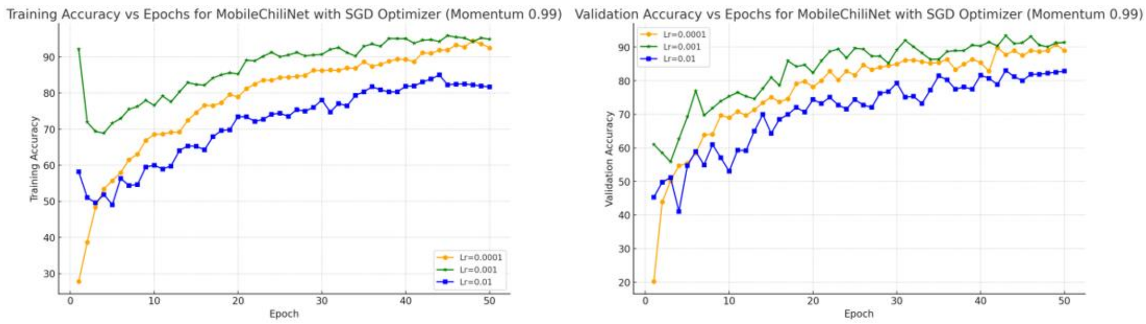


Figure 8. Training and validation accuracy of MobileChiliNet with SGD momentum 0.99

Figure 8 shows that changing the momentum in SGD from 0.9 to 0.99 did not significantly impact the accuracy improvement. With a learning rate of 0.001, the model achieved a highest training accuracy of 95.83% and a highest validation accuracy of 93.43%. The momentum change did not significantly enhance the model's performance and even resulted in a slight decrease in validation accuracy. Afterward, the experiment continued by switching the optimizer to AdamW, and the training results for this optimizer are shown in Figure 9.

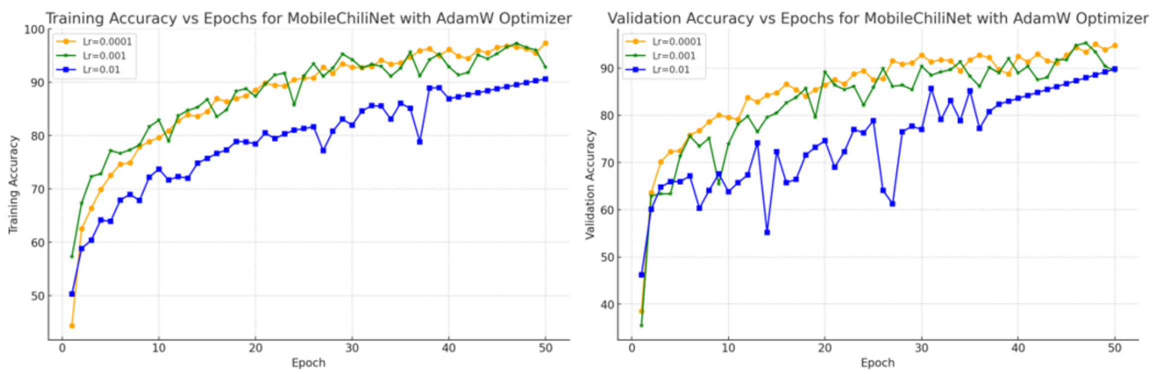


Figure 9. Training and validation accuracy of MobileChiliNet with AdamW

In Figure 9, the graph shows that using a learning rate of 0.001 with the AdamW optimizer provided more stable results and higher accuracy compared to other learning rate values. The highest training accuracy reached 97.30%, while the highest validation accuracy was 95.31%. Although the validation accuracy is comparable to the results obtained using AdaGrad, the performance with AdamW was more stable throughout the training process. Next, the experiment was continued by switching the optimizer to RMSProp, and the accuracy results for each epoch using RMSProp are shown in Figure 10.

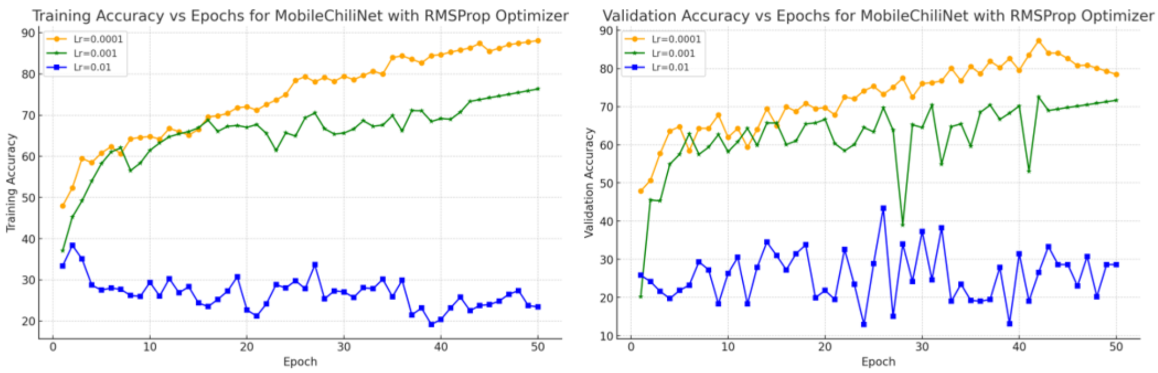


Figure 10. Training and validation accuracy of MobileChiliNet with RMSProp

As shown in Figure 10, the training and validation accuracy of MobileChiliNet when optimized using RMSProp was very low. The highest accuracy achieved with a learning rate of 0.0001 was 88.08% for training accuracy and 87.32% for validation accuracy. This indicates that RMSProp is not suitable for chili leaf disease classification. The results of all optimizer experiments, including AdaGrad, Adam, SGD, AdamW, and RMSProp, are presented in Table 7.

Table 7. Accuracy comparison of optimizer usage

Optimizer	Learning rate	Training accuracy (%)	Validation accuracy (%)
Adagrad	0.0001	64.28	63.85
	0.001	91.77	87.09
	0.01	99.18	95.31
Adam	0.0001	96.30	94.84
	0.001	97.12	94.37
	0.01	89.60	85.45
SGD Momentum=0.9	0.0001	92.89	89.91
	0.001	94.42	91.08
	0.01	95.36	94.13
SGD Momentum=0.99	0.0001	94.42	90.85
	0.001	95.83	93.43
	0.01	84.96	83.10
AdamW	0.0001	97.36	95.07
	0.001	97.30	95.31
	0.01	90.64	89.80
RMSProp	0.0001	88.08	87.32
	0.001	76.33	72.54
	0.01	38.37	43.43

From Table 7, it can be seen that the AdaGrad optimizer provided the best results for MobileChiliNet with a learning rate of 0.01, achieving a training accuracy of 99.18% and a validation accuracy of 95.31%. However, fluctuations caused instability during training. In contrast, AdamW offered better stability with the same validation result of 95.31%. On the other hand, RMSProp yielded much lower results, with unsatisfactory training and validation accuracy, indicating that this optimizer is less suitable for chili leaf disease classification in this model.

The observed fluctuations in optimizer performance can be attributed to differences in how each algorithm handles gradient updates. SGD relies on a fixed learning rate and simple momentum, which can lead to unstable convergence, especially when navigating noisy or complex loss surfaces. In contrast, AdamW combines adaptive learning rates with weight decay regularization, allowing it to adjust learning rates individually for each parameter and prevent overfitting through effective regularization. This makes AdamW more resilient to oscillations during training and better suited for deep architectures like MobileChiliNet. The stable performance of AdamW across epochs, as shown in the training curves, highlights its ability to converge more smoothly than SGD, which tends to be sensitive to learning rate and momentum settings.

3.2. Comparison of learning rate schedules

After testing MobileChiliNet with several optimizers, AdamW was selected as the more stable optimizer. To further improve the performance of MobileChiliNet, we implemented several learning rate schedules such as PolynomialDecay, CosineAnnealingLR, ExponentialLR, ReduceLROnPlateau, and CyclicLR [35], [36]. These schedules dynamically adjust the learning rate during training to enhance the accuracy and stability of the model. The accuracy results of MobileChiliNet using AdamW in combination with these learning rate schedulers are presented in Table 8, showing the performance variation of each approach.

Table 8. Comparison of learning rate scheduler usage

Learning rate scheduler	Training accuracy	Validation accuracy
PolynomialDecay	99.76	96.48
CosineAnnealingLR	99.65	96.24
ExponentialLR	98.82	95.30
StepLR	94.71	92.25
ReduceLROnPlateau	99.29	95.07
CyclicLR	80.55	81.45

Table 8 shows a comparison of the accuracy of MobileChiliNet with various learning rate schedulers. PolynomialDecay achieved the best accuracy with a training accuracy of 99.765% and a validation accuracy of 96.4789%, followed by CosineAnnealingLR with a validation accuracy of 96.2441%. ExponentialLR and ReduceLROnPlateau also performed well, but with slightly lower validation accuracy, at 95.3052% and 95.0704% respectively. StepLR produced lower accuracy compared to the other schedules, while CyclicLR gave the lowest result, showing unstable performance with a validation accuracy of only 81.4554%.

4. CONCLUSION

This research focuses on the urgent need for a more accurate and computationally efficient model for chili leaf disease classification. Existing models, such as ResNet50 and VGG16, have limitations in both accuracy and complexity, particularly in multi-class classification tasks in agriculture. MobileChiliNet is designed to address these issues by reducing computational load through pruning techniques and hyperparameter optimization, making it suitable for implementation in low-cost smart agricultural systems. With a significant accuracy improvement of up to 96.48% and better generalization capabilities, MobileChiliNet successfully addresses the challenges of achieving high classification performance in resource-constrained agricultural environments. This model can assist farmers in detecting diseases early, preventing disease spread, and minimizing crop losses. The implementation of MobileChiliNet also supports the achievement of sustainable agricultural practices in line with the SDGs. For future research, several directions are recommended to enhance the practical adoption and scientific contribution of this work: i) investigate the scalability and robustness of MobileChiliNet across different field conditions, crop sizes, and deployment environments; ii) extend the model for real-time deployment on edge devices, such as smartphones or IoT modules, to enable direct use in farming operations; iii) incorporate additional environmental factors into the dataset (e.g., lighting variation, leaf maturity, and camera resolution) to improve the model's robustness and generalizability; and iv) explore the applicability of MobileChiliNet to other crops or mixed-class datasets, which would validate its utility for broader agricultural disease diagnosis tasks.

ACKNOWLEDGMENT

The authors also express their gratitude to DRTPM and the Center for Research, Community Service, and International Publication (P3MPI) of Universitas Medan Area for their continuous support.

FUNDING INFORMATION

This work was supported by the Directorate of Research, Technology, and Community Service (DRTPM), Directorate General of Higher Education, Ministry of Education and Culture of the Republic of Indonesia, through the 2024 research grant under contract No. 103/E5/PG.02.00.PL/2024, along with derivative contracts 020/LL1/AL.04.03/2024 and 52/P3MPI/01.3.2/VI/2024.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sayuti Rahman	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Marischa Elveny	✓	✓		✓	✓	✓			✓	✓				✓
Marwan Ramli	✓	✓			✓				✓			✓		✓
Dionikxon Manurung		✓	✓		✓		✓	✓	✓				✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known financial, personal, or professional interests that could influence, or be perceived as potentially influencing, the work reported in this paper. This study was conducted independently, and the findings are solely intended to contribute to academic knowledge and the advancement of smart agriculture research.

INFORMED CONSENT

No human participants were involved in this study, and no personal or identifiable information was collected or analyzed. Therefore, informed consent was not required or applicable. The research focused exclusively on image datasets of chili leaves obtained from publicly available sources, ensuring that no ethical concerns related to human subjects were involved.

ETHICAL APPROVAL

This paper does not involve people or animals; no investigation has involved human subjects. Therefore, the authors did not seek approval from any institutional review board.

DATA AVAILABILITY

The dataset used in this study is openly available from Mendeley Data at https://doi.org/10.1016/j.dib.2024.110524, reference number [24]. Additional derived data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] O. Khalid *et al.*, "Comparative study of trust and reputation systems for wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 6, pp. 669–688, Jun. 2013, doi: 10.1002/sec.597.

[2] T. H.-Pérez, M. del R. G.-García, M. E. Valverde, and O. P.-López, "Capsicum annuum (hot pepper): an ancient Latin-American crop with outstanding bioactive compounds and nutraceutical potential. A review," *Comprehensive Reviews in Food Science and Food Safety*, vol. 19, no. 6, pp. 2972–2993, Nov. 2020, doi: 10.1111/1541-4337.12634.

[3] D. Sukmawati, E. Dasipah, and A. Nurdin, "Changes in subsidized fertilizer policy on factors of production and farm income of red chili (*Capsicum Annuum L*) in Cianjur Regency," *Greenation International Journal of Tourism and Management*, vol. 1, no. 3, pp. 246–252, Sep. 2023, doi: 10.38035/gijtm.v1i3.79.

[4] H. Sinaga, M. Nurminah, and O. Abira Rajagukguk, "The effect of utilizing palm oil brown sugar and red chili extract (*Capsicum annuum L.*) in the production of sweet and spicy palm kernel meal sauce," *Advances in Food Science, Sustainable Agriculture and Agroindustrial Engineering*, vol. 6, no. 3, pp. 267–279, Sep. 2023, doi: 10.21776/ub.afssaae.2023.006.03.6.

[5] S. Nazeer, T. T. R. Afzal, Sana, M. Saeed, S. Sharif, and M. Z.-Ul-haq, "Chili pepper," in *Essentials of Medicinal and Aromatic Crops*, Cham: Springer International Publishing, 2023, pp. 855–885, doi: 10.1007/978-3-031-35403-8_33.

[6] Y. I. Aprilia, N. Khuriyati, and A. C. Sukartiko, "Classification of chili powder (*Capsicum annuum L.*) antioxidant activity based on near infrared spectra," *Food Research*, vol. 5, no. S2, pp. 51–56, Jun. 2021, doi: 10.26656/fr.2017.5(S2).008.

[7] E. O. Fenibo, G. N. Ijoma, and T. Matambo, "Biopesticides in sustainable agriculture: a critical sustainable development driver governed by green chemistry principles," *Frontiers in Sustainable Food Systems*, vol. 5, Jun. 2021, doi: 10.3389/fsufs.2021.619058.

[8] P. R. Shingote *et al.*, "An overview of chili leaf curl disease: molecular mechanisms, impact, challenges, and disease management strategies in Indian subcontinent," *Frontiers in Microbiology*, vol. 13, Jun. 2022, doi: 10.3389/fmicb.2022.899512.

[9] M. T. Sundari, Darsono, J. Sutisno, and E. Antriyandarti, "Analysis of chili supply in Indonesia," *IOP Conference Series: Earth and Environmental Science*, vol. 1200, no. 1, Jun. 2023, doi: 10.1088/1755-1315/1200/1/012036.

[10] K. Nazari, M. J. Ebadi, and K. Berahmand, "Diagnosis of alternaria disease and leafminer pest on tomato leaves using image processing techniques," *Journal of the Science of Food and Agriculture*, vol. 102, no. 15, pp. 6907–6920, Dec. 2022, doi: 10.1002/jsfa.12052.

[11] M. Ramli, S. Rahman, and R. B. Syah, "ConciseCarNet: convolutional neural network for parking space classification," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 4, pp. 4158–4168, Dec. 2024, doi: 10.11591/ijai.v13.i4.pp4158-4168.

[12] H. I. Peyal *et al.*, "Plant disease classifier: detection of dual-crop diseases using lightweight 2D CNN architecture," *IEEE Access*, vol. 11, pp. 110627–110643, 2023, doi: 10.1109/ACCESS.2023.3320686.

[13] V. Singh, A. Chug, and A. P. Singh, "Classification of beans leaf diseases using fine tuned CNN model," *Procedia Computer Science*, vol. 218, pp. 348–356, 2022, doi: 10.1016/j.procs.2023.01.017.

[14] C. Öztürk, M. Taşyürek, and M. U. Türkdamar, "Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 4, Feb. 2023, doi: 10.1002/cpe.7542.

[15] R. Kumar, A. Chug, and A. P. Singh, "Plant leaf diseases severity estimation using fine-tuned CNN models," in *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*, Mar. 2023, pp. 1–6, doi: 10.1109/ISCON57294.2023.10111948.

[16] V. Prithviraj and S. Rajkumar, "Magnitude-based weight-pruned automated convolutional neural network to detect and classify the plant disease," in *Lecture Notes in Networks and Systems*, vol. 612, 2023, pp. 617–636, doi: 10.1007/978-981-19-9228-5_53.

[17] B. Wang, C. Zhang, Y. Li, C. Cao, D. Huang, and Y. Gong, "An ultra-lightweight efficient network for image-based plant disease and pest infection detection," *Precision Agriculture*, vol. 24, no. 5, pp. 1836–1861, Oct. 2023, doi: 10.1007/s11119-023-10020-0.

[18] M. J. Karim, M. O. F. Goni, M. Nahiduzzaman, M. Ahsan, J. Haider, and M. Kowalski, "Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM," *Scientific Reports*, vol. 14, no. 1, Jul. 2024, doi: 10.1038/s41598-024-66989-9.

[19] A. F. K. Hasbollah, Z. M. Zin, N. Ibrahim, and R. F. R. Suleiman, "Green chilli leaf disease detection using convolution neural networks," *Journal of Green Engineering*, vol. 10, no. 12, pp. 13005–13019, 2020.

[20] N. N. A. Loti, M. R. M. Noor, and S. W. Chang, "Integrated analysis of machine learning and deep learning in chili pest and disease identification," *Journal of the Science of Food and Agriculture*, vol. 101, no. 9, pp. 3582–3594, Jul. 2021, doi: 10.1002/jsfa.10987.

[21] A. Patil and K. Lad, "Feature selection for chili leaf disease identification using GLCM algorithm," in *Smart Innovation, Systems and Technologies*, vol. 251, 2022, pp. 359–365, doi: 10.1007/978-981-16-3945-6_35.

[22] C. J. Entuni, T. M. A. Zulcaffle, and K. H. Ping, "Classification of capsicum leaf disease from a complex cluster of leaves using an improved multiple layers ShuffleNet CNN model," *International Journal of Advanced Technology and Engineering Exploration*, vol. 10, no. 102, pp. 515–533, May 2023, doi: 10.19101/IJATEE.2022.10100509.

[23] Z. Gulzar, S. Chandu, and K. Ravi, "Classification and analysis of chilli plant disease detection using convolution neural networks," in *Lecture Notes in Networks and Systems*, vol. 798 LNNS, 2023, pp. 677–696, doi: 10.1007/978-981-99-7093-3_45.

[24] M. P. Aishwarya and A. P. Reddy, "Dataset of chilli and onion plant leaf images for classification and detection," *Data in Brief*, vol. 54, Jun. 2024, doi: 10.1016/j.dib.2024.110524.

[25] V. K. Pratap and N. S. Kumar, "High-precision multiclass classification of chili leaf disease through customized EffecientNetB4 from chili leaf images," *Smart Agricultural Technology*, vol. 5, Oct. 2023, doi: 10.1016/j.atech.2023.100295.

[26] S. A. Amjad, T. Anuradha, T. M. Datta, and U. M. Babu, "Chilli leaf disease detection using deep learning," in *Communications in Computer and Information Science*, vol. 2054 CCIS, 2024, pp. 81–89, doi: 10.1007/978-3-031-56703-2_7.

[27] Y. Gulzar, "Fruit image classification model based on MobileNetV2 with deep transfer learning technique," *Sustainability*, vol. 15, no. 3, Jan. 2023, doi: 10.3390/su15031906.

[28] S. Kumar, S. Pal, V. P. Singh, and P. Jaiswal, "Performance evaluation of ResNet model for classification of tomato plant disease," *Epidemiologic Methods*, vol. 12, no. 1, Jan. 2023, doi: 10.1515/em-2021-0044.

[29] P. K. Das, "Leaf disease classification in bell pepper plant using VGGNet," *Journal of Innovative Image Processing*, vol. 5, no. 1, pp. 36–46, Mar. 2023, doi: 10.36548/jiip.2023.1.003.

[30] R. Kursun, E. T. Yasin, and M. Koklu, "Classification of sugarcane leaf disease with AlexNet model," *Proceedings of International Conference on Intelligent Systems and New Applications*, Apr. 2024, doi: 10.58190/icisna.2024.86.

[31] Y. Zhou, C. Fu, Y. Zhai, J. Li, Z. Jin, and Y. Xu, "Identification of rice leaf disease using improved ShuffleNet V2," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 4501–4517, 2023, doi: 10.32604/cmc.2023.038446.

[32] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[33] B. Swiderski, S. Osowski, G. Gwardys, J. Kurek, M. Slowinska, and I. Lugowska, "Random CNN structure: tool to increase generalization ability in deep learning," *Eurasip Journal on Image and Video Processing*, vol. 2022, no. 1, p. 3, Dec. 2022, doi: 10.1186/s13640-022-00580-y.

[34] T. Chauhan, H. Palivela, and S. Tiwari, "Optimization and fine-tuning of DenseNet model for classification of COVID-19 cases in medical imaging," *International Journal of Information Management Data Insights*, vol. 1, no. 2, Nov. 2021, doi: 10.1016/j.jjimei.2021.100020.

[35] Y. Xiong, L. C. Lan, X. Chen, R. Wang, and C. J. Hsieh, "Learning to schedule learning rate with graph neural networks," *ICLR 2022-10th International Conference on Learning Representations*, 2022.

[36] J. Konar, P. Khandelwal, and R. Tripathi, "Comparison of various learning rate scheduling techniques on convolutional neural network," in *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2020*, Feb. 2020, pp. 1–5, doi: 10.1109/SCEECS48394.2020.94.

BIOGRAPHIES OF AUTHORS

Dr. Sayuti Rahman is lecturer at the University of Medan Area, has recently earned his Ph.D. degree from Syiah Kuala University in Banda Aceh. Throughout his teaching career, Sayuti has demonstrated a strong interest in exploring the domains of digital imagery and machine learning. As a result, his current focus revolves around research related to extracting information from digital images, particularly in the realm of Computer Vision. Python serves as the primary programming language extensively utilized in his research pursuits. Currently, Sayuti is actively involved in a research project dedicated to the development of an intelligent parking system, with a specific emphasis on optimizing convolutional neural networks (CNN) for effective parking space classification. He was born on June 18, 1987, in Bukit Gantung, situated in the Wampu District of Langkat Regency. He can be contacted at email: sayutirahman@staff.uma.ac.id.

Marischa Elveny received (bachelor's) degree in information technology at the Universitas Sumatera Utara and earned his M. Kom. (master) degree in Informatics Engineering, Universitas Sumatera Utara. Doctorate (Ph.D.) at the Universitas Sumatera Utara. Currently working as a lecturer in the Faculty of Computer Science and Information Technology at the Universitas Sumatera Utara. His research interests are artificial intelligence, data science and computational intelligence. She can be contacted via email: marischaelveny@usu.ac.id.

Prof. Marwan Ramli Marwan is a Professor of Applied Mathematics and an active professor and researcher at Department of Mathematics and Mathematics Graduated Study Program Universitas Syiah Kuala. Prof. Marwan was born in Tanjung Pinang, Kepulauan Riau, on November 25, 1971. He received his Ph.D. in Applied Mathematics from the Bandung Institute of Technology, specializing on Extreme Water Waves Generation. In 1999, he received his master's degree in mathematics from the Bandung Institute of Technology in Indonesia. In 1996, he received his BSc in Mathematics from Syiah Kuala University in Indonesia. He can be contacted at email: marwan.math@unsyiah.ac.id.

Dionikxon Manurung is an undergraduate student in Informatics at Universitas Medan Area. He has a strong interest in the fields of artificial intelligence and deep learning, with a focus on exploring intelligent systems and their applications in solving real-world problems. He is passionate about research and continuously seeks to expand his knowledge through academic and practical experiences in AI. He can be contacted at email: dionikxonmanurung@gmail.com.