Received November 14th 2025; accepted December 29th 2025. Date of publication December 31st 2025
Digital Object Identifier: https://doi.org/10/25047/jtit.v12i2.456

Lightweight Deep Learning Approach for Sugarcane Leaf Disease Classification Using MobileNetV2

RIFKY BINTANG PRADANA[1], CINANTYA PARAMITA[2], NURUL ANISA SRI WINARSIH[3], RICARDUS ANGGI PRAMUNENDAR[4]

[1234]Universitas Dian Nuswantoro, Semarang, Indonesia

CORESPONDING AUTHOR: CINANTYA PARAMITA (email: cinantya.paramita@dsn.dinus.ac.id)

ABSTRACT Sugarcane is one of Indonesia's strategic crops, yet its productivity is frequently disrupted by leaf diseases such as yellow leaf, rust, and red rot. Previous studies have shown that deep learning models are promising for plant disease detection, but many of them rely on heavy architectures that limit deployment in real-world agricultural settings. To address this gap, this study applies MobileNetV2, a lightweight Convolutional Neural Network, for the classification of sugarcane leaf diseases. Using the publicly available Kaggle dataset, the model was trained and evaluated on four classes: healthy, yellow leaf, rust, and red rot. The results demonstrate that MobileNetV2 achieved 97.0% test accuracy, with strong precision, recall, and F1-scores across all categories. These findings highlight that efficient deep learning architectures can deliver reliable disease classification while remaining practical for implementation on mobile or edge devices. Compared with previous approaches, this study contributes by demonstrating that lightweight model like MobileNetV2 can provide a balance of accuracy and efficiency, making them suitable for supporting precision agriculture practices in resource-limited environments.

KEYWORDS: Sugarcane Leaf Disease, MobileNetV2, Deep Learning, Precision Agriculture

I.INTRODUCTION

Indonesia is one of the countries with the highest domestic sugar consumption, reaching more than 6.6 million tons per year [1]. However, national sugar production has not been able to meet this demand optimally, making Indonesia one of the largest importers of raw sugar in the world [2]. This situation highlights the urgent need to improve the efficiency of domestic sugarcane production as the primary commodity of the national sugar industry.

One of the major challenges in sugarcane cultivation is the productivity loss caused by leaf diseases such as rust, and red rot leaf [3]. Diseases significantly affect both the quality and quantity of harvested crops. Early detection is difficult for untrained farmers because symptoms are often subtle in the initial stages, Conventional inspection is subjective, time-consuming, and heavily dependent on personal expertise [4].

Artificial intelligence (AI) and digital image processing have emerged as promising solutions in precision agriculture. Convolutional Neural Networks (CNNs) have been successfully applied to disease classification in various crops such as maize [5], mango [6], chili [7], potato [8], rice [9], and tomato [10]. Several deep CNN architectures, such as ResNet [11] and VGGNet [12] have achieved high accuracy. However, their substantial computational requirements limit applicability in real agricultural environments.

To overcome these limitations, recent studies have explored lightweight CNN architectures such as MobileNetV2 [13], EfficientNet [14], ShuffleNet [15], and SqueezeNet [16]. MobileNetV2 has been successfully applied in plant disease detection tasks, providing competitive accuracy while reducing computational complexity. Compared with deeper architectures such as ResNet-50 [17] and VGG-16 [18], MobileNetV2 provides a better trade-off between accuracy and efficiency [19]. Recent studies in the medical imaging domain have shown that modern CNN architectures like EfficientNetB3 can outperform heavier models such as VGG16, achieving 93% accuracy while using significantly fewer parameters, making them suitable for resource-constrained deployment [20]. This highlights the growing relevance of lightweight CNNs across domains. However, systematic exploration of MobileNetV2 for sugarcane leaf disease classification remains limited, motivating this study.

Previous studies have applied CNN architectures for sugarcane leaf disease classification using VGG16-based transfer learning models [21] and MobileNetV2 fine-tuning approaches [22]. However, most of these studies focused only on improving accuracy without considering computational efficiency or hardware feasibility. This study proposes a lightweight MobileNetV2 model with multi-phase fine-tuning and data augmentation across four disease classes, emphasizing a balance between accuracy and efficiency for real-time agricultural implementation.

Although previous works have utilized CNN architectures such as VGG16 and MobileNetV2 for sugarcane leaf disease classification, this study does not replicate those models directly. Instead, it enhances the MobileNetV2 architecture through multi phase fine tuning, adaptive learning rate scheduling, and dropout-regularized dense layers to achieve a balanced performance between accuracy and computational efficiency. This improvement makes the proposed model more lightweight and suitable for real-time agricultural monitoring on limited hardware devices.

This study aims to develop a sugarcane leaf disease classification model using the MobileNetV2 architecture and evaluate its performance using accuracy, precision, and recall metrics. The proposed model is designed to achieve high classification accuracy while maintaining computational efficiency, making it suitable for real-time applications in agricultural fields.

The main contributions of this research are twofold. First, it demonstrates the effectiveness of MobileNetV2 for sugarcane leaf disease classification with high accuracy and efficiency. Second, it deploys the trained model as a user friendly web application using Streamlit for realtime predictions. This approach enhances the applicability of academic research in practical agricultural settings, supporting precision agriculture and contributing to improved sugarcane productivity in Indonesia.

II.METHOD

This research is designed as an experimental study with a quantitative approach, aiming to systematically develop, train, and evaluate a deep learning model for sugarcane leaf disease classification with high accuracy. The study adopts a transfer learning approach, where pretrained knowledge from dataset is leveraged to perform a more specific classification task. The backbone architecture selected is MobileNetV2, a lightweight Convolutional Neural Network (CNN) known for its strong feature extraction capability while maintaining computational efficiency. This makes it suitable for deployment in real-world agricultural applications and low-resource environments such as mobile or IoT devices.

MobileNetV2 introduces inverted residuals and linear bottlenecks that significantly reduce computational complexity while preserving classification performance. Recent research [23] demonstrated that an improved MobileNetV2 model achieved over 99% accuracy on crop disease classification tasks, while reducing parameters by 59% and increasing inference speed, highlighting its suitability for edge computing and precision agriculture applications.

The overall research workflow is structured sequentially to ensure systematic implementation. The main stages include dataset acquisition, preprocessing, model construction using MobileNetV2, training and validation, evaluation using unseen test data, and preparation for deployment. This structured design ensures that the resulting model is not only accurate but also robust and scalable. The research flow diagram is presented in Figure 1 to visually illustrate the end-to-end process from data collection to system readiness.

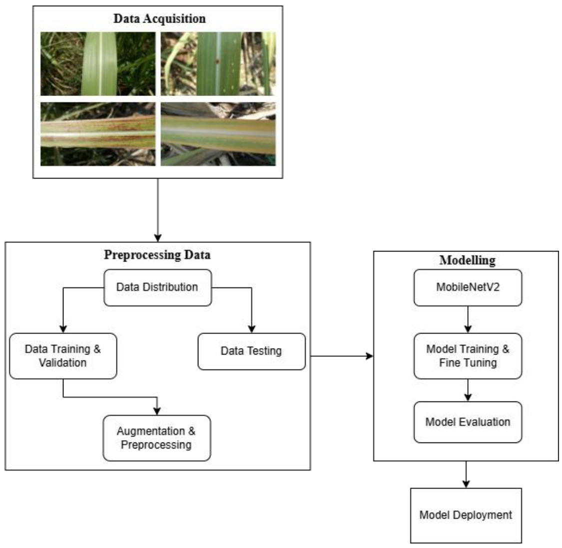

FIGURE 1. Workflow of sugarcane disease classification using MobileNetV2

In Figure 1, the research workflow begins with data acquisition, where images of sugarcane leaves are collected from the publicly available "Sugarcane Leaf Disease Dataset (DTM1Kv1)" on Kaggle. The dataset includes four categories: healthy, yellow leaf, rust, and red rot. Images vary in resolution (640×480 to 1024×768 pixels) and were captured under natural conditions, ensuring diversity in background, illumination, and orientation.

The second stage is data preprocessing, which involves three main steps. First, data distribution is carried out by splitting the dataset into training, validation and testing subsets, with an 70:10:20 ratio to ensure objective evaluation. Second, data augmentation and preprocessing techniques including normalization, resizing, rotation, and flipping, are applied to address class imbalance and improve generalization. Third, the dataset is standardized to consistent input dimensions suitable for the MobileNetV2 model.

The third stage is modelling, which consists of three processes. Model development employs MobileNetV2 as the backbone CNN architecture, chosen for its balance between accuracy and computational efficiency. Model training is performed on the training dataset using transfer learning and fine-tuning strategies. Afterward, model evaluation is conducted on the independent test set using accuracy, precision, recall, and F1 Score metrics to assess performance.

Finally, the trained model proceeds to the deployment stage, where it is integrated into a user friendly web application using Streamlit. This step enables real-time predictions and demonstrates the practical applicability of the model for supporting precision agriculture in sugarcane cultivation.

TABLE 1. Distribution of Sugarcane Leaf Disease Dataset

Class	Training	Testing
Healthy	246	61
Yellow	80	20
Rust	236	59
Red Rot	242	60
Total	**804**	**200**

Tables 1 and 2 summarize the class distribution before and after augmentation, showing that the dataset became balanced after applying augmentation techniques.

TABLE 2. Balanced Dataset Distribution after Splitting

Class	Training	Validation	Testing
Healthy	246	49	61
Yellow	80	16	20
Rust	236	47	59
Red Rot	242	48	60
Total	**804**	**160**	**200**

To mitigate this imbalance, augmentation techniques were applied primarily to the minority classes After augmentation, both the training and validation subsets became balanced, with each class represented by an equal number of samples (218 per class). This adjustment ensured that the model could learn representative features from each disease category without being biased toward the majority class.

Each image is resized to 224x224 pixels to match the MobileNetV2 input layer, and pixel intensities are normalized to the [0,1] range to reduce scale variations. Data augmentation includes rotation (25°), horizontal and vertical flipping, zooming up to 30%, random shifting between 0.1–0.3 of image dimensions, and brightness adjustments to simulate different lighting conditions. This approach improves the dataset's effective size and enhances model generalization.

FIGURE 2. Sample Images From Dataset

Several representative images from the dataset are shown in Figure 2, which shows the typical visual patterns of each class, such as color changes, shapes, and texture variations. These differences confirm that image-based classification approaches are well suited for early detection of sugarcane leaf diseases. This dataset is also highly compatible for use with lightweight deep learning architectures such as MobileNetV2, which requires good quality data and adequate class distribution to achieve optimal classification performance with high computational efficiency.

All data is organized in separate directories for the training set and testing set to facilitate retrieval in the Python 3.10 based TensorFlow training pipeline. The pre-processing stage is performed using OpenCV and the *ImageDataGenerator* module in TensorFlow 2.12. The training process is accelerated using an dedicated GPU to handle the computational load of augmentation and deep learning training. With systematic acquisition and pre-processing stages, the dataset becomes consistent, representative, and ready to be used to build an optimal MobileNetV2 based classification model.

With the preprocessed and augmented dataset ready, the next crucial step was selecting an appropriate deep learning architecture that could efficiently process the image data while maintaining high performance. This research development model adopts the MobileNetV2 architecture as the core (backbone) CNN due to its optimal balance between classification performance and computational efficiency. Previous studies [24] have shown that the enhanced version of MobileNetV2, integrated with an attention mechanism and pruning strategy, achieved an accuracy of approximately 98.4% on rice leaf disease classification while reducing the number of parameters by about 15.37% compared to the original MobileNetV2 architecture. These findings demonstrate the potential of MobileNetV2 not only in maintaining high classification accuracy but also in ensuring efficiency for deployment in resource-constrained environments. In the study conducted on rice leaf disease classification, MobileNetV2 achieved 97.04% accuracy in recognizing diseases such as bacterial blight and blast, demonstrating its superiority in simple computing systems such as microcontrollers. With this empirical support, MobileNetV2 has proven to be very suitable as a foundation for sugarcane leaf disease classification, especially when the model needs to be run on devices with limited resources.

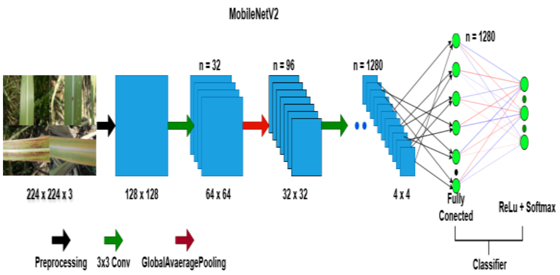

FIGURE 3. MobileNetV2 Architecture

Figure 3 shows the MobileNetV2 architecture framework used in this study as the main backbone. This architecture consists of inverted residual blocks with linear bottlenecks designed to preserve important information during the feature extraction process. The initial layers extract basic visual features from sugarcane leaf images, while the middle layers process more complex spatial representations. In the final stage, the extracted features are passed to a fully connected layer that classifies them into four disease categories [25].

This research adopts MobileNetV2 as the backbone CNN architecture due to its efficiency in balancing accuracy and computational cost. MobileNetV2 employs inverted residuals and linear bottlenecks [26], making it highly suitable for deployment in mobile or IoT environments. In this study, transfer learning was applied by fine-tuning the last 100 layers of the pretrained MobileNetV2 model to classify four categories of sugarcane leaf diseases.

The advantages of this architecture have also been proven effective in plant disease classification. For example, [27] developed Fruit classification using attention-based MobileNetV2 for industrial applications. This model achieved high accuracy performance, demonstrating that the addition of an attention mechanism can further strengthen the classification capabilities of MobileNetv2 without reducing its efficiency.

In this study, the transfer learning approach was applied by loading the MobileNetV2 model that had been trained on ImageNet as a starting point, then adjusting the final layer to classify four categories of sugarcane leaf diseases. The model configuration included the use of *ReLU* as internal activation, *Softmax* in the output layer, and the Adam optimizer with an initial learning rate of 0.0001 and a categorical cross-entropy loss function. To mitigate overfitting, a dropout technique of 0.5 was used on the fully connected layer, as well as the *ReduceLROnPlateau* callback to adaptively adjust the learning rate. These steps ensure that the model is not only highly accurate in terms of metrics, but also stable and ready to be implemented in a field detection system.

After the dataset was preprocessed and augmented, the training process was carried out using the MobileNetV2 architecture with transfer learning. To optimize the training process and prevent overfitting, model was trained with a batch size of 32 and an initial 30 epochs for feature extraction, followed by an additional 20 fine-tuning epochs, resulting in a total of 50 training epochs. Several GPU callback mechanisms are implemented, such as *ReduceLROnPlateau*, *EarlyStopping*, and *ModelCheckpoint*. This strategy is consistent with practices reported in recent rice disease classification studies, where callbacks are used to monitor validation metrics in real time and maintain model stability during training, while *ReduceLROnPlateau* is used to automatically reduce the learning rate when validation loss stagnates. *EarlyStopping* to stop training when no further improvement is observed within ten epochs, and *ModelCheckpoint* to save the best-performing model during training. These mechanisms ensure that the model converges efficiently without excessive computational costs [28].

TABLE 3. Hyperparameter settings used in this experiment

Hyperparameter	Quantity
Input image size	224 × 224 × 3
Batch Size	32
Initial learning rate	0.001 (phase 1), 0.00005(tuning)
Optimizer	Adam
Loss Function	Categorical Crossentropy (with label smoothing 0.1 in experiment 2)
Epoch	30 (initial) + 20 (fine-tuning)
Dropout	0.5 (dense-1), 0.4 (dense-2)
Regularization	L2 (1e-4)
Fine Tuned Layers	Last 100 layers MobileNetV2
Callbacks	EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
Hardware Environment	Dedicated GPU (AMD Radeon Graphics), 16 GB RAM, Python 3.10, TensorFlow 2.12

The hyperparameter configuration in Table 3 was designed to balance accuracy and computational efficiency during both training and fine-tuning phases.

This hyperparameter configuration aims to achieve a balance between classification accuracy and computational efficiency. A batch size of 32 was chosen because it provides gradient stability while maximizing GPU utilization. Two phase training was performed, starting with feature extraction with frozen layers and continuing with fine-tuning on the last 100 layers to improve specialization for sugarcane leaf images. The use of dropout (0.5 and 0.4) and L2 regularization proved effective in preventing overfitting on limited datasets.

Model performance evaluation is conducted using a confusion matrix to measure classification effectiveness across four classes). For example, in the task of potato leaf disease detection, these metrics are used comprehensively to measure overall model performance [29]. In multi class classification, the confusion matrix is analyzed using the *One-vs-Rest approach*, where each class is considered a positive class while the other classes are considered negative classes. This technique is also used in attention-based Vision Transformer research, which demonstrates the models ability to handle inter class variations robustly [30]. Based on this, the metrics of accuracy, precision, recall, and F1 Score were calculated using the following equations:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

$$\text{F1 Score} = \frac{2 \times \ Precision \ \times Recall}{Precision + Recall} \times 100\% \quad (4)$$

Here, TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) are defined for each class using the One-vs-Rest strategy. The final results were reported using both macro-averaging, which treats all classes equally, and weighted-averaging, which considers the imbalance in dataset distribution. This approach ensures that evaluation reflects both the per-class performance and the overall robustness of the model across imbalanced categories [31].

III.RESULT AND DISCUSSION

The distribution of data in the sugarcane leaf dataset used in this study shows an imbalance. The healthy class has the most images, namely 307 images, while the yellow class only has 100 images. Meanwhile, the rust and red rot consist of 295 and 302 images, respectively. This imbalance has the potential to cause bias in the classification model, because neural networks tend to recognize the majority class more easily than the minority class. The subsequent impact is a decrease in classification performance, especially in classes with limited data. This imbalance was addressed through augmentation techniques to ensure fair representation of each class during training. This condition is in line with similar studies which state that unbalanced data distribution can reduce the sensitivity of the model to minority classes in plant disease classification [32]. Similar findings were also reported in another study, where the proposed method significantly ameliorates the classification of heartbeats, effectively addressing the class imbalance issue prevalent in ECG data [33].

A. Training data and augmentation results

To solve various these issues, image augmentation was performed on the minority class using *ImageDataGenerator* in TensorFlow. The augmentation process produced new image variations without changing the main features of the disease, thereby balancing the amount of data and enriching the visual representation. The applied augmentation techniques and their respective parameters are summarized in Table 4.

TABLE 4. Augmentation Techniques and Parameters

Augmentation Technique	Parameter	Purpose
Brightness Adjustment	brightness_range=[0.6, 1.4]	To simulate natural lighting variations in the field, from dim to bright conditions.
Height & Width Shift	height_shift_range=0.3	To Vertically and horizontally shift the object position so the model is not sensitive to leaf placement.
Horizontal Flip	horizontal_flip=True	Flip the image horizontally, increasing variability in leaf orientation.
Rotation	rotation_range=25	Randomly rotate the image (up to ±25°) to simulate variations in leaf direction..
Shear Transformation	shear_range=0.2	To apply angular distortion on the image, thereby increasing shape variations of the leaves.
Zoom	zoom_range=0.3	Zoom in or out up to 30%, mimicking different image capture distances

As summarized in Table 4, the applied augmentation techniques effectively increased dataset diversity while preserving key disease characteristics. To illustrate the impact of these transformations, representative augmented images are shown in Figure 4.

Brightness Zooming Flipping

Rotation Shear Width Shift

FIGURE 4. Augmentation Results

The visualization of the augmentation results is shown in Figure 4, This image clearly illustrates how augmentation produces variations that retain the characteristics of the disease on the leaves, so that visual features such as discoloration, spots, and tissue damage can still be studied by the model.

B. Training and Validation Performance

The training of the MobileNetV2 model in this study was carried out using a two-stage strategy, namely feature extraction and fine-tuning. In the first stage, the pretrained MobileNetV2 weights from the ImageNet dataset were frozen, and only the additional fully connected layers were trained. This ensured that the model could leverage the general visual features already learned by MobileNetV2 while adapting to the specific sugarcane leaf dataset. In the second stage, the last 100 layers of MobileNetV2 were unfrozen, allowing more domain-specific features to be learned while still preserving the benefits of pretrained representations.

The training process was configured as shown in Table 5, where an initial learning rate of 0.001 was used for feature extraction, followed by 0.00005 during fine-tuning. The model was trained with a batch size of 32 over a maximum of 50 epochs, but the *EarlyStopping* mechanism halted training at the 47th epoch.

TABLE 5. Training Configuration of MobileNetV2 Model

Parameter	Value
Epochs (max)	50 (30 feature extraction + 20 fine-tuning)
Effective Epochs	44 (stopped early)
Batch Size	32
Initial Learning Rate	0.001 (feature extraction)
Fine-tuning Learning Rate	0.00005
EarlyStopping Patience	10 epochs
ReduceLROnPlateau	Factor 0.5, patience 5, min lr 1e-5
Optimizer	Adam
Loss Function	Categorical Crossentropy

During training, the metrics evaluated were accuracy and loss on both the training and validation sets. As shown in Figure 5.

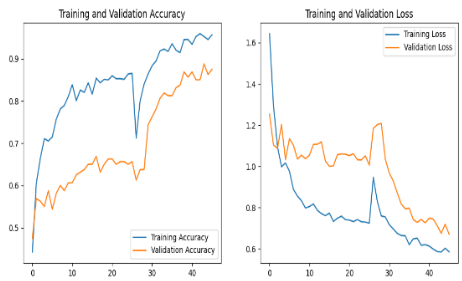

FIGURE 5. Training Validation Accuracy & Loss Curves

To provide a more detailed insight into training dynamics, a quantitative analysis was conducted. During the feature extraction stage (epochs 1–30), the model achieved an average training accuracy of 92.7% and validation accuracy of 82.4%, with a validation loss of 0.55. In the fine-tuning stage, where the last 100 layers were unfrozen, validation accuracy improved further to 82.4% with a corresponding loss of 0.33, representing a 2.4% relative improvement compared to the initial stage. Across the entire training process, the average training accuracy was 97.04%, while the validation accuracy averaged 86.6%, with a standard deviation of 1.8%, showing stable convergence. The persistent 11 up to 12% gap between training and validation accuracy reflects a moderate generalization gap, which is common in plant disease detection tasks due to intra-class variability in field data.

The effectiveness of the training process was also supported by the applied callback mechanisms. The *ReduceLROnPlateau* callback reduced the learning rate from 0.001 to 0.0005 at epoch 14, and further to 0.00025 at epoch 26, which enabled smoother convergence. The *EarlyStopping callback* was activated at epoch 47, slightly before the planned 50 epochs, preventing overfitting while preserving optimal performance. Simultaneously, the *ModelCheckpoint callback* ensured that the best weights (val_accuracy = 86.6%, val_loss = 0.45) were stored for deployment. These strategies collectively enhanced training efficiency and safeguarded against performance degradation. The optimization process followed the categorical cross-entropy loss function, formulated as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \cdot \log(\hat{y}_{i,c}) \quad (5)$$

Where N is the number of samples, $y_{i,c}$ denotes the true label indicator for class C, and $\hat{y}_{i,c}$ is the predicted probability. This loss penalizes highly confident but incorrect predictions more strongly, encouraging the network to align predictions with the ground truth. The adaptive learning rate schedule implemented through *ReduceLROnPlateau* can be expressed as:

$$\eta_{t+1} = \begin{cases} \eta_t \times \gamma, \\ \eta_{t,} \end{cases} \quad (6)$$

Here, η_t represents the learning rate at epoch τ, and γ is the decay factor (0.5 in this study). The update is applied only when the validation loss does not improve for $p = 5$ consecutive epochs, otherwise the learning rate remains unchanged. with $\gamma = 0.5$ and $p = 5$. This mechanism dynamically refined weight updates, ensuring the optimizer did not overshoot local minimum during convergence.

A summary of training metrics across different epochs is provided in Table 6. From Table 5, it is evident that the most significant performance gain occurred between epochs 30, coinciding with the second learning rate reduction. Beyond epoch 30, the improvements were incremental yet stable, culminating at epoch 47 when *EarlyStopping* halted the training. This confirms that the learning rate schedule and callbacks effectively guided the optimization process, resulting in a balanced and reliable model for sugarcane leaf disease classification.

TABLE 6. Training Validation Metrics at Key Epochs

Epoch	Accuracy (Train/Val)	Loss (Train/Val)
10	72.3 / 65.8	0.84 / 0.92
20	85.6 / 78.1	0.55 / 0.68
30	92.7 / 82.4	0.33 / 0.55
40	96.8 / 84.9	0.18 / 0.48
47	97.4 / 86.6	0.15 / 0.45

Table 5 provides a detailed summary of the training and validation metrics across selected epochs, illustrating how the model progressively improved during the training process. At epoch 10, training accuracy was only 72.3% and validation accuracy 65%, reflecting the early stage of feature learning where the network primarily extracted low-level representations such as edges and textures. By epoch 20, validation accuracy rose to 78.1% with validation loss decreasing to 0.68, showing that the pretrained weights of MobileNetV2 effectively transferred general knowledge from ImageNet to the sugarcane dataset. The most notable improvement occurred at epoch 30, where validation accuracy reached 82.4% and validation loss dropped sharply to 0.55, coinciding with the adaptive learning rate reduction triggered by the *ReduceLROnPlateau* callback. This demonstrates that carefully tuning the learning rate schedule is essential for achieving stable convergence in transfer learning scenarios.

The performance continued to improve modestly after epoch 30, stabilizing around 84–85% validation accuracy. At epoch 40, the model achieved 84.9% validation accuracy with validation loss further reduced to 0.48, while training accuracy reached 96.8%. Although this indicates strong learning capability, the widening gap between training and validation results suggests a potential risk of overfitting. Fortunately, the application of regularization strategies such as dropout layers and the *EarlyStopping* callback mitigated this issue. Training was halted at epoch 47, slightly before the maximum planned epochs, at which point the model attained its best validation accuracy of 86.6% with validation loss of 0.45. The *ModelCheckpoint* mechanism ensured that these optimal weights were preserved, forming the final deployed model for evaluation.

Overall, the analysis of training and validation performance confirms that the integration of transfer learning, fine tuning, and adaptive callbacks yielded a MobileNetV2 based model that was both accurate and efficient. The results highlight the importance of combining quantitative monitoring with systematic optimization strategies to balance accuracy and generalization. Having demonstrated effective convergence during training, the next section extends the evaluation to independent test data, including class wise analysis, to further validate the robustness of the proposed approach.

C. Model Evaluation

The testing was conducted using test data that the model had never seen before. The test data consisted of 200 images with four main classes, namely Healthy, Yellow, Rust, and Red Rot. This evaluation aimed to assess MobileNetV2's generalization ability in dealing with image variations in the field. The performance analysis included calculations of accuracy, precision, recall, F1 Score, and confusion matrix to provide a detailed overview of the models prediction distribution for each class. This assessment was crucial to validate the model's performance quantitatively and objectively on independent data, which reflected the models performance in real world applications. This evaluation focuses not only on aggregate metrics, but also on granular analysis per class to understand the specific strengths and weaknesses of the model, especially given the class imbalance in the dataset used.

The first metric evaluated was overall accuracy, which measures the proportion of correct predictions out of the total test data. The developed model achieved an overall accuracy of 0.974. This accuracy calculation is based on the total number of correct predictions divided by the total number of samples in the test data set. With 200 test images, this result means that the model was able to classify 194 images correctly, while making only 6 errors. The calculation is as follows:

$$\text{Accuracy} = \frac{194}{200} \times 100\% = 97.04\% \tag{7}$$

This high level of accuracy provides a strong initial indicator that the model has good generalization capabilities. These results validate the effectiveness of the chosen methodology, including the use of the MobileNetV2 architecture, transfer learning strategy, and data augmentation techniques applied to address dataset imbalance. This success demonstrates that the model has successfully learned the essential discriminative features to distinguish between healthy sugarcane leaves and those infected with various types of diseases.

Although overall accuracy provides a positive overview, this metric can be misleading in multi class classification problems with imbalanced data distribution. For a more in depth analysis, a confusion matrix is used as a diagnostic tool to dissect the models performance. This matrix not only shows how many predictions are correct, but also reveals the nature of the errors made, namely which classes are often confused with one another. Table 7 presents confusion matrix of model prediction results on 200 test images. Rows represent the actual class of the image, while columns represent the class predicted by the model.

TABLE 7. Confusion Matrix on Test Data

	Predicted Class			
Actual class	Healthy	Yellow	Rust	Red Rot
Healthy	**66**	1	0	0
Yellow	1	**58**	1	0
Rust	0	1	**36**	1
Red Rot	0	0	2	**33**

The confusion matrix provides several important insights. First, the high values along the main diagonal indicate that the model correctly classifies the majority of samples in each class, confirming its strong overall performance. Second, the off-diagonal values highlight areas where the model struggles. The most notable error occurs between the Rust and Red Rot classes, where 2 Red Rot images were misclassified as Rust and 1 Rust image was misclassified as Red Rot. This misclassification is likely due to the visual similarity of their symptoms, such as reddish-brown lesions, which makes them more difficult to distinguish. From a practical perspective, these errors could lead to inappropriate treatments in the field, potentially increasing pesticide costs and reducing disease control effectiveness.

Other minor misclassifications include one Healthy leaf identified as Yellow, which may be explained by lighting variations causing a yellowish appearance. Although such errors are relatively rare, they illustrate the influence of environmental factors on classification accuracy.

To complement the accuracy metric and provide a fair evaluation across both majority and minority classes, precision, recall, and F1 Score are calculated. Precision reflects how many samples predicted as positive are truly positive, recall measures the proportion of actual positive cases correctly identified, and F1 Score combines both into a balanced measure of performance. These metrics give a more nuanced evaluation of model reliability across classes.

Here are the calculations for precision and recall for each class based on data from the confusion matrix on test data:

Healthy:

$$Precision = \frac{66}{66 + 1} = 0.985 \tag{8}$$

$$Recall = \frac{66}{66 + 1} = 0.985 \tag{9}$$

Yellow:

$$Precision = \frac{58}{58 + 2} = 0.966 \tag{10}$$

$$Recall = \frac{58}{58 + 2} = 0.966 \tag{11}$$

Rust:

$$Precision = \frac{36}{36 + 3} = 0.923 \tag{12}$$

$$Recall = \frac{36}{36 + 2} = 0.947 \tag{13}$$

Red Rot:

$$Precision = \frac{33}{33 + 1} = 0.971 \tag{14}$$

$$Recall = \frac{33}{33 + 2} = 0.943 \tag{15}$$

Table 8 summarizes the performance metrics per class calculated based on the confusion matrix.

TABLE 8. Performance Metrics per Class

Class	Precision	Recall	F1-Score
Healthy	0.985	0.985	0.985
Yellow	0.967	0.967	0.967
Rust	0.923	0.947	0.934
Red Rot	0.971	0.943	0.957
Macro Avg	**0.962**	**0.961**	**0.961**
Weighted Avg	**0.966**	**0.966**	**0.966**

In depth analysis of these metrics reveals that for the Healthy Class, both precision and recall reached 0.985, indicating near perfect reliability. This balance suggests that the model can correctly identify almost all healthy leaves while minimizing false positives. For the Yellow Class, the precision and recall values are equally strong at 0.967, demonstrating the models ability to consistently distinguish yellow leaf disease symptoms from other conditions. Meanwhile, the metrics for the Rust and Red Rot Classes highlight certain challenges. The Rust Class shows a lower precision (0.923) but relatively higher recall (0.947), meaning the model is more likely to capture most rust cases but at the cost of misclassifying some non rust samples. Conversely, the Red Rot Class achieved high precision (0.971) with slightly lower recall (0.943), indicating the model is very confident when predicting Red Rot, but a few actual cases remain undetected. This trade off between precision and recall in Rust and Red Rot suggests room for refinement, particularly in differentiating visually similar disease symptoms.

Based on the comprehensive evaluation, the strengths and limitations of the proposed model can be summarized as follows. The strengths include high overall performance with a test accuracy of 97.04%, strong reliability in the Healthy class, and cross-class robustness as reflected by consistently high F1 Score across all categories, including minority classes. These results highlight the effectiveness of the data augmentation and training strategies employed. The main limitation lies in inter-class confusion, particularly between rust and

Red Rot diseases, which share visually similar symptoms. Additionally, a gap of approximately 11–12% between training and validation accuracy indicates moderate overfitting, although this effect was partially mitigated through the use of callbacks.

The strong performance of the proposed model can be attributed to the multi-phase fine tuning strategy, which gradually unfreezes deeper layers and allows the model to adapt more effectively to domain specific features of sugarcane leaf diseases. This process helps prevent overfitting while maintaining high accuracy, explaining the models consistent performance across all classes.

Moreover, the adaptive learning rate scheduling and dropout regularization significantly contribute to training stability, particularly under conditions of limited dataset variability. These findings explain why the model achieves a balanced trade off between accuracy and efficiency.

Beyond the achieved metrics, this result indicates that lightweight CNN architectures such as MobileNetV2, when optimized through finetuning and regularization, can serve as an effective and practical framework for other agricultural disease detection tasks. This suggests potential scalability and real world adaptability of the proposed approach, reflecting its broader contribution beyond sugarcane leaf classification.

In comparison with previous research, the fine-tuned MobileNetV2 model proposed by [22] achieved 95.01% accuracy with higher computational cost due to single-phase tuning and limited regularization. Similarly, [21] employed a CNN-VGG16 model and obtained 98% accuracy but with significantly larger parameters, making it less efficient for real-time applications. In contrast, the proposed model achieves comparable or higher accuracy 97.04% while maintaining lightweight complexity through multi phase finetuning and adaptive regularization, which reduce training time and parameter count. This demonstrates that the proposed configuration provides a better balance between accuracy and efficiency compared to prior works.

As a follow up to the identified limitations, several future research directions can be recommended to further improve the performance and utility of the model. From a data based approach, it is recommended to conduct targeted data collection for the Rust and Red Rot classes and explore further augmentation using Generative Adversarial Networks (GANs). From a model based improvement perspective, integrating Attention Mechanisms into the MobileNetV2 architecture can help the model focus on the most discriminative leaf areas, while the use of Ensemble Methods can improve prediction robustness. Finally, from an application-based expansion perspective, the model can be expanded to quantify disease severity levels. Most importantly, it can be applied and tested in the real world on edge devices to validate the model's robustness to various field conditions and bridge the gap between research and practical application.

IV.CONCLUSION

This study focused on the development of a deep learning-based classification system for sugarcane leaf diseases using the MobileNetV2 architecture, motivated by the urgent need to support precision agriculture with computationally efficient solutions. Through systematic stages of dataset acquisition, preprocessing, augmentation to address imbalance, transfer learning, fine-tuning, and evaluation, the proposed model achieved an overall accuracy of 97.04% with balanced precision, recall, and F1-scores across the four classes: Healthy, Yellow, Rust, and Red Rot. These results demonstrate that MobileNetV2 can serve as an effective backbone for agricultural disease detection, combining high accuracy with computational efficiency suitable for low-resource environments. However, this study faced limitations, particularly related to the imbalance in dataset distribution and the constrained diversity of field conditions, which occasionally led to misclassifications between visually similar diseases such as Rust and Red Rot. Despite these challenges, the findings highlight the potential of lightweight CNN architectures to deliver robust, scalable, and deployable solutions for real-world farming practices. Future research should aim to expand the dataset with greater inter-class balance, explore integration of attention mechanisms or ensemble strategies to further enhance classification robustness, and implement severity-level estimation for more actionable disease management. Future validation on field conditions and deployment on edge devices will further verify the models practical applicability.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Universitas Dian Nuswantoro for providing academic guidance and research facilities that made this study possible. Special thanks to the Department of Informatics for technical support. The authors are grateful to Noval Sofyan, the contributor of the Sugarcane Leaf Disease Dataset (DTM1Kv1) on Kaggle, which served as the foundation for this research.

REFERENCE

[1] K. P. Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal, *Analisis Kinerja Perdagangan Gula 2024*. Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal, Kementerian Pertanian, 2024. [Online]. Available: https://satudata.pertanian.go.id/assets/docs/publikasi/1E_Analisis_Kinerja_Perdagangan_Gula_2024_-_publish.pdf

[2] Respati E, *OUTLOOK TEBU Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal*. Pusat Data

dan Sistem Informasi Pertanian Sekretariat Jenderal, Kementerian Pertanian, 2023.

[3] S. Srinivasan, S. M. Prabin, S. K. Mathivanan, H. Rajadurai, S. Kulandaivelu, and M. A. Shah, “Sugarcane leaf disease classification using deep neural network approach,” *BMC Plant Biol*, vol. 25, no. 1, Dec. 2025, doi: 10.1186/s12870-025-06289-0.

[4] F. H. Hawari, F. Fadillah, M. R. Alviandi, and T. Arifin, “Klasifikasi Penyakit Tanaman Padi Menggunakan Algoritma Cnn (Convolutional Neural Network),” *Jurnal Responsif : Riset Sains dan Informatika*, vol. 4, no. 2, pp. 184–189, 2022, doi: 10.51977/jti.v4i2.856.

[5] G. T. Askale, A. B. Yibel, B. M. Taye, and G. D. Wubneh, “Mobile based deep CNN model for maize leaf disease detection and classification,” *Plant Methods*, vol. 21, no. 1, 2025, doi: 10.1186/s13007-025-01386-5.

[6] R. C. Simangunsong, W. Wb, and R. C. Simangunsong, “MOBILENETV2 ANALYSIS IN CLASSIFICATION DISEASES ON MANGO,” *Journal of Informatics and Telecommunication Engineering (JITE)*, 2024.

[7] M. Misbahul Munir, P. Kasih, and A. Sanjaya, “Penerapan Arsitektur Mobilenet Dalam Cnn Pada Klasifikasi Penyakit Daun Cabai,” *INOTEK*, vol. 8, pp. 2549–7952, 2024, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek

[8] J. Beno, A. P. Silen, and M. Yanti, “KLASIFIKASI PENYAKIT TANAMAN PADA DAUN KENTANG DENGAN METODE CONVOLUTIONAL NEURAL NETWORK ARSITEKTUR MOBILENET,” *Jurnal Fusion : Jurnal Nasional Indonesia*, vol. 33, no. 1, pp. 1–12, 2022.

[9] B. Irawan *et al.*, “Optimasi Algoritma Adam untuk Peningkatan Akurasi Klasifikasi Penyakit Daun Padi Berbasis CNN Optimization Adam Algorithm for Improving the Accuracy of Rice Leaf Disease Classification Based on CNN,” vol. 13, no. 105, pp. 738–743, 2024.

[10] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, “Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification,” *IEEE Access*, vol. 10, no. July, pp. 68868–68884, 2022, doi: 10.1109/ACCESS.2022.3187203.

[11] M. A. N. Hidayat, Kusrini, and Hanafi, “Convolutional Neural Network Pada Identifikasi Varian Tanaman Anggur Menggunakan Resnet-50,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 3, pp. 61–70, 2023, [Online]. Available: https://jurnal.mdp.ac.id/index.php/jatisi/article/download/4789/1531

[12] Moh. A. Hasan, Y. Riyanto, and D. Riana, “Grape leaf image disease classification using CNN-VGG16 model,” *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 4, pp. 218–223, Oct. 2021, doi: 10.14710/jtsiskom.2021.14013.

[13] H. Wang, S. Qiu, H. Ye, and X. Liao, “A Plant Disease Classification Algorithm Based on Attention MobileNet V2,” *Algorithms*, vol. 16, no. 9, 2023, doi: 10.3390/a16090442.

[14] A. L. Rukuna, F. U. Zambuk, A. Y. Gital, and U. M. Bello, “Citrus diseases detection and classification based on efficientnet-B5,” *Systems and Soft Computing*, vol. 7, no. August 2024, p. 200199, 2025, doi: 10.1016/j.sasc.2025.200199.

[15] Y. N. Yu *et al.*, “Citrus Pest Identification Model Based on Improved ShuffleNet,” *Applied Sciences (Switzerland)*, vol. 14, no. 11, pp. 1–13, 2024, doi: 10.3390/app14114437.

[16] L. Xu, X. Chen, and X. Yang, “Tourism image classification based on convolutional neural network SqueezeNet ——Taking Slender West Lake as an example,” *PLoS One*, vol. 19, no. 1 January, pp. 1–16, 2024, doi: 10.1371/journal.pone.0295439.

[17] E. Prasiwiningrum and Adyanata Lubis, “Classification Of Palm Oil Maturity Using CNN (Convolution Neural Network) Modelling RestNet 50,” *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 4, no. 3, pp. 983–999, 2024, doi: 10.51454/decode.v4i3.822.

[18] rahayu deny danar dan alvi furwanti Alwie, A. B. Prasetio, R. Andespa, P. N. Lhokseumawe, and K. Pengantar, “KLASIFIKASI CITRA PENYAKIT DAUN TANAMAN PADI MENGGUNAKAN CNN DENGAN ARSITEKTUR VGG-19,” *Jurnal Ekonomi Volume 18, Nomor 1 Maret201*, vol. 2, no. 1, pp. 41–49, 2020.

[19] M. B. Kurniawan and E. Utami, “Performance Comparison of ResNet50, VGG16, and MobileNetV2 for Brain Tumor Classification on MRI Images,” *Sistemasi*, vol. 14, no. 2, p. 767, 2025, doi: 10.32520/stmsi.v14i2.5054.

[20] C. Paramita, S. Rakasiwi, P. N. Andono, and G. F. Shidik, “Deep Learning-Based Eye Disorder Classification : A K-Fold Evaluation of EfficientNetB and VGG16 Models,” vol. 12, no. 3, pp. 441–452, 2025, doi: 10.15294/sji.v12i3.26257.

[21] W. L. Pratitis, K. Kurniasari, and H. Al Fata, “Classification of Spotted Disease on Sugarcane Leaf Image Using Convolutional Neural Network Algorithm,” *JTECS : Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*, vol. 3, no. 2, p. 117, 2023, doi: 10.32503/jtecs.v3i2.3433.

[22] S. Agustiani, R. Aryanti, S. K. Wildah, Y. T. Arifin, S. Marlina, and T. Misriati, “Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2,” vol. 4, no. 4, pp. 150–157, 2024.

[23] J. Lu, X. Liu, X. Ma, J. Tong, and J. Peng, “Improved MobileNetV2 crop disease identification model for intelligent agriculture,” *PeerJ Comput Sci*, vol. 9, 2023, doi: 10.7717/peerj-cs.1595.

[24] H. Nugroho, J. X. Chew, S. Eswaran, and F. S. Tay, “Resource-optimized cnns for real-time rice disease detection with ARM cortex-M microprocessors,” *Plant Methods*, vol. 20, no. 1, 2024, doi: 10.1186/s13007-024-01280-6.

[25] S. Duhan, P. Gulia, N. S. Gill, and E. Narwal, “RTR_Lite_MobileNetV2: A lightweight and efficient model for plant disease detection and classification,” *Curr Plant Biol*, vol. 42, no. August 2024, p. 100459, 2025, doi: 10.1016/j.cpb.2025.100459.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[27] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, “Fruit classification using attention-based MobileNetV2 for industrial applications,” *PLoS One*, vol. 17, no. 2 February, pp. 1–21, 2022, doi: 10.1371/journal.pone.0264586.

[28] S. Lamba *et al.*, “A novel fine-tuned deep-learning-based multi-class classifier for severity of paddy leaf diseases,” *Front Plant Sci*, vol. 14, no. September, 2023, doi: 10.3389/fpls.2023.1234067.

[29] J. H. Sinamenye, A. Chatterjee, and R. Shrestha, “Potato plant disease detection: leveraging hybrid deep learning models,” *BMC Plant Biol*, vol. 25, no. 1, 2025, doi: 10.1186/s12870-025-06679-4.

[30] E. T. Baek, “Attention Score-Based Multi-Vision Transformer Technique for Plant Disease Classification,” *Sensors*, vol. 25, no. 1, 2025, doi: 10.3390/s25010270.

[31] C. Paramita and U. D. Nuswantoro, “Performance Evaluation of YOLOv8 Models for Multi-Class Skin Lesion Detection from Dermoscopic Images,” 2025, doi: 10.1109/SIML65326.2025.11080819.

[32] T. Miftahushudur, H. M. Sahin, B. Grieve, and H. Yin, "A Survey of Methods for Addressing Imbalance Data Problems in Agriculture Applications," *Remote Sens (Basel)*, vol. 17, no. 3, pp. 1–31, 2025, doi: 10.3390/rs17030454.

[33] E. R. Subhiyakto *et al.*, "Evaluation of Resampling Techniques in CNN-Based Heartbeat Classification," *Ingenierie des Systemes d'Information*, vol. 29, no. 4, pp. 1323–1332, 2024, doi: 10.18280/isi.290408.