P-ISSN: 2656-1743 | E-ISSN: 2656-1735 **JURNAL RISET INFORMATIKA**
DOI: https://doi.org/10.34288/jri.v8i3.542 Vol. 8, No. 3. Juni 2026
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

DETECTION OF SUGARCANE LEAF DISEASES USING MOBILENETV3LARGE-BASED TRANSFER LEARNING FOR MOBILE APPLICATIONS

Frida Nur Cahyani[-1], Salamun Rohman Nudin[-2]

Study Program of Informatics Management
Faculty of Vocational Studies
Universitas Negeri Surabaya
frida.22036@mhs.unesa.ac.id[-1], salamunrohman@unesa.ac.id[-2]

Abstract

Sugarcane is one of Indonesia's key plantation commodities with a critical role in fulfilling national sugar demand and supporting bioethanol production. However, sugarcane productivity remains low due to leaf diseases that reduce crop quality and yields, while slow or inaccurate identification accelerates their spread. This study proposes and develops a mobile-based sugarcane leaf disease detection system using transfer learning with the MobileNetV3Large architecture to classify 11 disease classes. Two dataset scenarios were applied: Scenario 1 using the SLD Dataset with 6,748 images and Scenario 2 combining the SLD and Sugarcane Smut datasets totaling 14,804 images. Each scenario was trained under three optimizer configurations: Adam, RMSprop, and SGD, to identify the best-performing combination. Results show that Adam achieved the highest validation accuracy in both scenarios, reaching 94.24% in Scenario 1 and 97.43% in Scenario 2, with corresponding test accuracies of 94.91% and 97.31% respectively. The final model was deployed as a Flutter-based mobile application capable of performing real-time disease detection through image upload or camera capture, providing an accessible tool for farmers to identify sugarcane leaf diseases efficiently.

Keywords: Sugarcane; Leaf Disease Detection; MobileNetV3Large; Transfer Learning; Mobile Application

Abstrak

Tebu merupakan salah satu komoditas perkebunan unggulan Indonesia yang berperan penting dalam pemenuhan kebutuhan gula nasional sekaligus mendukung produksi bioetanol. Namun, produktivitas tebu masih rendah akibat serangan penyakit daun yang menurunkan kualitas tanaman dan hasil panen, sementara identifikasi yang lambat atau tidak akurat dapat mempercepat penyebaran penyakit. Penelitian ini mengusulkan dan mengembangkan sistem deteksi penyakit daun tebu berbasis mobile menggunakan metode transfer learning dengan arsitektur MobileNetV3Large untuk mengklasifikasikan 11 kelas penyakit. Dua skenario dataset diterapkan: Skenario 1 menggunakan SLD Dataset dengan 6.748 gambar dan Skenario 2 menggabungkan SLD Dataset dengan Sugarcane Smut Dataset sebanyak 14.804 gambar. Setiap skenario dilatih dengan tiga konfigurasi optimizer: Adam, RMSprop, dan SGD, untuk menentukan kombinasi terbaik. Hasil penelitian menunjukkan bahwa optimizer Adam mencapai akurasi tertinggi pada kedua skenario, yaitu sebesar 94,91% pada Skenario 1 dan 97,31% pada Skenario 2, dengan performa paling kompetitif secara konsisten. Model terbaik kemudian diimplementasikan pada aplikasi mobile berbasis Flutter yang mampu melakukan deteksi penyakit secara real-time melalui fitur unggah gambar maupun kamera, sehingga dapat membantu petani mengidentifikasi penyakit daun tebu secara lebih efektif dan efisien.

Kata kunci: Tebu; Deteksi Penyakit Daun; MobileNetV3Large; Transfer Learning; Aplikasi Mobile

INTRODUCTION

Indonesia is an agrarian nation in which agriculture constitutes one of the primary pillars of the national economy (Azhari & Purnomo, 2022). The sector plays a critical role in sustaining national food availability while simultaneously serving as the principal source of livelihood for the majority of the rural population (Respati, 2022). The diversity of agricultural commodities cultivated under Indonesia's tropical climate positions the country as having considerable potential for developing renewable energy feedstocks from the agricultural sector (Mudhoffar & Magriasti, 2024).

Biomass derived from organic materials such as crop residues and agricultural waste is one of the renewable energy sources currently being developed (Nurdiansyah et al., 2024). Biomass can be converted into various energy forms, including liquid, gaseous, and solid fuels, to generate electricity and heat (Danso-Boateng & Achaw, 2022). Sugarcane (*Saccharum officinarum L.*) is primarily cultivated as a sugar crop; however, its by-products, including molasses, are also utilized as feedstocks for bioethanol production, thereby reducing dependence on fossil fuels (Parascanu et al., 2021). These dual functions mean that the management of sugarcane plantations bears a direct influence on Indonesia's food security and energy resilience.

National sugarcane productivity continues to be constrained by a range of foliar diseases that inflict substantial economic losses when not addressed at an early stage (Rajput et al., 2021). Diseases such as Brown Spot, Yellow Leaf, Smut, and Grassy Shoot represent major production constraints that pose considerable economic risks, necessitating accurate and timely early detection to prevent further yield deterioration (Strachan et al., 2022). Accordingly, the capacity for timely detection of sugarcane leaf diseases has become an urgent operational requirement for sustaining production continuity (Msomba et al., 2024).

Early detection of plant diseases is fundamental to preventing large-scale agricultural losses (Shoaib et al., 2023). Conventionally, sugarcane disease identification has largely relied on manual visual inspection, an approach that is highly dependent on observer expertise and inherently susceptible to error (Agustiani et al., 2024). Misidentification further leads to inappropriate pesticide application, with adverse consequences for both environmental integrity and farmer health (Sitompul et al., 2024). Machine learning and deep learning technologies offer automated solutions that are both faster and more accurate than conventional manual identification processes (Shafik et al., 2024).

This study developed a sugarcane leaf disease detection system based on the MobileNetV3Large architecture, employing transfer learning and fine-tuning strategies. This study contributes three aspects not collectively addressed in prior work: incorporating an additional Smut-specific dataset to improve the discriminative representation of the Smut class and reduce misclassification with Pokkah Boeng during mobile inference, applying a combined strategy of per-class augmentation, class weights, and Categorical Focal Loss to handle class imbalance, and validating the complete system through scenario-based testing on a Flutter-based mobile application with TFLite deployment. MobileNetV3Large was selected for its superior feature extraction capability relative to both MobileNetV3Small and MobileNetV2, owing to its Squeeze-and-Excitation (SE) mechanism, which enables selective attention to discriminative features across 11 visually diverse disease classes while remaining compatible with TFLite-based mobile deployment (A. T. Khan et al., 2023), an advantage particularly critical for detecting plant diseases with visually similar inter-class characteristics (Daphal & Koli, 2024). Although MobileNetV3Small is more compact in terms of model size, the Large variant has been shown to achieve higher classification accuracy for plant leaf diseases, with an accuracy of 92.81% and an F1-Score of 92.14% reported for strawberry leaf disease classification using MobileNetV3-Large, outperforming EfficientNet-B0 which achieved only 90.71% (Pramudhita et al., 2023). This finding is consistent with results reported in a related study (Kunduracioglu & Pacal, 2024). Furthermore, MobileNetV3Large deployed with a transfer learning approach has been shown to achieve accuracy exceeding 90% in rice leaf disease detection, affirming the architecture's adaptability across diverse agricultural commodities (Faqih et al., 2024). Compared to larger models such as ResNet50 or YOLOv8, MobileNetV3Large remains competitive in accuracy while being substantially more lightweight and well-suited for mobile device deployment (Amrulloh et al., 2024). Fine-tuning using ImageNet pre-trained weights allows the model to adapt its generalized visual representations to the specific characteristics of sugarcane leaf imagery (M. Xu et al., 2022). This transfer learning paradigm has been shown to be effective in addressing the constraints of limited labeled data and in accelerating training convergence within agricultural domains (Mazumder et al., 2023).

A number of prior studies have demonstrated promising results in the application of deep learning to sugarcane leaf disease detection. A fine-tuned MobileNetV2 model trained on a nine-class sugarcane disease dataset achieved a validation accuracy of 95.01% (Agustiani et al., 2024). A hybrid CNN-SVM approach attained an accuracy of 88.69% in classifying sugarcane leaf diseases, highlighting the potential of hybrid methods for this classification task (Yunizar et al., 2025). Data augmentation applied to YOLOv8

improved the mean Average Precision (mAP) from 40.3% to 50.5% (Amrulloh et al., 2024). EfficientNet-b6 achieved an accuracy of 93.39% in sugarcane leaf disease detection (Kunduracioglu & Pacal, 2024). Beyond foliar disease detection, a lightweight model SSV2-YOLO based on YOLOv5s was developed for detecting sugarcane aphid pests in unstructured field environments, demonstrating the viability of lightweight architectures for real-world agricultural deployment (W. Xu et al., 2023). Additionally, a transformer-based deep learning model integrated with a GAN module outperformed conventional models in plant leaf disease detection, further affirming the potential of efficient hybrid architectures for agricultural applications (Zhang et al., 2022).

Several limitations in previous studies still warrant attention. Models such as YOLOv8 and EfficientNet impose high computational demands, rendering them less suitable for deployment on mobile devices (Amrulloh et al., 2024). The CNN-SVM approach also faces challenges with respect to computational time at 618.47 seconds, making it suboptimal for real-time field application (Yunizar et al., 2025). Studies employing MobileNetV2 have similarly encountered difficulties in handling class imbalance (Agustiani et al., 2024). An algorithmic comparison further indicated that while Neural Network algorithms yielded the highest accuracy, no single machine learning approach was fully optimal in simultaneously addressing computational speed and accuracy on sugarcane leaf disease datasets (Wantoro et al., 2025). High computational overhead, slow inference time, and suboptimal class imbalance handling motivated this study to employ MobileNetV3Large with a structured fine-tuning strategy and comprehensive per-class disease evaluation as a lightweight solution for mobile device implementation.

Implementation of the proposed system is expected to support early, independent, and accurate sugarcane leaf disease detection directly in field environments. The proposed methodology has the potential to serve as a foundation for developing analogous systems for other strategically important agricultural commodities in support of sustainable farming practices and national food security. This study therefore aims to develop a mobile-based sugarcane leaf disease detection system using MobileNetV3Large with transfer learning, addressing class imbalance and inter-class misclassification, and validating the system through end-to-end deployment on a Flutter-based mobile application.

RESEARCH METHODS

MobileNetV3Large-based transfer learning was employed in this study to build an image classification model capable of identifying sugarcane leaf diseases. The research was structured into a sequence of interconnected stages, each building on the previous one to ensure a rigorous and reproducible development process. The stages include data collection, data preparation, model development, model evaluation, deployment, and system testing, as illustrated in Figure 1.

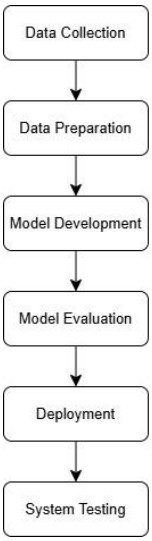


Figure 1. Research Flow

Data Collection

The dataset used in this study consists of sugarcane leaf images sourced from two publicly available repositories. The primary source is the Sugarcane Leaf Image Dataset published on Mendeley Data, comprising 6,748 images across 11 classes: Brown Spot, Sett Rot, Dry Leaf, Yellow Leaf, Brown Rust, Banded Chlorosis, Smut, Viral Disease (Mosaic), Pokkah Boeng, Healthy, and Grassy Shoot. The Smut class in this dataset lacked sufficient visual distinctiveness, potentially hindering discriminative feature learning (Kunduracioglu & Pacal, 2024). An additional Sugarcane Smut Dataset from Kaggle was incorporated to address this limitation, providing 8,056 images specifically for the Smut, Healthy, and Dry Leaf classes. The Smut class was supplemented with real additional images from a publicly available dataset to strengthen its visual representation. As no equivalent public dataset was available for Pokkah Boeng,

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.542
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

augmentation was applied as the primary strategy for that class. The incorporation of real Smut images increased the visual diversity and distinctiveness of the Smut class, making the inter-class boundary between Smut and Pokkah Boeng more defined and thereby improving the model's ability to differentiate between the two classes during inference. Two dataset scenarios were defined in this study. Scenario 1 relied solely on the primary dataset with a total of 6,748 images, whereas Scenario 2 merged both sources resulting in 14,804 images in total. The resulting class distribution across both sources was inherently imbalanced; this imbalance was subsequently addressed during data preparation and model training through augmentation, class weighting, and a focal loss function. To mitigate potential source-level distribution bias, images from both sources were merged at the class level prior to splitting, and a stratified 70/15/15 split was applied independently per class using a fixed random seed, ensuring that each class was proportionally represented across training, validation, and testing subsets regardless of source origin. The distribution of images per class across both repositories is presented in Table 1.

Table 1. Dataset Distribution

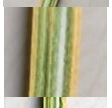
No.	Class	SLD Dataset	Smut Dataset
1.	Banded Chlorosis	471	0
2.	Brown Spot	1,722	0
3.	Brown Rust	314	0
4.	Dry Leaf	343	1,944
5.	Grassy Shoot	346	0
6.	Healthy	430	3,650
7.	Pokkah Boeng	297	0
8.	Sett Rot	652	0
9.	Smut	316	2,462
10.	Viral Disease	663	0
11.	Yellow Leaf	1,194	0

Data Preparation

Data preparation was performed to ensure that the dataset was properly standardized before model training. This stage covered image resizing, dataset splitting, data augmentation, and normalization. Image dimensions were standardized to 224×224 pixels in accordance with the input requirements of MobileNetV3Large. The dataset was divided into training, validation, and testing subsets using stratified random sampling to maintain proportional class representation, with a distribution of 70%, 15%, and 15%.

Offline data augmentation was applied exclusively to the training set and only targeted classes whose sample count fell below 1,000 images, bringing each of those classes up to a minimum of 1,000 samples. This threshold was selected because lower sample counts yield insufficient representation of visual features such as lesion patterns, discoloration, and leaf texture, and would necessitate excessively aggressive augmentation. Augmentation and oversampling strategies have been shown to significantly improve classification accuracy on imbalanced datasets (Ilham et al., 2024). This value is also consistent with the augmentation strategy applied on the same SLD dataset, where per-class augmentation to 1,000 samples yielded competitive classification performance across 11 disease classes (Srinivasan et al., 2025). In Scenario 1, the classes that received augmentation were Brown Rust, Pokkah Boeng, Banded Chlorosis, Grassy Shoot, Healthy, Dry Leaf, Smut, and Sett Rot. In Scenario 2, following the merging of both datasets, only Brown Rust, Pokkah Boeng, and Banded Chlorosis still required augmentation. The augmentation techniques applied are presented in Table 2. Pixel normalization was also applied using the preprocess_input function from TensorFlow Keras, which rescales pixel values to the range of −1 to 1 according to the MobileNetV3Large ImageNet pre-training standard (Pramudhita et al., 2023).

Table 2. Data Augmentation Parameters

No.	Parameter	Value	Image
1.	Horizontal Flip	Enabled	
2.	Rotation	±5%	
3.	Zoom	Up to 15%	
4.	Contrast	10%	
5.	Brightness	10%	

Model Development

The model development stage aimed to build a classification model using MobileNetV3Large with a transfer learning

approach. MobileNetV3Large is a compact CNN architecture optimized for deployment in environments with limited computational capacity, utilizing depthwise separable convolution and the hard-swish activation function to minimize processing overhead without sacrificing feature extraction quality (A. T. Khan et al., 2023). Applying transfer learning with ImageNet pre-trained weights enables the model to leverage robust visual features previously acquired from an extensive dataset, which is particularly beneficial when domain-specific training data is limited (Ady & Pramudwiatmoko, 2025). The overall architecture of MobileNetV3Large used in this study is presented in Figure 2.

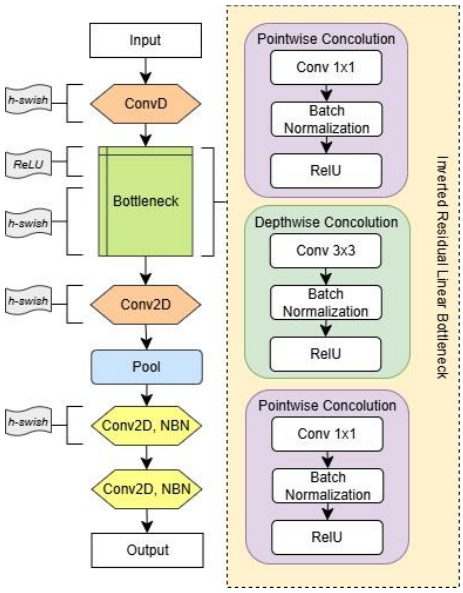


Figure 2. MobileNetV3Large Architecture

The backbone was loaded from TensorFlow Keras Applications with the top classification layer excluded. The first 160 layers were frozen to preserve low-level visual features such as edges and textures. This boundary was determined based on the architectural structure of MobileNetV3Large, in which layers up to index 160 correspond to early and mid-level feature extraction blocks that encode domain-agnostic representations transferable across datasets (A. T. Khan et al., 2023), while layers beyond this point encode higher-level features more responsive to fine-tuning on domain-specific data, and were therefore set as trainable to adapt to the characteristics of sugarcane leaf imagery (M. Xu et al., 2022). All BatchNormalization layers were kept frozen throughout to preserve running statistics derived from ImageNet pretraining, as updating these statistics on a domain-shifted dataset introduces training instability. The backbone begins with a Conv2D layer activated by hard-swish, followed by a series of Bottleneck blocks structured as an Inverted Residual Linear Bottleneck, and concludes with a Pooling layer and two Conv2D layers with Batch Normalization before the final output (Pramudhita et al., 2023).

A custom classification head was then appended to the backbone. The head begins with a GlobalAveragePooling2D layer that condenses spatial feature maps into a 960-dimensional vector, followed by a BatchNormalization layer to stabilize the learning process. Two fully connected Dense layers with ReLU activation of 512 and 256 units respectively were added, each followed by a Dropout layer with a rate of 0.3 to reduce overfitting (Shafik et al., 2024). The output layer employs Softmax activation with 11 nodes corresponding to the 11 disease classes.

The model was trained across six experimental combinations derived from two dataset scenarios and three optimizer configurations: Adam, RMSprop, and SGD. Categorical Focal Loss was adopted as the loss function to better handle the remaining class imbalance after augmentation, as it assigns lower weight to well-classified samples and higher weight to misclassified or hard-to-learn instances, thereby directing the model's attention toward underrepresented and visually ambiguous classes during training (Daphal & Koli, 2024). EarlyStopping and ReduceLROnPlateau callbacks were applied to stabilize convergence and prevent overfitting (Daphal & Koli, 2024). Class weights were additionally incorporated to further reinforce balanced learning across all classes. The complete training configuration is presented in Table 3.

Table 3. Training Configuration

Parameter	Value
Input Size	224 x 224 pixels
Batch Size	32
Optimizer	Adam, RMSprop, SGD
Loss Function	Categorical Focal Loss
Learning Rate	0.001
Epochs	50
Early Stopping	Patience = 5
ReduceLROnPlateau	Factor = 0.3, Patience = 4
Class Weight	Balanced

Model Evaluation

Model evaluation was performed using the test dataset through a confusion matrix, from which

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.542
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

four performance metrics were derived, including accuracy, precision, recall, and F1-score (Agustiani et al., 2024). Although initial model selection was based on accuracy, the final evaluation encompassed accuracy, precision, recall, and F1-score to ensure a comprehensive performance assessment. The formulas for each metric are defined as follows.

Accuracy represents the proportion of images correctly classified by the model across all evaluated predictions.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision quantifies how many of the model's positive predictions for each class were actually correct, reflecting the reliability of those predictions.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

Recall measures the model's ability to correctly identify all positive instances within each class.

$$recall = \frac{TP}{TP + FN} \quad (3)$$

F1-Score is the harmonic mean of Precision and Recall, offering a balanced performance measure that is especially relevant when class distributions are unequal.

$$F1\text{-}Score = 2x \frac{recall\ x\ precision}{recall + precision} \quad (4)$$

Deployment

The best-performing model was converted to TensorFlow Lite format with the extension .tflite to enable efficient inference on mobile devices with limited computational resources (F. Khan et al., 2023). The mobile application was developed using the Flutter framework to support cross-platform deployment, with the TFLite model integrated directly into the application pipeline. The application provides two image input methods by allowing users to capture an image directly through the device camera or select one from the gallery. Once an image is submitted, the model processes it and outputs the predicted disease class along with a confidence score. Detection history is stored locally using SQLite, allowing users to review previous identification results without requiring an internet connection.

System Testing

A black-box testing approach was applied to validate the functionality of the mobile application's primary features. Black-box testing evaluates system behavior through inputs and outputs without examining internal code, and is well-suited for validating functional requirements from the user perspective (Rifkiansyah & Azrino Gustalika, 2025). The aspects tested cover image selection from the gallery, image capture through the device camera, disease classification upon image submission, saving detection results to local storage, and retrieval of previously stored detection history.

RESULTS AND DISCUSSION

Model training was conducted across six experimental combinations derived from two dataset scenarios and three optimizer configurations. Scenario 1 utilized the SLD Dataset (6,748 images), while Scenario 2 employed the combined SLD and Smut Dataset (14,804 images). Each scenario was trained using Adam, RMSprop, and SGD, each with a learning rate of 1×10^{-3}. The training results for Scenario 1 are presented in Table 4.

Table 4. Training Results Scenario 1

Optimizer	Train (%)	Val (%)	Test (%)	Time (min)
Adam	93.68	94.24	94.91	19
RMSprop	93.14	94.24	93.74	20
SGD	91.09	91.26	93.25	34

Based on Table 4, the Adam and RMSprop optimizers achieved the same validation accuracy at 94.24%, followed by SGD at 91.26%. SGD required the longest training duration at 34 minutes despite yielding the lowest accuracy. The relatively close performance across all optimizers in Scenario 1 suggests that model generalization at this stage was more constrained by the limited dataset diversity than by optimizer choice. The accuracy and loss graphs for all three optimizers in Scenario 1 are presented in Figures 3 through 5.

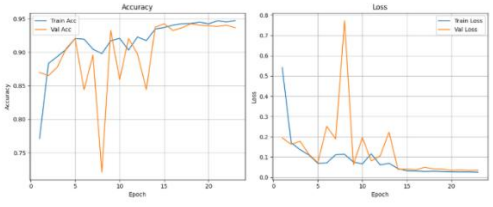


Figure 3. Accuracy and Loss Graph – Adam Optimizer Scenario 1

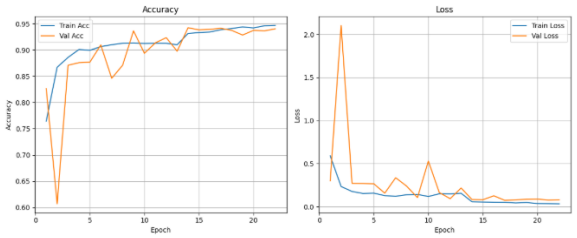


Figure 4. Accuracy and Loss Graph – RMSprop Optimizer Scenario 1

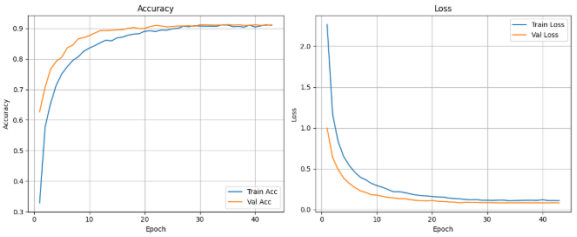


Figure 5. Accuracy and Loss Graph – SGD Optimizer Scenario 1

As shown in Figures 3 through 5, training accuracy across all three optimizers in Scenario 1 improved steadily and converged above 0.91. Mid-training fluctuations in validation accuracy and loss, particularly under Adam and RMSprop, are attributable to the interaction between the ReduceLROnPlateau callback and the adaptive momentum mechanisms inherent to these optimizers, which amplified gradient updates on hard-to-classify samples before reconvergence. SGD, lacking adaptive internal momentum, responded more smoothly to learning rate reductions, resulting in a more stable validation curve. A marginal divergence between training and validation metrics in the final epochs indicates a tendency toward mild overfitting, attributable to the limited class diversity within the single-source SLD Dataset. The per-class classification reports for Adam, RMSprop, and SGD in Scenario 1 are presented in Tables 5 through 7.

Table 5. Classification Report – Scenario 1 Adam Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9885	0.9923	0.9904	259
Sett Rot	1.0000	1.0000	1.0000	99
Dry Leaf	0.9623	0.9808	0.9714	52
Yellow Leaf	0.9779	0.9833	0.9806	180
Brown Rust	0.8727	1.0000	0.9320	48
Banded Chlorosis	0.8889	0.8889	0.8889	72
Smut	0.8529	0.6042	0.7073	48
Viral Disease	0.9432	0.8300	0.8830	100
Pokkah Boeng	0.7719	0.9565	0.8544	46
Healthy	0.9420	1.0000	0.9701	65
Grassy Shoot	0.9815	1.0000	0.9907	53
	Accuracy		0.9491	1022

Based on Table 5, the Adam optimizer in Scenario 1 achieved an overall test accuracy of 94.91%. Most classes performed well, with Sett Rot and Grassy Shoot reaching perfect scores. The Smut class recorded the lowest recall at 0.6042, reflecting insufficient training samples for that class within the SLD Dataset, while Pokkah Boeng showed lower precision due to visual overlap with neighboring classes. Brown Rust achieved a perfect recall of 1.0000 despite its limited support of 48 samples, suggesting that its visual characteristics were sufficiently distinct for the model to identify all positive instances correctly.

Table 6. Classification Report – Scenario 1 RMSprop Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9922	0.9807	0.9864	259
Sett Rot	1.0000	1.0000	1.0000	99
Dry Leaf	0.9107	0.9808	0.9444	52
Yellow Leaf	0.9778	0.9778	0.9778	180
Brown Rust	1.0000	0.7917	0.8837	48
Banded Chlorosis	0.7955	0.9722	0.8750	72
Smut	0.6610	0.8125	0.7290	48
Viral Disease	0.8969	0.8700	0.8832	100
Pokkah Boeng	0.9643	0.5870	0.7297	46
Healthy	0.9552	0.9846	0.9697	65
Grassy Shoot	0.9815	1.0000	0.9907	53
	Accuracy		0.9374	1022

Based on Table 6, the RMSprop optimizer achieved a test accuracy of 93.74%. Sett Rot and Grassy Shoot again achieved perfect F1-scores. Pokkah Boeng recorded the lowest recall at 0.5870, while the Smut class showed improved recall over Adam at 0.8125, though with lower precision at 0.6610. Brown Rust recorded a perfect precision of 1.0000 under RMSprop, indicating that all

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.542
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

predicted instances of that class were correct, though its recall dropped to 0.7917, reflecting a trade-off between precision and sensitivity.

Table 7. Classification Report – Scenario 1 SGD Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9768	0.9768	0.9768	259
Sett Rot	0.9900	1.0000	0.9950	99
Dry Leaf	0.9444	0.9808	0.9623	52
Yellow Leaf	0.9773	0.9556	0.9663	180
Brown Rust	0.9130	0.8750	0.8936	48
Banded Chlorosis	0.8272	0.9306	0.8758	72
Smut	0.6833	0.8542	0.7593	48
Viral Disease	0.9318	0.8200	0.8723	100
Pokkah Boeng	0.8158	0.6739	0.7381	46
Healthy	0.9403	0.9692	0.9545	65
Grassy Shoot	0.9811	0.9811	0.9811	53
	Accuracy		0.9325	1022

Based on Table 7, the SGD optimizer achieved the lowest test accuracy among Scenario 1 configurations at 93.25%. Per-class performance was more evenly distributed than the other optimizers, though the Smut class again posted the lowest F1-score at 0.7593. These results confirm that the performance gap across optimizers in Scenario 1 was primarily driven by class-level data limitations rather than optimizer behavior. Sett Rot remained the most reliably classified class with a recall of 1.0000, consistent with results observed under Adam and RMSprop.

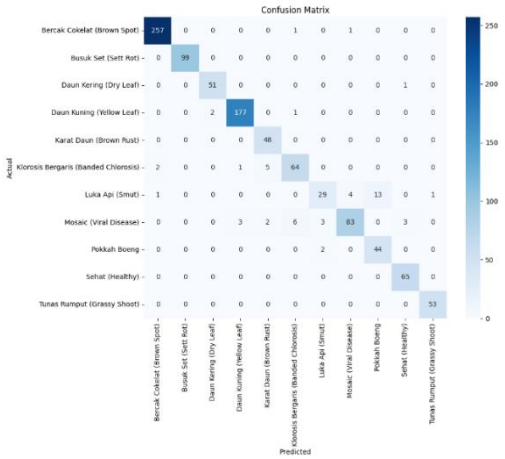


Figure 6. Confusion Matrix – Scenario 1 Adam

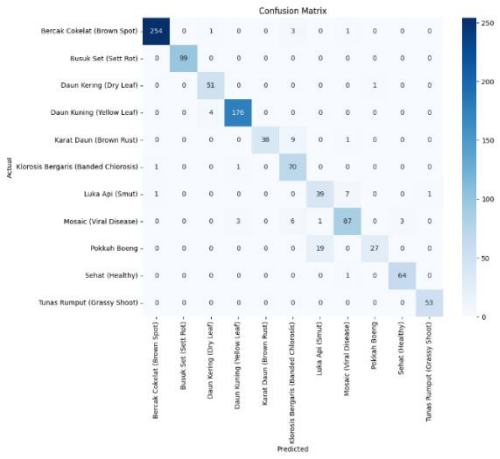


Figure 7. Confusion Matrix – Scenario 1 RMSprop

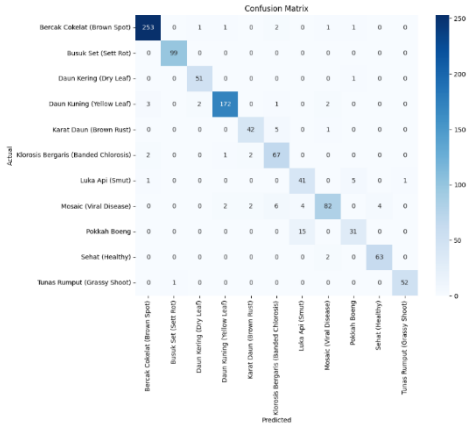


Figure 8. Confusion Matrix – Scenario 1 SGD

Figures 6 through 8 present the confusion matrices for all three optimizer configurations in Scenario 1. Brown Spot, Sett Rot, and Grassy Shoot were classified consistently across all optimizers, while Smut, Banded Chlorosis, and Pokkah Boeng exhibited the most notable misclassification, consistent with their lower F1-scores in Tables 5 through 7.

Table 8. Training Results Scenario 2

Optimizer	Train (%)	Val (%)	Test (%)	Time (min)
Adam	97.48	97.43	97.31	30
RMSprop	97.65	97.38	96.50	29
SGD	95.88	96.80	96.10	43

Based on Table 8, Adam achieved the highest validation accuracy at 97.43%, followed by RMSprop at 97.38% and SGD at 96.80%. Across all optimizer configurations, Scenario 2 consistently outperformed Scenario 1, confirming that the expanded and more balanced dataset directly

improved the model's generalization capability. SGD required the longest training time at 43 minutes while still producing the lowest validation accuracy, reinforcing the pattern observed in Scenario 1. The accuracy and loss graphs for all three optimizers in Scenario 2 are presented in Figures 9 through 11.

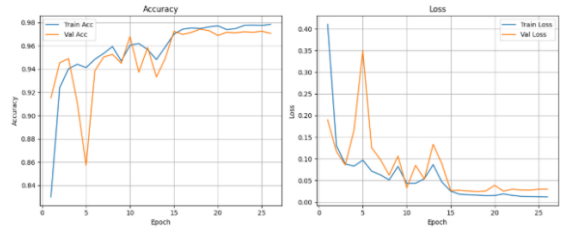


Figure 9. Accuracy and Loss Graph – Adam Optimizer Scenario 2

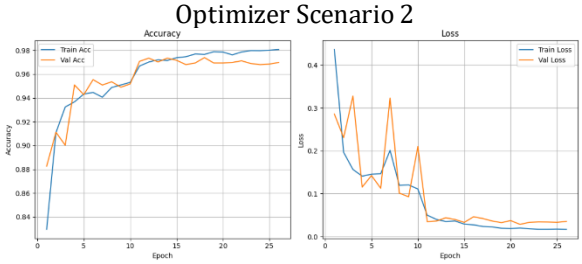


Figure 10. Accuracy and Loss Graph – RMSprop Optimizer Scenario 2

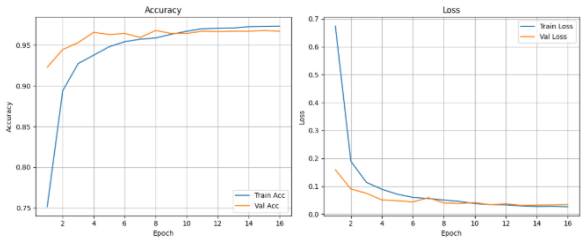


Figure 11. Accuracy and Loss Graph – SGD Optimizer Scenario 2

As shown in Figures 9 through 11, training accuracy across all three optimizers in Scenario 2 rose steeply and converged near 0.97 within fewer epochs. Adam and RMSprop exhibited minor early-epoch fluctuations consistent with the pattern observed in Scenario 1, though with faster reconvergence owing to the larger and more balanced dataset providing more consistent mini-batch distributions. SGD maintained a smoother convergence trajectory throughout. The validation accuracy closely mirrored the training curve with a minimal gap, and the loss settled at a consistently low level, indicating superior generalization relative to Scenario 1. The per-class classification reports for Adam, RMSprop, and SGD in Scenario 2 are presented in Tables 9 through 11.

Table 9. Classification Report – Scenario 2 Adam Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9922	0.9807	0.9864	259
Sett Rot	1.0000	1.0000	1.0000	99
Dry Leaf	0.9942	0.9971	0.9956	344
Yellow Leaf	0.9674	0.9889	0.9780	180
Brown Rust	0.7344	0.9792	0.8393	48
Banded Chlorosis	0.9048	0.7917	0.8444	72
Smut	0.9975	0.9450	0.9705	418
Viral Disease	0.9072	0.8800	0.8934	100
Pokkah Boeng	0.7667	1.0000	0.8679	46
Healthy	0.9967	0.9984	0.9976	612
Grassy Shoot	0.9815	1.0000	0.9907	53
	Accuracy		0.9731	2231

Based on Table 9, the Adam optimizer in Scenario 2 achieved a test accuracy of 97.31%. The Smut class reached an F1-score of 0.9705, a substantial gain over its Scenario 1 result of 0.7073, directly attributable to the increased training samples. Brown Rust and Banded Chlorosis recorded the lowest F1-scores at 0.8393 and 0.8444, stemming from visual overlap with other disease categories.

Table 10. Classification Report – Scenario 2 RMSprop Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9921	0.9691	0.9805	259
Sett Rot	1.0000	1.0000	1.0000	99
Dry Leaf	0.9857	1.0000	0.9928	344
Yellow Leaf	0.9719	0.9611	0.9665	180
Brown Rust	0.7800	0.8125	0.7959	48
Banded Chlorosis	0.8056	0.8056	0.8056	72
Smut	0.9826	0.9450	0.9634	418
Viral Disease	0.8318	0.8900	0.8599	100
Pokkah Boeng	0.7321	0.8913	0.8039	46
Healthy	1.0000	0.9984	0.9992	612
Grassy Shoot	0.9815	1.0000	0.9907	53
	Accuracy		0.9650	2231

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.542
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Based on Table 10, the RMSprop optimizer achieved a test accuracy of 96.50%. Sett Rot, Grassy Shoot, and Healthy maintained near-perfect scores. Brown Rust and Banded Chlorosis continued to yield lower F1-scores, consistent with the pattern observed under Adam.

Table 11. Classification Report – Scenario 2 SGD Optimizer

Class	Precision	Recall	F1-Score	Support
Brown Spot	0.9654	0.9691	0.9672	259
Sett Rot	1.0000	1.0000	1.0000	99
Dry Leaf	0.9942	0.9942	0.9942	344
Yellow Leaf	0.9551	0.9444	0.9497	180
Brown Rust	0.7321	0.8542	0.7885	48
Banded Chlorosis	0.8525	0.7222	0.7820	72
Smut	0.9851	0.9522	0.9684	418
Viral Disease	0.8208	0.8700	0.8447	100
Pokkah Boeng	0.7407	0.8696	0.8000	46
Healthy	0.9935	0.9984	0.9959	612
Grassy Shoot	0.9815	1.0000	0.9907	53
	Accuracy		0.9610	2231

Based on Table 11, the SGD optimizer achieved a test accuracy of 96.10%. Despite recording the lowest accuracy among Scenario 2 configurations, per-class performance remained competitive, with Smut reaching an F1-score of 0.9684. Brown Rust and Banded Chlorosis again posted the lowest scores, reinforcing that inter-class visual similarity remains the primary source of misclassification across all optimizer configurations in this scenario.

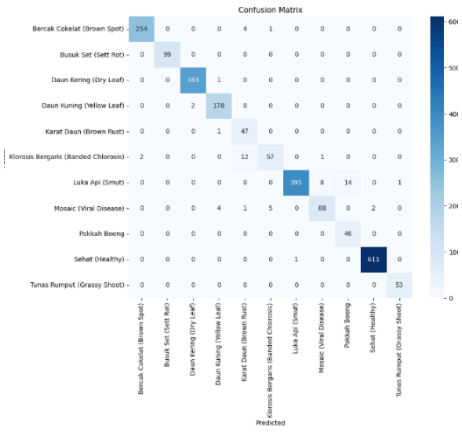


Figure 12. Confusion Matrix - Scenario 2 Adam

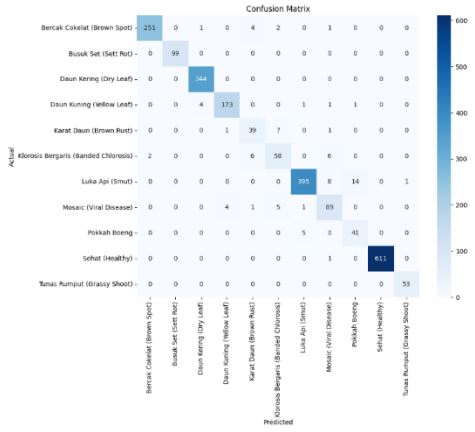


Figure 13. Confusion Matrix – Scenario 2 RMSprop

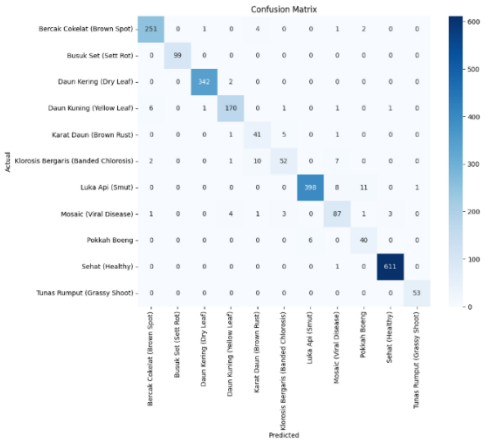


Figure 14. Confusion Matrix – Scenario 2 SGD

Figures 12 through 14 present the confusion matrices for all three optimizer configurations in Scenario 2. The Smut class showed a marked improvement in correct classification across all optimizers compared to Scenario 1, reflecting the impact of the expanded dataset. Brown Rust and Banded Chlorosis remained the primary sources of misclassification, consistent with their lower F1-scores in Tables 9 through 11. The evaluation metrics for all experimental combinations are summarized in Table 12 to facilitate a comprehensive cross-scenario comparison.

Table 12. Evaluation Metrics - All Experimental Combinations

Scenario	Optimizer	Accuracy	Precision	Recall	F1-Score
Scenario 1	Adam	0.9491	0.9256	0.9305	0.9244
	RMSprop	0.9374	0.9214	0.9052	0.9063
	SGD	0.9325	0.9074	0.9107	0.9068
Scenario 2	Adam	0.9731	0.9311	0.9601	0.9422
	RMSprop	0.9650	0.9229	0.9198	0.9200

SGD	0.9610	0.9110	0.9249	0.9165

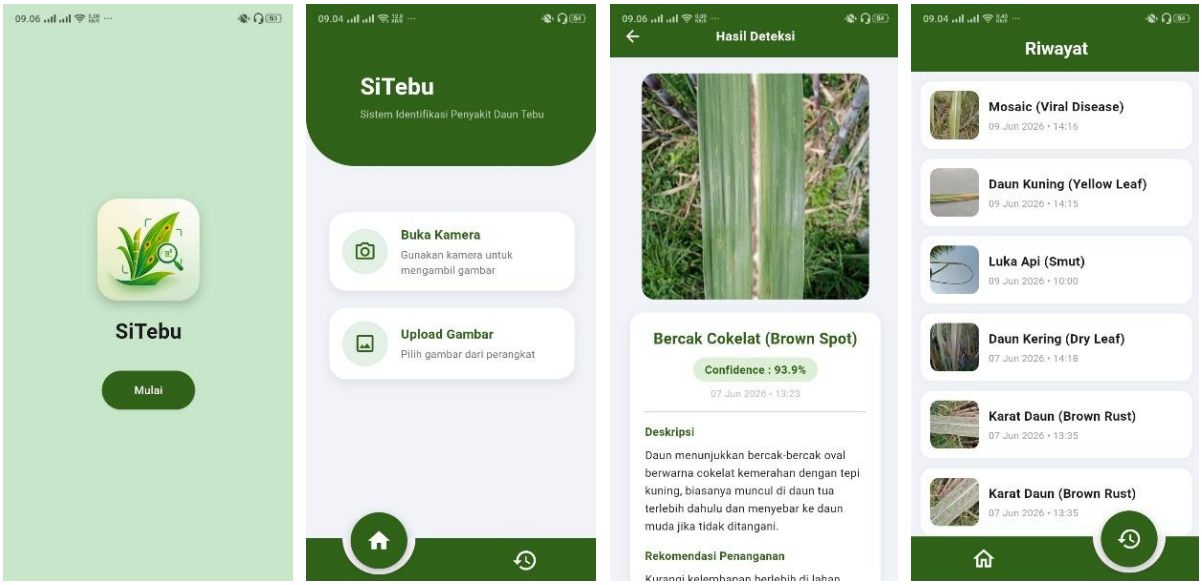
Based on Table 12, the Adam optimizer yielded the best performance in both scenarios, achieving test accuracies of 94.91% and 97.31% in Scenario 1 and Scenario 2 respectively, with corresponding F1-scores of 0.9244 and 0.9422. The most significant improvement across both scenarios was observed in the Smut class, where recall increased from 0.6042 to 0.9450 due to the addition of the Sugarcane Smut Dataset. These findings confirm that dataset diversity and volume have a greater influence on model performance than optimizer selection. Classes such as Brown Rust, Banded Chlorosis, and Pokkah Boeng proved challenging across both scenarios due to their visual similarity with other disease categories. Based on these results, the combination of the Adam optimizer with the combined dataset (Scenario 2) was designated as the best model and subsequently integrated into the mobile application.

The best-performing model was converted into TensorFlow Lite (.tflite) format and embedded into the Flutter-based mobile application. The application was designed to be accessible without requiring technical expertise from the user, presenting a simple and intuitive interface. The application supports image acquisition through direct camera capture or gallery-based selection. After submission, the system displays the predicted disease class along with a confidence score, a disease description, and recommended treatment steps.

Detection results are stored automatically in a local SQLite database, enabling users to review their history without an internet connection. The application interface is illustrated in Figure 15, consisting of four main screens: the Splash Screen (a), the Home Screen (b), the Result Screen (c), and the History Screen (d). Black-box testing was conducted to verify that all primary features of the application functioned correctly under each defined test scenario. The testing covered core functionalities from application startup to detection result storage and retrieval. The testing results are presented in Table 14.

Table 14. Black-Box Testing Results

Feature	Test Scenario	Expected Output	Result
Splash Screen	Application opened	Splash screen displayed	Pass
Home Page	Access home menu	Camera and upload menu appear	Pass
Camera	Capture image	Image successfully captured	Pass



(a) (b) (c) (d)

Figure 15. Mobile Application Interface, (a) Splash Screen, (b) Home Screen, (c) Result Screen, (d) History Screen

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i3.542
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Upload Image	Select image from gallery	Image displayed on screen	Pass
Detection	Click detect button	Label and confidence shown	Pass
History	Open history page	History records displayed	Pass
Delete History	Delete record	Record successfully deleted	Pass

Based on Table 14, all seven tested features functioned correctly and produced outputs consistent with the expected behavior. These results confirm that the application is functionally stable and ready for practical deployment in the field. Scenario-based detection testing was performed to evaluate the system's robustness under various conditions that may be encountered during real-world field use. The testing was structured around four main scenarios: background variation, image rotation, lighting condition, and capture distance. Table 15 summarizes the results of each scenario-based test.

Table 15. Detection Scenario Testing Results

No.	Test Scenario	Test Variation	Result
1.	Background Variation	Plain	95%
		Complex	89%
2.	Image Rotation	0°	95%
		45°	88%
		90°	84%
3.	Lightning Condition	Bright	94%
		Dark	89%
4.	Capture Distance	Near (10 cm)	94%
		Medium (15 cm)	88%
		Far (20 cm)	87%

Based on Table 15, scenario-based testing was conducted using Yellow Leaf as a representative disease class. Background variation testing showed that performance remained relatively robust across plain and complex backgrounds at 95% and 89% respectively. Rotational robustness was maintained across all tested orientations, confirming that the augmentation strategy effectively improved rotational invariance. Lighting conditions had a moderate impact on detection confidence at 94% and 89% for bright and dark conditions respectively. Regarding capture distance, near-to-medium range images produced the most reliable results, while confidence decreased gradually with increasing distance, reaching 87% at 20 cm. These findings suggest that providing users with guidance on optimal image capture practices would further improve real-world system performance.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This study successfully developed a mobile-based system capable of identifying sugarcane leaf diseases through transfer learning with the MobileNetV3Large architecture, covering 11 disease classes across two dataset scenarios. Scenario 1 utilized the SLD Dataset with 6,748 images, while Scenario 2 combined the SLD and Smut datasets totaling 14,804 images. Among all optimizer configurations tested, the Adam optimizer achieved the highest validation accuracy in both scenarios, reaching 94.24% in Scenario 1 and 97.43% in Scenario 2, with corresponding test accuracies of 94.91% and 97.31% respectively. The expanded dataset in Scenario 2 produced consistent accuracy gains across all optimizer configurations, confirming that expanding dataset volume and diversity directly improves the model's generalization capability. Per-class analysis revealed that the Smut class improved substantially from an F1-score of 0.7073 in Scenario 1 to 0.9705 in Scenario 2 following the incorporation of additional data, while visually distinguishable classes including Sett Rot and Grassy Shoot consistently achieved near-perfect scores in both scenarios. The best-performing model was deployed within a Flutter-based mobile application, providing image-based disease classification through camera capture or gallery upload, along with treatment recommendations and detection history stored locally via SQLite. All primary application features passed black-box testing, confirming functional readiness for field deployment.

Suggestion

The findings of this study provide several recommendations for future research. First, additional training data should be collected for classes with limited sample sizes and high visual similarity, particularly Banded Chlorosis, Brown Rust, and Pokkah Boeng, as these classes showed the weakest per-class performance and remain most susceptible to misclassification. Second, a more adaptive augmentation strategy, such as generative augmentation using GANs or diffusion-

based synthesis, may help address class imbalance more effectively than the offline augmentation approach applied in this study. Third, future work may explore alternative lightweight architectures such as EfficientNetLite or MobileNetV3Small as comparative baselines, to assess whether comparable accuracy can be achieved with reduced model size and inference latency on low-end mobile devices. Fourth, the application may be further developed to include real-time disease severity assessment and internet-based treatment recommendations, expanding its practical utility for farmers operating across diverse field conditions.

REFERENCES

Ady, R. P., & Pramudwiatmoko, A. (2025). APPLICATION OF TRANSFER LEARNING ON EFFICIENTNET-B0 ARCHITECTURE FOR AUTOMATIC ROOF TILE DAMAG CLASSIFICATION. *Jurnal Riset Informatika*, *8*(1), 63–73. https://doi.org/10.34288/jri.v8i1.450

Agustiani, S., Aryanti, R., Khotimatul Wildah, S., Arifin, Y. T., Marlina, S., & Misriati, T. (2024). Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu dengan Fine-Tuning MobileNetV2. *Journal of Informatics Management and Information Technology*, *4*(4), 150–157. https://doi.org/10.47065/jimat.v4i4.411

Amrulloh, I. T. A., Nurina Sari, B., Nur Padilah Informatika, T., Singaperbangsa Karawang, U., HSRonggo Waluyo, J., Telukjambe Timur, K., Karawang, K., & Barat, J. (2024). EVALUASI AUGMENTASI DATA PADA DETEKSI PENYAKIT DAUN TEBU DENGAN YOLOV8. *Jurnal Mahasiswa Teknik Informatika*, *8*(4), 7547–7552. https://doi.org/https://doi.org/10.36040/jati.v8i4.10267

Azhari, W. F., & Purnomo, D. (2022). Analisis input – output: Dampak sektor pertanian terhadap perekonomian, pendapatan rumah tangga, dan kesesempatan kerja. *Journal of Economics Research and Policy Studies*, *2*(3). https://doi.org/10.53088/jerps.v2i3.417

Danso-Boateng, E., & Achaw, O. W. (2022). Bioenergy and biofuel production from biomass using thermochemical conversions technologies—a review. *AIMS Energy*, *10*(4), 585–647. https://doi.org/10.3934/energy.2022030

Daphal, S. D., & Koli, S. M. (2024). Enhanced Classification of Sugarcane Diseases Through a Modified Learning Rate Policy in Deep Learning. *Traitement Du Signal*, *41*(1), 441–449. https://doi.org/10.18280/ts.410138

Faqih, R. R., Irsan, M., & Fathoni, M. F. (2024). Rice Plant Disease Detection System Using Transfer Learning with MobilenetV3Large. *Sinkron*, *8*(2), 805–812. https://doi.org/10.33395/sinkron.v8i2.13383

Ilham, I. M., Ernawati, S., & Indra, M. (2024). IMPROVING IMAGE CLASSIFICATION ACCURACY WITH OVERSAMPLING AND DATA AUGMENTATION USING DEEP LEARNING: A CASE STUDY ON THE SIMPSONS CHARACTERS DATASET. *Jurnal Riset Informatika*, *6*(4), 201–210. https://doi.org/10.34288/jri.v6i4.348

Khan, A. T., Jensen, S. M., Khan, A. R., & Li, S. (2023). Plant disease detection model for edge computing devices. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1308528

Khan, F., Zafar, N., Tahir, M. N., Aqib, M., Waheed, H., & Haroon, Z. (2023). A mobile-based system for maize plant leaf disease detection and classification using deep learning. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1079366

Kunduracioglu, I., & Pacal, I. (2024). Deep Learning-Based Disease Detection in Sugarcane Leaves: Evaluating EfficientNet Models. *Journal of Operations Intelligence*, *2*(1), 321–335. https://doi.org/10.31181/jopi21202423

Mazumder, M. K. A., Mridha, M. F., Alfarhood, S., Safran, M., Abdullah-Al-Jubair, M., & Che, D. (2023). A robust and light-weight transfer learning-based architecture for accurate detection of leaf diseases across multiple plants using less amount of images. *Frontiers in Plant Science*, *14*. https://doi.org/10.3389/fpls.2023.1321877

Msomba, B. H., Ndaki, P. M., & Joseph, C. O. (2024). Sugarcane sustainability in a changing climate: a systematic review on pests, diseases, and adaptive strategies. In *Frontiers in Agronomy* (Vol. 6). Frontiers Media SA. https://doi.org/10.3389/fagro.2024.1423233

Mudhoffar, K., & Magriasti, L. (2024). Ekonomi Politik Energi Terbarukan: Peluang dan Tantangan di Indonesia. *Multiverse: Open Multidisciplinary Journal*, *3*(1), 47–52. https://doi.org/10.57251/multiverse.v3i1.1382

Nurdiansyah, Setyani, M., Sespira, D., Anggiriani, F., Aqbal, J., Erlangga, M. B., Pratiwi, M. M. A., Meilani, D., Andri, R. Z., Triansyah, R. P., & Saputra, Y. (2024). 49.+Production_Nurdiansyah (1). *Jurnal Pengabdian Masyarakat Bangsa*, *2*(7), 2774–2780. https://doi.org/https://doi.org/10.59837/jpmba.v2i7.1334

Parascanu, M. M., Sanchez, N., Sandoval-Salas, F., Carreto, C. M., Soreanu, G., & Sanchez-Silva, L. (2021). Environmental and economic analysis of bioethanol production from sugarcane molasses and agave juice. *Environmental Science and Pollution Research*, *28*(45), 64374–64393. https://doi.org/10.1007/s11356-021-15471-4

Pramudhita, D. A., Azzahra, F., Arfat, I. K., Magdalena, R., & Saidah, S. (2023). Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, *9*(3), 522–534. https://doi.org/10.26555/jiteki.v9i3.26341

Rajput, M. A., Rajput, N. A., Syed, R. N., Lodhi, A. M., & Que, Y. (2021). Sugarcane smut: Current knowledge and the way forward for management. In *Journal of Fungi* (Vol. 7, Number 12). MDPI. https://doi.org/10.3390/jof7121095

Respati, E. (2022). *OUTLOOK TEBU 2022 Pusat Data dan Sistem Informasi Pertanian i OUTLOOK TEBU Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal-Kementerian Pertanian 2022*.

Rifkiansyah, H., & Azrino Gustalika, M. (2025). Design And Development Of A Ticket Booking Application Using Extreme Programming At Serayu Larangan. *Jurnal Riset Informatika*, *7*(4), 278–288. https://doi.org/10.34288/jri.v7i4.384

Shafik, W., Tufail, A., De Silva Liyanage, C., & Apong, R. A. A. H. M. (2024). Using transfer learning-based plant disease classification and detection for sustainable agriculture. *BMC Plant Biology*, *24*(1). https://doi.org/10.1186/s12870-024-04825-y

Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. In *Frontiers in Plant Science* (Vol. 14). Frontiers Media SA. https://doi.org/10.3389/fpls.2023.1158933

Sitompul, D., Lumbantobing, P., Manik, S., & Harefa, M. S. (2024). Optimasi Penggunaan Bio-Pestisida sebagai Pengganti Pestisida Kimia pada Pertanian di Kec. Percut Sei Tuan, Kab. Deli Serdang. *El-Mujtama: Jurnal Pengabdian Masyarakat* , *4*(2). https://doi.org/10.47467/elmujtama.v4i2.1281

Strachan, S., Bhuiyan, S. A., Thompson, N., Nguyen, N. T., Ford, R., & Shiddiky, M. J. A. (2022). Latent potential of current plant diagnostics for detection of sugarcane diseases. In *Current Research in Biotechnology* (Vol. 4, pp. 475–492). Elsevier B.V. https://doi.org/10.1016/j.crbiot.2022.10.002

Wantoro, A., Dwi Feriyanto, Arry Verdian, & Rakhmat Dedi Gunawan. (2025). Implementasi Algoritma Machine Learning untuk Deteksi Penyakit Daun Tebu: Analisis Perbandingan Kinerja. *Prosiding Seminar Nasional Pembangunan Dan Pendidikan Vokasi Pertanian*, *6*(1), 1683–1690. https://doi.org/10.47687/snppvp.v6i1.1902

Xu, M., Yoon, S., Jeong, Y., & Park, D. S. (2022). Transfer learning for versatile plant disease recognition with limited data. *Frontiers in Plant Science*, *13*. https://doi.org/10.3389/fpls.2022.1010981

Xu, W., Xu, T., Alex Thomasson, J., Chen, W., Karthikeyan, R., Tian, G., Shi, Y., Ji, C., & Su, Q. (2023). A lightweight SSV2-YOLO based model for detection of sugarcane aphids in unstructured natural environments. *Computers and Electronics in Agriculture*, *211*. https://doi.org/10.1016/j.compag.2023.107961

Yunizar, S. F., Sari, A. P., & Aditiawan, F. P. (2025). Pemanfaatan Model ResNet50 dan SVM untuk Klasifikasi Penyakit Daun Tebu. *CICES*, *11*(1), 94–104. https://doi.org/10.33050/cices.v11i1.3506

Zhang, Y., Wa, S., Zhang, L., & Lv, C. (2022). Automatic Plant Disease Detection Based on Tranvolution Detection Network With GAN Modules Using Leaf Images. *Frontiers in Plant Science*, *13*. https://doi.org/10.3389/fpls.2022.875693