SOURSOP LEAF DISEASE DETECTION WITH CNNs: FROM TRAINING TO DEPLOYMENT

Siti Hidayatullah Nuriadi[1], Erlin[2*], Alyauma Hajjah[3], Ramalia Noratama Putri[4]
[1,2,3,4]Department of Informatics Engineering, Institut Bisnis dan Teknologi Pelita Indonesia
siti@student.pelitaindonesia.ac.id, erlin@lecturer.pelitaindonesia.ac.id,
alyauma.hajjah@lecturer.pelitaindonesia.ac.id, ramalia.noratamaputri@lecturer.pelitaindonesia.ac.id

Abstract - Soursop (*Annona muricata*) is a valuable tropical fruit crop that is highly susceptible to leaf diseases caused by fungal, bacterial, and viral infections. These diseases can significantly impact crop yield and quality, posing challenges for farmers, especially when early detection is delayed. This study proposes an automated solution using Convolutional Neural Networks (CNNs) to detect soursop leaf diseases through image classification. A dataset of 400 labelled leaf images, including healthy and diseased leaves (Leaf Rust, Leaf Spot, and Sooty Mold), was collected and preprocessed for the dataset. Three CNN architectures—MobileNetV2, VGG19, and ResNet50—were evaluated based on accuracy, precision, recall, and F1-score. Among them, MobileNetV2 outperformed the others, achieving 73% accuracy, 72% precision, 65% recall, and 66% F1-score and demonstrated strong consistency across classes. The best-performing model was deployed using the Flask web framework, enabling users to upload soursop leaf images and receive instant disease classification along with suggested treatments and preventive measures. This study's novelty lies in the end-to-end pipeline, from model training to deployment via Flask, providing a ready-to-use solution for farmers.

Keywords – soursop, leaf disease detection, convolutional neural network, image classification, model deployment

I. INTRODUCTION

Soursop (*Annona muricata*) is a tropical fruit crop known for its nutritional value and wide use in traditional medicine. It is cultivated in many regions across Southeast Asia, Africa, and Latin America. However, soursop cultivation faces a critical challenge: leaf diseases caused by bacterial, fungal, or viral infections. These diseases can reduce crop productivity, compromise fruit quality, and lead to significant economic losses for farmers [1]. Conventional disease detection methods rely heavily on manual observation, which can be slow, inconsistent, and dependent on expert knowledge that is not always available in the field [2]. Inaccuracy or delay in diagnosing plant disease often leads to improper treatment or the uncontrolled spread of the infection. This creates an urgent need for an automated, fast, and reliable method to detect leaf diseases in their early stages [3].

In the last decade, Convolutional Neural Networks (CNNs) have revolutionized image analysis tasks, including those in precision agriculture. CNNs can learn complex visual patterns directly from raw image data, making them well-suited for classifying healthy and diseased plant leaves. While CNNs have been widely applied to common crops like potato [4], corn [5], and tomatoes [6], a Google Scholar search reveals fewer than 50 relevant studies focused on soursop leaf disease detection using deep learning. Unlike prior works on tomato and corn that often use large datasets or controlled settings, our study deals with a small, real-world dataset under varied conditions. There is a lack of specialized, automated systems for detecting leaf diseases in soursop plants, hindering effective crop management. Most existing solutions either do not support soursop or fail to bridge the gap between model development and practical deployment in real-world agricultural settings.

This study aims to develop a CNN-based model to classify soursop leaf images into healthy and diseased categories, evaluate the model's performance using standard accuracy metrics (precision, recall, F1-score), and deploy the trained model in a user-accessible format to demonstrate its real-world applicability. By addressing both the technical and practical aspects of plant disease detection, this study contributes to the growing field of smart agriculture, with the goal of supporting farmers in making faster and more accurate decisions to protect their crops.

II. SIGNIFICANCE OF THE STUDY

A. Literature Review

The application of deep learning in agriculture has grown significantly in recent years, particularly in the area of plant disease detection through image classification. Convolutional Neural Networks (CNNs), a class of deep learning models specialized for visual pattern recognition, have demonstrated high accuracy in diagnosing plant diseases from leaf images. Early work by Mohanty et al. [7] applied CNNs to a publicly available dataset of 54,306 images covering 26 diseases in 14 crop species, achieving over 99% accuracy in controlled conditions. Similarly, Ferentinos evaluated CNN performance in diagnosing plant diseases in real-time and under varied lighting conditions, confirming the robustness of CNN architectures such as AlexNet and VGGNet for agricultural use [8].

Recent studies have demonstrated the effectiveness of Convolutional Neural Networks (CNNs) in detecting plant leaf diseases across various crops. For tomatoes, CNN models have achieved high accuracy in classifying multiple leaf diseases, with one study reporting 95% accuracy using transfer learning techniques [9][10]. Similarly, a CNN-based approach for rice leaf disease classification, utilizing transfer learning with a VGG16 architecture, showed promising results despite limited dataset availability [11]. Another study exploring plant leaf disease classification achieved 95% accuracy using a CNN model trained on a dataset comprising various diseases [12]. These findings highlight the potential of deep learning in agricultural applications, particularly for early disease detection and crop management. However, challenges such as dataset imbalance and the need for real-world validation remain, suggesting areas for future research to enhance the practical utility of these models in diverse agricultural settings.

In terms of model deployment, recent research has focused on developing mobile and web-based applications for crop disease diagnosis using convolutional neural networks (CNNs). These solutions aim to provide farmers with accessible tools for plant health monitoring. Studies have demonstrated high accuracy rates for CNN models in controlled environments, with some achieving over 95% accuracy on test datasets [13][14]. However, real-world deployment presents challenges due to varying image quality, lighting conditions, and environmental factors, often resulting in decreased model performance [15][16]. To address these issues, researchers have explored lightweight models optimized for mobile devices, offline functionality, and transfer learning techniques [13][14]. Additionally, some studies have incorporated secondary models for input validation and explored progressive web applications to improve accessibility and user experience. Despite these advancements, bridging the gap between lab performance and real-world usability remains a significant challenge in the field. This tool directly benefits soursop farmers by enabling early disease detection even without expert assistance.

B. Research Method

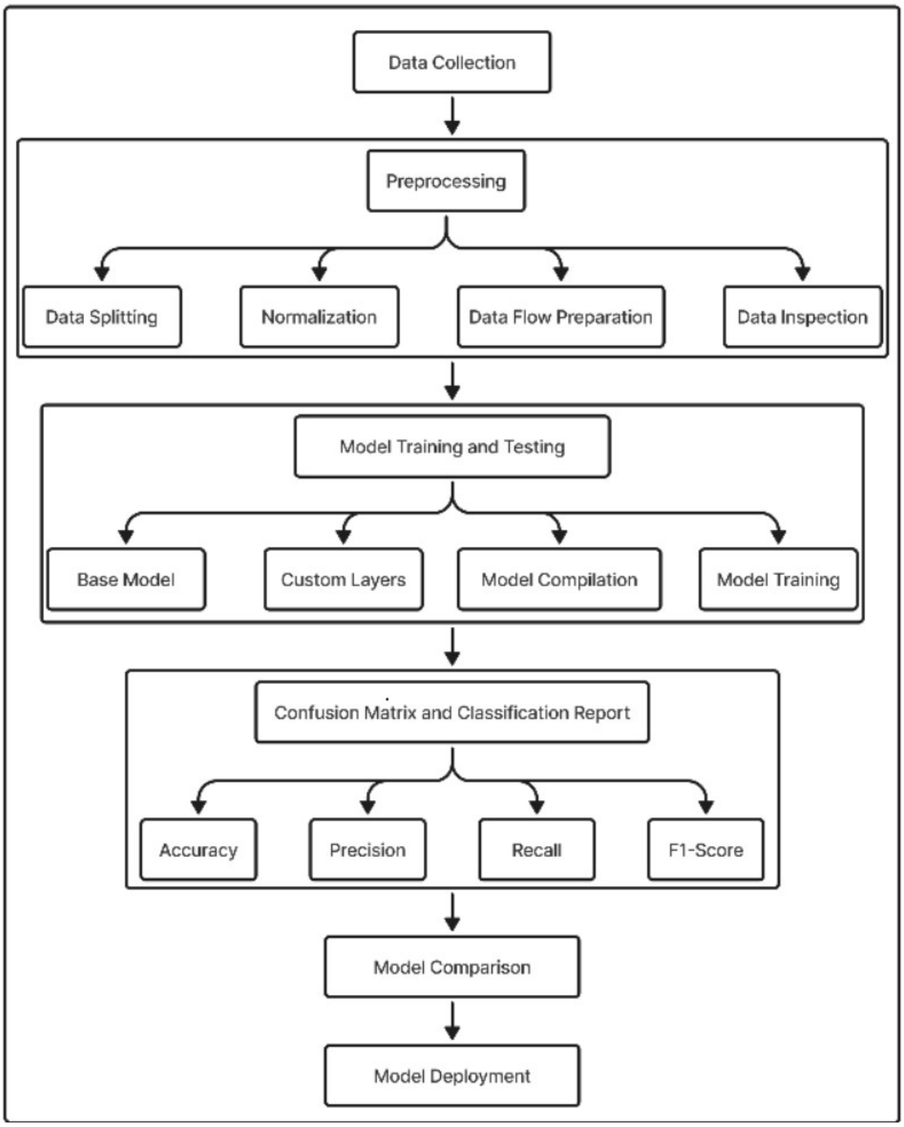


Figure 1. Research Method

This section describes the methodological framework used to develop, evaluate, and deploy a deep learning model for soursop leaf disease detection as depicted in Figure 1. The research process is structured into several key phases, including data collection, preprocessing, model design and training, performance evaluation, model comparison, and deployment.

1. Data Collection

In this phase, images of soursop leaves were gathered from field. This includes collecting samples of both healthy and diseased leaves under various lighting and environmental conditions to ensure diversity and realism in the dataset. All collected data was manually labeled by experts from Pest and Disease Control at Dinas Pangan Tanaman Pangan dan Hortikultura Riau to ensure there were no classification errors, thereby guaranteeing that the training and testing processes run effectively. Table I provides detailed descriptions of the four classes of soursop leaf conditions used in this study: Leaf Rust, Leaf Spot, Sooty Mold, and Healthy. Each entry includes the disease name (in both English and Indonesian) and its defining characteristics. This classification framework is essential for training and evaluating the leaf disease detection model.

TABEL I
DESCRIPTION OF SOURSOP LEAF DISEASE AND HEALTHY LEAF CHARACTERISTICS

Name of Disease	Description
Leaf Rust (Karat Daun)	Leaf rust is caused by the fungus *Phakopsora Pachyrhizi*, particularly during the dry season [17], and is characterized by rust-colored spots and premature leaf drop. This disease is also known as red rust and is caused by the green algae *Cephaleuros Virescens*. It is characterized by round, brown patches with a velvety texture.
Leaf Spot (Bercak Daun)	Leaf spot disease is caused by the fungus *Pestalotiopsis glandicola*, which produces gray spots, or *Phyllosticta annonikola*, which causes brown spots. This disease is widespread and can affect plants at all growth stages, both in nurseries and in the field.
Sooty Mold (Embun Jelaga)	Sooty mold is caused by the fungus *Capnodium sp*., with symptoms appearing as a black coating on leaves and stems that can be easily peeled off, while the underlying leaf tissue remains green.
Healthy (Daun Sehat)	The healthy leaf images represent undamaged, uniformly green foliage with no visible indications of disease. These samples function as the baseline reference for comparative analysis against diseased leaf images.

2. Data Pre-processing

Before model training begins, the dataset undergoes a series of preprocessing steps to ensure data quality and model readiness. First, data splitting is performed by dividing the dataset into three subsets: training, validation, and testing. This division allows the model to learn from one subset, tune its parameters on another, and finally be evaluated on unseen data to measure generalization performance. Next, normalization is applied by scaling pixel values to a [0,1] range using rescale=1./255. This step standardizes input values and accelerates the training process by preventing large gradients. In the data flow preparation phase, the dataset is organized into batches using data generators that efficiently feed images into the model during training, validation, and testing. Lastly, data inspection is conducted to verify that each subset contains an appropriate number of samples and that class labels are distributed correctly. This step ensures dataset integrity and helps identify potential issues such as class imbalance or mislabeling.

3. Model Training and Testing

In this study, MobileNetV2, ResNet50, and VGG19 are employed as the base models. These architectures are initialized with pretrained weights from the ImageNet dataset, allowing the model to leverage previously learned visual features. To preserve these learned features, the initial layers of each base model are frozen, meaning they are set as non-trainable during the training process. On top of the base models, custom layers are added to tailor the network for the specific task of soursop leaf disease classification. These include layers such as Conv2D, MaxPooling2D, Dropout**,** Flatten, and a final Dense layer with a softmax activation function to produce output across four target classes. The model is then compiled using the Adam optimizer, with categorical crossentropy as the loss function, which is suitable for multi-class classification tasks. Finally, the model is trained using the training dataset and is validated on the validation dataset to monitor performance and adjust parameters during learning. This process ensures that the model generalizes well and is not overfitting to the training data.

4. Confusion Matrix and Classification Report

To assess the performance of the trained model, a confusion matrix and a classification report are used. These evaluation tools provide a detailed breakdown of the model's predictions across all classes. The classification report includes key metrics such as precision, recall, F1-score, and accuracy for each individual class, offering a comprehensive view of how well the model distinguishes between healthy and diseased soursop leaves. The confusion matrix further visualizes the number of correct and incorrect predictions, making it easier to identify which classes the model tends to misclassify. Together, these metrics offer a thorough and reliable evaluation of the model’s real-world classification capabilities.

5. *Model Comparison*

After evaluating the performance of each CNN architecture, a model comparison is conducted to identify the most effective model for soursop leaf disease classification. The performance of MobileNetV2, ResNet50, and VGG19 is compared using key evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics provide insight into each model's ability to correctly classify images across all categories. By analyzing the results side by side, the comparison highlights which model offers the best balance of accuracy and generalization. The model with the highest and most consistent performance is then selected for deployment in the final system.

6. *Model Deployment*

Once the best-performing model is identified, the final step is model deployment, which involves integrating the trained model into a practical and accessible system for real-world use. The selected model is exported and deployed in a format suitable for application in the field, such as a web interface, mobile application, or lightweight embedded system. This deployment allows users to upload or capture leaf images and receive instant disease classification results. The goal of this phase is to bridge the gap between research and practical implementation by transforming the model from a laboratory solution into a usable tool that supports timely decision-making in crop management.

III. RESULTS AND DISCUSSION

1. *Data Collection Result*

TABEL II
DISTRIBUTION OF SOURSOP LEAF IMAGES IN EACH CLASS

Class	Number of Images
Leaf Rust	100
Leaf Spot	100
Sooty Mold	100
Healthy	100
Total	400

Table II shows the distribution of soursop leaf images across different classes. Each class represents a specific condition: Leaf Rust, Leaf Spot, Sooty Mold, and Healthy. There are 100 images allocated to each class, totaling 400 images in the dataset.

Figure 2 displays sample visualizations of a soursop leaf dataset, categorized into four distinct classes based on leaf condition. Leaf Spot samples show leaves with small to medium dark spots, indicating localized damage or disease. Leaf Rust leaves exhibit rusty discoloration, usually in orange or brownish patches, often caused by fungal infection. Healthy leaves appear vibrant green with no visible signs of damage, representing the ideal condition. Sooty Mold samples are marked by a dark, soot-like coating on the leaf surface, typically resulting from fungal growth on insect-secreted honeydew. Each row in the image corresponds to one of these categories, serving as a visual reference for distinguishing leaf conditions in classification or diagnostic tasks.

Figure 2. Visual representation of soursop leaf conditions across four classes

2. *Data Pre-processing*

A. *Data Splitting*

TABEL III
DATASET SPLITTING

Data Type	Percentage	Number
Training Data	80%	320
Testing Data	10%	40
Validation Data	10%	40
Total		400

The dataset splitting used in this study is shown in Table III. Total of 400 images, the dataset is divided into three subsets: training data (80% or 320 images) used to train the model, testing data (10% or 40 images) used to evaluate the model's performance after training, and validation data (10% or 40 images) used during training to fine-tune parameters and reduce overfitting. This distribution ensures a balanced setup that supports effective training, validation, and evaluation of the model. The result of this split is three subsets: train_df (80% of the data for training), test_df (10% of the data for testing), and valid_df (10% of the data for validation). Each subset is then further processed to generate data in batch format, ready for use.

B. *Normalization*

All images used in the training and testing processes are normalized. This normalization step aims to convert the pixel values of the images from the original range of [0, 255] to a range of [0, 1]. This is a crucial step to accelerate model training and improve its stability. Normalization is performed by applying the parameter `rescale=1./255`, which automatically divides each pixel value by 255, resulting in more standardized data that is ready for processing by the model. Figure 3 shows a code snippet of how normalization is applied.

```
gen=ImageDataGenerator(rescale=1./255)
validgen=ImageDataGenerator(rescale=1./255)
testgen=ImageDataGenerator(rescale=1./255)
```

Figure 3. A code snippet of data normalization

C. Data Flow Preparation

To simplify data processing during training and testing, a generator was created to stream the data in batch form. This process uses ImageDataGenerator, which enables the dataset to be processed incrementally in batches of a specified size. This approach allows the model to handle large datasets without running out of memory, making the training process more efficient and stable. Initial parameters are used to configure the data settings. The image dimensions are set to a height and width of 224 pixels each, with 3 color channels for RGB images. These parameters are defined in the variables img_shape and img_size, which determine the input format for the model. Additionally, the batch_size is set to 64, indicating the number of samples processed in each iteration during training or validation.

D. Data Inspection

Verifying the number of samples in each subset and ensuring that the classes in the dataset are correctly defined is a crucial step. This process helps confirm that the data distribution across classes is accurate and free from issues. A visualization of sample soursop leaf images for each class, after going through the processes of dataset splitting, normalization, data flow preparation, and data inspection, is shown in Figure 4.

Figure 4. Sample Visualization of Soursop Leaf Dataset by Class After Preprocessing

3. Training and Testing Model

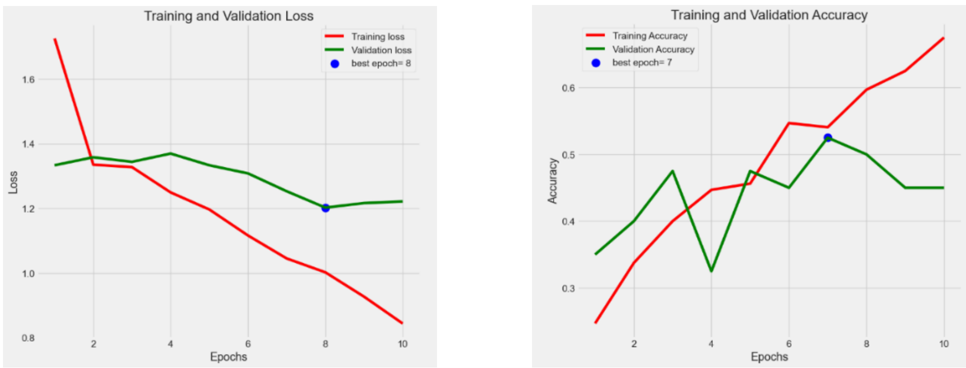


Figure 5. (a) Training and Validation Loss of VGG19

(b) Training and Validation Accuracy of VGG19

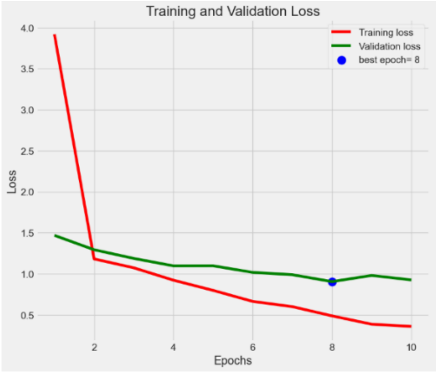
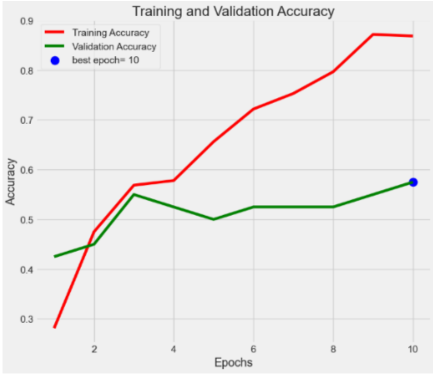


Figure 6. (a) Training and Validation Loss of MobileNetV2



(b) Training and Validation Accuracy of MobileNetV2

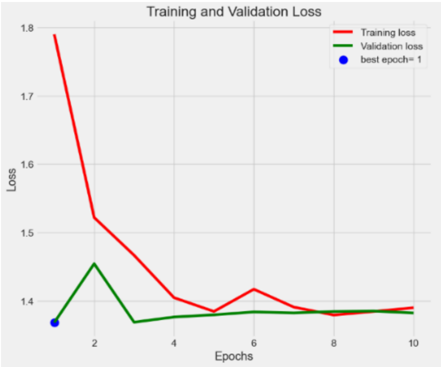
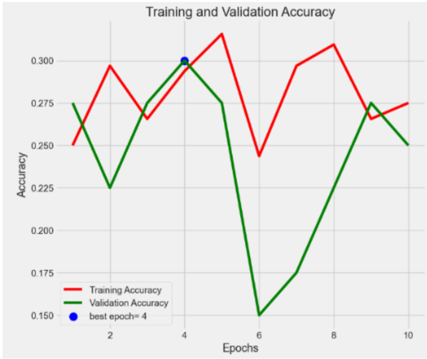


Figure 7. (a) Training and Validation Loss of ResNet50



(b) Training and Validation Accuracy of ResNet50

Figure 5 shows the performance of the VGG19 model across 10 training epochs. Subplot (a) depicts the training and validation loss. The training loss (red) decreases steadily, while the validation loss (green) fluctuates slightly but trends downward. The lowest validation loss occurs at epoch 8 (blue dot), indicating the model's most optimal state before overfitting risks increase. Subplot (b) presents training and validation accuracy. The training accuracy improves consistently, while validation accuracy varies. The best validation result appears at epoch 7, suggesting this is when the model generalizes most effectively.

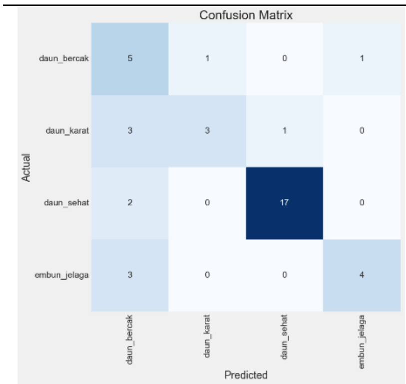
Figure 6 illustrates MobileNetV2 training performance over the same period. In subplot (a), training loss declines sharply early on and continues to improve, while validation loss also decreases, though more gradually. The lowest validation loss is marked at epoch 8. Subplot (b) shows that training accuracy rises steadily to nearly 90%. In contrast, validation accuracy fluctuates, peaking at epoch 10. This gap suggests possible overfitting during later stages of training.

Figure 7 presents results from ResNet50. In subplot (a), training loss drops early and stabilizes, while validation loss remains relatively flat. The best result is at epoch 1. Subplot (b) reveals unstable training and validation accuracy, with a significant drop around epoch 6. The best validation accuracy is seen at epoch 4, indicating inconsistency and a need for further tuning.

4. *Confusion Matrix and Classification Report*

TABEL IV
CONFUSION MATRICES AND CLASSIFICATION REPORT FOR THREE DIFFERENT MODEL ARCHITECTURE OF CNN

MobileNetV2

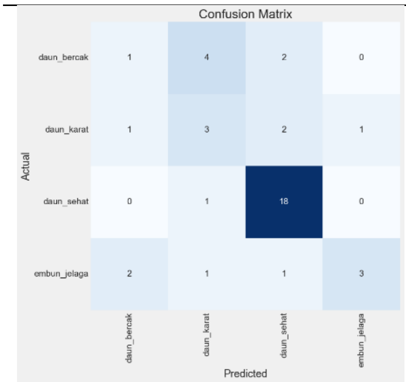



```
Classification Report:
----------------------
              precision    recall  f1-score   support

 daun_bercak       0.38      0.71      0.50         7
  daun_karat       0.75      0.43      0.55         7
  daun_sehat       0.94      0.89      0.92        19
embun_jelaga       0.80      0.57      0.67         7

    accuracy                           0.73        40
   macro avg       0.72      0.65      0.66        40
weighted avg       0.79      0.72      0.74        40
```

VGG19

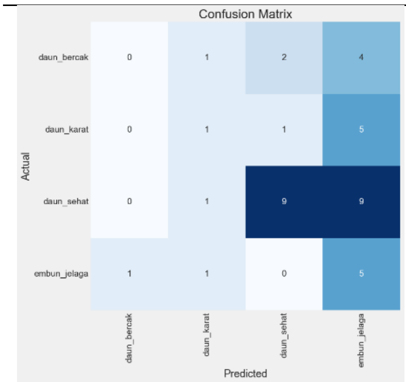



```
Classification Report:
----------------------
              precision    recall  f1-score   support

 daun_bercak       0.25      0.14      0.18         7
  daun_karat       0.33      0.43      0.38         7
  daun_sehat       0.78      0.95      0.86        19
embun_jelaga       0.75      0.43      0.55         7

    accuracy                           0.62        40
   macro avg       0.53      0.49      0.49        40
weighted avg       0.61      0.62      0.60        40
```

ResNet50




```
Classification Report:
----------------------
              precision    recall  f1-score   support

 daun_bercak       0.00      0.00      0.00         7
  daun_karat       0.25      0.14      0.18         7
  daun_sehat       0.75      0.47      0.58        19
embun_jelaga       0.22      0.71      0.33         7

    accuracy                           0.38        40
   macro avg       0.30      0.33      0.27        40
weighted avg       0.44      0.38      0.37        40
```

Table IV presents the confusion matrices and classification reports for three different CNN architectures—MobileNetV2, VGG19, and ResNet50—evaluated on the soursop leaf dataset. These metrics are used to assess each model's performance in classifying the four leaf conditions: *daun_bercak* (leaf spot), *daun_karat* (leaf rust), *daun_sehat* (healthy), and *embun_jelaga* (sooty mold). Among the models, MobileNetV2 demonstrates the best overall performance with an accuracy of 82.5%, showing particularly high precision and recall in identifying healthy leaves (*daun_sehat*) with an F1-score of 0.89, and a solid performance across other classes. VGG19 follows with 62% accuracy, showing a more balanced performance but lower precision and recall in identifying leaf spot and leaf rust. ResNet50 performs the weakest, with an overall accuracy of 38%, and struggles especially with distinguishing leaf spot and leaf rust, as indicated by low recall and F1-scores. Misclassifications often occurred between Leaf Spot and Leaf Rust, likely due to similar visual patterns, suggesting the need for more diverse training data. These results suggest that MobileNetV2 is the most reliable architecture for classifying soursop leaf conditions in this dataset. Compared to [10], which achieved 95% accuracy on tomato datasets with 5000+ images, our model reaches 73% on just 400 images, demonstrating efficiency with limited data.

5. Model Comparison

Table V presents a comparison of the performance of three CNN architectures—MobileNetV2, VGG19, and ResNet50—based on four evaluation metrics: accuracy, precision, recall, and F1-score (macro average). Among the models, MobileNetV2 achieved the highest overall performance, with an accuracy of 73%, precision of 72%, recall of 65%, and an F1-score of 66%, making it the most reliable model for classifying soursop leaf conditions. VGG19 follows with moderate results—62% accuracy and an F1-score of 49%—showing potential but with lower consistency across classes. ResNet50 showed the weakest performance, with only 38% accuracy and an F1-score of 27%, indicating challenges in both precision and generalization. This comparison clearly highlights MobileNetV2 as the best-performing model in this study, suitable for real-world deployment in soursop leaf disease detection.

TABEL V
MODEL PERFORMANCE COMPARISON

Model	Accuracy	Macro Avg		
		Precision	Recall	F1-Score
MobileNetV2	73%	72%	65%	66%
VGG19	62%	53%	49%	49%
ResNet50	38%	30%	33%	27%

6. Model Deployment

To implement MobileNetV2, the best-performing model, selected based on its highest accuracy among the evaluated models, into a web-based application, the deployment process is carried out using Flask, a lightweight micro web framework. Flask enables the development of simple yet functional web applications. The trained model is integrated into the Flask application to receive soursop leaf images as input, process them using the disease detection model, and display the analysis results on a web page. This Flask application serves as the bridge between the model and the user interface. It handles HTTP requests from users, loads the pre-trained model, and uses it to analyze uploaded leaf images. The results of the analysis—such as the type of leaf disease or the health status are then presented to users in an informative format.

Figure 8 shows the result of deploying a web-based soursop leaf disease detection system. In this example, the model successfully predicts the disease as leaf_rust with a confidence level of 96%. The interface allows users to upload a soursop leaf image, and the model analyzes the image to provide both the predicted disease and recommended actions. The solution section includes preventive measures, such as pruning infected parts, maintaining good drainage, and proper planting practices, as well as treatment suggestions using pesticides containing active ingredients like zineb, copper oxychloride, fermate, or dithane. The deployment demonstrates not only the model's predictive capability but also its potential for practical, real-time decision support in plant disease management.

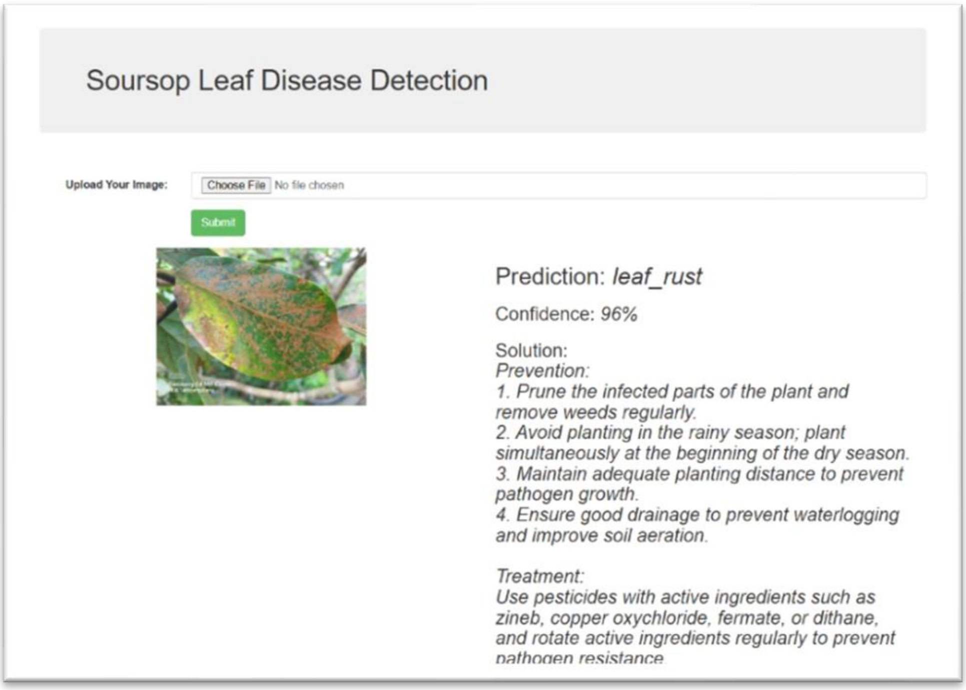


Figure 8. Model Deployment Result for Soursop Leaf Disease Prediction

IV. CONCLUSION

This study demonstrates the development, evaluation, and deployment of a Convolutional Neural Network-based model for detecting soursop leaf diseases. By collecting a balanced dataset of 400 images across four categories, Leaf Rust, Leaf Spot, Sooty Mold, and Healthy, the model was trained and tested using three well-known CNN architectures: MobileNetV2, VGG19, and ResNet50. Among them, MobileNetV2 achieved the highest performance, with an accuracy of 73%, macro precision of 72%, and F1-score of 66%, outperforming the other models in both evaluation metrics and prediction confidence.

The results show that MobileNetV2 not only performs well in terms of accuracy but also demonstrates strong generalization ability with consistent prediction confidence across all classes. This model was selected for deployment in a web-based application using Flask, allowing users to upload leaf images and instantly receive disease classification along with actionable treatment suggestions. The deployment phase bridges the gap between model development and practical use, providing a real-world tool that supports farmers in identifying and managing leaf diseases more effectively. Overall, the study offers a complete, end-to-end solution for soursop leaf disease detection, combining deep learning accuracy with field-ready accessibility to support smart agriculture practices. One key limitation is the small dataset size (400 images), which may limit generalizability. Future work should focus on data augmentation, field testing, and optimizing the model for mobile deployment. This study provides not only an effective classification model but also a deployable system that bridges the lab-to-field gap in smart agriculture.

REFERENCES

[1] P. U. González-Ruíz, A.V.; Palomino-Hermosillo, Y.A.; Balois-Morales, R.; Ochoa-Jiménez, V.A.; Casas-Junco, P.P.; López-Guzmán, G.G.; Berumen-Varela, G.; Bautista-Rosales, "Pathogenic Fungi Associated with Soursop Fruits (Annona muricata L.) during Postharvest in Nayarit, Mexico," *Horticulturae*, vol. 7, pp. 1–11, 2021.

[2] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *J. Big Data*, vol. 11, no. 5, pp. 1–43, 2024.

[3] S.Sailaja, E.V.N.Jyothi, and M.Kranthi, "DeepLeaf: Automated Plant Disease Diagnosis using Deep Learning Approach," *Int. J. Sci. Methods Comput. Sci. Eng.*, vol. 01, no. 01, pp. 17–23, 2024.

[4] Erlin, I. Fuadi, R. N. Putri, D. Nasien, Gusrianty, and D. Oktarina, "Deep Learning Approaches for Potato Leaf Disease Detection: Evaluating the Efficacy of Convolutional Neural Network Architectures," *Rev. d'Intelligence Artif.*, vol. 38, no. 2, pp. 717–727, 2024.

[5] F. Rajeena P. P, A. S. U, M. A. Moustafa, and M. A. S. Ali, "Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach," *Electronics*, vol. 12, no. 8, pp. 1–18, 2023.

[6] C. Vengaiah, "CNN Model Suitability Analysis for Prediction of Tomato Leaf Diseases," *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*. 2023.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, no. September, pp. 1–10, 2016.

[8] Konstantinos P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2022.

[9] Bhoomika and G. Verma, "Advancing Tomato Crop Health: A CNN Approach to Leaf Disease Detection," in *8th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2024 - Proceedings*, 2024, pp. 979–983.

[10] A. A. Garba, V. Jain, and K. Singla, "Tomato Leaf Disease Detection Using CNN Models," in *Proceedings - 4th International Conference on Technological Advancements in Computational Sciences, ICTACS 2024*, 2024, pp. 1504–1507.

[11] V. S. M. S. Balapuram Jayanth , A C Girish , D Kannan , D Charan Kumar , Pandreti Praveen , R Karunia Krishna Priya , N Vijaya Kumar, "Rice Leaf Diseases Classification Using CNN with Transfer Learning," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. IV, pp. 1–14, 2025.

[12] Y. Nurlanuly, "Classification of Plant Leaf Diseases Using Convolutional Neural Networks: A Study of Various Diseases," in *Electrical-Electronics and Biomedical Engineering Conference, ELECO 2024 - Proceedings*, 2024, pp. 1–5.

[13] S.-K. Kim, W. Hur, and J.-G. Ahn, "A Progressive Web Application for Mobile Crop Disease Diagnostics Based on Transfer Learning," *J. Korea Acad. Coop. Soc.*, vol. 23, no. 2, pp. 22–29, 2022.

[14] A. Rathod, D. Panchal, N. Minocha, and D. Bhatt, "Crop Disease Detection Using Lightweight Deep Learning Model for Smartphone," *Int. J. Sci. Technol.*, vol. 16, no. 2, pp. 1–10, 2025.

[15] R. Amanda *et al.*, "A Mobile-Based Deep Learning Model for Cassava Disease Diagnosis," *Front. Plant Sci.*, vol. 10, 2021.

[16] M. Thalor, Y. Chavhan, and S. Mate, "Performance Comparison of CNN Models for Tomato Disease Detection using Image-based data in Both Controlled and Real-world Conditions," *Curr. Agric. Res. J.*, vol. 13, no. 1, pp. 105–111, 2025.