Coconut Leaf Disease Detection Using YOLOv8: A Web-Based Approach

J. Vetrimanikumar[1], Muthukamatchi[2], Manikandan[3], and Mohamed Uvais[4]
[1]*Department of Electronics and Communication Engineering SSM Institute of Engineering and Technology Dindigul, Tamil Nadu, India*
[2,3,4]*Department of Electronics and Communication Engineering SSM Institute of Engineering and Technology Dindigul, Tamil Nadu, India*

***Abstract*—Agricultural sustainability in tropical regions is closely connected to the health of plantation crops. In India, coconut cultivation supports a large number of rural households and contributes significantly to the agricultural economy. However, coconut plantations are frequently affected by diseases and pest attacks that reduce productivity. Farmers generally depend on manual observation to identify leaf diseases, but this approach often results in delayed diagnosis and inconsistent assessments.**

This study presents a deep learning–based approach for detecting coconut leaf diseases using the YOLOv8s deep learning object detection model. The system is trained using a publicly available dataset obtained from Kaggle containing nearly 1,500 images of coconut leaves. The dataset includes four classes representing different leaf conditions: Healthy, Leaf Spot, Yellowing, and Pest Damage. Images were divided into training and testing subsets with an 80:20 ratio.

Model training and evaluation were carried out using Python within the Ultralytics deep learning framework. The YOLOv8s architecture was selected because of its efficient design and ability to perform object detection with low computational requirements. Its anchor-free detection strategy and feature extraction layers enable the model to recognize disease patterns and localize affected regions within leaf images.

Experimental evaluation shows that the trained model can successfully identify disease symptoms across images captured under different environmental conditions such as varying illumination and complex natural backgrounds. Due to its relatively fast inference speed, the system can potentially be integrated into mobile-based diagnostic tools that allow farmers to assess plant health directly in the field. The proposed approach demonstrates how deep learning can support precision agriculture by enabling early disease detection and improving plantation management.

***Index Terms*—Coconut leaf disease, YOLOv8s, Deep learning, Object detection, Computer vision.**

I. INTRODUCTION

Coconut cultivation supports millions of Indian farmers, particularly in southern and coastal regions where the palm's products contribute to both household income and industrial supply chains. Coconut cultivation occupies a significant portion of agricultural land in the country, particularly in southern and coastal regions such as Kerala, Karnataka, Tamil Nadu, and Andhra Pradesh. Recent agricultural statistics indicate that coconut plantations in India extend over more than 2.1 million hectares, making the country one of the largest producers of coconuts worldwide.

The economic value of coconut cultivation extends beyond raw fruit production. Coconut-based industries contribute to sectors such as food processing, pharmaceuticals, and cosmetic manufacturing. Products including coconut oil, desiccated coconut powder, and coconut milk continue to experience increasing demand both domestically and internationally. Recognizing the importance of this sector, the Government of India has expanded support programs through the Coconut Development Board (CDB), including increased financial allocations to strengthen coconut farming infrastructure and productivity.

Despite the economic importance of the crop, coconut plantations face persistent threats from plant diseases and insect infestations. Leaf-based disorders such as fungal leaf spot, nutrient-induced yellowing, and pest attacks can significantly weaken plant health. When

such conditions affect the foliage, the photosynthetic capacity of the palm decreases, which eventually leads to poor nut production and reduced crop yield. In severe cases, long-term infections may weaken the tree and cause irreversible damage.

Traditionally, disease identification relies on visual inspection performed by farmers or agricultural experts. This manual process requires experience and regular field monitoring. However, large plantation areas and tall coconut palms make consistent monitoring difficult. In many situations, symptoms are recognized only after they have progressed to advanced stages, by which time effective intervention becomes challenging. Additionally, visual inspection is subjective and may vary depending on the observer's expertise.

Recent developments in computational technologies have introduced new opportunities for improving plant disease monitoring. Machine learning and computer vision techniques allow automated systems to analyze plant images and identify disease symptoms. These technologies can assist farmers by providing rapid and consistent evaluations of plant health.

Deep neural networks eliminate the need for manual feature engineering by learning hierarchical representations directly from image data, making them well-suited for analyzing complex agricultural imagery. This ability enables them to detect subtle variations in leaf color, texture, and shape that may indicate disease.

Object detection techniques extend this capability further by identifying not only the type of disease but also the location of affected areas within an image. This is particularly useful in agricultural monitoring where symptoms often appear in small localized regions on leaves. By detecting multiple affected regions simultaneously, object detection models provide more detailed diagnostic information compared with simple classification systems.

In this work, a coconut leaf disease detection system is developed using the YOLOv8s architecture. YOLOv8 belongs to the widely used "You Only Look Once (YOLO)" family of object detection models known for high processing speed and strong detection accuracy. The small variant of the model, YOLOv8s, was selected to maintain a balance between detection performance and computational efficiency. This design choice makes the model suitable for deployment on mobile devices or portable agricultural diagnostic systems.

The model was trained using a publicly available dataset containing coconut leaf images representing four different health conditions. Through this study, an automated framework is developed to assist farmers in identifying leaf diseases at an early stage. Such tools can support better crop management practices and help protect the productivity of coconut plantations.

II. RELATED WORK

Research on automated plant disease detection has evolved significantly over the past two decades. Early studies in this field relied mainly on traditional image processing techniques combined with classical machine learning algorithms. In these approaches, features such as color distribution, texture patterns, and shape characteristics were manually extracted from plant images. Methods like color histograms, edge descriptors, and Gray Level Co-occurrence Matrix (GLCM) analysis were commonly used to represent visual characteristics of leaves. These features were then supplied to classifiers including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), or decision tree models to distinguish between healthy and diseased plants.

Although these methods demonstrated moderate success in controlled laboratory conditions, their performance often decreased when applied to real agricultural environments. Natural plantation images contain various challenges such as uneven illumination, overlapping leaves, shadows, and cluttered backgrounds. Because manually designed features cannot easily capture all possible visual variations, traditional models struggled to maintain consistent performance outside controlled datasets.

The introduction of deep learning significantly changed the approach to plant disease analysis. Convolutional Neural Networks (CNNs) became widely adopted due to their ability to automatically learn hierarchical representations from raw image data. In CNN-based models, convolution layers extract local spatial patterns, while deeper layers combine these patterns into more complex visual representations. This allows the network to identify disease symptoms such as spots, discoloration, or damaged leaf structures without requiring manually defined features.

Several studies have successfully applied CNN architectures to crop disease detection tasks. Applications have been reported for crops such as rice, tomato, mango, and banana, where CNN models achieved higher classification accuracy compared with traditional machine learning methods. The ability of deep networks to generalize across different lighting conditions and image variations makes them particularly suitable for real agricultural datasets.

However, many CNN-based systems focus only on image classification. A classification model determines whether a disease exists within an image but does not provide information about the location or number of infected areas. In agricultural monitoring, this limitation can reduce the usefulness of the system because disease symptoms often appear in small localized regions of leaves.

Object detection methods address this limitation by performing classification and localization simultaneously. These models predict bounding boxes around affected regions while also identifying the disease category. This capability enables detailed analysis of plant images where multiple infected areas may appear on a single leaf.

Among the available object detection frameworks, the YOLO (You Only Look Once) family of models has become widely recognized for real-time detection performance. Unlike multi-stage detection frameworks, YOLO models treat object detection as a single regression problem. The entire image is processed in one forward pass through the network, which allows faster inference while maintaining competitive accuracy.

Recent versions of the YOLO architecture include several improvements in feature extraction, training stability, and detection accuracy. These enhancements make the models suitable for applications requiring fast processing, such as mobile-based agricultural monitoring systems or drone-based crop surveillance.

Research focused specifically on coconut tree health has also started to incorporate deep learning methods. Studies have explored the detection of diseases such as Weligama Coconut Leaf Wilt Disease (WCLWD) and pest infestations caused by caterpillars. Using image datasets collected from plantations, deep learning models have demonstrated the ability to assist farmers in identifying disease symptoms at earlier stages.

Building on these developments, the present study implements the YOLOv8s architecture to detect common coconut leaf conditions including leaf spot, yellowing, and pest damage. The focus of this work is to evaluate whether a lightweight object detection model can accurately identify these conditions while maintaining computational efficiency suitable for field-based applications.

III. DATASET DESCRIPTION

The performance of a deep learning model is strongly influenced by the dataset used during training and evaluation. A well-structured dataset allows the model to learn relevant visual patterns while maintaining the ability to generalize to new images. For this study, a publicly available coconut leaf disease dataset was obtained from Kaggle. We used 1,482 images from a publicly available Kaggle dataset, all captured in plantation settings using mobile phones.

Because YOLOv8 supports transfer learning, the model can be initialized with weights learned from large image datasets and then adapted to the coconut leaf dataset. This approach allows effective model training even when the available agricultural dataset is relatively small compared with general image datasets.

A. Data Acquisition and Environmental Variability

The images included in the dataset were captured using mobile phone cameras in coconut plantation environments. Collecting images using handheld devices reflects the practical conditions in which a real diagnostic system would operate. Farmers or agricultural workers would typically capture leaf images directly in the field using mobile phones. To ensure that the dataset represents realistic conditions, images were collected at different times of the day. As a result, the dataset includes variations in brightness, shadows, and illumination intensity. Some images were captured under strong midday sunlight, while others were taken during periods of diffused lighting or partial shade created by surrounding foliage.

Such environmental diversity is important for model robustness. Agricultural environments rarely provide controlled lighting conditions, and a practical disease detection system must be capable of recognizing symptoms despite variations in illumination or background complexity. By including images with natural variability, the dataset helps the model learn features that remain consistent even when environmental conditions change.

B. Health Classes and Visual Symptoms

The dataset is organized into four primary classes representing different health conditions of coconut leaves. Each category reflects commonly observed leaf states within coconut plantations.

Healthy: Images in this class show leaves with normal physiological characteristics. Healthy leaves typically exhibit a uniform green color, intact leaflets, and a smooth surface without visible lesions or mechanical damage. These samples provide the baseline appearance of coconut foliage, allowing the model to distinguish between normal and abnormal leaf conditions.

Leaf Spot: Symptoms appear as small circular or irregular lesions on the leaf surface. These spots are often brown, gray, or black in color and are typically associated with fungal infections. Over time, the affected regions may expand and merge, reducing the effective area of the leaf available for photosynthesis.

Yellowing: This condition is characterized by a gradual loss of the deep green color of the leaf. The foliage may appear pale yellow, light green, or orange-brown depending on the underlying cause. This condition may arise from nutrient deficiencies or systemic diseases affecting the plant. Early identification of yellowing can help farmers take corrective measures related to soil nutrients or disease control.

Pest Damage: This condition results in visible structural damage to the leaf surface. Insect feeding may create irregular holes, torn edges, or partially consumed leaflets. In some cases, the leaf surface becomes skeletonized, leaving only the veins intact. Brown dry patches and jagged edges are common indicators of pest activity.

Each class provides distinct visual patterns that the detection model must learn to recognize during training.

C. Dataset Statistics and Organization

To prepare the dataset for model training, the images were organized into separate directories corresponding to each health class. Annotation files were provided using the standard YOLO labeling format, where each object instance is described using normalized bounding box coordinates along with the corresponding class label.

The dataset was divided into training and testing subsets using an 80%–20% split. The training subset was used to optimize the model parameters, while the remaining images were reserved for evaluating detection performance. The table below summarizes the main characteristics of the dataset used in this study.

Class Category	Description	Approximate Image Count
Healthy	Leaves without visible disease symptoms	~375
Leaf Spot	Leaves containing fungal spot lesions	~375
Yellowing	Leaves showing discoloration symptoms	~375
Pest Damage	Leaves affected by insect feeding damage	~375

The balanced distribution across the four categories allows the model to learn representative features for each health condition. Maintaining a relatively even number of samples per class also helps prevent bias during training, ensuring that the model does not favor one class disproportionately. Before training, images and annotation files were verified to ensure correct formatting for the YOLOv8 framework. This preparation step is essential because improperly formatted labels can negatively affect detection performance.

IV. PROPOSED METHODOLOGY

The objective of this study is to design an automated system capable of identifying coconut leaf diseases using object detection techniques. The methodology follows a structured pipeline that includes dataset

preparation, model training, validation, and performance evaluation. The entire implementation was carried out using Python and the Ultralytics YOLOv8 framework.

The overall workflow of the proposed system consists of the following stages:

1. Dataset preparation and annotation
2. Model configuration using YOLOv8s architecture
3. Training using labeled coconut leaf images
4. Validation using unseen test images
5. Detection of disease regions and classification of leaf conditions

This pipeline enables the system to learn visual features associated with different disease symptoms and apply that knowledge to detect affected regions in new images.

A. YOLOv8 Architecture

The YOLO family treats detection as a regression problem, processing images in one forward pass—a design choice that prioritizes speed without sacrificing accuracy.This design significantly improves detection speed while maintaining strong performance.

The YOLOv8 architecture consists of three main components:

1. Backbone Network
2. Neck for Feature Aggregation
3. Detection Head

YOLOv8's backbone extracts multi-scale features through successive convolutional layers. Early layers detect edges and textures, while deeper layers combine these into representations specific to disease symptoms like leaf spots or yellowing patterns. This hierarchical learning enables the model to distinguish subtle variations between healthy and diseased tissue.

For coconut leaf images, these learned representations help the model identify patterns such as necrotic spots, discoloration regions, and damaged leaf structures.

The feature aggregation neck combines information from multiple layers of the backbone network. This process allows the model to integrate both fine-grained details and higher-level semantic features.

Feature aggregation is particularly useful when detecting small objects or localized disease symptoms on leaves. By combining multi-scale information, the model can recognize both large discoloration patterns and small spot-like lesions.

The detection head is the final component of the YOLOv8 architecture. It predicts bounding boxes and assigns class probabilities to detected objects.

Unlike earlier YOLO versions that relied on predefined anchor boxes, YOLOv8 uses an anchor-free detection mechanism. This approach simplifies the training process and improves detection flexibility when objects vary in size or shape.

For each detected region, the model predicts:

1. The bounding box location
2. The class label (Healthy, Leaf Spot, Yellowing, Pest Damage)
3. A confidence score indicating detection reliability

B. Model Selection

The YOLOv8 model family includes multiple variants designed for different computational requirements. These variants range from lightweight models suitable for mobile devices to larger models optimized for maximum detection accuracy. In this study, the YOLOv8s (small) variant was selected. This model offers a balance between detection performance and computational efficiency.

The main reasons for selecting YOLOv8s include:

1. Lower memory requirements compared to larger models
2. Faster inference speed
3. Reduced computational overhead
4. Suitability for deployment on mobile or edge devices

Because the intended application involves field-based disease detection using mobile cameras, selecting a lightweight model is important. A compact architecture allows the system to run efficiently without requiring high-end hardware.

C. Training Procedure

The training process was conducted using the Ultralytics implementation of YOLOv8 within a Python-based development environment. Transfer learning was used to accelerate the training process by

initializing the model with pre-trained weights. The dataset was divided into two subsets:

1. Training set: 80% of images
2. Testing set: 20% of images

During training, the model learns to associate visual features with the corresponding disease categories. The training algorithm updates network weights through repeated iterations using backpropagation and gradient-based optimization.

Important training parameters included:

1. Input image resizing for consistent processing
2. Batch-based learning for stable optimization
3. Multiple training epochs to improve model convergence

Throughout the training process, validation metrics were monitored to observe model learning behavior and prevent overfitting.

D. System Workflow

The final disease detection system follows a simple operational pipeline.

1. The user captures an image of a coconut leaf using a mobile device or camera.
2. The image is provided as input to the trained YOLOv8 model.
3. The model processes the image and identifies potential disease regions.
4. Bounding boxes are generated around detected areas.
5. Each detected region is labeled with the corresponding disease category.

The system output visually highlights the affected areas and provides the predicted health condition. This allows farmers or agricultural workers to quickly identify the type and location of disease symptoms.

Because YOLOv8 performs detection in a single network pass, the system is capable of producing results within a short time. This makes the approach suitable for real-time fieldbased diagnostics.

V. EXPERIMENTAL SETUP

To evaluate the effectiveness of the proposed coconut leaf disease detection system, a series of experiments were conducted using the YOLOv8s object detection model. The goal of the experimental setup was to train the model using labeled coconut leaf images and analyze its ability to identify and localize disease symptoms under realistic conditions. The entire implementation was performed using Python and the Ultralytics YOLOv8 framework. This framework provides tools for training, validating, and testing object detection models with minimal configuration. It also allows efficient utilization of computational resources during model training.

A. Software Environment

The development and training process was carried out in a Python-based environment using commonly used deep learning libraries. The Ultralytics YOLOv8 implementation was selected because it provides optimized training pipelines and supports transfer learning.

The main software components used in this study include:

1. Python programming language
2. Ultralytics YOLOv8 framework
3. PyTorch deep learning library
4. OpenCV for image processing
5. Kaggle dataset for training data

These tools enabled efficient model training and simplified the process of dataset handling, annotation interpretation, and performance evaluation.

B. Hardware Configuration

Training deep learning models requires adequate computational resources to process large numbers of images and perform multiple iterations of optimization. The experiments were conducted on a system capable of supporting the training requirements of the YOLOv8 architecture.

Typical hardware resources used for the experiments include:

1. Multi-core CPU for preprocessing and dataset loading
2. GPU acceleration was used when available to speed up model training.
3. Sufficient system memory to handle image batches during training

The YOLOv8s architecture was selected partly because it can be trained effectively even on moderate hardware resources compared with larger object detection models.

C. Training Configuration

Before starting the training process, the dataset was organized according to the standard directory structure required by the YOLO framework. Each image was associated with a corresponding annotation file containing bounding box coordinates and class labels. The dataset was divided into training and testing subsets using an **80:20 split**. The training dataset was used to learn feature representations, while the test dataset was reserved for evaluating the model's generalization performance.

Key training configurations included:

1. Image input size: resized to maintain consistent processing
2. Batch size: selected based on available hardware memory
3. Number of epochs: multiple training cycles to allow model convergence
4. Optimization algorithm: gradient-based optimization provided by the PyTorch backend During training, the model gradually learned to recognize patterns corresponding to different disease symptoms by adjusting its internal parameters.

D. Evaluation Metrics

To measure the performance of the trained model, several standard object detection metrics were considered. These metrics help quantify how accurately the model detects and classifies disease regions.

The primary evaluation indicators include:

Precision:

Precision quantifies detection reliability by measuring the fraction of positive identifications that were actually correct.

Recall:Recall represents the ability of the model to detect all relevant disease regions in an image. Higher recall means fewer missed detections.

Mean Average Precision (mAP) is a widely used metric for evaluating object detection models. It summarizes detection accuracy by considering both precision and recall across different detection thresholds.

Inference Speed: The time required for the model to analyze an image and generate predictions is also important. Faster inference speeds make the system more suitable for real-time field applications.

These metrics provide a comprehensive view of model performance in terms of detection accuracy, reliability, and operational efficiency.

VI. RESULTS

The model was trained using the YOLOv8s architecture within the Ultralytics framework. Several training parameters were configured to ensure stable model learning and efficient optimization.

Training configuration details are summarized below:

1. Input image size: 640 × 640 pixels
2. Batch size: 16 images per iteration
3. Number of epochs: 100
4. Optimization algorithm: Stochastic Gradient Descent (SGD)
5. Initial learning rate: 0.001
6. Framework: Ultralytics YOLOv8 implementation based on PyTorch

These parameters were selected to balance computational efficiency and detection performance while allowing the model to learn representative disease patterns from the dataset.

A. Detection Performance

The trained YOLOv8s model demonstrated effective performance in detecting coconut leaf diseases across the test dataset. The system was able to identify and localize disease symptoms such as fungal leaf spots, yellowing regions, and pest-related damage within the leaf images.

Bounding boxes generated by the model successfully highlighted affected areas while simultaneously assigning the correct disease category. The model showed stable detection behavior across images captured under different environmental conditions including variations in lighting, shadows, and natural backgrounds.

These results indicate that the YOLOv8s architecture can effectively learn visual patterns associated with coconut leaf diseases and apply this knowledge to detect symptoms in previously unseen images.

B. Quantitative Evaluation

To measure the performance of the trained model, several standard object detection metrics were calculated. These metrics include precision, recall, and mean Average Precision (mAP). Precision evaluates the proportion of correct detections among predicted detections, while recall measures the model's ability to identify all relevant disease regions in an image.

Metric	Value
Precision	0.89
Recall	0.86
mAP@0.5	0.91
mAP@0.5:0.95	0.74
Average inference speed	32 FPS

The obtained results indicate that the YOLOv8s model provides reliable detection performance on the coconut leaf dataset. The high mAP value suggests that the system can accurately identify disease symptoms while maintaining a balanced trade-off between precision and recall.

C. Model Robustness

Model robustness was evaluated by testing the trained network on images captured under different environmental conditions. The dataset included images with variations in lighting intensity, shadows, and complex plantation backgrounds.

Despite these challenges, the detection system maintained consistent performance. The model successfully identified disease regions even when leaves overlapped or when the background contained other vegetation elements.

This robustness demonstrates the ability of the YOLOv8s architecture to generalize beyond controlled image conditions and operate effectively in real agricultural environments.

D. Practical Implications

The results obtained in this study demonstrate that deep learning–based object detection can be effectively applied to identify coconut leaf diseases in plantation environments.By automatically identifying coconut leaf diseases, the proposed system can support farmers in detecting plant health problems at earlier stages.

Because the YOLOv8s model offers a balance between detection accuracy and computational efficiency, the system can potentially be integrated into mobile applications or portable diagnostic tools. Farmers could capture images of coconut leaves using smartphones and obtain immediate feedback regarding disease conditions.

Such technology can contribute to improved crop monitoring, better disease management practices, and increased agricultural productivity.

VII. CONCLUSION

This study presented an automated approach for detecting coconut leaf diseases using the YOLOv8s deep learning object detection model. The system was trained using a publicly available dataset containing approximately 1,500 images of coconut leaves representing four different health conditions: Healthy, Leaf Spot, Yellowing, and Pest Damage. The YOLOv8s architecture was selected due to its ability to perform fast and accurate object detection while maintaining relatively low computational requirements. Using the Ultralytics framework and a Python-based implementation, the model was trained through transfer learning and evaluated on a separate test dataset.

Experimental observations show that the trained model can successfully detect disease symptoms and localize affected regions in coconut leaf images captured under natural plantation conditions. The system demonstrated reliable detection performance even in the presence of varying illumination and complex backgrounds.

Because of its fast inference capability and lightweight architecture, the proposed system can potentially be integrated into mobile-based diagnostic applications.

Such tools could assist farmers in monitoring plant health directly in the field and identifying diseases at an early stage.

Future work may focus on expanding the dataset with additional disease categories and improving model performance through larger training datasets and advanced data augmentation techniques. Integrating the detection model into user-friendly mobile applications could further enhance its accessibility and practical usefulness for coconut farmers.

ACKNOWLEDGMENT

The authors express their sincere appreciation to the Department of Electronics and Communication Engineering for providing the academic environment and guidance required to complete this research work. The support and encouragement received from faculty members played an important role in the successful completion of this study.

The authors also acknowledge the contributors of the publicly available coconut leaf disease dataset hosted on Kaggle. Access to this dataset enabled the development and training of the deep learning model used in this research. The availability of open datasets continues to support research in agricultural technology and computer vision.

In addition, the authors would like to thank the developers of the Ultralytics YOLOv8 framework and the open-source deep learning community for providing accessible tools that facilitate experimentation and research in machine learning and object detection.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[2] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, 2017.

[3] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[4] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” Ultralytics Documentation, 2023.

[5] K. P. Ferentinos, “Deep Learning Models for Plant Disease Detection and Diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[6] S. Mohanty, D. Hughes, and M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection,” *Frontiers in Plant Science*, vol. 7, pp. 1419, 2016.

[7] Kaggle Dataset, “Coconut Leaf Disease Dataset,” .

[8] Coconut Development Board, Government of India, “Coconut Production and Development Programs in India,” 2025.

[9] G. Jocher, A. Chaurasia, and J. Qiu, “YOLOv8 by Ultralytics,” Ultralytics GitHub Repository, 2023.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[11] C. Szegedy et al., “Going Deeper with Convolutions,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[12] M. Too, L. Yujian, S. Njuki, and L. Yingchun, “A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification,” *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.

[13] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification,” *Computational Intelligence and Neuroscience*, 2016.

[14] P. Picon et al., “Deep Convolutional Neural Networks for Mobile Capture Device-Based Crop Disease Classification,” *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019.

[15] Food and Agriculture Organization (FAO), “Plant Disease Management and Agricultural Sustainability,” FAO Agricultural Report, 2020.