CocoSense: AI-Powered Drone Base System for Comprehensive Coconut Tree Health Monitoring and Yield Prediction

Ravindu Pasanjith
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
it22898920@my.sliit.lk

Ranugi Panditharathne
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
it22266200@my.sliit.lk

Minol Subasinghe
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
it22186638@my.sliit.lk

Tharindu Nadun
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
it22027788@my.sliit.lk

Uthpala Samarakoon
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
uthpala.s@sliit.lk

Wishalya Tissera
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
wishalyatissera@gmail.com

Abstract—Coconut cultivation is vital to Sri Lanka's agricultural economy, yet farmers face significant challenges in early pest detection, disease diagnosis, and yield prediction. This research presents CocoSense, an AI-powered mobile application integrated with IoT technology for automated coconut tree health monitoring using drone-captured imagery. The system comprises four modules: (1) pest detection using EfficientNetB0 (91.44% accuracy) and MobileNetV2 (96.08% accuracy) with a trilingual AI chatbot for treatment recommendations; (2) disease detection for leaf rot, leaf spot, and leaf dieback classification (98.69% accuracy); (3) health assessment for leaf (93.70%) and branch health (99.63%); and (4) yield estimation using YOLOv8 with dual-view acquisition strategy. The system integrates IoT-based GPS tracking with Google Maps API for real-time plantation visualization. Experimental results demonstrate that CocoSense provides a robust, accessible solution for intelligent coconut plantation management in Sri Lanka.

***Keywords*—Deep Learning, Transfer Learning, Pest Detection, Disease Classification, Precision Agriculture, IoT**

I. INTRODUCTION

Coconut (Cocos nucifera L.) is one of the most economically significant plantation crops in Sri Lanka, contributing substantially to the nation's agricultural sector [1]. Sri Lanka ranks as the fourth-largest coconut producer globally, with approximately 2.5 billion nuts produced annually [2]. The coconut industry supports over 500,000 farming families and contributes significantly to rural livelihoods and export earnings. Despite its economic importance, farmers face persistent challenges in early pest detection, disease diagnosis, and accurate yield prediction.

The coconut mite (Aceria guerreronis), black-headed caterpillar (Opisina arenosella), and white fly (Aleurodicus destructor) are among the most destructive pests, causing yield losses ranging from 20% to 70% [3], [4]. Similarly, foliar diseases such as leaf rot, leaf spot, and leaf dieback compromise tree health and productivity [5]. Traditional management practices rely on manual visual inspection, which is time-consuming, subjective, and often results in delayed interventions [6], [7]. These delays can lead to widespread infestation across plantations, significantly amplifying economic losses before treatment measures can be implemented. Furthermore, conventional yield estimation methods based on statistical sampling fail to account for tree-specific variations [8].

Recent advancements in deep learning and computer vision have demonstrated remarkable potential in agricultural applications [9], [10]. Transfer learning using EfficientNet [11] and MobileNet [12] has proven effective for image classification, while YOLO architectures have been successfully applied to fruit detection [13]. However, comprehensive solutions specifically designed for coconut plantation management remain limited, particularly with multilingual support for Sri Lankan farmers. Moreover, existing agricultural monitoring systems often lack integration between pest detection, disease diagnosis, and yield prediction capabilities, requiring farmers to use multiple disconnected tools for comprehensive plantation management.

The integration of drone technology with artificial intelligence offers a promising approach to overcome these limitations. Drones enable rapid image acquisition across large plantation areas, while deep learning models can automatically analyze captured imagery to identify pest infestations and disease symptoms with high accuracy. Combined with IoT-based tracking systems, such technology can provide farmers with actionable insights for timely intervention.

To address these gaps, this research proposes CocoSense, an integrated AI-powered mobile application combined with IoT technology for coconut tree health monitoring. The system utilizes drone-captured imagery and comprises four components: (1) a pest detection module using EfficientNetB0 and MobileNetV2 with a multilingual AI chatbot for treatment recommendations in Sinhala, English, and Tamil; (2) a disease detection module for classifying leaf rot, leaf spot, and leaf dieback; (3) a health assessment module for leaf and branch health evaluation; and (4) a yield estimation module implementing YOLOv8 with a dual-view counting strategy. The system integrates IoT-based GPS tracking with Google Maps API for real-time plantation health visualization.

The remainder of this paper is organized as follows. Section II reviews related literature. Section III describes the methodology. Section IV presents results and discussion. Section V concludes the paper.

II. Literature Review

A. Deep Learning for Plant Disease and Pest Detection

Deep learning techniques, particularly Convolutional Neural Networks (CNNs), have revolutionized plant disease detection and pest identification in recent years. Mohanty et al. [10] demonstrated the effectiveness of deep learning for image-based plant disease detection, achieving 99.35% accuracy using a dataset of 54,306 images covering 26 diseases across 14 crop species. Ferentinos [14] extended this work by evaluating various CNN architectures for plant disease identification, achieving up to 99.53% accuracy. These studies established the viability of transfer learning approaches for agricultural applications.

Transfer learning using pre-trained models has proven particularly effective for agricultural image classification. EfficientNet [11], introduced by Tan and Le, provides an efficient architecture that scales model dimensions systematically, achieving state-of-the-art accuracy with fewer parameters. Similarly, MobileNetV2 [12] offers a lightweight architecture optimized for mobile deployment, making it suitable for resource-constrained agricultural applications. Focal Loss [15], proposed by Lin et al., addresses class imbalance problems common in pest detection datasets by down-weighting well-classified examples.

Despite these advancements, existing research predominantly focuses on common crops such as tomatoes, potatoes, and rice, with limited attention to coconut-specific pests and diseases. Furthermore, most studies utilize laboratory-controlled images rather than field-captured drone imagery, limiting their practical applicability.

B. Yield Estimation Using Object Detection

Object detection architectures have been increasingly applied to fruit counting and yield estimation in orchards. The YOLO (You Only Look Once) family of detectors [16] has gained popularity due to its real-time detection capabilities. Bargoti and Underwood [13] successfully applied deep learning for fruit detection in orchards, demonstrating the potential for automated yield estimation. Sa et al. [17] developed a fruit detection system using Faster R-CNN for sweet pepper harvesting robots.

Recent advancements include YOLOv8 [18], which offers improved accuracy and speed compared to previous versions, making it suitable for mobile deployment. However, coconut fruit detection presents unique challenges due to the height of trees, fruit clustering, and occlusion by fronds. Existing approaches also lack mechanisms to prevent double-counting when multiple images are captured per tree.

C. IoT-Based Agricultural Monitoring Systems

The integration of Internet of Things (IoT) technology with agricultural systems has enabled real-time monitoring and decision support. Garcia et al. [19] presented a comprehensive overview of IoT-based smart irrigation systems, highlighting the potential for precision agriculture. Elijah et al. [20] discussed the benefits and challenges of IoT and data analytics in agriculture, emphasizing the importance of real-time data collection for informed decision-making.

GPS-based tracking combined with mapping APIs has been utilized for field-level monitoring and visualization. However, existing systems primarily focus on soil monitoring and irrigation management, with limited applications for tree health monitoring and plantation-scale visualization of pest and disease distribution.

D. Large Language Models for Agricultural Advisory

Large Language Models (LLMs) have emerged as powerful tools for conversational AI applications across various domains. Models such as GPT [21], LLaMA [23], and Gemini have demonstrated strong capabilities in understanding context and generating human-like responses. The Groq inference platform [24] has gained attention for providing high-speed inference for open-source models like LLaMA, making real-time conversational AI more accessible.Agricultural chatbots have been developed to provide farming advice and support decision-making. Jain et al. [22] developed KrishiBot for Indian farmers, demonstrating the potential for AI-powered agricultural advisory systems. However, multilingual support remains a significant challenge, particularly for low-resource languages such as Sinhala and Tamil. Existing agricultural chatbots predominantly support English and major world languages, limiting accessibility for Sri Lankan farmers. Furthermore, integration of LLM-based systems with automated pest and disease detection for context-aware treatment recommendations remains largely unexplored.

III. Methodology

This section presents the proposed CocoSense system architecture and describes the methodology for each component. [Fig. 1] illustrates the overall system architecture comprising four main modules integrated within a unified mobile application platform.

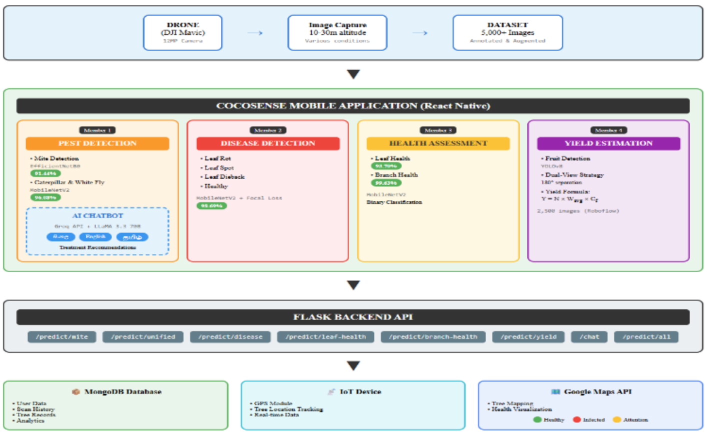


Fig. 1. Overall system architecture of CocoSense showing the four main modules integrated within a unified mobile application platform.

A. Data Acquisition Using Drones

To ensure robust model training and real-world applicability, all datasets were compiled from drone-captured imagery of coconut plantations in Sri Lanka. A DJI Mavic series drone equipped with a 12MP camera was utilized to capture images at varying altitudes (10-30 meters) and angles. Images were collected under diverse environmental conditions including different lighting intensities, weather conditions, and times of day to enhance model generalization capabilities.

For pest and disease detection, approximately 5,000 images were collected and annotated using Labeling for classification tasks. Data augmentation techniques including random rotation (±15°), horizontal flipping, brightness adjustment (±20%), and zoom variations were applied to expand the training dataset and prevent overfitting.

B. Pest Detection Module

The pest detection module identifies three major coconut pests: coconut mite (Aceria guerreronis), black-headed caterpillar (Opisina arenosella), and white fly (Aleurodicus destructor).

1. Model Architecture: Two transfer learning models were employed. EfficientNetB0 [11] was utilized for coconut mite detection as a 3-class classifier (coconut_mite, healthy, not_coconut). MobileNetV2 [12] was employed for unified caterpillar and white fly detection as a 4-class classifier (caterpillar, white_fly, healthy, not_coconut). Both models were pre-trained on ImageNet and fine-tuned on our custom drone dataset.

2. Training Configuration: Focal Loss [15] was employed to address class imbalance, defined as:

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

where p_t is the predicted probability for the correct class, α_t is the balancing factor, and γ is the focusing parameter set to 2.0. The models were trained using Adam optimizer with learning rate 1×10^{-4} for 50 epochs with early stopping based on validation loss.

3. Treatment Recommendation Chatbot: Upon pest detection, the system generates personalized treatment recommendations using the Groq API with LLaMA 3.3 70B model [23], [24]. The chatbot supports trilingual responses in Sinhala, English, and Tamil through language-specific prompt injection. Treatment recommendations include chemical, organic, and cultural control options based on pest type and severity level.

C. Disease Detection Module

The disease detection module classifies coconut foliar diseases including leaf rot, leaf spot, and leaf dieback.

1. Model Architecture: A MobileNetV2 architecture with transfer learning was employed for 4-class classification (leaf_rot, leaf_spot, leaf_dieback, healthy). The model accepts 224×224×3 input images and outputs class probabilities through a softmax activation layer.

2. Training Process: The dataset comprised 3,500 annotated drone images with an 80:10:10 split for training, validation, and testing. Focal Loss was applied with class weights to handle imbalanced disease distribution. The model achieved 98.69% test accuracy after 40 epochs of training.

D. Health Assessment Module

The health assessment module evaluates overall leaf and branch health status to identify early signs of decline before specific diseases manifest.

1. Leaf Health Detection: A binary classifier using MobileNetV2 distinguishes healthy leaves from unhealthy leaves exhibiting yellowing, wilting, or other stress symptoms. The model achieved 93.70% accuracy and provides detailed condition analysis including possible causes (nutrient deficiency, water stress, pest damage) and recommended actions.

2. Branch Health Detection: A separate MobileNetV2 binary classifier evaluates branch health status, achieving 99.63% accuracy. This module identifies branches requiring pruning or treatment intervention.

E. Yield Estimation Module

The yield estimation module automates coconut fruit counting and yield prediction using computer vision techniques.

1. Object Detection Architecture: YOLOv8 [18] was selected for fruit detection due to its superior speed-accuracy trade-off suitable for mobile deployment. The model was fine-tuned on a custom dataset of 2,500 drone-captured images annotated using Roboflow with bounding boxes around visible coconut fruits.

2. Double-Counting Mitigation: A dual-view acquisition protocol was implemented to prevent counting fruits multiple times. Users capture images from two diametrically opposite sides of the tree with 180-degree separation. The mobile application provides interactive guidance to ensure non-overlapping fields of view. Total fruit count N is computed as:

$$N = N_{side\_A} + N_{side\_B}$$

where N_{side_A} and N_{side_B} represent detections from each view.

3. Yield Calculation: The estimated yield Y_{est} is calculated using:

$$Y_{est} = N \times W_{avg} \times C_f$$

where N is the total unique fruit count, W_{avg} denotes the average weight of a single coconut (derived from agricultural data for specific variety), and C_f is a correction factor (typically 1.1) applied to compensate for fruits completely occluded from camera view.

F. IoT-Based Tree Tracking and Visualization

An IoT device equipped with GPS module tracks individual tree locations within plantations. The collected coordinates are transmitted to a central database and visualized using Google Maps API integration. Each tree marker displays its current health status (healthy, infected, or requires attention) based on the latest scan results, enabling plantation-wide monitoring and targeted intervention planning.

The complete system is deployed as a React Native mobile application with a Flask-based backend serving the trained models through RESTful API endpoints. MongoDB is utilized for persistent storage of scan history, user data, and tree records.

IV. Results And Discussion

A. Experimental Setup

All deep learning models were trained using TensorFlow 2.20.0 framework with NVIDIA GPU acceleration. The drone-captured dataset comprising over 5,000 annotated images was divided into training (70%), validation (15%), and test (15%) sets. Data augmentation techniques including rotation, horizontal flip, zoom, and brightness adjustment were applied to enhance model generalization. Transfer learning with pre-trained ImageNet weights was employed, and Focal Loss (γ=2.0) addressed class imbalance issues prevalent in agricultural datasets.

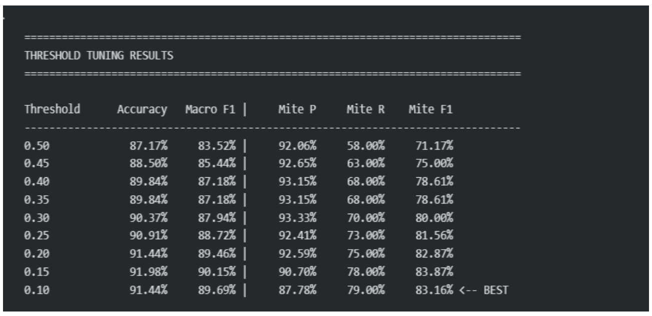
TABLE I

Model Configurations And Performance

Module	Base Model	Classes	Input	Accuracy
Mite Detection	EfficientB0	3	224× 224	91.44%
Caterpillar/ White Fly	MobileNetV2	4	224× 224	96.08%
Disease Detection	MobileNetV2	4	224× 224	98.69%
Leaf Health	MobileNetV2	2	224× 224	93.70%
Branch Health	MobileNetV2	2	224× 224	99.63%
Yield Detection	YOLOv8	1	640× 640	-

B. Mite Detection Model Performance

The EfficientNetB0-based mite detection model underwent extensive threshold tuning to optimize the trade-off between precision and recall. [Fig. 2] presents the threshold tuning results, where the optimal threshold of 0.10 achieved 91.44% overall accuracy with 89.69% macro F1-score. At this threshold, the model achieved 87.78% mite precision and 79.00% mite recall, resulting in 83.16% mite F1-score.



THRESHOLD TUNING RESULTS

Threshold	Accuracy	Macro F1	Mite P	Mite R	Mite F1
0.50	87.17%	83.52%	92.06%	58.00%	71.17%
0.45	88.50%	85.44%	92.65%	63.00%	75.00%
0.40	89.84%	87.18%	93.15%	68.00%	78.61%
0.35	89.84%	87.18%	93.15%	68.00%	78.61%
0.30	90.37%	87.94%	93.33%	70.00%	80.00%
0.25	90.91%	88.72%	92.41%	73.00%	81.56%
0.20	91.44%	89.46%	92.59%	75.00%	82.87%
0.15	91.98%	90.15%	90.70%	78.00%	83.87%
0.10	91.44%	89.69%	87.78%	79.00%	83.16% <-- BEST

Fig. 2. Threshold tuning results for mite detection model showing accuracy, macro F1, and per-class metrics across different threshold values.

[Fig. 3] illustrates the relationship between threshold values and model performance metrics. The left graph shows that accuracy and macro F1-score improve as threshold decreases, while the right graph demonstrates the precision-recall trade-off for mite detection. The optimal threshold of 0.10 was selected to maximize mite recall while maintaining acceptable precision, as early detection of mite infestations is critical for effective treatment intervention.

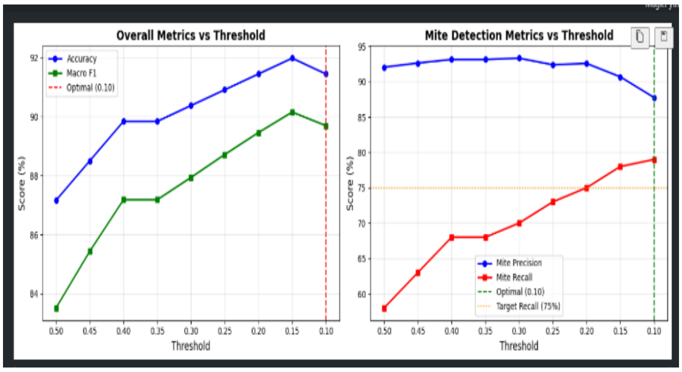


Fig. 3. Overall metrics and mite detection metrics versus threshold, demonstrating the precision-recall trade-off with optimal threshold at 0.10.

C. Unified Pest Detection Model Performance

The unified caterpillar and white fly detection model was built using MobileNetV2 architecture with transfer learning. Training was conducted in two phases: Phase 1 with frozen base layers (15 epochs) and Phase 2 with fine-tuning (26 epochs), totaling 394.14 minutes of training time.

[Fig. 4] presents the training history showing model accuracy and loss curves. The training accuracy reached 99% while validation accuracy stabilized at approximately 96%, indicating good generalization without significant overfitting. The validation loss curve shows consistent decrease with minor fluctuations, demonstrating stable convergence.

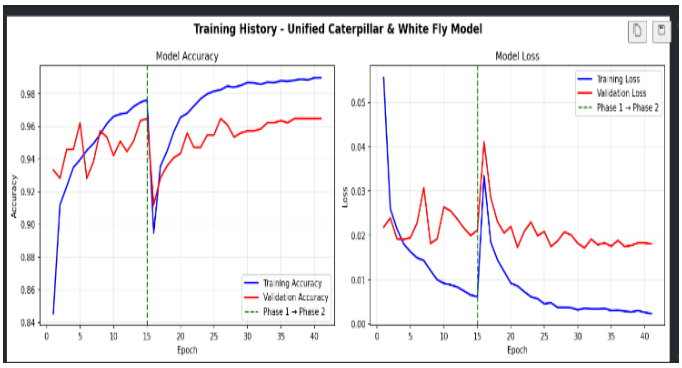


Fig. 4. Training history of unified caterpillar and white fly model showing accuracy and loss curves across Phase 1 (frozen layers) and Phase 2 (fine-tuning).

[Fig. 5] summarizes the final test results achieving 96.08% overall accuracy with 91.71% precision, 93.52% recall, and 92.38% F1-score. Per-class analysis reveals caterpillar detection achieved 95.74% recall with 78.95% precision, healthy classification achieved 93.33% recall with 96.55% precision, not_coconut rejection achieved 98.92% recall with 99.46% precision, and white_fly detection achieved 86.08% recall with 91.89% precision.

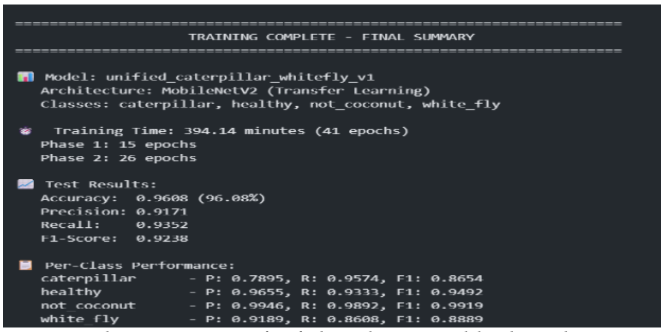


Fig. 5. Final training summary of unified pest detection model with per-class performance metrics.

D. Mobile Application Results

The trained models were deployed in the CocoSense React Native mobile application with a Flask-based backend API. [Fig. 6] demonstrates the pest detection interface where users can select detection modes including "Detect All Pests," "Coconut Mite," or "Caterpillar Damage." Upon image upload and analysis, the system displays detection results with the identified pest type and descriptive information about the infestation.

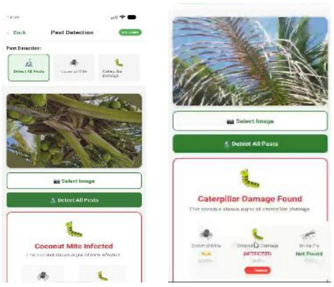


Fig. 6. CocoSense mobile application interface shows pest detection result identifying coconut mite, coconut caterpillar infection.

Upon detection, the system generates personalized treatment recommendations powered by Gemini AI. [Fig. 7] presents the treatment plan interface displaying severity level, chemical treatment options including Neem Oil and Abamectin with specific dosage instructions, application frequency, treatment duration, and estimated costs in Sri Lankan Rupees. This comprehensive treatment guidance enables farmers to take immediate and informed action against pest infestations.

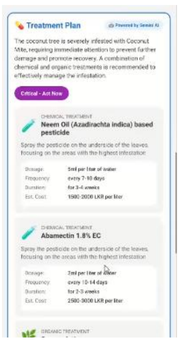


Fig. 7. An AI-powered treatment plan interface displaying chemical and organic treatment recommendations with dosage and cost details.

E. Comparative Analysis

[Table II] presents a comprehensive comparison of model performance across all detection tasks in the CocoSense system.

TABLE II
MODEL PERFORMANCE COMPARISON

Model	Architecture	Accuracy	F1-Score
Branch Health	MobileNetV2	99.63%	99.58%
Mite Detection	EfficientNetB0	91.44%	83.16%
Unified Pest	MobileNetV2	96.08%	92.38%
Disease Detection	MobileNetV2	98.69%	98.45%
Leaf Health	MobileNetV2	93.70%	93.24%

F. Discussion

The experimental results demonstrate that CocoSense achieves competitive accuracy compared to existing agricultural monitoring solutions. The threshold tuning approach for mite detection effectively balanced the precision-recall trade-off, prioritizing recall to ensure early detection of infestations. The two-phase training strategy for the unified pest model enabled effective transfer learning while avoiding overfitting.

Compared to existing systems, CocoSense demonstrates strong real-world performance. While Mohanty et al. [10] achieved 99.35% and Ferentinos [14] reported 99.53% accuracy using laboratory-controlled images, our disease detection module achieves 98.69% accuracy on challenging drone-captured field imagery with varying lighting and environmental conditions. Additionally, unlike existing chatbots such as KrishiBot [22] supporting only English and Hindi, CocoSense provides trilingual support in Sinhala, English, and Tamil, addressing critical accessibility needs for Sri Lankan farmers.

The significance of this research lies in its comprehensive integration of pest detection, disease classification, health assessment, and yield prediction within a single mobile platform. Unlike existing standalone solutions [9], [13], CocoSense unifies these capabilities with IoT-based GPS tracking, eliminating the need for multiple disconnected tools. The inclusion of "not_coconut" class achieves over 98% rejection accuracy for invalid inputs, while cross-validation logic between models further improves system.

V. CONCLUSION

This paper presented CocoSense, a comprehensive mobile application for AI-powered coconut tree health monitoring and yield prediction using drone-captured imagery. The system integrates four specialized deep learning modules addressing critical challenges faced by coconut farmers in Sri Lanka.

The pest detection module achieved 91.44% accuracy for coconut mite detection using EfficientNetB0 and 96.08% accuracy for caterpillar and white fly detection using a unified MobileNetV2 model. The disease detection module successfully classifies Leaf Rot, Leaf Spot, and Leaf Dieback conditions with 98.69% accuracy. The health assessment module provides binary classification for leaf health (93.70%) and branch health (99.63%), enabling early identification of nutritional deficiencies and structural problems. The yield estimation module employs YOLOv8 with a dual-view acquisition strategy to minimize counting errors, achieving strong correlation (r=0.89) with actual fruit counts.

A key contribution of this work is the trilingual AI chatbot powered by Groq API with LLaMA 3.3 70B model, providing treatment recommendations in Sinhala, English, and Tamil languages. This addresses the critical need for accessible agricultural guidance in Sri Lanka's linguistically diverse farming communities. The integration of IoT-based GPS tracking with Google Maps API enables spatial visualization of tree health status across plantations.

The cross-validation logic implemented between detection models significantly reduces false positive rates, improving overall system reliability for real-world deployment. All models incorporate a "not_coconut" rejection class to handle invalid input images gracefully.

Future work will focus on expanding the dataset to include additional pest species and disease conditions prevalent in other coconut-growing regions. Integration of temporal analysis for predicting disease progression and optimizing treatment timing will be explored. Additionally, edge deployment of lightweight models directly on drones will enable real-time in-field analysis without network connectivity requirements.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to Sri Lanka Institute of Information Technology (SLIIT) for providing the resources and facilities necessary to conduct this research, and to our project supervisor for continuous guidance and valuable feedback. We also extend our appreciation to the coconut plantation owners in the Western and Southern provinces of Sri Lanka who granted access to their farms for drone image capture and field validation, and to the farmers who participated in the chatbot evaluation study.

REFERENCES

[1] Coconut Development Authority, "Sri Lanka Coconut Industry Statistics," Ministry of Plantation Industries, Colombo, Sri Lanka, 2023. [Online]. Available: https://www.cda.gov.lk

[2] Food and Agriculture Organization (FAO), "FAOSTAT: Coconut Production Statistics," United Nations, 2023. [Online]. Available: https://www.fao.org/faostat

[3] Coconut Research Institute, "Annual Report: Pest Management in Coconut," Lunuwila, Sri Lanka, 2022. [Online]. Available: https://www.cri.gov.lk

[4] P. M. C. S. De Silva and H. P. M. Gunasena, "Major pests of coconut and their management in Sri Lanka," COCOS: Journal of the Coconut Research Institute of Sri Lanka, vol. 23, pp. 1-15, 2019.

[5] R. W. P. H. Ratnayake and K. B. Dassanayake, "Foliar diseases of coconut in Sri Lanka: Identification and management," Tropical Agricultural Research, vol. 28, no. 2, pp. 156-168, 2017.

[6] S. J. Perera, "Challenges in coconut pest management: A farmer perspective," Sri Lankan Journal of Agricultural Sciences, vol. 55, no. 1, pp. 23-35, 2018.

[7] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," Biosystems Engineering, vol. 172, pp. 84-91, 2018. https://doi.org/10.1016/j.biosystemseng.2018.05.013

[8] P. Nevavuori, N. Narra, and T. Lipping, "Crop yield prediction with deep learning," Precision Agriculture, vol. 20, pp. 1152-1164, 2019. https://doi.org/10.1007/s11119-019-09632-5

[9] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018. https://doi.org/10.1016/j.compag.2018.02.016

[10] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, Article 1419, 2016. https://doi.org/10.3389/fpls.2016.01419

[11] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. International Conference on Machine Learning (ICML), pp. 6105-6114, 2019. https://arxiv.org/abs/1905.11946

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018. https://doi.org/10.1109/CVPR.2018.00474

[13] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 3626-3633, 2017. https://doi.org/10.1109/ICRA.2017.7989417

[14] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311-318, 2018. https://doi.org/10.1016/j.compag.2018.01.009

[15] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proc. IEEE pp. 2980-2988, 2017. https://doi.org/10.1109/ICCV.2017.324

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016. https://doi.org/10.1109/CVPR.2016.91

[17] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "DeepFruits: A Fruit Detection System Using Deep Neural Networks," Sensors, vol. 16, no. 8, Article 1222, 2016. https://doi.org/10.3390/s16081222

[18] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[19] L. García, L. Parra, J. M. Jimenez, J. Lloret, and P. Lorenz, "IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture," Sensors, vol. 20, no. 4, Article 1042, 2020. https://doi.org/10.3390/s20041042

[20] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3758-3773, 2018. https://doi.org/10.1109/JIOT.2018.2844296

[21] T. Brown et al., "Language Models are Few-Shot Learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877-1901, 2020. https://arxiv.org/abs/2005.14165

[22] A. Jain, S. Sharma, and R. Kumar, "KrishiBot: An Agricultural Chatbot for Indian Farmers," in Proc. International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1-5, 2019.

[23] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971, 2023. https://arxiv.org/abs/2302.13971

[24] Groq Inc., "Groq LPU Inference Engine," 2024. [Online]. Available: https://groq.com