Deployment of a Deep Learning Model for the Automated Diagnosis of Thai Rubber Leaf Diseases via the LINE Platform

Puriwat Lertkrai, Benjamin Chanakot, and Nattapong Kaewboonma*

Faculty of Management Technology, Rajamangala University of Technology Srivijaya, Nakhon Si Thammarat, Thailand
Email: puriwat.l@rmutsv.ac.th (P.L.); benjamin.c@rmutsv.ac.th (B.C.); nattapong.k@rmutsv.ac.th (N.K.)
*Corresponding author

Manuscript received December 24, 2024; revised February 19, 2025; accepted April 9, 2025; published July 9, 2025

***Abstract*—This research presents an approach for real-time detection of rubber leaf diseases using the YOLOv8 deep learning model, integrated with the LINE messaging platform. The system enables users to submit rubber leaf images via LINE, where they are processed by an Artificial Intelligence (AI)—powered backend hosted on Heroku, utilizing Docker and Flask for scalability and efficiency. The YOLOv8 model was trained on a dataset comprising three classes: Healthy, New Disease, and Powdery Mildew. It achieved an overall mAP50 of 57.9% and an mAP50-95 of 42.8%, demonstrating strong performance in detecting healthy leaves (mAP50: 98.3%) but lower accuracy in identifying Powdery Mildew (mAP50: 22.4%), likely due to significant class imbalance. User testing involved 50 beta testers who submitted 500 images through the chatbot, yielding a detection accuracy of 72.4%, a misclassification rate of 5.4%, and an average response time of 2.5 seconds. Key performance metrics included a macro-average precision of 78.2%, recall of 73.0%, and an F1 score of 75.5%. User feedback highlighted satisfaction with the system's ease of use and response speed, though improvements were suggested in handling misclassifications and providing treatment recommendations. These results indicate that the system offers a robust and scalable solution for rubber leaf disease detection, with potential for further optimization.**

***Keywords*—image data processing, image data diagnosis, image detection, deep learning, rubber leaf disease**

I. Introduction

The Hevea brasiliensis Muell, commonly known as the rubber tree, is a vital component of both the agricultural and timber sectors, particularly in Southeast Asia. Originally native to the Amazon Basin, it was introduced to Southeast Asia in the early 20th century, where it now thrives as a major plantation crop. Key rubber-producing countries such as Indonesia, Malaysia, and Thailand collectively supply approximately 72% of the world's rubberwood. Rubber trees are typically tapped for latex for 25 to 30 years before being harvested for their wood. Rubberwood is highly valued in woodworking due to its favorable properties, light colour, and cost-effectiveness, making it a popular alternative to other tropical hardwoods in the furniture and construction industries. Despite being a by-product of latex production, rubberwood has become a significant export commodity, particularly in Malaysia, where it plays a crucial role in the national economy. The ongoing expansion of rubberwood plantations and advances in Hevea clones are expected to sustain supply levels and meet the growing demand for rubberwood products. However, challenges remain, particularly regarding the availability of high-quality logs and the effective management practices among smallholders [1].

The rubber industry has evolved significantly, especially following the early 20th-century boom driven by trading companies [2]. Natural rubber remains globally significant due to its widespread applications across various sectors. Extensive studies by [3] have examined trends in natural rubber production, consumption, and pricing both in Association of Southeast Asian Nations (ASEAN) countries and globally. Rubber plantations are crucial in agriculture, providing latex and supporting livelihoods [4, 5]. However, their expansion has negatively impacted biodiversity and ecosystems [6], with evidence of reduced ecosystem functionality, including lower biomass and plant diversity compared to forests. Leaf diseases also affect rubber production; in China, for example, climate factors and powdery mildew have reduced latex yields by 20% [7], while in Cameroon, diseases caused by *Fusarium oxysporum* and *Pestalotiopsis microspora* have led to similar reductions [8]. *Corynespora* leaf fall, caused by *Corynespora cassiicola*, is particularly severe, reducing yields by up to 45% [9]. Effective management of foliar diseases such as crown rust and leaf spot is essential for minimising yield losses and maintaining crop quality [10].

Global climate change, along with rising disease and pest outbreaks, poses significant risks to rubber production. A shortage of plant disease specialists and limited access to advanced diagnostic tools exacerbate these challenges for rubber farmers, who often lack the expertise for timely interventions. Consequently, there is a critical need for simple and effective tools to help farmers manage these issues. The widespread use of camera-equipped mobile devices has spurred the development of numerous mobile applications for image analysis, including those focused on agricultural and plant diseases, available on platforms like the Google Play Store. Several applications use artificial intelligence for disease diagnosis; for instance, Planty-Plant Disease Detector [11] allows users to capture images of affected plants for analysis, providing diagnostic results and recommendations. Similarly, Plantix-Your Crop Doctor [12] aids in identifying pests and diseases across 30 major crops, detecting over 400 types of plant damage. Despite these advancements, automated plant disease diagnosis apps remain limited, with many designed for specific countries' plant species and diseases, addressing diverse needs and purposes.

Developing and maintaining a standalone mobile application can be challenging due to potential variations in behaviour based on the user's device OS version. This study explores the development of a system that leverages the LINE mobile application, a widely used social communication platform in Thailand, to avoid Operating System (OS) version dependency. The system employs a LINE Bot—an automated chatbot integrated into a LINE account—for diagnosing rubber leaf diseases. This bot serves as an automated interface for user interaction, with information sent to the LINE Bot processed by a disease diagnosis engine that detects disease indicators in images. While previous research, such as that by [13], has examined various image processing techniques for object detection, contemporary studies emphasise deep learning methods using neural networks. Training these models involves adjusting weights to minimise a loss function, typically binary cross-entropy for binary classification or categorical cross-entropy for multi-class problems. For detection tasks requiring both classification and localisation, a regression loss is combined with cross-entropy loss.

This paper extends our previous research [14], which evaluated object detection techniques for diagnosing and predicting rubber leaf diseases. Our study found that YOLOv8 outperformed other deep learning models—Faster Region-based Convolutional Neural Networks (R-CNN), RetinaNet, and Mask R-CNN—in classifying rubber leaf diseases. We introduce a rubber leaf disease diagnosis system that utilizes images captured in field settings via the LINE mobile application. This user-centric system enables rubber farmers to detect diseased areas from field images without requiring sample preparation or specialized tools. Additionally, the system fosters a communication network between farmers and researchers, facilitating knowledge exchange and improving diagnostic accuracy. Our system provides preliminary disease diagnoses in real-time, operating continuously, 24 hours a day.

II. Literature Review

The rapid advancement of deep learning and image processing technologies has revolutionised plant disease detection, offering innovative solutions to enhance agricultural productivity and disease management. Traditional methods of diagnosing plant diseases relied heavily on manual inspections and basic image analysis via mobile applications. However, the introduction of sophisticated deep learning models has enabled more accurate, efficient, and real-time detection capabilities. This transition from rudimentary diagnostic tools to advanced AI-driven systems marks a pivotal development in precision agriculture, empowering farmers to respond to disease outbreaks more effectively and sustainably. In this section, we review the evolution of plant disease detection technologies, focusing on the progression from early mobile applications to the integration of cutting-edge deep learning models, such as YOLO and Mask R-CNN, into modern agricultural practices.

The implementation of deep learning models for the automated diagnosis of Thai rubber leaf diseases through the LINE platform aims to improve agricultural productivity and disease management. Early mobile applications, such as Rice Doctor and riceXpert, provided essential diagnostic tools by offering farmers text-based information and basic image analysis. However, these applications lacked advanced interactivity and automation [15, 16]. With the rise of deep learning, more sophisticated models like Convolutional Neural Networks (CNNs) have been adopted, enabling more precise and faster disease detection from images. Specifically, the YOLO (You Only Look Once) model, particularly its YOLOv8 version, represents a significant advancement in the field, offering real-time object detection capabilities that surpass the speed and accuracy of models such as Faster R-CNN and RetinaNet [17, 18].

One notable example is the LINE Bot-based rice disease detection platform, which leverages YOLOv3 to assist Thai farmers by diagnosing diseases directly from images taken in the field. This system not only provides real-time diagnoses but also facilitates communication between farmers and specialists, enhancing overall disease management [13]. Similar advancements have been made with the Mask R-CNN model, which extends object detection capabilities by incorporating instance segmentation, making it a powerful tool for identifying and localizing plant diseases within images [19]. Furthermore, refining training datasets and improving model parameters have been crucial in enhancing detection accuracy. For example, systematic data augmentation and the removal of noisy data have improved the performance of models like YOLOv3, making them more reliable under diverse environmental conditions [20]. Additionally, research into IoT-based systems for plant disease monitoring has gained momentum, with real-time data from sensors and cameras being analyzed by AI models to provide continuous monitoring and early warning systems for disease outbreaks [21].

III. Materials and Methods

In this section, we outlined the process for preparing the training data for rubber leaf disease detection. Section III-B covers the training of the object detection model and the refinement process. Finally, Section III-C provides an overview of the LINE Bot system, followed by detailed explanations of its components.

A. Data Preparation

The Rubber Leaf Disease Dataset was meticulously compiled, consisting of 734 rubber leaf samples collected from various sources, including the Rubber Authority of Thailand, the Rubber Research Center, and the Faculty of Natural Resources at Prince of Songkla University. Each sample was carefully evaluated by domain experts with specialized knowledge of rubber leaf diseases, ensuring accurate labeling and classification. The classification of different rubber leaf diseases was based solely on their distinct external visual characteristics. Images were systematically captured in natural settings, with many samples photographed using a variety of smartphone models with different operating systems. This study focused on three specific categories of rubber leaves: those affected by new diseases, those with powdery mildew disease, and healthy leaves [14], as shown in Table 1.

Our dataset consists of 734 images divided into three categories: 205 images of powdery mildew disease, 259 images of a newly identified disease, and 270 images of healthy leaves. The images were then annotated by the rubber specialists or researchers trained by them, with the diseased areas clearly labeled. For the training phase, we organized the dataset into distinct subsets, allocating 70% of the data for training, 20% for validation, and 10% for testing. An examination of sample distribution across categories emphasizes the dataset's considerable scope. To further enhance the training set, we applied an image data augmentation technique, creating modified versions of existing images to increase the dataset size to 1536 images. This augmentation process serves a dual role, expanding the dataset while also acting as a regularization method to reduce the risk of overfitting.

Table 1. Typical symptoms of rubber leaf

Rubber Leaf	Typical Symptom
Healthy	Vibrant and disease-free rubber leaves are characterized by a rich green hue, showcasing a smooth surface devoid of any pathological markings, with well-defined veins readily discernible
Powdery Mildew Disease	At the onset, these lesions manifest as diminutive, dispersed silver-white spots, characterized by a web-like arrangement of hyphae, distributed across the leaf's upper or lower surface. Subsequently, these lesions progress to encompass the entire leaf. As the lesions mature, the powdery mildew spots transform into circular, ringworm-like formations with a white appearance. Concurrently, the leaf's surface undergoes desiccation and takes on a yellow hue, culminating in its eventual detachment.
New Disease	In the early stages of symptom development, a discernible bruised lesion emerges beneath the leaf, accompanied by the appearance of a circular yellowing on the leaf's upper surface within the same vicinity. Subsequently, this area undergoes expansion, with the wound edges darkening and transitioning into a brown, desiccated tissue that ultimately fades into pale white lines. The wound typically maintains a roughly circular shape, without a surrounding yellow halo; multiple points of affliction may converge, forming a larger wound. In cases of severe symptomatology, the leaves will exhibit yellowing and eventual abscission.

B. Rubber Leaf Disease Detection Using Deep Learning Models

An overview of the object detection model's training and refinement process is illustrated in Fig. 1. The process began with planning how to prepare the data and selecting the appropriate model. For simplicity, this paper refers to the deep learning model as "the model." The initial planning phase was informed by a review of relevant literature. Subsequently, models were selected, a training dataset was prepared, and experiments were conducted, leading to iterative refinements of the model's performance.

Data preparation began with gathering images of infected rubber leaves from actual rubber fields or by cultivating infected leaves in a controlled environment, such as a paddy field. The data collection process was influenced by seasonal factors. The Rubber Authority of Thailand's Rubber Research Center (hereafter referred to as rubber specialists) was responsible for ensuring data quality. They provided, examined, and validated the disease images, categorizing them based on typical symptoms displayed on the leaves. Rubber leaf diseases, such as Powdery Mildew Disease and New Disease, can generally be classified by common symptoms. However, different pathogens often cause distinct physical signs, which the specialists could visually differentiate.

In object classification, model performance is typically measured using accuracy, which compares the predicted class to the ground-truth class. However, object detection models must not only classify objects correctly but also localize them accurately within an image. To evaluate this, several performance metrics are commonly used. Precision, defined as TP / (TP + FP), measures the proportion of detected objects that are correct, whereas recall, calculated as TP / (TP + FN), assesses the model's ability to detect all relevant objects in an image [22]. Additionally, the F1-score, a harmonic mean of precision and recall, provides a single performance measure that balances both metrics.

Mean Average Precision (mAP) is another crucial metric for object detection, particularly mAP@0.5 and mAP@0.5:0.95, which evaluate detection performance across different Intersection over Union (IoU) thresholds. While mAP@0.5 considers a fixed IoU threshold of 0.5, mAP@0.5:0.95 averages performance over multiple IoU values from 0.5 to 0.95, offering a more comprehensive evaluation of model robustness [23, 24]. Beyond these metrics, it is essential that the predicted bounding box closely aligns with the actual object's location. IoU measures the ratio of overlap between the predicted and ground-truth bounding boxes relative to their union and is widely used for this purpose [25].

The PASCAL Visual Object Classes Challenge (VOC) challenge introduced the concept of mAP, which computes precision and recall for each object class at an IoU threshold of 0.5. Later, the Common Objects in Context (COCO) dataset introduced stricter evaluation standards with mAP@0.5:0.95, reflecting real-world performance more accurately [22, 26].

Additionally, inference speed is a key consideration for real-time applications and is typically measured by Frames Per Second (FPS), which determines how many images can be processed per second. Faster models with higher FPS are preferable for real-time chatbot applications, where latency is a concern. Together, these evaluation metrics provide a comprehensive assessment of an object detection model's accuracy, efficiency, and real-world applicability, guiding researchers in selecting the most suitable model for deployment.

$$Precision = \frac{TP}{(TP+\quad)} \tag{1}$$

$$Recall = \frac{TP}{(TP+FN)} \tag{2}$$

$$F = 2\,x\,\frac{Precision\ x\ Recall}{Precision+Rec} \tag{3}$$

$$mAP@0.5 = \sum_{i=1}^{N} AP_i\,(0.5) \tag{4}$$

$$mAP@0.5{:}\,0.95 = \frac{1}{10}\sum_{i=1}^{N} mAP@t \tag{5}$$

$$FPS = \frac{1}{Inference\ Time(seconds)} \tag{6}$$

C. Model Evaluation

For model selection, we trained YOLO models using our dataset, and the results are presented in Table 2.

Table 2. Yolo models metrics comparison

Metrics	YOLOv11n	YOLOv10n	YOLOv9t	YOLOv8n
mAP@0.5	0.6949	0.6459	0.6974	0.6863
mAP@0.5:0.95	0.4944	0.4740	`0.5015	0.4946
Precision	0.6520	0.6214	0.6552	0.6402
Recall	0.6889	0.6467	0.7290	0.7206
F1-Score	0.6700	0.6338	0.6901	0.6780
Inference Time	2.9056	2.6304	3.5094	6.2885
FPS	0.3442	0.3802	0.2850	0.1590
Model Size (MB)	5.2304	5.4980	4.4349	5.9718
Training Time (s)	1800.0	2188.8	3290.4	1526.4

An object detection model within a chatbot application necessitates careful evaluation of several critical factors, including accuracy, inference speed, and computational efficiency. The comparison of YOLO models is shown in Figs. 1–4.

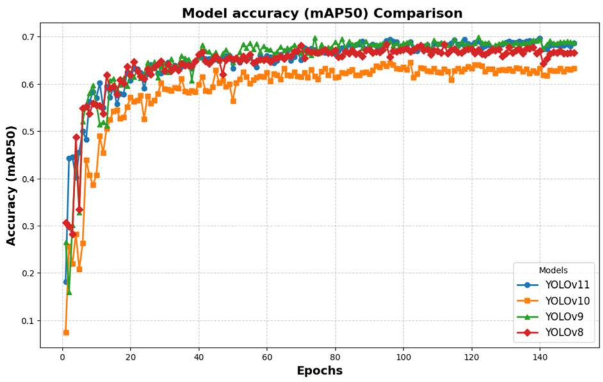

Fig. 1. mAP@0.5 comparison.

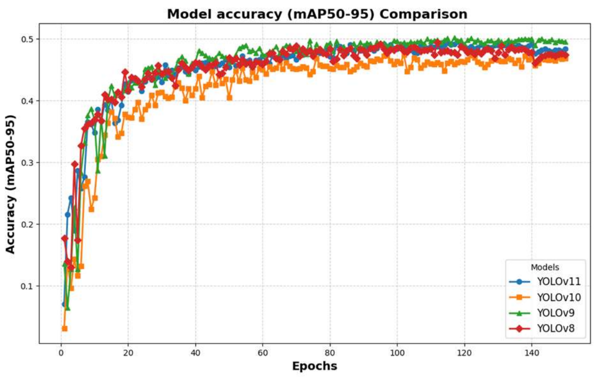

Fig. 2. mAP@0.5:0.95 comparison.

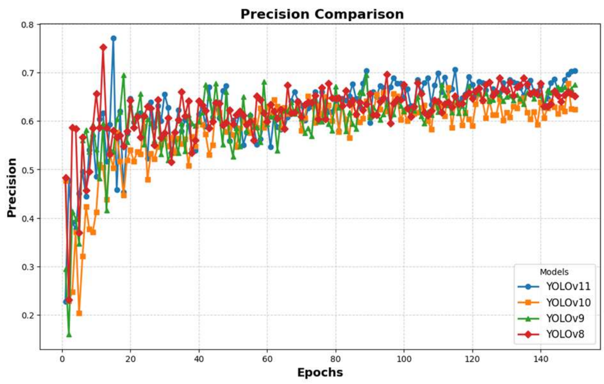

Fig. 3. Precision comparison.

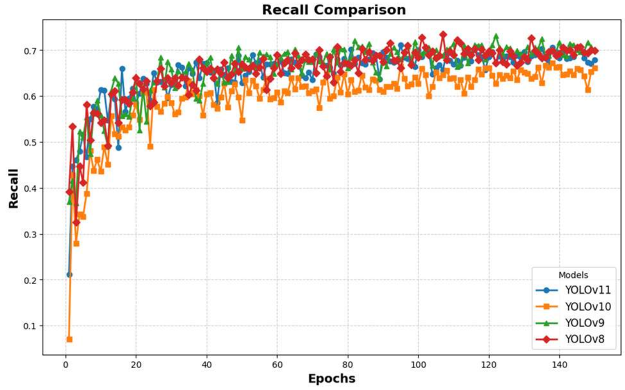

Fig. 4. Recall comparison.

Among the evaluated models, YOLOv8n emerges as the most suitable choice due to its optimal balance across key metrics. It achieves an mAP@0.5 of 0.6863 and an mAP@0.5:0.95 of 0.4946, demonstrating competitive detection performance while maintaining a strong balance between precision (0.6402) and recall (0.7206). Furthermore, its F1-score of 0.6780 affirms its robust detection capability, ensuring reliable performance in real-world chatbot applications. While YOLOv9t offers marginally higher accuracy (mAP@0.5 = 0.6974), its increased inference time and computational overhead render it less efficient for deployment. In contrast, YOLOv8n features a lightweight architecture, reduced processing time, and storage efficiency—crucial attributes for real-time chatbot operations [24, 25].

A key advantage of YOLOv8n is its suitability for real-time applications that demand low-latency responses. Despite having the highest inference time among the evaluated models (6.29 ms), its overall efficiency is enhanced by a compact model size of 5.97 MB, making it ideal for deployment in cloud-based or edge environments where memory constraints are a concern. Additionally, its training time of 1526.4 seconds, the shortest among the tested models, indicates a lower computational burden, allowing for faster retraining and adaptation to evolving datasets. Moreover, its seamless integration into chatbot frameworks via the Ultralytics package simplifies deployment, making it an attractive choice for researchers and developers [27].

YOLOv8n has also demonstrated high adaptability across various computer vision tasks, including object detection, image segmentation, and classification. Its ability to maintain high detection accuracy even with relatively small training datasets further enhances its applicability in research settings, where extensive labeled data may be unavailable [28]. While YOLOv9t achieves the highest detection performance, it comes at the cost of increased training time (3290.4 seconds) and higher inference latency (3.51 ms), which may hinder its suitability for chatbot interactions requiring immediate responses. Therefore, YOLOv8n presents the most balanced option, offering an optimal trade-off between accuracy, efficiency, and real-time processing capability for our dataset. An example of testing with unseen images is presented in Fig. 5.

Fig. 5. Example of rubber leafs disease detection with unseen images.

In the present study, the dataset was refined by expanding data diversity and eliminating low-quality or erroneous samples. Details of this refinement process are discussed in the experimental results section. Following these enhancements, the model was retrained and its performance rigorously evaluated with feedback from rubber disease specialists. Once the specialists approved the model's performance, it was deployed. This refinement process is iterative, and further improvements are anticipated based on the model's performance in real-world conditions, where rubber farmers will use the LINE Bot system in the field.

D. Design and Model Deployment

This section explains how the LINE Messaging API enables users to submit images of rubber leaves, which are then sent as detection requests to an API endpoint hosted on Heroku. The backend infrastructure, containerised with Docker, runs a Flask application that handles image processing and interacts with an AI model trained for disease classification.

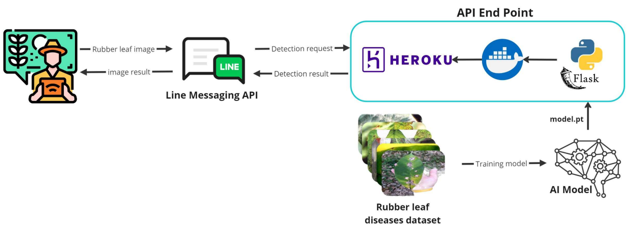

Fig. 6. LINE Messaging API architecture.

After the inference process is complete, the Flask application sends the detection results back to the LINE API, which then relays the information to the user as illustrated in Fig. 6. This architecture ensures scalability, efficiency, and provides a real-time, user-friendly solution for detecting rubber leaf diseases.

IV. Result and Discussion

The YOLOv8n model was trained on a plant health dataset containing three classes: Healthy, New Disease, and Powdery Mildew. The model achieved an overall mAP50 of 57.9% and an mAP50-95 of 42.8%. Class-wise, the model performed best on the Healthy class, with high precision (86.2%), recall (96.4%), and an mAP50 of 98.3%. Performance was moderate for the New Disease class, with a precision of 55.4% and recall of 58.9%, yielding an mAP50 of 53.1%. However, the model struggled with Powdery Mildew, exhibiting lower precision (40.1%) and recall (26.5%), with an mAP50 of just 22.4%. These results are likely influenced by significant class imbalance in the training dataset, which contains 1451 labeled instances for Powdery Mildew, 1298 for New Disease, and only 310 for the Healthy class. This imbalance may explain the high precision and recall for the underrepresented Healthy class, while the relative abundance of Powdery Mildew instances resulted in poorer performance for that category.

For testing on the LINE platform, 50 beta testers evaluated the rubber leaf disease detection model. The model demonstrated promising results, categorizing images into three classes: Healthy, Powdery Mildew, and New Disease. A total of 500 images were processed, yielding an overall accuracy of 72.4% and a misclassification rate of 5.4%. The model excelled in identifying New Disease, with 140 true positives, while common misclassifications occurred between Powdery Mildew and New Disease, highlighting visual similarities that impacted performance, as shown in Fig. 2.

Response time metrics indicated an average response time of 2.5 seconds, with the fastest responses at 1.3 seconds under low server load and a maximum of 4.1 seconds during peak usage. User feedback was overwhelmingly positive, with an average satisfaction rating of 4.5 out of 5. Most users found the chatbot easy to use, rated the response speed favorably, and expressed confidence in the prediction quality, particularly after removing background classifications, as shown in Table 3.

Table 3. Beta testing results

Metric	Results
Total users	50
Total images processed	500
accuracy	72.4%
Average response Time	2.5 seconds
Misclassification Rate	5.4% (27 images)
User Satisfaction	4.5/5
Precision (Macro avg)	78.2%
Recall (Macro avg)	73.0%
F1 Score (Macro avg)	75.5%

Constructive feedback focused on the need for improved clarity in misclassifications, particularly between “Powdery Mildew” and “New Disease.” Users expressed a desire for additional resources regarding treatment options for detected conditions. These insights suggest potential avenues for enhancing model usability and user experience in future iterations.

The beta testing results table provides an overview of key performance metrics for a deep learning model designed for leaf disease diagnosis. The test involved 50 users and processed a total of 500 images, providing a reasonable sample size to evaluate the model’s functionality in identifying and classifying diseases. The model achieved an accuracy of 72.4%, indicating that 72.4% of its predictions were correct. Although this represents a promising initial performance, there is room for improvement, particularly if higher accuracy is essential for practical deployment in agricultural settings where precision is critical.

Moreover, the model demonstrated an average response time of 2.5 seconds, which highlights its efficiency—a key factor for real-time applications. Additionally, the misclassification rate was relatively low, at 5.4%, suggesting that the model is effective in reducing errors. Nevertheless, further reducing this rate would enhance the model’s reliability. In terms of user experience, satisfaction scored high at 4.5 out of 5, indicating that the model was generally well-received, especially regarding ease of use, reliability, and practical benefits from an end-user perspective.

Regarding the model’s classification performance, precision on a macro average basis was recorded at 78.2%, reflecting the model’s capacity to minimize false positives. This attribute is particularly important in scenarios where false positives—incorrectly identifying healthy leaves as diseased—could result in unnecessary interventions. Furthermore, the recall rate, at 73.0% (macro average), reflects the model’s ability to identify true positives or correctly detect diseased leaves, which is essential to prevent undiagnosed infections from spreading. The F1 score, which balances precision and recall, was calculated at 75.5%, indicating a reasonable trade-off between detecting true positives and avoiding false positives.

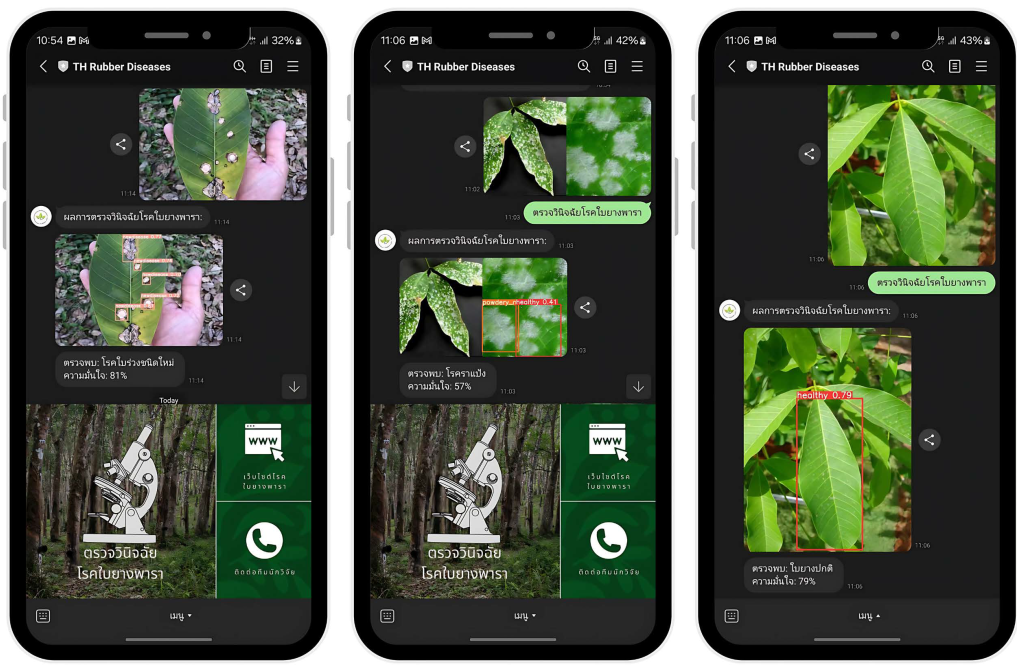

Fig. 7. Example of chatbot rubber disease detection results using LINE platform.

The beta testing results suggest that the model performs at a satisfactory level, exhibiting moderate accuracy, high user satisfaction, and a balanced F1 score. However, there remains potential to improve accuracy, recall, and misclassification rates to enhance the model's reliability for practical use in detecting leaf diseases. The efficient response time of 2.5 seconds further supports its suitability for real-world applications, where timely detection and action are crucial. Nonetheless, it is important to acknowledge certain limitations of this beta test. For example, the devices used for testing varied significantly in quality, especially in terms of the cameras and processors used. Furthermore, many of the testers were farmers who may lack the technical skills required to effectively utilize the rubber leaf disease diagnosis application, which could have influenced the evaluation results. Fig. 7 provides an example of the chatbot's detection results on the LINE platform, illustrating how users receive real-time feedback on disease identification. These results demonstrate a solid foundation for the model, with potential areas for enhancement identified through its performance metrics.

V. Conclusion

In conclusion, this research successfully demonstrated the feasibility of integrating a YOLOv8-based model into the LINE platform for real-time rubber leaf disease detection. The model achieved an overall accuracy of 72.4%, excelling in identifying "New Disease" but facing challenges in distinguishing between "Powdery Mildew" and "New Disease," likely due to visual similarities and dataset imbalance. Performance metrics, including a macro-average precision of 78.2%, recall of 73.0%, and an F1 score of 75.5%, highlight the model's solid foundation. The deployment architecture, utilizing the LINE API, Docker, Flask, and Heroku, enabled efficient and scalable real-time responses, with an average response time of 2.5 seconds. User testing with 50 beta testers yielded positive feedback, with a satisfaction rating of 4.5 out of 5. While the overall results were promising, improvements in handling misclassifications and providing clearer treatment guidance were identified as areas for future enhancement.

For future work, we plan to implement individual chat functionality and integrate a location tagging feature to better utilize disease spread information. Additionally, we are acquiring more diverse training data and refining the disease detection model to analyze a broader range of disease types, with the goal of covering most diseases commonly found in Thailand. As system usage continues to grow, both anticipated and unforeseen challenges may arise. To address this, we are developing a system maintenance assistant application to facilitate the management of the LINE Bot and chat community for system administrators. Furthermore, we are designing a feature to automatically extract and analyze dialogues from rubber specialists, using this information to enhance the model's performance further.

Conflict of Interest

The authors declare no conflicts of interest. All co-authors have reviewed and approved the contents of the manuscript, and there are no financial interests to disclose. We certify that this submission is original work and is not under review by any other publication.

Author Contributions

P.L. conceived and designed the analysis, collected the data, contributed data or analysis tools, trained pre-trained models, and performed the model analysis. B.C. performed the analysis, contributed data or analysis tools, and wrote the manuscript. N.K. conceived and designed the analysis, wrote the manuscript, and contributed to project coordination and supervision. All authors had approved the final version.

Funding

We sincerely acknowledge the generous support of the Fundamental Fund, Thailand Science Research and Innovation 2023 (Grant No. 180997), which played a pivotal role in facilitating our research.

Acknowledgment

Our gratitude extends to the Rubber Authority of Thailand and the Rubber Research Center for providing the essential rubber leaf image dataset, a critical resource without which this study would not have been possible. Furthermore, we deeply appreciate the invaluable technical support and expertise of Dr. Akadej Udomchaiporn, Head of the KMITL Digital Analytics and Intelligence Center, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, and Assoc. Prof. Dr. Narit Thaochan, Faculty of Natural Resources, Prince of Songkla University. Their expert guidance has been instrumental in shaping the direction and success of this research.

References

[1] W. Killmann. (April 2001). *The rubber plantation resource*. [Online]. Available: https://www.fao.org/4/ac126e/ac126e00.htm

[2] B. Linneweh, "Global trading companies in the commodity chain of rubber between 1890 and the 1920s," *Business History*, vol. 65, no. 5, pp. 863–879, Dec. 2019. doi: 10.1080/00076791.2019.1693544

[3] Y. C. Fong, A. A. Khin, and C. S. Lim, "Conceptual review and the production, consumption and price models of the natural rubber industry in selected ASEAN countries and world market," *Asian Journal of Economic Modelling*, vol. 6, no. 4, pp. 403–418, Jan. 2018. doi: 10.18488/journal.8.2018.64.403.418

[4] A. K. Singh *et al.*, "A global review of rubber plantations: Impacts on ecosystem functions, mitigations, future directions, and policies for sustainable cultivation," *The Science of the Total Environment*, vol. 796, 148948, Jul. 2021. doi: 10.1016/j.scitotenv.2021.148948

[5] Q. Yang, S. Yu, H. Zhong, T. Liu, E. Yao, Y. Zhang, H. Zou, and W. Du, "Gas products generation mechanism during co-pyrolysis of styrene-butadiene rubber and natural rubber," *Journal of Hazardous Materials*, vol. 401, 123302, 2021. https://doi.org/10.1016/j.jhazmat.2020.123302

[6] H. M. Jayathilake, C. De La Porte, J. W. J. Chang, D. P. Edwards, and L. R. Carrasco, "Transnational evidence for socio-economic factors affecting income and plantation expansion into natural habitats in smallholder rubber," *Resources Conservation & Recycling Advances*, vol. 18, 200161, May 2023. doi: 10.1016/j.rcradv.2023.200161

[7] D.-L. Zhai, P. Thaler, F. R. Worthy, and J. Xu, "Rubber latex yield is affected by interactions between antecedent temperature, rubber phenology, and powdery mildew disease," *International Journal of Biometeorology*, vol. 67, no. 10, pp. 1569–1579, Jul. 2023. doi: 10.1007/s00484-023-02515-2

[8] N.-N. Godswill, N. T. Bernadette-Sorele, D. D. C. Ulrich, Y. Aoudou, and N. N. A. I. Claire, "Effect of two fungicides on mycelial growth of fungi causing leaf blight and wilt of the rubber plant (Hevea brasiliensis Muell Arg.) in Cameroon," *Journal of Agricultural Studies*, vol. 8, no. 4, p. 111, Aug. 2020. doi: 10.5296/jas.v8i4.17476

[9] F. Umoh and E. L. Fashoranti, "Corynespora leaf fall of *Hevea brasilensis*: Challenges and prospect," *African Journal of Agricultural Research*, vol. 13, no. 40, pp. 2098–2103, Oct. 2018. doi: 10.5897/ajar2018.13352

[10] J. I. Dietz, M. Schierenbeck, and M. R. Simón, "Impact of foliar diseases and its interaction with nitrogen fertilization and fungicides mixtures on green leaf area dynamics and yield in oat genotypes with different resistance," *Crop Protection*, vol. 121, pp. 80–88, Mar. 2019. doi: 10.1016/j.cropro.2019.03.017

[11] Planty: Plant disease detector—apps on Google play. [Online]. Available: https://play.google.com/store/apps/details?id=com.ethix.planty&pcampaignid=web_share

[12] Plantix—your crop doctor—apps on Google play. [Online]. Available: https://play.google.com/store/apps/details?id=com.peat.GartenBank&pcampaignid=web_share

[13] P. Temniranrat, K. Kiratiratanapruk, A. Kitvimonrat, W. Sinthupinyo, and S. Patarapuwadol, "A system for automatic rice disease detection from rice paddy images serviced via a Chatbot," *Computers and Electronics in Agriculture*, vol. 185, 106156, Apr. 2021. doi: 10.1016/j.compag.2021.106156

[14] N. Kaewboonma, P. Lertkrai, B. Chanakot, and J. Lertkrai, "Thai rubber leaf disease classification using deep learning techniques," in *Proc. 2023 6th Artificial Intelligence and Cloud Computing Conference,* 2023, pp. 84–91. doi: 10.1145/3639592.3639605

[15] LucidMobile. (2017). Rice Doctor. [Online]. Available: https://play.google.com/

[16] National Rice Research Institute. (2020). riceXpert. [Online]. Available: https://play.google.com/

[17] J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," arXiv Print, arXiv: 1804.02767, Jan. 2018. doi: 10.48550/arxiv.1804.02767

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," arXiv Print, arXiv: 1506.01497, Jan. 2015. doi: 10.48550/arxiv.1506.01497

[19] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, Jun. 2018. doi: 10.1109/tpami.2018.2844175

[20] K. Kiratiratanapruk, P. Temniranrat, A. Kitvimonrat, W. Sinthupinyo, and S. Patarapuwadol, "Using deep learning techniques to detect rice diseases from images of rice fields," *Lecture Notes in Computer Science*, pp. 225–237, Jan. 2020. doi: 10.1007/978-3-030-55789-8_20

[21] C. Mattihalli, E. Gedefaye, F. Endalamaw, and A. Necho, "Plant leaf diseases detection and auto-medicine," *Internet of Things*, vol. 1–2, pp. 67–73, Sep. 2018. doi: 10.1016/j.iot.2018.08.007

[22] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jun. 2014. doi: 10.1007/s11263-014-0733-5

[23] T.-Y. Lin *et al*., "Microsoft COCO: Common objects in context," in *Proc. Computer vision–ECCV 2014: 13th European conference, Lecture Notes in Computer Science*, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48

[24] R. Joseph, D. Santosh, G. Ross, and F. Ali, "You only look once: Unified, real-time object detection," arXiv Print, arXiv: 1506.02640, Jan. 2015. doi: 10.48550/arxiv.1506.02640

[25] G. Jocher, A. Chaurasia, and J. Qiu. (Jan. 01, 2023). YOLO by Ultralytics. GitHub. [Online]. Available: https://github.com/ultralytics/ultralytics

[26] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc.* 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 3296–3297. doi: 10.1109/CVPR.2017.351

[27] M. M. A. Parambil *et al*., "Navigating the YOLO landscape: A comparative study of object detection models for emotion Recognition," *IEEE Access*, vol. 12, pp. 109427–109442, Jan. 2024. doi: 10.1109/access.2024.3439346

[28] F. Hermens, "Automatic object detection for behavioural research using YOLOv8," *Behavior Research Methods*, vol. 56, no. 7, pp. 7307–7330, May 2024. doi: 10.3758/s13428-024-02420-5