Real-Time Plant Disease Detection Using YOLOv8 on a Low Resources Device

Usama Mohiuddin[†], Emad A. Mohammed[†,*]
[†] Wilfrid Laurier University, 75 University Ave W, Waterloo, ON N2L 3C5

Abstract

The agricultural industry faces challenges from plant diseases, threatening crop yields and food security. Early detection is crucial, especially in greenhouses where precision monitoring enhances plant health. This study develops a deep learning-based disease detection system on an autonomous robot for real-time identification and intervention. Initially, classification models like YOLOv8, ResNet, MobileNet, EfficientNet, and a Custom Convolutional Neural Network (CNN) were trained on the PlantVillage dataset, achieving high accuracy but lacking localization capability. To address this, YOLOv8 was used for both classification and localization. The trained YOLOv8 model, deployed on the ROSMASTER X3 PLUS robot with an edge device such as the NVIDIA Jetson Orin NX, achieved 90% accuracy using 100 testing images. The system aims to integrate an automated spraying mechanism for targeted treatment, reducing pesticide use and improving greenhouse sustainability. Additionally, a cloud-connected backend service is being developed to track plant health over time, logging detected diseases and treatment history to optimize intervention strategies.

Keywords: Plant Disease Detection, Greenhouse Automation, Deep Learning Models, YoloV8, Edge Computing, Jetson Orin NX

1. Introduction

Deep learning models have been utilized in many industrial applications [1]. However, they are underutilized in areas such as the agriculture sector. The agricultural sector is the backbone of many economies, yet it remains vulnerable to numerous challenges, with plant diseases ranking among the most severe [2]. These diseases, caused by pathogens such as fungi, bacteria, and viruses, can spread rapidly and devastate crops, leading to food shortages, price surges, and economic instability. Traditionally, farmers rely on manual inspection for disease identification, a method that is labour-intensive, subjective, and prone to inaccuracies. Early diagnosis is particularly challenging due to the subtle and overlapping visual symptoms of different diseases, which can lead to misclassification even among experienced agronomists. The issue is exacerbated in regions with limited access to agricultural expertise, resulting in delayed intervention and uncontrolled disease outbreaks.

Compounding these challenges are the scale of modern farms, environmental variability, and the increasing demand for sustainable farming practices. Many farms lack the resources to conduct frequent, large-scale disease monitoring, necessitating both efficient and scalable solutions. Advances in machine learning (ML) and computer vision offer promising approaches to automating plant disease detection, allowing for improved accuracy and rapid diagnosis. However, existing AI-based methods have significant deployment challenges. Many state-of-the-art models require cloud computing resources, leading to latency issues and dependence on stable internet connectivity, often unavailable in remote agricultural regions [3]. Additionally, on-device deep learning solutions face computational constraints, limiting their effectiveness in real-time inference scenarios.

This study presents the ROSMASTER X3 PLUS, an AI-powered robot for real-time plant disease detection and intervention. Equipped with an NVIDIA Jetson Orin NX, it performs on-device classification and localization, eliminating reliance on external computing. The

[*]Corresponding author email: emohammed@wlu.ca

robot autonomously patrols crops, captures high-resolution images, and detects diseases with low-latency inference. It also features a robotic arm for precise pesticide spraying, reducing overuse and optimizing resource efficiency. This research aims to develop an AI-driven precision agriculture system that integrates disease detection with targeted treatment, enhancing greenhouse automation and crop health management.

2. Literature Review

Study	Model(s)	Contribution	Advantages	Disadvantages
Joshi et al. (2025) [3]	YOLOv8 + Hybrid Data Augmentation	Precision diagnosis of tomato diseases using YOLOv8	Robust against class imbalances, high accuracy	Computationally expensive, lacks real-time object detection
Peng and Wang (2022) [4]	YOLOv5	Object detection for small plant disease spots	Improved small-object recognition, real-time detection	High computational cost, needs dataset-specific tuning
Rahman et al. (2025) [5]	MobileNet	Lightweight model for real-time plant disease detection	Faster inference, suitable for mobile and edge devices	Lower accuracy compared to deeper CNNs
Bhagat et al. (2024) [6]	Lite-MDC (lightweight CNN)	Real-time detection on pigeon pea crops with a new field dataset	Achieves 94.14% accuracy with only 2.2M parameters at 33 FPS	Focused on classification, not object detection or robotic deployment
Joseph et al. (2024) [7]	MobileNet, Inception, Xception	Real-time datasets for rice, wheat, and maize using complex backgrounds	Dataset includes disease stages, real-life images, and supports object detection	Focused on grains; limited robotic or real-time system integration
Khalid et al. (2023) [8]	YOLOv5	Detection of healthy and unhealthy leaves (Money plant, Bell Pepper) using exclusive and public datasets	Achieved 93% accuracy with real-time FPS; supports mobile deployment	Focused on binary classification, lacks robotic integration or multi-class scaling

Table 1. Comparative analysis of deep learning models for plant disease detection, highlighting models, datasets, key contributions, advantages, and limitations.

Over the past decade, deep learning-based plant disease detection has seen significant advancements, particularly with the widespread adoption of CNN-based classification models and object detection frameworks. The PlantVillage dataset [9] has played a crucial role as a benchmark for evaluating these models, providing a diverse set of labelled images of both healthy and diseased plant leaves. Early research primarily focused on CNN-based classifiers, such as VGG, Inception, and MobileNet (Table 1), which demonstrated high classification accuracy but could not localize diseased regions within an image. This limitation led to the adoption of object detection models, particularly YOLO (You Only Look Once), which integrates classification and spatial localization in a single inference step.

CNN-based models such as MobileNet demonstrated high classification performance but lacked real-time applicability for autonomous agricultural monitoring [5]. To address this, researchers transitioned toward object detection frameworks capable of simultaneously classifying and localizing diseased regions. Peng and Wang (2022) [4] applied YOLOv5 to PlantVillage images and achieved enhanced detection of small disease spots, although the model's high computational demand limited deployment on edge devices. Khalid et al. (2023) [8] addressed this challenge by using YOLOv5 for binary classification optimized for mobile deployment across various crops.

More recent developments, such as YOLOv8, improve accuracy and inference speed with a lighter architecture suitable for real-time plant disease detection. Joshi et al. (2025) [3] demonstrated its performance using hybrid data augmentation techniques on the PlantVillage dataset, achieving high classification accuracy. However, its deployment on low-power edge devices remained underexplored. To address real-world deployment challenges, Bhagat et al. (2024) [6] introduced Lite-MDC, a compact CNN designed for real-time performance on resource-constrained hardware. Joseph et al. (2024) [7] focused on dataset realism by capturing disease images under varying field conditions, improving the generalizability of trained models.

While these works emphasize efficient models and realistic data collection, they do not integrate robotic intervention or operate fully within controlled environments like greenhouses. In contrast, our approach builds upon this foundation by leveraging YOLOv8 and the PlantVillage dataset, deploying it on an edge device mounted on a mobile robot. This combination enables real-time detection, spatial localization, and scalability for precision agriculture applications in controlled and semi-controlled environments.

The literature can be grouped chronologically to illustrate the evolution of approaches:

(1) **Initial Studies (2015–2017):** These focused on traditional CNN architectures like ResNet and VGG, establishing benchmarks for plant disease classification.
(2) **Efficiency-Driven Models (2018–2020):** EfficientNet and MobileNet were introduced to address the challenges of deploying deep learning models in edge environments.
(3) **Real-Time Detection (2020–Present):** The emergence of YOLO-based object detection enabled simultaneous disease classification and localization, improving real-world deployment.
(4) **Optimization for Deployment (2021–Present):** Recent efforts focused on optimizing YOLO models for edge devices, bridging the gap between academic research and real-world applications.

While studies such as Joshi et al. (2025) [3] and Peng and Wang (2022) [4] have demonstrated YOLO-based disease detection on PlantVillage images, these models were not optimized for real-time inference on edge devices like the Jetson Orin NX. Furthermore, prior research has not explored the integration of object detection with autonomous robotic intervention systems, such as targeted pesticide spraying using robotic arms.

This study evaluates deep learning models, including YOLOv8, for real-time plant disease detection and deploys an optimized version on the Yahboom ROSMASTER X3 PLUS. By optimizing YOLOv8 for edge computing on the Jetson Orin NX, it enables scalable AI-driven disease monitoring and intervention in greenhouse environments.

3. Data

The dataset used in this study is a condensed version of the publicly available PlantVillage Dataset [9], which was sourced from Kaggle. The original dataset comprises 54,305 images across 13 crop species, including apples, blueberries, grapes, oranges, peaches, peppers, potatoes, soybeans, strawberries, and tomatoes. These images are classified into healthy and diseased categories, with each disease further labelled according to its corresponding crop. The dataset is widely recognized for its high-quality labelled data, making it a benchmark dataset for plant disease classification tasks.

This study extracted a subset of the dataset to focus on three primary crops: PepperBell, Tomato, and Potato. These crops were selected based on their prevalence in controlled agricultural environments and susceptibility to common plant diseases. The resulting dataset includes 20,639 images, categorized into infected and healthy samples. Table 2 provides a breakdown of the dataset distribution.

Crop	Infected Samples	Healthy Samples
PepperBell	997	1478
Tomato	14525	1591
Potato	2000	152

Table 2. Distribution of infected and healthy samples for each crop.

Each image in the dataset is RGB, with a resolution of 256x256 pixels, making it well-suited for deep learning models such as CNN's. The dataset was pre-split into training,

validation, and testing subsets, following an 80:10:10 ratio. This split ensures a balanced evaluation, with sufficient training data, while keeping separate validation and test sets for unbiased model assessment.

Despite being well-curated, the dataset presents real-world challenges that must be considered before deploying models in practical settings:

(1) **Controlled Environment Bias:** The images were captured under uniform lighting conditions, which differs from real-world agricultural environments where lighting variations, shadows, and occlusions exist
(2) **Limited Background Complexity:** The dataset primarily consists of isolated leaves on plain backgrounds, whereas real-world applications require models to detect diseases in cluttered backgrounds with multiple overlapping leaves.
(3) **Class Imbalances:** Some categories have significantly fewer healthy samples than diseased ones, which may affect model generalization and require data augmentation techniques.

To mitigate class imbalance in the dataset, particularly between infected tomato and healthy potato samples, we applied targeted data augmentation to underrepresented classes. Techniques included rotations, flips, brightness adjustments, and zooming to synthetically increase sample diversity. Class-aware sampling was also used during training to ensure balanced representation within batches. These strategies helped improve recall and F1-scores for minority classes and reduced model bias toward dominant categories.

Figure 1. Images from the PlantVillage dataset showcasing healthy and diseased leaves, including Tomato Yellow Leaf Curl Virus and Late Blight. These images highlight dataset variability, essential for robust YOLO model training and evaluation

Preprocessing was essential to prepare the data for training machine learning models effectively. Key preprocessing steps included:

(1) **Normalization:** All pixel values were scaled to the range [0, 1] to ensure consistent input distributions and stable training.
(2) **Augmentation:** Techniques such as random rotations (0°–30°), horizontal flipping, brightness adjustments, zooming (10–20%), and random cropping were applied to increase dataset diversity and prevent overfitting artificially.

(3) **Resizing:** Images were resized to 80×80 pixels to reduce computational overhead while retaining important visual features.
(4) **Shuffling:** The dataset was randomly shuffled to ensure balanced training batches and reduce bias. (Figure 1)

These preprocessing steps helped classification models generalize better to new, unseen images, particularly under varied lighting and environmental conditions.

Unlike classification models, object detection models such as YOLOv8 in object detection mode required bounding box annotations for identifying diseased regions within an image. The preprocessing steps for YOLOv8 object detection included:

(1) **Dataset Labeling:** Each image was manually annotated to generate bounding boxes around diseased leaf areas. Labels were converted to YOLO format (*class_id x_center y_center width height*), and a custom script automated annotation using adaptive thresholding and contour detection.
(2) **Bounding Box Generation:** Grayscale conversion was applied to simplify image processing, followed by adaptive thresholding to detect disease-affected regions. Contour detection was implemented to locate the largest leaf area, which was then enclosed in a bounding box.
(3) **Data Splitting:** The object detection dataset was structured into YOLO's standard format with separate `images/train`, `images/valid`, `labels/train`, and `labels/valid` folders. An 80:10:10 train-validation-test split was applied to ensure fair distribution.
(4) **Data Augmentation:** YOLO-specific augmentation techniques such as random cropping, scaling, brightness adjustments, and horizontal flips were applied to enhance model robustness under diverse real-world conditions.
(5) **Final Dataset Preparation:** The labelled dataset was stored in YOLO-compatible format, and a `data.yaml` file was generated to define the dataset structure for YOLOv8 training.

To optimize deep-learning models, this study leveraged CNN-based automatic feature extraction, eliminating the need for manual feature engineering [6]. Pretrained models such as YOLOv8, EfficientNet, and ResNet effectively captured disease symptoms, including leaf texture, colour variations, and lesion patterns.

To address dataset limitations, augmentation techniques were applied to improve model generalization. Random rotations, brightness adjustments, and horizontal flips mitigated the effects of lighting variations and occlusions [7]. Random cropping and zooming also helped focus on disease patterns rather than background noise. Class imbalance was addressed using oversampling and synthetic augmentation to ensure fair learning across disease categories [3].

For the custom CNN model, preprocessing was optimized to enhance disease-specific feature extraction. Convolutional layers, max-pooling, and fully connected layers were fine-tuned based on best practices in deep learning [8]. The model effectively captured leaf vein structures, lesion distribution, and discoloration patterns.

YOLOv8 required bounding box annotations for object detection. A combination of adaptive thresholding and contour detection facilitated systematic annotation, reducing manual effort while ensuring consistency. The dataset was structured in YOLO's standard format, optimizing it for real-time inference [6].

This study ensured efficient model training and deployment on resource-constrained edge devices by applying targeted preprocessing and dataset structuring. Using a condensed PlantVillage dataset enabled accurate disease detection while addressing real-time inference challenges, aligning with prior research [7].

4. Methods

The primary research question driving this study is: **How can deep learning models be optimized and evaluated to identify the most accurate and efficient model for real-time plant disease detection on edge devices, such as the Jetson Orin NX?**

Multiple deep-learning models were trained and evaluated for plant disease classification and object detection to answer this question. The study hypothesizes that YOLOv8, when integrated into the Yahboom ROSMASTER X3 PLUS robot, will outperform other models, including custom CNNs and pre-trained architectures, in accuracy and real-time computational efficiency due to its edge-specific optimization. This hypothesis is based on the established advantages of YOLO models, which prioritize speed and accuracy, making them highly suitable for real-time agricultural applications [3, 4].

The study tests this hypothesis by conducting a comparative analysis of six models: Custom CNN, Inception, MobileNet, ResNet, EfficientNet, and YOLOv8. These models were trained on a Macbook Air M3 16GB RAM with a high-performance GPU to determine the best-performing model based on classification accuracy, computational efficiency, and inference speed, training each model with 50 epochs. The highest-performing model was further optimized and deployed on the Jetson Orin NX for real-time inference. The overall experimental workflow is illustrated in Figure 2, where the models undergo training, performance evaluation, and selection before deployment on an autonomous robot for real-time plant disease detection.

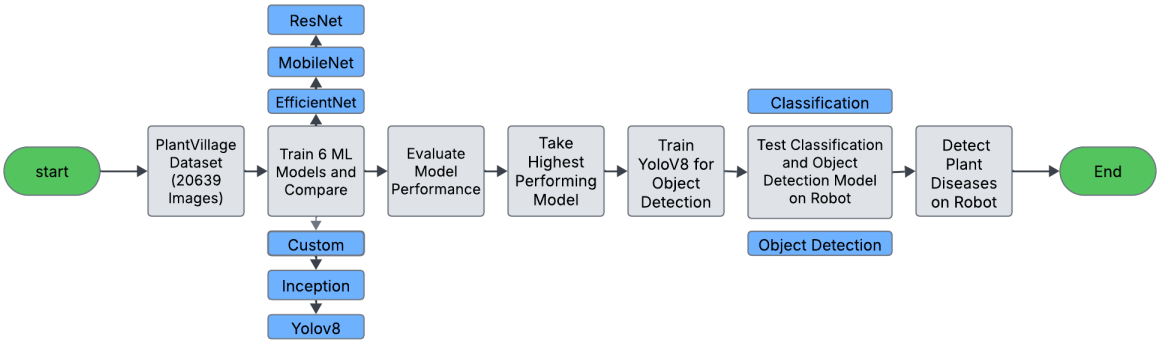


Figure 2. Overview of the experimental plant disease detection and classification workflow. The process begins with training six deep learning models—Custom CNN, Inception, MobileNet, ResNet, EfficientNet, and YOLOv8—on the PlantVillage dataset. Model performance is evaluated based on accuracy, precision, recall, F1-score, and inference time. The highest-performing model is selected for further training in object detection using YOLOv8. The optimized classification and object detection models are then deployed on the Yahboom ROSMASTER X3 PLUS robot for real-time plant disease detection, demonstrating the feasibility of deep learning in precision agriculture

This section outlines the two major experimental phases: comparing classification models and developing an object detection model for plant disease detection. The first phase trains and evaluates multiple deep-learning models for plant disease classification. The second phase builds upon these findings to develop an optimized object detection model to localize disease regions within an image.

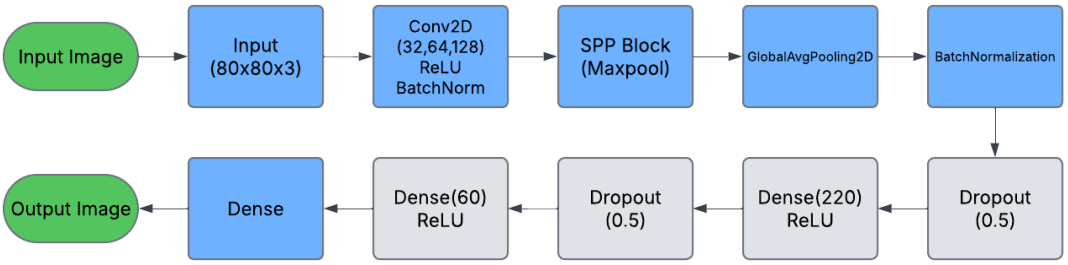


Figure 3. YOLOv8 classification model training flowchart. The input image is processed through convolutional layers to extract features. These features are refined using pooling and normalization steps, followed by dense layers to make predictions.

To determine the most accurate and computationally efficient classification model, 6 deep learning models were trained and evaluated using similar layers corresponding to Figure 3 . The following steps were undertaken:

(1) **Dataset Preparation:** The dataset consisted of plant images categorized into different disease classes splitting the dataset at 80% training and 20% validation. Each disease type was stored in separate classes, and a structured DataFrame was created to map file paths to corresponding labels. This format facilitated efficient data retrieval during training and testing.
(2) **Image Preprocessing:** Prior to training, preprocessing techniques were applied. Images were resized to a dimension of 80x80x3 to maintain uniformity. Pixel values were normalized scaling to the range [0,1] to ensure stable training. Techniques such as random rotation, horizontal flipping, zooming and brightness (10% - 20%) adjustments were applied to make the data more accurate.
(3) **Model Selection:** Six models were selected for comparison: Custom CNN, Inception V3, MobileNet, ResNet50, EfficientNet and YOLOV8
(4) **Model Training:** Each model was trained for 50 epochs using a batch size of 64 and the Adam optimizer with a learning rate 0.0005. Cross-entropy and early Stopping were employed to optimize the CNN model
(5) **Performance Evaluation:** The following models were evaluated using Accuracy, Loss, F1, Precision, Recall and Inference Time

Following the classification model evaluation, the next phase involved creating an object detection model for disease localization. YOLOv8 yielded the highest performance when compared to the classification models. The primary goal was to train the YOLOv8 model in object detection mode to identify and localize infected leaves within plant images. The following steps were performed:

(1) **Dataset Preparation:** The PlantVillage dataset, consisting of 20,904 images, was utilized for training and validation. The dataset was split into 80% training and 20% validation to ensure a balanced evaluation. A custom script was developed to generate bounding boxes around the diseased regions to automate the annotation process, thereby converting the dataset into a YOLO-compatible format. The annotated dataset was structured into separate directories for images and labels, with each label corresponding to a specific disease class.
(2) **Model Training:** The YOLOv8s model from Ultralytics was trained using the newly annotated dataset. The training process was configured with 20 epochs, an image size of 640×640 pixels, and a patience value of 10 to ensure optimal

performance while preventing overfitting. The model leveraged pre-trained weights to accelerate convergence and improve detection accuracy.

(3) **Model Testing:** Once trained, the YOLOv8 object detection model was deployed on the Jetson Orin NX to evaluate its real-time inference capabilities. The model was tested on various plant images to verify its ability to detect and localize diseased regions, ensuring it performed well under real-world conditions.

(4) **Performance Evaluation:** The performance of the trained YOLOv8 model was compared across different hardware configurations. Specifically, the inference speed and detection accuracy were evaluated on a 16GB RAM ThinkPad L14 and the Jetson Orin NX without CUDA optimization. The results provided insights into the model's computational efficiency, accuracy, and real-time feasibility when deployed on edge devices.

Following this structured approach, the YOLOv8 object detection model was successfully trained and deployed, demonstrating its effectiveness in automated plant disease localization. The results highlight its potential for real-time agricultural applications, with insights into performance trade-offs and areas for future optimization.

Although this study focuses on deploying existing deep learning models for real-time inference, future work will explore ways to adapt and optimize these models for resource-constrained environments. This includes pruning unimportant layers, applying post-training quantization, and evaluating lightweight architectures such as Tiny-YOLOv8 and YOLO-Nano. These techniques aim to reduce memory usage, model size, and power consumption—making them more suitable for edge devices like Jetson Nano or Raspberry Pi. Such efforts will enhance the practical scalability of the system and contribute toward real-time, energy-efficient AI solutions in agriculture.

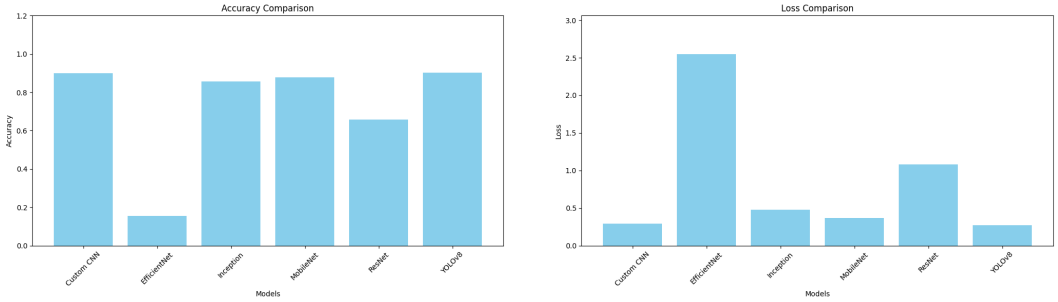
5. **Results**

The performance of various deep learning models for plant disease classification was evaluated based on accuracy, loss, precision, recall, F1-score, and inference speed. Table 3 summarizes the results, highlighting the trade-offs between accuracy and computational efficiency. YOLOv8 demonstrated the highest accuracy (98.69%) and lowest loss (0.0363), making it the most suitable for real-time plant disease detection. In contrast, EfficientNet performed poorly, suggesting it is unsuitable for this task. Custom CNN and MobileNet exhibited competitive performance but fell short of YOLOv8's efficiency in real-time inference.

Models	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1**	**Inference Time**
Custom	0.2873	0.9049	0.9205	0.9049	0.9054	0.54 ms/img
EfficientNet	2.5502	0.1555	0.0242	0.1555	0.0419	3.06 ms/img
Inception	0.4796	0.8557	0.8554	0.8557	0.8544	3.65 ms/img
MobileNet	0.3709	0.8777	0.8755	0.8777	0.8743	1.81 ms/img
ResNet	0.7816	0.7504	0.7508	0.7504	0.7388	7.48 ms/img
YOLOv8	0.0363	0.9869	0.9870	0.9869	0.9869	1.70 ms/img

Table 3. Performance comparison of classification models on the PlantVillage dataset. YOLOv8 achieves the highest accuracy and lowest loss, making it the best candidate for real-time applications.

The performance of these models during training is further analyzed in Figure 4 and Figure 5, where YOLOv8 exhibited rapid convergence and the lowest loss among all models. This suggests that the model effectively learned the features necessary for plant disease classification, unlike EfficientNet, which struggled significantly.



(a) Comparison of model accuracy during training. *(b)* Comparison of model loss values.

Figure 4. Accuracy and loss comparison of different models trained on the PlantVillage dataset. YOLOv8 showed faster convergence and lower loss, making it the most effective.

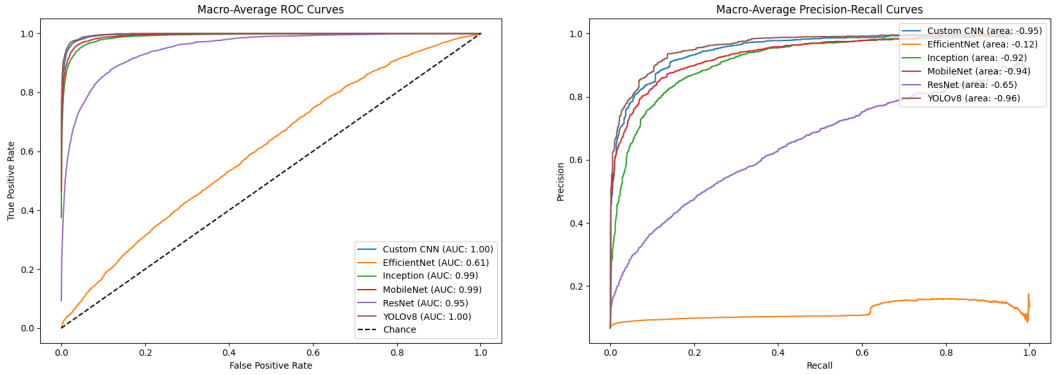


Figure 5. Comparison of model performance using ROC and Precision-Recall curves. The ROC curve () illustrates the trade-off between sensitivity and the false positive rate, while the Precision-Recall curve () evaluates model performance in handling imbalanced data, highlighting precision against recall.

Following model training, the classification model was tested on unseen samples from the PlantVillage dataset. Figure 6 shows the training progress, where YOLOv8 achieved a validation accuracy of 98% after 40 epochs. The effectiveness of YOLOv8 in classification is further illustrated in Figure 7, where the model correctly identifies plant diseases with high confidence.

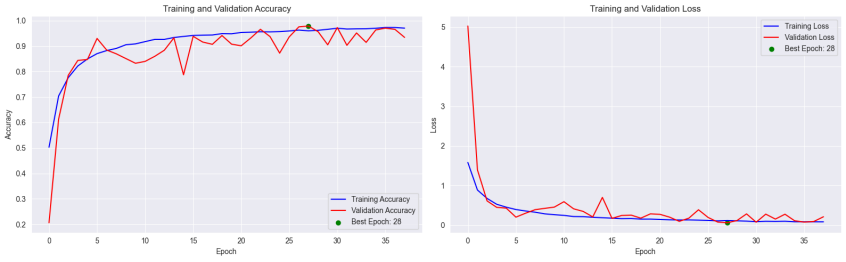


Figure 6. Training performance of YOLOv8 classification model with transfer learning, achieving a training accuracy of 0.96 and validation accuracy of 0.98 after 40 epochs.

Figure 7. YOLOv8 classification model tested on 20% of the PlantVillage dataset. The model correctly identifies plant diseases with high confidence.

For real-world evaluation, YOLOv8 was deployed on the Jetson Orin NX. The model was tested in a controlled setup, identifying multiple diseased leaves on a wall-mounted test environment, as shown in Figure 8. Further, the classification and object detection models were tested on the Jetson Orin NX, with sample results in Figure 9.

Inference speed is crucial for real-time applications. Figure 10 compares the FPS performance of YOLOv8 running on a Jetson Orin NX and a ThinkPad L14. While the ThinkPad achieves higher FPS, the Jetson remains a viable low-power alternative.

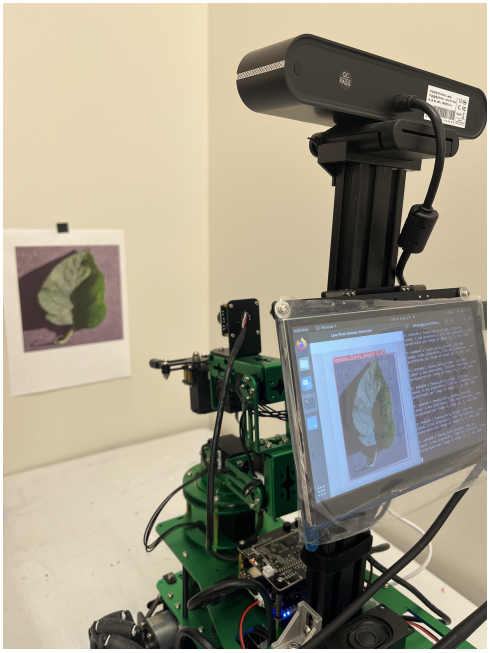
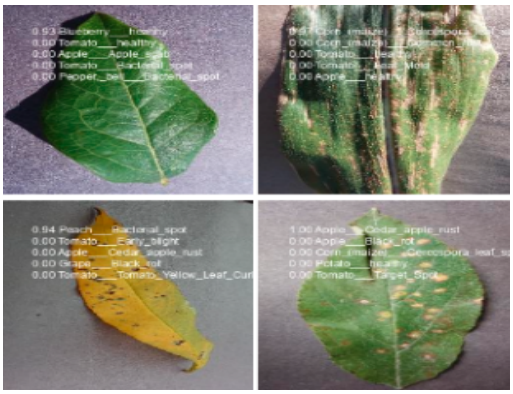


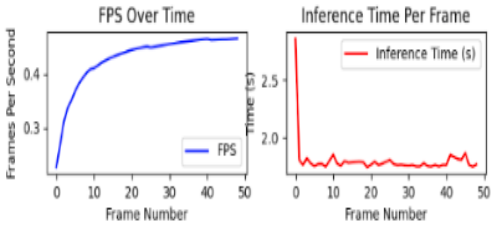
Figure 8. Real-time plant disease detection using YOLOv8 on the Yahboom ROSMaster X3 PLUS robot. The model detects "Tomato Early Blight" with 93% confidence, showcasing its effectiveness in identifying diseased leaves in a controlled environment.



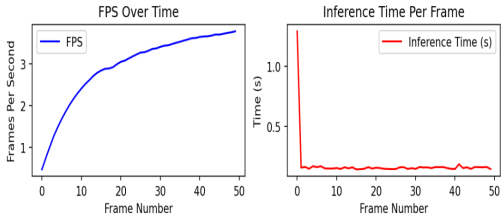
(a) YOLOv8 classification model trained on Jetson Orin NX using 50 epochs without CUDA (50 hours).

(b) YOLOv8 object detection model deployed on Jetson Orin NX to detect multiple plant leaves.

Figure 9. This test was conducted in a simplified indoor setup and does not fully represent the complexity of real-world agricultural environments.



(a) FPS over time for Jetson Orin NX 16GB.



(b) FPS over time for ThinkPad L14 16GB .

Figure 10. FPS comparison between Jetson Orin NX and ThinkPad L14 for YOLOv8 inference.

The Jetson Orin NX 16GB used in this study delivers 100 TOPS of compute and 102 GB/s memory bandwidth, making it capable of real-time inference with models like YOLOv8. During testing, it operated at 10–15W, with short bursts up to 25W under load. A 26,800 mAh portable power bank supported continuous operation for up to 5 hours, demonstrating feasibility for short-term greenhouse use. These specs make the device suitable for edge AI tasks, though longer deployments may benefit from power optimization or external sources.

6. Conclusion

This study demonstrates the feasibility of deploying deep learning models for real-time plant disease detection on edge devices such as the Jetson Orin NX. By evaluating multiple classification models on the PlantVillage dataset, YOLOv8 emerged as the most effective, achieving 90% accuracy with the lowest inference time. The model was successfully deployed on the Yahboom ROSMASTER X3 PLUS robot, enabling real-time disease detection and localization in greenhouse environments. This contribution enhances precision agriculture by integrating deep learning into low-power, edge-based robotic systems for automated crop monitoring.

While the proposed system offers a real-time, autonomous solution, it has several limitations. The Jetson Orin NX's hardware constraints impact inference speed and memory,

which may hinder scalability across larger or more complex agricultural settings. Furthermore, although the PlantVillage dataset provides high-quality annotated samples, it was collected under controlled conditions with uniform lighting, clean backgrounds, and minimal occlusion. As a result, the model has limited exposure to real-world complexities such as overlapping leaves, shadows, variable lighting, and background clutter. This gap can lead to degraded performance in actual greenhouse or field deployments, with an increased risk of false positives or missed detections. Therefore, relying solely on controlled datasets may limit model generalization and robustness in practical applications.

To address this, future work will involve collecting plant disease data under real-world conditions and applying domain adaptation techniques to better align training and deployment environments. In addition, we will focus on optimizing inference speed through quantization and pruning to reduce computational load while maintaining accuracy. A cloud-connected backend for real-time monitoring and analytics will enhance disease tracking and deliver actionable insights to farmers. Integration of automated intervention systems, such as targeted pesticide spraying, will support closed-loop decision-making in agriculture. Beyond plant disease detection, this approach can be extended to yield prediction, pest monitoring, and crop health assessment, making it suitable for a wide range of smart farming applications.

References

[1] S. Muhammad, M. M. Umar, S. Khan, N. A. Alrajeh, and E. A. Mohammed. "Honesty-based social technique to enhance cooperation in social internet of things". In: *Applied Sciences* 13.5 (2023), p. 2778.

[2] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, and S. Singh. "Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges". In: *Archives of Computational Methods in Engineering* (2022). https://doi.org/10.1007/s11831-021-09588-5.

[3] K. Joshi, S. Hooda, A. Sharma, H. Sonah, R. Deshmukh, N. Tuteja, S. S. Gill, and R. Gill. "Precision diagnosis of tomato diseases for sustainable agriculture through deep learning approach with hybrid data augmentation". In: *Current Plant Biology* 41 (2025), p. 100437. DOI: 10.1016/j.cpb.2025.100437. URL: https://doi.org/10.1016/j.cpb.2025.100437.

[4] Y. Peng and Y. Wang. "Leaf disease image retrieval with object detection and deep metric learning". In: *Frontiers in Plant Science* 13 (2022). DOI: 10.3389/fpls.2022.963302. URL: https://www.frontiersin.org/articles/10.3389/fpls.2022.963302/full.

[5] K. N. Rahman, S. C. Banik, R. Islam, and A. A. Fahim. "A Real Time Monitoring System for Accurate Plant Leaves Disease Detection Using Deep Learning". In: *Crop Design* 4 (2025), p. 100092. DOI: 10.1016/j.cropd.2024.100092.

[6] S. Bhagat, M. Kokare, V. Haswani, P. Hambarde, T. Taori, P. Ghante, and D. Patil. "Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop". In: *Smart Agricultural Technology* 7 (2024), p. 100408. DOI: 10.1016/j.atech.2024.100408.

[7] D. S. Joseph, P. M. Pawar, and K. Chakradeo. "Real-Time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning". In: *IEEE Access* 12 (2024), pp. 16310–16333. DOI: 10.1109/ACCESS.2024.3358333.

[8] M. Khalid, M. S. Sarfraz, U. Iqbal, M. U. Aftab, G. Niedbała, and H. T. Rauf. "Real-Time Plant Health Detection Using Deep Convolutional Neural Networks". In: *Agriculture* 13.2 (2023), p. 510. DOI: 10.3390/agriculture13020510.

[9] D. P. Hughes and M. Salathé. *PlantVillage Dataset*. https://www.kaggle.com/datasets/emmarex/plantdisease. Accessed on: 2025-01-30. 2015. arXiv: 1511.08060. URL: https://www.kaggle.com/datasets/emmarex/plantdisease.