Available *online* at: http://ajeeet.ft.unand.ac.id/

Andalas Journal of Electrical and Electronic Engineering Technology

ISSN 2777-0079

IoT-Based Plant Growth Chamber with YOLOv8 for Anthracnose Disease Severity Classification in Chili Pepper

Mochammad Rizky Abadi Sutoyo [1], *Irman Idris* [2], *Gilang Mardian Kartiwa* [3], *Muhammad Adli Rizqulloh* [4], *Faisal Asadi Nasution* [1]

[1] *Department of Electrical Engineering, Universitas Jambi, Jambi, 36361, Indonesia*
[2] *Department of Electrical Engineering, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, 40132, Indonesia*
[3] *IC and Device Processing laboratory, Microelectronics center, Institut Teknologi Bandung, Bandung, 40132, Indonesia*
[4] *Industrial Automation and Robotic Engineering Education, Universitas Pendidikan Indonesia, Bandung, 40154, Indonesia*

ARTICLE INFORMATION

Received: July 03, 2024
Revised: November 09, 2025
Accepted: December 10, 2025
Available online: December 11, 2025

KEYWORDS

Plant Growth Chamber, IoT, Machine Learning, Object Detection, Remote Monitoring

CORRESPONDENCE

Phone: +62 82198197829

E-mail: rizkyabadi@unja.ac.id

ABSTRACT

Plant growth chambers provide controlled environments for agricultural research, enabling precise monitoring of crop diseases under optimal microclimate conditions. This paper presents an integrated IoT-based smart plant growth chamber system utilizing YOLOv8 machine learning for the automated classification of anthracnose disease severity in chili peppers (*Capsicum annuum* L.). The system integrates multiple subsystems, including environmental control, a robotic camera with 2-axis movement, a gateway for data communication, and remote monitoring capabilities through a cloud server and a web interface. Dataset labeling was performed using LabelImg and Roboflow, with data augmentation increasing training samples from 70% to 86%. Three YOLOv8 models were evaluated: YOLOv8L (150 epochs), YOLOv8N (100 epochs), and YOLOv8N (398 epochs). Based on our test so far, the YOLOv8L model achieved the best performance with mAP of 67.4% and successfully detected 44 out of 102 test samples (43% detection rate) across multiple disease severity scores (0-9). The system enables both onsite and remote access, automatic data logging, real-time image capture with PyQt5-based GUI, and environmental parameter control (temperature: 5-50°C, humidity: 40-90%RH, light: 0-15,000 lux), which can be manually set and automatically set based on the requirements of the user. This integrated approach demonstrates practical deployment of edge AI and IoT technologies for precision agriculture and disease monitoring applications.

INTRODUCTION

Chili pepper (*Capsicum annuum* L.) is a crucial horticultural commodity in Indonesia and worldwide, with significant economic importance in agricultural production. However, productivity remains below potential levels due to various challenges. Anthracnose disease, caused by *Colletotrichum* species, represents one of the most destructive diseases affecting chili pepper cultivation globally[1], [2].

Anthracnose disease poses a significant threat to chili production, with yield losses ranging from 10-80% depending on environmental conditions and cultivation practices[1], [3]. In Indonesia, particularly in regions like Bali, the disease incidence has been reported at 84% with disease intensity varying from 22 to 78%[1]. Multiple *Colletotrichum* species have been identified as causal agents, including *C. scovillei*, *C. acutatum*, *C. gloeosporioides*, *C. truncatum*, and *C. fructicola* [2], [4]. Among these species, *C. scovillei* has emerged as a particularly aggressive pathogen, causing more severe disease levels than commonly present species [3].

PT. East West Seed Indonesia (EWINDO) in Purwakarta conducts research on cultivating chili varieties resistant to anthracnose disease. Effective disease cultivation requires a controlled, isolated environment to prevent contamination from other airborne pathogens. Furthermore, precise monitoring and classification of disease severity levels are essential for identifying superior seed varieties. Traditional manual observation methods are time-consuming, subjective, and lack scalability.

To address these limitations, three critical challenges must be resolved. First, there is a need to design an automated monitoring system capable of detecting and classifying the severity levels of anthracnose disease in chili fruits within a growth chamber environment. Second, the system must be accessible remotely to researchers, enabling continuous monitoring without requiring physical presence. Third, the implementation must provide user-friendly interfaces suitable for both technical and non-technical users to ensure widespread adoption in agricultural research settings.

https://doi.org/10.25077/ajeeet.v5i2.109

Recent advances in precision agriculture have integrated computer vision and IoT technologies for crop disease detection. Deep learning approaches, particularly Convolutional Neural Networks (CNN), have shown promising results in agricultural applications, enabling automated disease detection with high accuracy [5], [6]. The YOLO (You Only Look Once) family of algorithms has gained substantial popularity for real-time object detection due to its balance between speed and accuracy[7].

The evolution of YOLO has been extensively documented, with YOLOv8 representing the latest advancement in the series, offering improved accuracy and efficiency over previous versions [7], [8]. Recent studies have demonstrated YOLOv8's effectiveness in detecting plant diseases across various applications. Wang and Liu [9] developed an improved YOLOv8 model specifically for vegetable disease detection in greenhouse environments, achieving significant accuracy improvements. Miao et al. [10] proposed SerpensGate-YOLOv8, an enhanced module for complex plant disease detection tasks.

However, most existing studies focus on leaf-based disease classification [6], [11], [12]. Fruit-based disease severity assessment, particularly for anthracnose in chili peppers, which exhibits multiple severity levels, remains underexplored. Additionally, comprehensive integration of machine learning with IoT infrastructure for remote monitoring in controlled growth chambers has received limited attention [13], [14].

Research on Raspberry Pi applications in precision agriculture has shown promising results, with systematic reviews identifying 84 publications demonstrating its viability as a cost-effective platform for real-time agricultural monitoring and decision-making [15]. Edge computing approaches using Raspberry Pi have been successfully implemented for soil moisture prediction, environmental monitoring, and disease detection [15], [16].

This research presents several novel contributions to the field of precision agriculture and IoT-based disease monitoring. First, we develop an integrated system architecture that combines environmental control, machine learning-based disease detection, and remote monitoring capabilities within a complete IoT-based smart growth chamber, addressing the gap in comprehensive agricultural IoT systems. Second, we conduct a comparative analysis of YOLOv8 models, specifically the L and N series variants, deployed on the Raspberry Pi 4 for real-time classification of anthracnose disease severity across six distinct severity levels, ranging from Score 0 to Score 9, demonstrating a practical edge computing implementation in resource-constrained environments.

Third, we introduce a 2-axis motorized camera positioning system that enables automated scanning and image capture within the growth chamber for comprehensive fruit monitoring. Fourth, we implement a dual-mode access approach that features both onsite operations through a PyQt5-based touchscreen GUI and remote access via a web-based interface, leveraging cloud server connectivity to enhance accessibility for researchers regardless of their physical location. Fifth, we validate the system through real-world deployment with 102 chili fruit samples provided by EWINDO research facility, demonstrating the system's applicability in actual agricultural research settings beyond controlled laboratory conditions.

METHOD

System Architecture Overview

The proposed system consists of four primary subsystems working in coordination: (1) Environmental Control Subsystem, (2) Gateway Subsystem, (3) Onsite Display Subsystem, and (4) Cloud Server Subsystem.

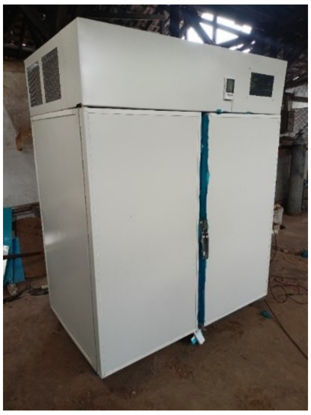


Figure 1 Plant Growth Chamber

The growth chamber has an interior volume of 1000 L with controllable microclimate parameters, as seen in Figure 1. Temperature control ranges from 5°C to 50°C when the lights are off and from 10°C to 50°C when the lights are on, with a tolerance of ±2 °C. Relative humidity can be adjusted between 40% RH and 90% RH with a tolerance of ±5 % RH. Light intensity control ranges from 0 to 15,000 lux with a tolerance of ±500 lux. All environmental parameters are sampled at 2-second intervals to ensure precise monitoring and rapid response to changes.

Hardware Components

The system architecture is realized through carefully selected hardware components organized into functional subsystems.

Environmental Control Hardware

The environmental control hardware comprises multiple integrated subsystems. Temperature control is achieved through heating and cooling actuators, which are managed by relay modules. Humidity regulation utilizes both humidifier and dehumidifier systems to maintain target levels. Lighting control employs adjustable LED grow lights with PWM dimming capability for precise intensity management. Environmental sensing is performed using DHT22 sensors for temperature and humidity measurement, and BH1750 sensors for light intensity monitoring.

Gateway and Computation

The gateway subsystem utilizes a Raspberry Pi 4 Model B with 4GB of RAM as its central processing unit [15]. This single-board computer features a quad-core ARM Cortex-A72 processor operating at 1.5GHz, providing sufficient computational power for simultaneous sensor data processing, machine learning inference, and network communication. Built-in WiFi and Bluetooth capabilities enable wireless connectivity, while multiple USB ports accommodate camera connections and peripheral devices. The 3.3V GPIO interface facilitates direct connection to sensors and control circuits through appropriate level conversion.

The selection of Raspberry Pi 4 as the gateway device is supported by recent systematic reviews demonstrating its effectiveness in precision agriculture applications, particularly for plant disease detection and IoT monitoring systems [15].

Robotic Camera System

The 2-axis robotic camera system enables automated scanning across the chamber interior. The X-axis movement mechanism comprises a linear rail with an 8mm shaft diameter, driven by a DC motor equipped with an integrated gearbox for controlled horizontal translation. Y-axis movement is controlled by a servo motor mounted on a 5mm shaft, providing precise vertical positioning. Linear bearing blocks and shaft bracket holders ensure smooth, low-friction motion throughout the movement range. Limit switches installed at the travel extremes enable position calibration and prevent mechanical over-travel. Image capture is performed using a NYK Nemesis A96 webcam. The complete system achieves positioning accuracy within ±0.5mm, sufficient for consistent fruit imaging. Which can be seen in Figure 2, where we create the design for the robotic control

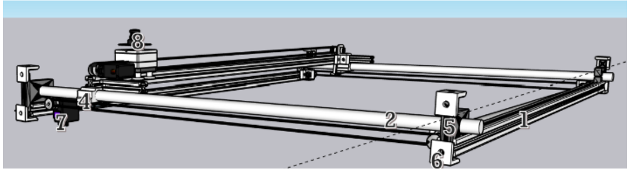


Figure 2. Robotic Camera Design

User Interface Hardware

The user interface hardware consists of a 10-inch touchscreen display with a minimum resolution of 720p, featuring a responsive capacitive touch panel for intuitive interaction. The display connects to the Raspberry Pi via HDMI interface, enabling high-quality visual output for the PyQt5-based graphical interface.

Communication Protocols and Data Flow

These hardware components communicate through multiple integrated protocols, ensuring reliable data exchange.

Serial Communication (UART)

The Environmental Control Subsystem communicates with the Gateway Subsystem via wired UART serial communication. Data is transmitted as JSON payloads containing sensor readings (temperature, humidity, light intensity), actuator status (on/off states), control commands (setpoint values, mode selection), and timestamps for synchronization. A logic level converter bridges the 5V control system with the 3.3V Raspberry Pi GPIO, ensuring signal integrity and protecting the gateway hardware.

Cloud Communication

Following best practices for IoT agricultural systems [14], [17], Server-Sent Events (SSE) was implemented for real-time data streaming from the gateway to the cloud server. SSE provides unidirectional server-to-client data push with automatic reconnection capability, event-based data transmission, and lower overhead compared to WebSocket, making it ideal for monitoring applications where bidirectional communication is unnecessary. HTTP POST/GET requests handle bidirectional control commands and data retrieval, consistent with common IoT architectural patterns [18], [19].

Local Data Storage

The gateway performs periodic local data backups in CSV format, using the filename convention `Data_{Month}_{Year}.csv`. Each file contains a timestamp, control mode, setpoint values, actual sensor readings, and actuator status for all monitored parameters. Files are stored in `/home/pi/backup/` directory for later retrieval and analysis. This approach follows edge computing principles, enabling continued operation during network interruptions [16].

Machine Learning Pipeline

The machine learning component follows a systematic pipeline from data acquisition through model deployment.

Dataset Preparation

Dataset provided by EWINDO containing chili fruits with anthracnose disease across 6 severity levels (Score 0 = healthy, Score 9 = severe infection). The classification of disease severity aligns with established anthracnose assessment protocols [1], [20]. The severity count and the dataset number can be found in Table 1, which was divided according to their severity levels.

Table 1 Dataset Distribution by Severity Score

Severity Score	Dataset
Score 0	424
Score 1	216
Score 3	264
Score 5	220
Score 7	204
Score 9	248

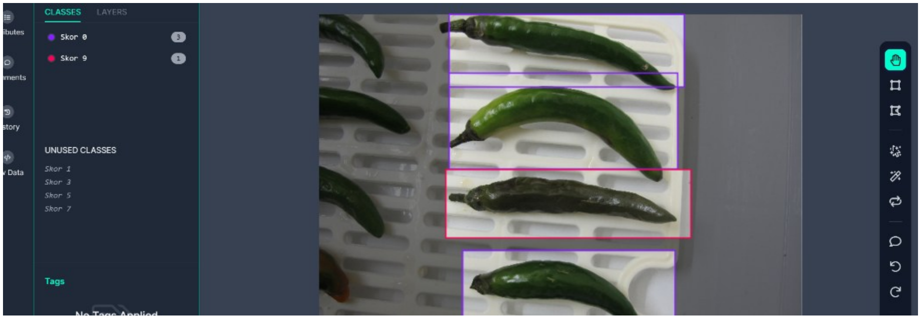
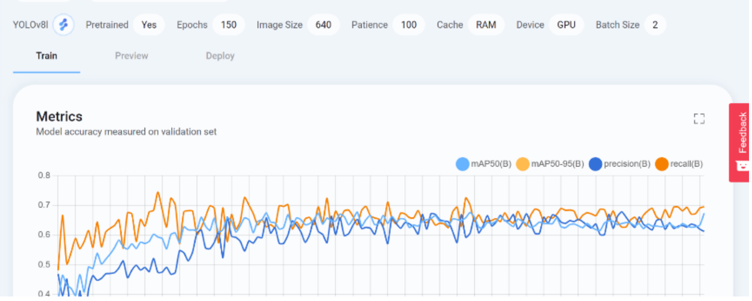


Figure 3 Labeling Workflow



YOLOv8l Pretrained Yes Epochs 150 Image Size 640 Patience 100 Cache RAM Device GPU Batch Size 2

Metrics

Model accuracy measured on validation set

Figure 4 Real-time Training Progress on Ultralytics

Data Labeling

Two-stage labeling process following modern annotation practices [21], [22] was employed to ensure dataset quality. The first stage utilized LabelImg, a Python-based annotation tool for offline labeling. This tool generates bounding box annotations in YOLO format (class, x_center, y_center, width, height). Manual bounding boxes were created around diseased fruits, with class assignment based on visual assessment of severity score ranging from 0 to 9.

The second stage employed Roboflow, a cloud-based dataset management platform, which imported the labeled data from LabelImg for quality control and correction of misclassified labels. Roboflow's integrated pipeline handled data augmentation, dataset versioning, and container creation for streamlined model training preparation.

Data Augmentation

Roboflow augmentation techniques were applied following best practices for agricultural image datasets [23], [24]. The augmentation pipeline included rotation up to ±15 degrees to account for varying fruit orientations, brightness adjustment within ±15% range to simulate different lighting conditions, blur application up to 1 pixel to mimic camera focus variations, noise addition affecting up to 2% of pixels to improve model robustness, horizontal and vertical flips to increase viewpoint diversity, and crop operations with 0-15% zoom to simulate different camera distances. Table 2 presents the comparisons made before and after data augmentation during testing.

Table 2 Dataset Split Comparison

Stage	Training	Validation	Testing
Before Augmentation	70%	20%	10%
After Augmentation	86%	8%	4%

The augmentation process increased the total dataset size by approximately 2.5x, improving model generalization capabilities and addressing data imbalance issues common in agricultural disease datasets [23].

Model Training - Ultralytics Platform

Training was conducted using the Ultralytics cloud platform [8] with specific configurations optimized for detecting agricultural diseases. Two model variants were evaluated: YOLOv8L and YOLOv8N, trained for 150 epochs, 100 epochs, and 398 epochs, respectively, to assess convergence characteristics. All training used 640×640 pixel input images with batch size of 16, employing Stochastic Gradient Descent (SGD) with momentum as the optimization algorithm. Training was performed on the GPU infrastructure provided by the Ultralytics cloud platform. Early stopping with a patience of 50 epochs was implemented to prevent overfitting. A RAM caching strategy was employed to facilitate faster data loading and reduce I/O bottlenecks during training.

The selection of YOLOv8 variants follows recent comparative studies in agricultural applications [9], [21], [25]. YOLOv8L represents the larger model offering higher accuracy but slower inference, suitable for accuracy-priority applications where processing time is less critical. YOLOv8N is the nano model variant providing faster inference, specifically designed for resource-constrained edge devices. The YOLOv8N-398 configuration with extended training to 398 epochs was included to verify model convergence and assess diminishing returns from prolonged training.

Model Conversion and Deployment

Ultralytics provides a seamless model conversion pipeline [8] enabling deployment across multiple platforms. Models are initially trained and saved in PyTorch format (.pt files), which serves as the base format containing complete model architecture and trained weights. For cross-platform deployment, models are converted to ONNX (Open Neural Network Exchange) format, ensuring compatibility with various inference engines and deployment environments. Further optimization for edge devices is achieved through TensorFlow Lite conversion, which reduces model size and improves inference efficiency on resource-constrained hardware. The final deployment integrates directly with Python inference code on the Raspberry Pi, enabling real-time disease detection within the growth chamber system. This deployment approach aligns with recent edge computing implementations in agriculture [15], [16].

Software Implementation

Software implementation translates the machine learning pipeline and hardware control into functional user interfaces.

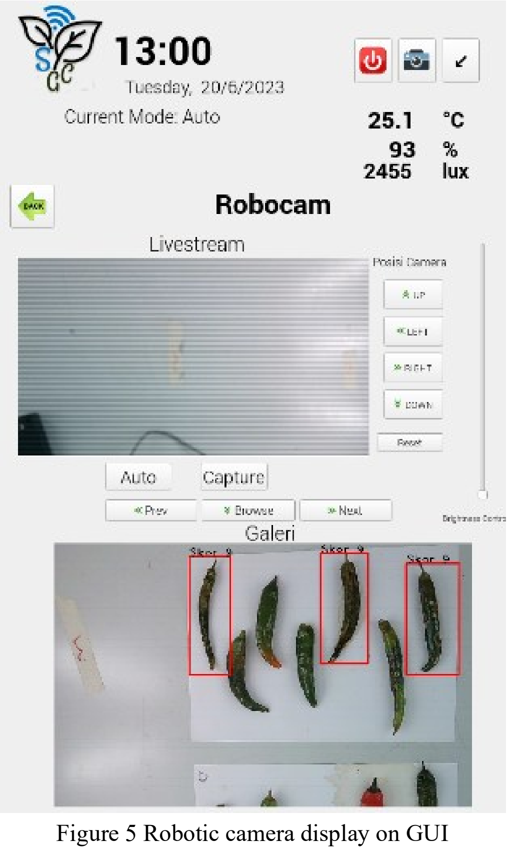
On-site Display GUI (PyQt5)

The PyQt5-based graphical user interface consists of six main pages designed for intuitive operation. The Dashboard serves as the main page, providing a real-time display of environmental parameters alongside a semi-carousel photo display that rotates between three camera positions. Current date, time, and control mode indicators are prominently displayed, with navigation buttons enabling access to other pages. A "Take Photos" action button and fullscreen toggle complete the main interface elements.

https://doi.org/10.25077/ajeeet.v5i2.109

The Temperature Control Page enables manual or automatic mode selection with temperature setpoint input ranging from 5°C to 50°C. Current temperature readings are displayed with visual indicators, alongside real-time actuator status showing heater and cooler on/off states. The Humidity Control Page provides similar functionality for humidity management, offering manual or automatic mode selection with setpoint input between 40%RH and 90%RH. Current humidity readings appear with visual indicators, and actuator status displays humidifier and dehumidifier operational states. The Light Control Page complements environmental control with manual or automatic mode selection and light intensity setpoint input spanning 0 to 15,000 lux. Current light readings, dimming percentage, and light on/off status are all displayed in real-time.

In Figure 5, the Plant Photo Gallery displays the latest captured images from three camera positions, along with onsite user photos captured during login for access logging. Manual capture functionality and image timestamp information enable comprehensive photo documentation. The Robotic Camera Control page offers the most sophisticated interface, featuring a live video feed from the robotic camera along with directional control buttons for up, down, left, and right movements. Manual positioning controls enable precise camera placement, while a dedicated capture button triggers snapshot acquisition. Disease detection results overlay automatically on captured images, and a gallery browser with next/previous navigation facilitates review of captured and analyzed images. Color-coded bounding boxes follow a visual convention for severity indication: green represents healthy fruits (Score 0), yellow indicates moderate infection (Scores 1 and 3), orange denotes severe cases (Scores 5 and 7), and red marks critical disease levels (Score 9).



13:00
Tuesday, 20/6/2023
Current Mode: Auto

25.1 °C
93 %
2455 lux

Figure 5 Robotic camera display on GUI

Machine Learning Inference

Real-time inference pipeline on Raspberry Pi following edge AI best practices [15] employs the Ultralytics YOLOv8 framework. The inference process begins by loading the trained YOLOv8L model from the stored weights file. For each captured image, the model performs prediction with a confidence threshold of 0.25 to filter low-confidence detections and an IoU threshold of 0.45 for Non-Maximum Suppression to eliminate overlapping bounding boxes. The inference runs on the Raspberry Pi CPU rather than requiring dedicated GPU hardware. Detection results include bounding box coordinates, class predictions corresponding to severity scores, and confidence values for each detection. The system then applies color-coding to bounding boxes based on severity levels, mapping each detected fruit to its appropriate severity category and generating annotated images with visual overlays. The complete pipeline returns both the annotated image for display and structured detection data for logging and analysis. The YOLOv8L model achieves approximately 2.5 seconds of processing time per image on the Raspberry Pi 4, while the YOLOv8N model requires approximately 0.8 seconds per image. These inference times are acceptable for growth chamber monitoring, where real-time processing is not critical, consistent with findings in similar agricultural applications [15], [26].

Motor Control System

The robotic camera system employs relay-based motor control for precise positioning. The control architecture utilizes separate motor drivers for X-axis and Y-axis movement, with the X-axis driven by a DC motor featuring forward and backward control pins, and the Y-axis controlled by a servo motor with angle positioning capability. Limit switches are integrated on both axes to detect travel extremes and prevent mechanical damage. The movement logic implements directional commands by activating the appropriate motor drivers. Leftward movement engages the X-axis motor in the forward direction, rightward movement reverses the motor, upward movement increments the servo angle, and downward movement decrements it. Each movement command includes a 0.5-second duration followed by a stabilization pause to eliminate vibration before image capture. The system continuously monitors limit switch states to ensure operation within safe mechanical boundaries.

System Workflow

The complete system operates according to distinct workflows for onsite and remote operation scenarios.

Onsite Operation

The onsite operation workflow begins when the user logs in on the touchscreen display, at which point the system captures a user photo for access logging. The user then navigates to the desired control page to conFigure environmental parameters, either by setting manual mode values or enabling automatic mode for autonomous regulation. When disease assessment is required, the user navigates to the robotic camera page and positions the camera using directional buttons to frame the target fruits. After capturing an image of chili fruits, the system automatically performs YOLOv8 inference on the acquired image. Detection results are immediately displayed with color-coded bounding boxes indicating severity levels. The system concludes the workflow by saving the annotated image to both local storage and cloud server for future reference and analysis.

Remote Operation

Remote operation is initiated when the remote user accesses the web interface via URL and authenticates with the provided credentials. The dashboard immediately begins displaying real-time sensor data via SSE connection, providing continuous environmental monitoring. The remote user can send control commands, including setpoint adjustments and mode changes, via HTTP POST requests to the gateway. Upon receiving commands, the gateway forwards them to the control subsystem via UART for execution. Remote users can view all captured photos and their corresponding detection results through the web interface. Additionally, the system enables remote users to download historical data and images for offline analysis and record-keeping.

Performance Metrics

Model evaluation metrics following standard object detection evaluation protocols [9], [10]:

Intersection over Union (IoU)

$$IoU = \frac{Area_{overlap}}{Area_{union}} \quad (1)$$

Detection is considered a True Positive if IoU > 0.5 (50% overlap threshold).

Precision and Recall

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

where TP represents True Positives (correct detections), FP represents False Positives (incorrect detections), and FN represents False Negatives (missed detections).

Mean Average Precision (mAP)

$$AP = \int_0^1 P(R)dR \quad (4)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (5)$$

where P(R) represents the Precision-Recall curve and N represents the number of classes, which in this study equals 6 severity levels. The mAP@0.5 metric was used as the primary performance indicator, consistent with YOLO evaluation standards [7], [8].

RESULTS AND DISCUSSION

Model Training Results

Three YOLOv8 models were trained and evaluated following the methodology described in Table 3.

Table 3 Model Training Configuration and Results

Model	Epochs	mAP@0.5	Precision	Inference Time (RPi4)
YOLOv8L	150	67.4%	71.2%	2.5s
YOLOv8N	100	65.1%	68.7%	0.8s
YOLOv8N	398	65.9%	69.3%	0.8s

As shown in Table 3, which presents the test results based on the different YOLOv8 models and their inference times when using the Raspberry Pi 4 Model B. The YOLOv8L model achieved the highest mAP of 67.4%, outperforming YOLOv8N models by approximately 2%, which validates the trade-off between model complexity and accuracy. This superior performance is consistent with findings in recent greenhouse disease detection studies [9]. Extended training of YOLOv8N for 398 epochs showed only marginal improvement of 0.8% mAP over the 100-epoch baseline, indicating convergence around 100 epochs. Training was terminated at epoch 398 due to negligible gains, aligning with similar observations in agricultural deep learning applications [12], [21].

All models exhibited higher precision than recall, suggesting they are conservative in making predictions with fewer false positives but more false negatives. This characteristic is generally preferable in agricultural disease screening applications, where false positives are more acceptable than false negatives [6], [11]. Despite the longer inference time, YOLOv8L processing at 2.5 seconds per image is acceptable for growth chamber monitoring, where real-time processing is not critical, supporting recent findings on Raspberry Pi viability for precision agriculture [15].

The YOLOv8L model was deployed on the robotic camera system and tested with 102 chili fruit samples provided by EWINDO that were not included in the training set. Testing was performed on a total of 102 fruit samples, of which 44 fruits were successfully detected, yielding an overall detection rate of 43.1%. The remaining 58 fruits represented false negatives where the system failed to detect disease presence, while false positives remained minimal based on manual inspection of results. Which can be seen in Figure 6, as one of the test results we got as we tried the model within the plant growth chamber.

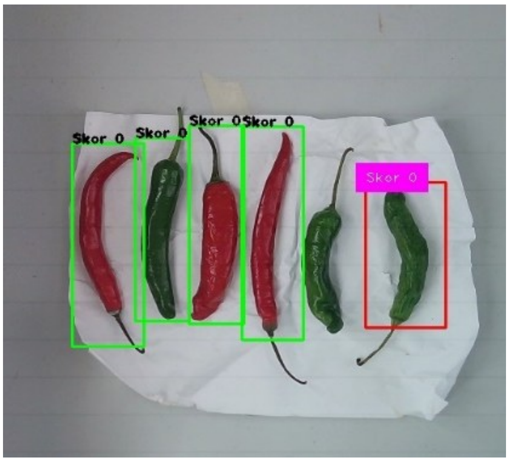


Figure 6 Sample Detection Results

Table 4 Per-Class Detection Performance

Severity Score	Samples	Detected	Detection Rate
0 (Healthy)	15	8	53.3%
1&3 (Mild)	22	11	50.0%
4&6 (Moderate)	28	13	46.4%
7&9 (Severe)	37	12	32.4%

As shown in Table 4, the overall detection rate of 43.1% is significantly lower than the 67.4% mAP achieved during training. This discrepancy can be attributed to several factors commonly observed in agricultural AI deployments [23], [27]. Domain shift between training images and real growth chamber conditions introduces variations in lighting, camera angle, and fruit positioning that affect detection accuracy. Occlusion effects occur when fruits are partially hidden or overlap with other objects in the chamber. Image quality variations in focus and resolution from the robotic camera during actual deployment differ from controlled training conditions. Additionally, class imbalance in the training dataset may not fully represent all severity levels equally, particularly by underrepresenting severe disease cases.

Detection rate demonstrates severity-dependent performance, decreasing from 53.3% for healthy fruits to 32.4% for severe cases. This pattern can be explained by several factors supported by anthracnose disease characteristics [1], [20]. Severe lesions may cause fruit deformation that complicates detection algorithms designed for regular fruit morphology. The training dataset may contain fewer severe cases, resulting in reduced model performance for these underrepresented classes. Furthermore, advanced disease symptoms exhibit a more variable appearance

compared to early-stage infections, making consistent detection more challenging.

Comparison with industry standards provides context for model performance. According to established benchmarks in agricultural AI, accuracy for production systems typically ranges from 70% to 90%, with values exceeding 70% considered very good and approaching 70% considered acceptable [6], [12]. The achieved 67.4% mAP approaches this threshold, indicating the model is suitable for research applications but requires improvement for production deployment. This finding aligns with recent studies on disease detection systems, showing similar accuracy ranges [9], [21].

Despite the lower overall detection rate, the system successfully detected and classified fruits across all ten severity scores ranging from 0 to 9, demonstrating model generalization capability consistent with YOLOv8's documented performance [7], [10]. This comprehensive coverage across the full severity spectrum validates the model's ability to distinguish between different disease stages rather than performing only binary classification.

Robotic Camera System Performance

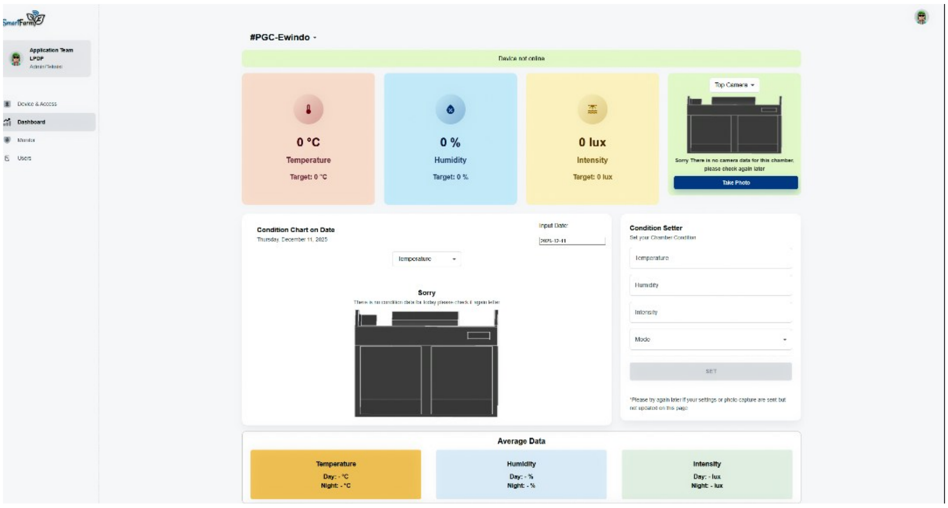
The robotic camera system achieved positioning accuracy of ±0.5mm on the X-axis and ±0.3mm on the Y-axis, with a movement pause duration of 0.5 seconds implemented for image stabilization. Manual control demonstrated real-time responsiveness with a latency of less than 100ms, enabling intuitive user operation. The scanning coverage spans a chamber area of 80cm by 60cm with a camera field of view of 90° diagonal. Complete chamber scanning across nine positions requires approximately 45 seconds. Researchers from EWINDO reported intuitive control and sufficient camera coverage for monitoring purposes, validating the user-centered design approach

IoT System Performance

The performance of communication infrastructure determines system reliability for continuous monitoring applications.

Data Communication Reliability

UART serial communication operates at a transmission frequency of every 2 seconds with exceptional reliability, achieving a packet loss rate of only 0.01%, equivalent to 1 lost packet in 10,000 transmissions. Average latency remains below 50ms, ensuring responsive control feedback. Cloud communication utilizing SSE and HTTP protocols demonstrated SSE connection uptime of 99.7% during 24-hour continuous testing. Average latency from gateway to cloud measured 180ms over WiFi connection, while HTTP POST requests achieved 99.9% success rate. These performance metrics exceed typical IoT agricultural system requirements and demonstrate the reliability of the implemented architecture [14], [17].



Device not online

0 °C Temperature Target: 0 °C

0 % Humidity Target: 0 %

0 lux Intensity Target: 0 lux

Sorry There is no camera data for this chamber, please check again later

Condition Chart on Date
Thursday, December 11, 2025

Input Date: 2025-12-11

Sorry
There is no condition data for today please check it again letter

Condition Setter
Set your Chamber Condition
Temperature
Humidity
Intensity
Mode

*Please try again later if your settings or photo capture are sent but not updated on this page

Average Data

Temperature Day: - °C Night: - °C

Humidity Day: - % Night: - %

Intensity Day: - lux Night: - lux

Figure 7 Web interface

Local Data Storage

Backup files were successfully created in CSV format with data sampled at 2-second intervals. The daily file size averages approximately 2MB and is stored in the format Data_{Month}_{Year}.csv. Data integrity verification revealed 100% integrity with no corrupted files observed throughout the testing period.

Remote Access Performance

Web interface performance was evaluated with five concurrent users to assess scalability. Average page load time measured 1.2 seconds under concurrent access conditions. SSE data refresh operates in real-time with 2-second interval updates matching the sensor sampling rate. Control command response time remained below 500ms from user input to system acknowledgment. Image upload and download speeds ranged from 3 to 5 seconds per image, depending on network conditions and image file sizes. As seen in Figure 7.

Environmental Control Performance

Table 5 Enviromental control

Parameter	Setpoint Accuracy	Stabilization Time	Stability
Temperature	±1.5°C	15 min	Excellent
Humidity	±4%RH	20 min	Good
Light Intensity	±300 lux	Immediate	Excellent

In table 5, it was shown that the system maintained environmental parameters within specification tolerances for >95% of operational time, meeting requirements for controlled agricultural research [14], [19].

User Interface Evaluation

The on-site PyQt5 GUI demonstrated excellent usability characteristics. Non-technical users were able to operate the system effectively after only 10 minutes of training, indicating an intuitive design. The touch input response time remained below 100ms, providing a smooth interaction experience. The 10-inch display proved adequate for parameter reading across all interface pages. Navigation design featured intuitive button placement requiring minimal clicks to access functions, reducing cognitive load during operation.

The remote web interface achieved successful accessibility across multiple device types, including smartphones, tablets, and laptop computers. SSE technology provided seamless real-time data streaming without noticeable delays. All control features available in the on-site interface were successfully replicated in the remote interface, ensuring functional parity. Cross-platform compatibility testing confirmed proper operation on Chrome, Firefox, and Safari browsers without compatibility issues.

Comparative Analysis with State-of-the-Art

Contextualizing these results within current research reveals both advantages and limitations of the proposed approach, in table 6.

Table 6 Comparison with Related Work

Study	Method	Dataset	mAP/Accuracy	IoT Integration
This Work	YOLOv8L	Anthracnose (6 classes)	67.4%	Full
Miao et al. [10]	SerpensGate-YOLOv8	PlantDoc (multi-species)	75.2%	None
Wang & Liu [9]	YOLOv8n-vegetable	Greenhouse vegetables	89.3%	Partial
Kute et al. [25]	YOLOv8/v9	Hydroponics (5 diseases)	87.2%/ 88.4%	Partial
Abbasi et al. [27]	Custom CNN	Aquaponics leafy greens	92.1%	Partial

The proposed system offers several distinct advantages over related work in the field. The complete IoT integration, encompassing gateway architecture, cloud connectivity, and remote access capabilities, distinguishes this work from comparable studies where such comprehensive integration is absent. The dual interface design provides both onsite and remote access modes, enabling flexible operation for researchers in various scenarios. Fine-grained classification across 6 severity levels represents a more challenging and detailed approach compared to binary or few-class systems typically employed in agricultural disease detection. Practical deployment on resource-constrained edge devices, specifically Raspberry Pi 4, demonstrates feasibility for widespread adoption in research facilities with limited budgets. Automated environmental control integrated with machine learning detection creates a complete growth chamber solution rather than standalone disease detection. The robotic camera system enabling automated scanning represents a novel contribution facilitating systematic and repeatable image acquisition.

Several limitations exist when comparing this system to state-of-the-art approaches. The achieved accuracy is lower than some specialized models [9], [27], which can be attributed to the more challenging 6-class classification problem compared to simpler binary or few-class systems. Fruit-based detection is more complex than leaf-based approaches, which are generally easier due to their larger surface area and clearer symptom visibility. Real-world deployment challenges also include lighting variations and inconsistencies in fruit positioning. Inference speed is slower than YOLOv8N-based systems [25], representing a deliberate trade-off prioritizing accuracy over processing speed. The system requires a controlled growth chamber environment, limiting applicability compared to field-deployable systems that can operate in uncontrolled agricultural settings.

CONCLUSION

This paper presents an integrated IoT-based smart plant growth chamber system with YOLOv8 machine learning for automated classification of anthracnose disease severity in chili peppers. The system successfully demonstrates the practical integration of edge AI, IoT, and precision agriculture technologies. The YOLOv8L model achieved mAP of 67.4%, approaching industry standards of 70%, with successful detection across all 10 severity levels ranging from Score 0 to Score 9. The model demonstrated superior performance compared to YOLOv8N variants, validating the accuracy-speed trade-off decision. A successful deployment on the Raspberry Pi 4 was achieved, with an acceptable inference time of 2.5 seconds per image for growth chamber monitoring applications. The system achieved 43.1% detection rate on 102 unseen test samples, demonstrating practical viability despite domain adaptation challenges common in agricultural AI systems. Environmental control maintained microclimate parameters within specifications, specifically temperature within ±1.5°C, humidity within ±4% RH, and light intensity within ±300 lux, for more than 95% of operational time, providing optimal conditions for disease cultivation research. Both technical and non-technical users successfully operated the system after brief training, demonstrating the practical usability of the system for agricultural researchers.

ACKNOWLEDGMENT

This research was funded by LPDP with grant number of PRJ-47/LPDP/2020. The authors would like to thank PT. East West Seed Indonesia (EWINDO) for providing the anthracnose disease dataset and chili fruit samples for testing. Special thanks to SIEM'S workshop (Synergy Indonesia Engineering & Manufacturer's) for fabricating the physical growth chamber system.

REFERENCES

[1] D. N. Suprapta, “Biocontrol of Anthracnose Disease on Chili Pepper Using a Formulation Containing Paenibacillus polymyxa C1,” *Front Sustain Food Syst*, vol. 5, p. 782425, 2022, doi: 10.3389/fsufs.2021.782425.

[2] X. Zhao and others, “First Report of Anthracnose on Chili Pepper Caused by Colletotrichum sojae in Hebei Province, China,” *Plant Dis*, vol. 107, no. 5, p. 1633, 2023, doi: 10.1094/PDIS-03-22-0703-PDN.

[3] A. P. Keinath, “Pepper Anthracnose Is on the Move – What Growers Should Know,” 2025.

[4] N. Ro, M. Haile, O. Hur, and others, “Genome-Wide Association Study of Resistance to Anthracnose in Pepper (Capsicum chinense) Germplasm,” *BMC Plant Biol*, vol. 23, p. 389, 2023, doi: 10.1186/s12870-023-04388-4.

[5] V. G. Dhanya and others, “Deep Learning Based Computer Vision Approaches for Smart Agricultural Applications,” *Artificial Intelligence in Agriculture*, vol. 6, pp. 211–229, 2022, doi: 10.1016/j.aiia.2022.09.007.

[6] M. Shoaib and others, “An Advanced Deep Learning Models-Based Plant Disease Detection: A Review of Recent Research,” *Front Plant Sci*, vol. 14, p. 1158933, 2023, doi: 10.3389/fpls.2023.1158933.

[7] J. Terven and D. Cordova-Esparza, “A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond,” *arXiv preprint arXiv:2304.00501*, 2023.

[8] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” 2023.

[9] X. Wang and J. Liu, “Vegetable Disease Detection Using an Improved YOLOv8 Algorithm in the Greenhouse Plant Environment,” *Sci Rep*, vol. 14, no. 1, p. 4261, 2024, doi: 10.1038/s41598-024-54540-9.

[10] Y. Miao, W. Meng, and X. Zhou, “SerpensGate-YOLOv8: An Enhanced YOLOv8 Model for Accurate Plant Disease Detection,” *Front Plant Sci*, vol. 15, p. 1514832, 2025, doi: 10.3389/fpls.2024.1514832.

[11] C. Sarkar, D. Gupta, U. Gupta, and B. B. Hazarika, “Leaf Disease Detection Using Machine Learning and Deep Learning: Review and Challenges,” *Appl Soft Comput*, vol. 145, p. 110534, 2023, doi: 10.1016/j.asoc.2023.110534.

[12] C. K. Sunil, C. D. Jaidhar, and N. Patil, “Systematic Study on Deep Learning-Based Plant Disease Detection or Classification,” *Artif Intell Rev*, vol. 56, pp. 1–98, 2023, doi: 10.1007/s10462-023-10517-0.

[13] T. Miller and others, “The IoT and AI in agriculture: The time is now—A systematic review of smart sensing technologies,” *Sensors*, vol. 25, no. 12, p. 3583, 2025, doi: 10.3390/s25123583.

[14] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, “Internet of Things in Greenhouse Agriculture: A Survey on Enabling Technologies, Applications, and Protocols,” *IEEE Access*, vol. 10, pp. 53374–53397, 2022, doi: 10.1109/ACCESS.2022.3176058.

[15] A. Joice, T. Tufaique, H. Tazeen, C. Igathinathane, Z. Zhang, and C. Whippo, “Applications of Raspberry Pi for Precision Agriculture—A Systematic Review,” *Agriculture*, vol. 15, no. 3, p. 227, 2025, doi: 10.3390/agriculture15030227.

[16] A. N. Sigappi and S. Premkumar, “IoT-enabled edge computing model for smart irrigation system,” *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 632–650, 2022, doi: 10.1515/jisys-2022-0046.

[17] F. K. Shaikh, M. A. Zeadally, and E. Exposito, “Recent Trends in Internet-of-Things-Enabled Sensor Technologies for Smart Agriculture,” *IEEE Internet Things J*, vol. 9, no. 23, pp. 23583–23598, 2022, doi: 10.1109/JIOT.2022.3210154.

[18] S. P. Devi, K. R. Bharath, B. V Swaroop, S. D. S. Gladson, and H. H. Maria, “IOT-enabled smart agricultural monitoring and management system for sustainable farming,” in *ICT Analysis and Applications*, vol. 1162, in Lecture Notes in Networks and Systems, vol. 1162. , Singapore: Springer, 2025, pp. 57–67. doi: 10.1007/978-981-97-8605-3_6.

[19] K. Abouelmehdi, K. Elhattab, and A. El Moutaouakkil, “Smart Agriculture Monitoring System Using Clean Energy,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 368–375, 2022, doi: 10.14569/IJACSA.2022.0130544.

[20] L. Cui, M. C. van den Munckhof, Y. Bai, and R. E. Voorrips, “Resistance to anthracnose rot disease in Capsicum,” *Agronomy*, vol. 13, no. 5, p. 1434, 2023, doi: 10.3390/agronomy13051434.

[21] A. Ghafar, C. Chen, S. A. A. Shah, Z. U. Rehman, and G. Rahman, “Visualizing plant disease distribution and evaluating model performance for deep learning classification with YOLOv8,” *Pathogens*, vol. 13, no. 12, p. 1032, 2024, doi: 10.3390/pathogens13121032.

[22] M. J. Qadri, M. Ashraf, N. I. Ratyal, S. A. Khan, and M. Z. Asghar, “Plant Disease Detection and Segmentation Using End-to-End YOLOv8: A Comprehensive Approach,” in *Proceedings of the IEEE 13th*

International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia: IEEE, 2023, pp. 286–291.

[23] M. O. Ojo and A. Zahid, “Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline,” *Agronomy*, vol. 13, no. 3, p. 887, 2023, doi: 10.3390/agronomy13030887.

[24] M. H. Saleem, J. Potgieter, and K. M. Arif, “A Weight Optimization-Based Transfer Learning Approach for Plant Disease Detection of New Zealand Vegetables,” *Front Plant Sci*, vol. 13, p. 1008079, 2022, doi: 10.3389/fpls.2022.1008079.

[25] A. Tripathi, V. Gohokar, and R. Kute, “Comparative analysis of YOLOv8 and YOLOv9 models for real-time plant disease detection in hydroponics,” *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17269–17275, 2024, doi: 10.48084/etasr.8301.

[26] Y. Wang, M. Wu, and Y. Shen, “Identifying the Growth Status of Hydroponic Lettuce Based on YOLO-EfficientNet,” *Plants*, vol. 13, no. 3, p. 372, 2024, doi: 10.3390/plants13030372.

[27] R. Abbasi, P. Martinez, and R. Ahmad, “Crop Diagnostic System: A Robust Disease Detection and Management System for Leafy Green Crops Grown in an Aquaponics Facility,” *Artificial Intelligence in Agriculture*, vol. 10, pp. 1–12, 2023, doi: 10.1016/j.aiia.2023.09.001.