World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

Check for updates

# Bell pepper detection system using YOLOv8n

Mayumi Serote [1,*], John Kevin Callora [1], Jeremy Cayabyab [1], Vivien Agustin [2] and Ronald Fernandez [2]

[1] *Department of Information Technology, Jesus Reigns Christian College, Manila, Philippines.*

[2] *La Consolacion University Philippines.*

World Journal of Advanced Research and Reviews, 2026, 30(02), 1864-1873

Publication history: Received on 15 April 2026; revised on 21 May 2026; accepted on 23 May 2026

Article DOI: https://doi.org/10.30574/wjarr.2026.30.2.1458

## Abstract

Traditional crop monitoring in Philippine agriculture heavily relies on manual inspection. This process is highly susceptible to human error, labor-intensive, and prone to causing significant financial losses due to inconsistent ripeness assessment. While computer vision and deep learning offer promising non-contact solutions for precision agriculture, many existing object detection frameworks require substantial computational resources, limiting their deployment on portable, field-ready hardware. This study addresses these challenges by developing a lightweight, real-time detection and classification system for local bell peppers (*Sultan F1*) utilizing the YOLOv8n variant architecture. Following an Agile development methodology—encompassing system requirements, design, testing, and deployment—the system was trained on an annotated dataset of raw images captured in an indoor environment, categorized into "monitor" and "ripe" maturity stages. The developed system integrates computer vision with a user-friendly monitoring dashboard to display real-time classification and ripeness outputs, optimizing inference speed and resource efficiency. The results demonstrate the viability of implementing edge AI on low-resource, portable platforms to mitigate post-harvest losses. Ultimately, this lightweight architecture provides a foundational framework for future advancements in smart harvesting, autonomous agricultural robotics, and automated crop monitoring systems. Both the training and evaluation results showed that YOLOv8n achieved nearly perfect performance across metrics such as F1-score, precision, recall, and mean Average Precision (mAP). The confidence curves of the metrics also verified the model's accuracy in detecting the bell pepper (Sultan F1) in terms of its maturity level: green pepper and red pepper (ripe) or transition pepper (monitor).

**Keywords:** Computer Vision; YOLOv8n; Artificial Intelligence; Bell Pepper Detection; Edge AI; Smart Agriculture

## 1. Introduction

The rapid advancement of artificial intelligence (AI) has significantly influenced the agricultural sector, particularly in the development of smart farming technologies designed to improve productivity, efficiency, and crop management. In the Philippines, agriculture remains one of the primary sources of livelihood, making the adoption of innovative technologies essential for sustainable food production. Bell pepper (Sultan F1), a commonly cultivated crop, requires careful monitoring to determine its maturity and quality before harvesting. However, traditional crop monitoring practices still depend heavily on manual observation and visual inspection performed by farmers. Although these conventional methods are widely practiced, they are often time-consuming, labor-intensive, and prone to inconsistencies caused by human error, environmental conditions, and subjective judgment in identifying the ripeness level of crops. These problems may lead to inaccurate harvesting decisions, reduced crop quality, post-harvest losses, and financial setbacks for farmers.

One major problem encountered in traditional farming is the difficulty in consistently assessing the ripeness and condition of bell pepper crops across large farming areas. Manual monitoring becomes inefficient when dealing with a

* Corresponding author: Mayumi Serote

high volume of crops because it requires continuous human labor and attention. Furthermore, farmers may experience challenges in accurately distinguishing immature bell peppers from ripe ones due to variations in color, lighting conditions, and physical appearance. Existing agricultural monitoring systems also present limitations because many require expensive hardware, high computational power, and complex processing systems, making them less practical and less accessible for small-scale or portable agricultural applications. As a result, there is a need for an efficient, lightweight, and real-time crop detection system that can support farmers in improving crop monitoring and harvesting decisions.

With the continuous growth of smart agriculture, computer vision and deep learning technologies have emerged as promising solutions for automated crop analysis and monitoring. Computer vision systems are capable of processing visual information and identifying objects based on characteristics such as color, texture, size, and shape. Among the existing object detection models, YOLOv8n is recognized as a lightweight and efficient architecture suitable for real-time agricultural applications. Unlike previous YOLO variants, YOLOv8n is optimized for faster inference speed while maintaining reliable detection accuracy using minimal system resources. These features make it highly suitable for portable agricultural monitoring systems that require real-time image detection and classification.

This study utilizes an annotated image dataset of bell pepper (Sultan F1) captured in an indoor environment to train the YOLOv8n model in detecting and classifying bell peppers according to their ripeness stages, specifically immature and ripe. The system integrates deep learning, image processing, and computer vision technologies to perform real-time crop detection and display the classification results through a monitoring dashboard. Through this approach, the study aims to address the limitations of traditional crop monitoring methods by providing a lightweight, portable, and AI-assisted agricultural detection platform capable of improving accuracy and efficiency in bell pepper ripeness assessment. Specifically, the study aims to develop a computer vision-based detection system for bell pepper (Sultan F1), classify the crop according to its maturity level using YOLOv8n, evaluate the model's effectiveness for real-time agricultural monitoring, and provide a foundation for future developments in automated object monitoring, autonomous robotics, and smart harvesting systems.

## 2. Materials and Methods

### 2.1. Related Literature

Several studies have recently emerged focusing on the detection and classification of crops using machine learning technologies. As technology emerged, these innovations provided significant support for farming and reduced the burden of manual labor [4]. This study demonstrates how artificial intelligence can improve the accuracy and efficiency of monitoring bell peppers (Sultan F1). The models trained in this study utilized a dataset sourced from Roboflow, a platform used for preparing and managing computer vision datasets for deep learning applications [5]. Computer vision is also one of the most advanced technologies, from detection and monitoring to advanced analysis, and it is changing the way agriculture operates by enabling non-contact, scalable solutions [6]. Alongside this, the object detection system not only identifies objects but also analyzes their conditions, providing effective solutions to common challenges in agriculture [7]. Vision-based intelligent system incorporate with computer vision, deep learning and machine learning allow for automated visual data interpretation for decision-making enabling real-time applications across various domains, including agriculture [8]. In precision agriculture, these systems not only support automated crops but also help to monitor ripeness assessment, and disease detection, reducing reliance on manual inspection and minimizing human error. Among modern object detection approaches, models such as YOLOv8 are widely used due to their high accuracy and real-time processing capability [9]. To improve the reliability of computer vision in algorithm tasks for image classification and identification, Convolutional Neural Networks (CNNs) demonstrate evolution with new improvements and techniques that help achieve high performance across various fields from visual data [10].

### 2.2. Methodology

This study aims to develop a bell pepper detection system that uses YOLOv8n, which proponents will design, build, and test for further use. A developmental research design of Agile Model applied, (1) including system requirements and data acquisition, (2) design and development, (3) testing, and (4) deployment. This model method was selected because the study involves developing a small-scale bell pepper detection system, including software and hardware components, to be integrated with an AI-trained model.

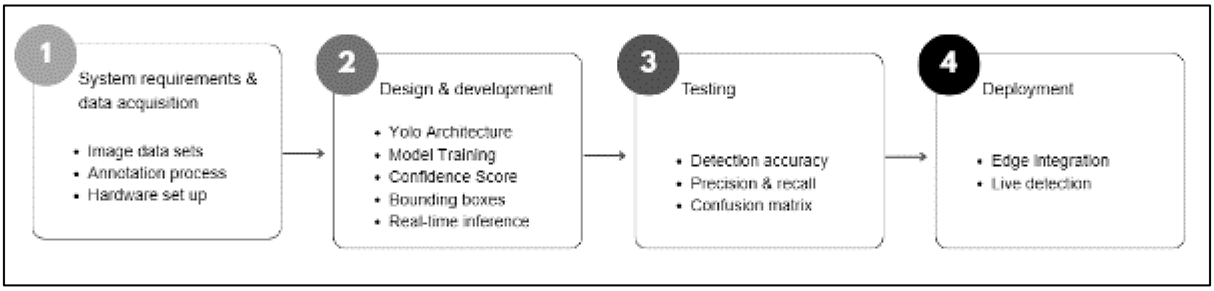


**Figure 1** Developmental Design of the Proposed System.

Figure 1 illustrates the four phases of the developmental design of the bell pepper (Sultan F1) detection system. The four phases used in this study involve the allocation of image datasets, the annotation process, and the hardware setup, all of which are then collected and analyzed. Following, training models with the YOLO (You-Only-Look-Once) architecture that achieve positive results in confidence scores and real-time inference. Then, the detection accuracy, precision & recall, and the confusion matrix of the trained YOLOv8n model are evaluated to assess its reliability for the proposed system. And the fourth phase, where the Raspberry Pi 5 edge integration is being used while also detecting the bell pepper (Sultan F1) live.

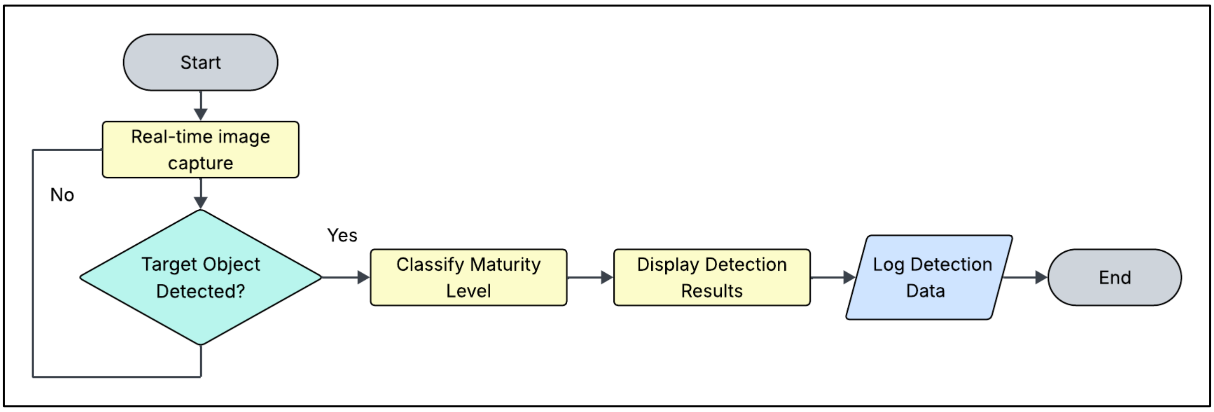


**Figure 2** Flowchart of the Bell Pepper Detection System

Figure 2 is a presentation of a comprehensive flowchart of the proposed bell pepper (Sultan F1) detection system. The workflow starts with the live camera capture, then proceeds with object detection. If the target object is indeed detected, it moves on to classify the maturity level of the bell pepper (Sultan F1), whether it is green pepper, red pepper (ripe), or transition pepper (monitor). After analyzing the maturity level of the pepper, the results will be displayed on the system's dashboard, and the log detection data will be stored.

### 2.3. Image Dataset and Annotation Process

The primary dataset of the bell pepper (Sultan F1) consists of a total of 1,048 images with a resolution of 1920x1080 px; the captured images were under an indoor environment with artificial lighting. The dataset types of data contain both the raw images, labels, and three classes of bell pepper (Sultan F1): *green_pepper* and *red_pepper* (ripe), and *transition_pepper* (monitor). These images were captured from different possible angles to support the model generalization—each image in the dataset was also annotated with bounding box labels to identify the bell pepper class, color, and whether it's for still monitoring or it's ripe.

### 2.4. System Requirements for Proposed System

The YOLOv8n, which is the nano variant of the YOLO (You-Only-Look-Once) model, was selected as the core detection architecture for the proposed system. With this variant in use, it is well-suited to a portable, resource-constrained project such as the bell pepper detection system. This model is also trained using the Ultralytics YOLO framework, which involves feeding it with the dataset—image labeling of bell pepper (Sultan F1) from Roboflow. The Geany Programmer's Editor on Raspberry Pi 5 serves as the primary tool for writing Python 3, HTML, and CSS scripts for image processing, AI inference, and the dashboard interface. Both OpenCV and Libcamera will be used for live camera capture, frame-by-frame image processing, and to draw boxes around detected bell peppers. Training in the model will be on Google Colab, where the allotted training time will be halved compared to training on a laptop, which can take days.

**Table 1** Software Requirements

| Component | Category | Description |
|---|---|---|
| YOLOv8n | AI model | Used for object and fruit maturity detection |
| Ultralytics YOLO | Framework | Model training and testing |
| Roboflow | Dataset annotation | Bell pepper (Sultan F1) image labeling |
| Geany Programmer's Editor | Code editor | Used for developing a detection system |
| Python 3 | Programing language | Image processing and AI inference |
| HTML, CSS | Frontend languages | Dashboard Interface |
| OpenCV | Computer vision library | For draw boxes, text, and display |
| Libcamera | Raspberry library | Captures frames from camera module 3 |
| Google Colab | Training platform | For training in the model detection |

**Table 2** Hardware Requirements

| Component | Specification |
|---|---|
| Laptop | AMD Ryzen 5 3550 with Radeon Vega Mobile Gfx and 16gb RAM |
| Portable Monitor | 15.6 inch |
| Raspberry Pi 5 | 4gb Ram 64gb SD card |
| Raspberry Pi Zero Camera Cable | 22-pin connector for the Pi Zero side and a 15-pin connector for the Camera module side |
| Raspberry Pi Camera Module 3 | CMOS 12-megapixel image sensor (Sony IMX708) with 11.9 MP Resolution and Phase Detection Autofocus |

The development of this project required hardware components chosen to support the demands of real-time AI-based fruit detection and embedded in the system deployment. A Laptop equipped with an AMD Ryzen 5 3550 processor, a Radeon Vega Mobile Gfx GPU, and 16GB RAM served as the primary development machine for model training, testing, and overall system development. A 15.6-inch Portable Monitor was used to display the workspace during deployment and testing. The Raspberry Pi 5, configured with 4GB RAM and a 64GB SD card, acted as the main embedded computing unit responsible for running the detection system on-site. To establish a reliable connection between the camera and the board, a Raspberry Pi Zero Camera Cable with a 15-pin connector on the Raspberry Pi 5 side and a 22-pin connector on the camera module side was used. Lastly, Image capture was handled by the Raspberry Pi 5 Camera Module 3, which features a CMOS 12-megapixel image sensor (Sony IMX708) with 11.9 MP resolution and Phase Detection Autofocus, ensuring clear and accurate image acquisition for the detection pipeline.

## 3. Results and Discussion

**Table 3** Training Results of the Yolov8n of the Proposed System

| Model (all) | Class | Box(P) | R | mAP50 |
|---|---|---|---|---|
| YOLOv8n | green_pepper | 0.998 | 0.973 | 0.994 |
| | transition_pepper | 0.976 | 1 | 0.994 |
| | red_pepper | 0.995 | 1 | 0.995 |

The training process of the YOLOv8n model was conducted on Google Colab using a Tesla T4 GPU, which provided sufficient computational resources for efficient model training without requiring a local hardware setup. A total of 67 epochs were completed in approximately 2.2 hours, during which the model progressively learned to identify and classify bell peppers by ripeness stage. The YOLOv8n architecture consists of 73 layers with approximately 3 million parameters, making it a lightweight model that operates at 8.1 GFLOPs and is well-suited for deployment on resource-constrained embedded systems such as the Raspberry Pi 5. Upon completion of training, the model demonstrated strong validation performance across all ripeness classes. For green peppers, the model achieved a precision of 0.998, a recall of 0.973, and an mAP50 of 0.994. The transition peppers class achieved a precision of 0.976, a perfect recall of 1.0, and an mAP50 of 0.994. Similarly, the red peppers class attained a precision of 0.995, a recall of 1.0, and an mAP50 of 0.995. Overall, the model achieved a precision of 0.99, a recall of 0.991, and a mean Average Precision at a 50% overlap threshold (mAP50) of 0.995, indicating near-perfect detection and classification performance. Additionally, the model achieved an inference speed of 7.5ms per image, confirming its suitability for real-time detection applications in agricultural environments.

The YOLOv8n-based model in this study uses multiple metrics and visualizations output that are generated during the training and validation of the object detection. The analysis of confidence curves that includes the F1, precision, recall, precision-recall, confusion matrices, graphs, and predictions is all included with the model being used.

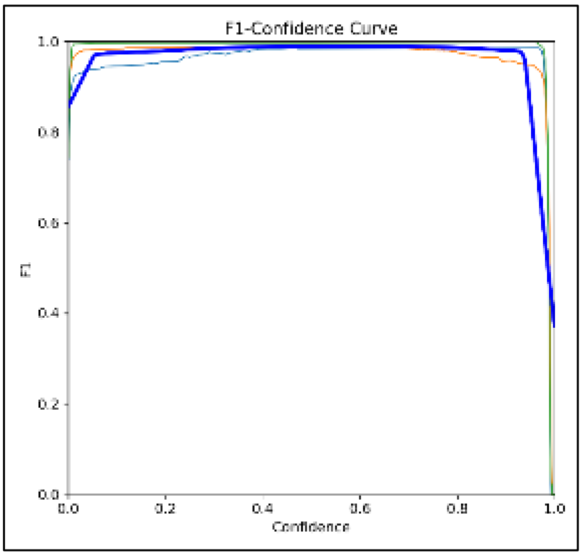


**Figure 3** F1-Confidence Curve

Figure 3 illustrates the mode's confidence threshold and the overall detection performance of the proposed system. The F1 score remained near 0.99, which is the mean of the precision and recall in the confidence range roughly from 0.05 to 0.93, indicating that the YOLOv8n model is in a stable performance, even though there are some changes to the confidence, but the optimal point of F1=0.99 at confidence 0.634.

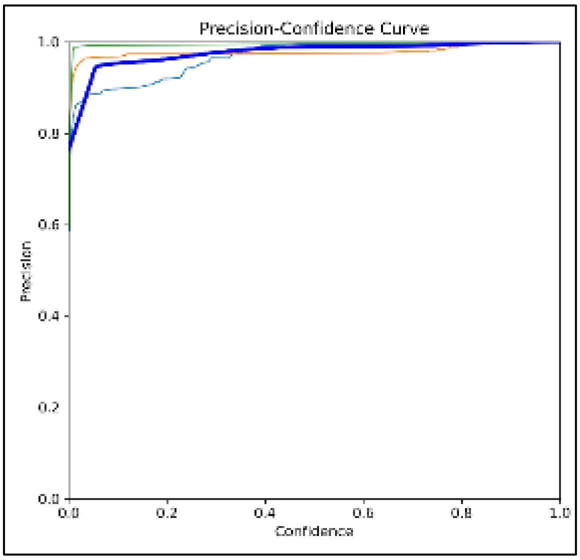


**Figure 4** Precision Confidence Curve

Figure 4 precision confidence reaches 1.00 at a high confidence, which means that the accuracy of the model detection is high. Aside from the minor overlapping of the three classes: green pepper, red pepper, and transition pepper, it still demonstrates a strong classification ability and minimal false-positive classification.

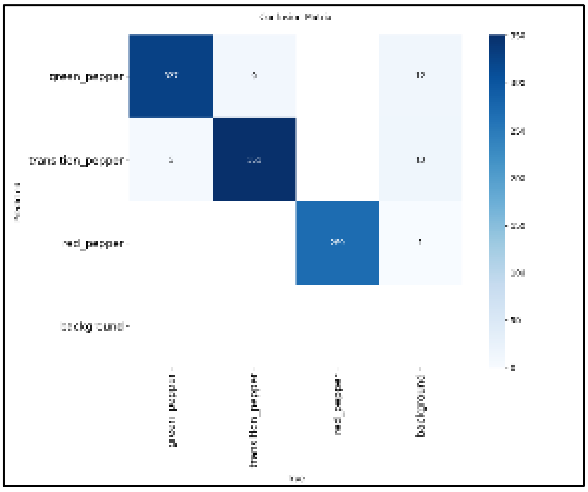


**Figure 5** Confusion Matrix

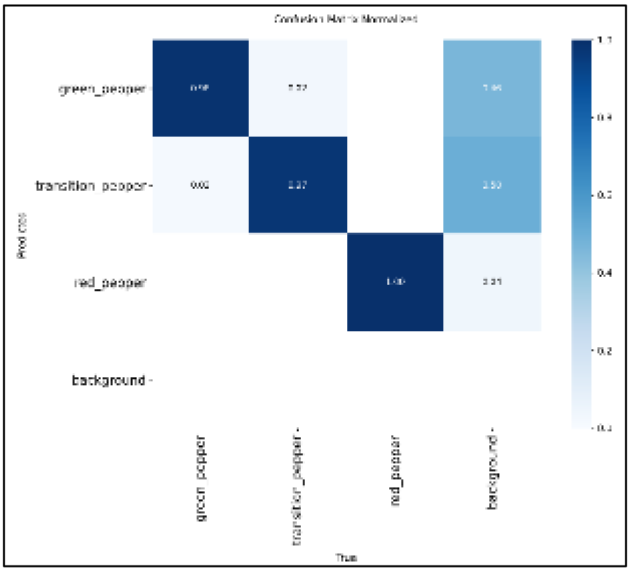


**Figure 6** Confusion Matrix Normalized

Figure 5 and 6 shows the confusion between the green pepper and the transition pepper, which is to be expected since the transition pepper is slightly similar to the green pepper class. The results of the confusion matrix on the bell peppers background, but there is no confusion that involves the red peppers.

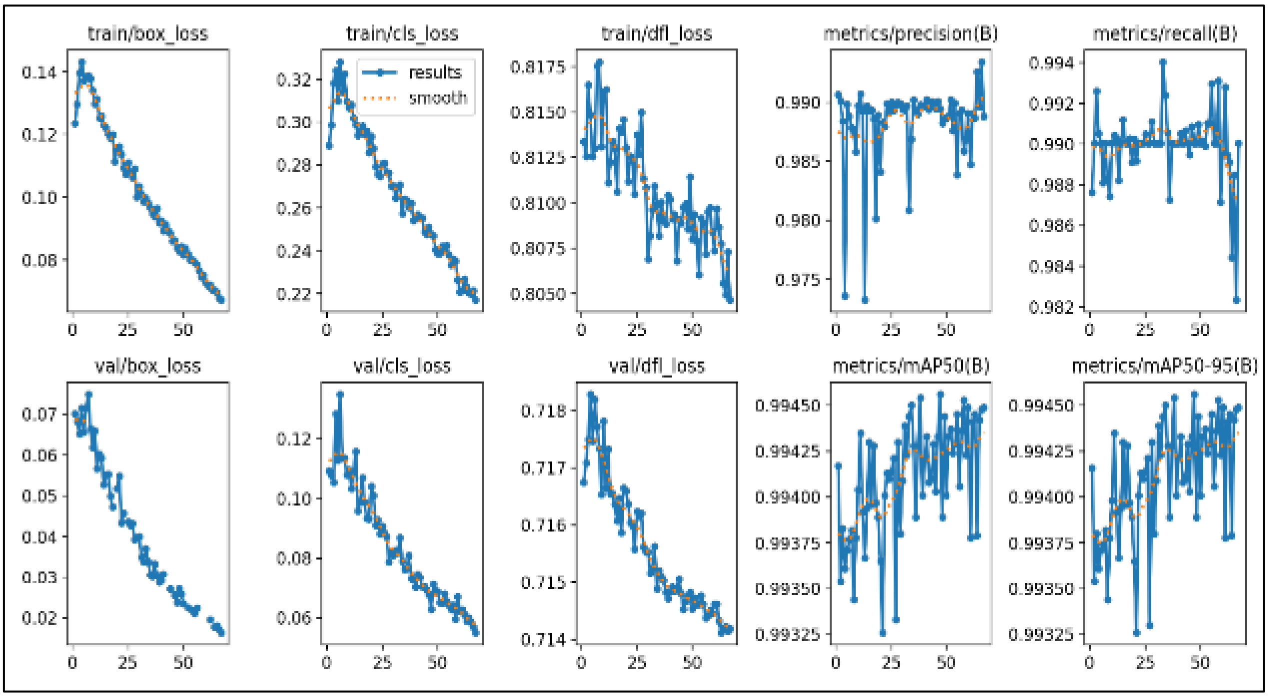


**Figure 7** Results

Figure 7 presents the performance of the training analysis, indicating that the training classification, box, and distribution of focal loss are gradually declining over successful epochs. The distribution analysis of the dataset's annotated classes, where there are 1,828 instances of green pepper, 1,950 instances of the transition pepper, and 1,633 instances of red pepper, indicates that the annotated classes were balanced. The model used achieved a mAP50 ≈ 0.9943 and mAP50-95 ≈ 0.9943; it detected the peppers and localized them accurately. The batch visualizations verified the different augmentation techniques that include mosaic augmentation, scaling, and diverse lighting, which enhanced the model against different environments.

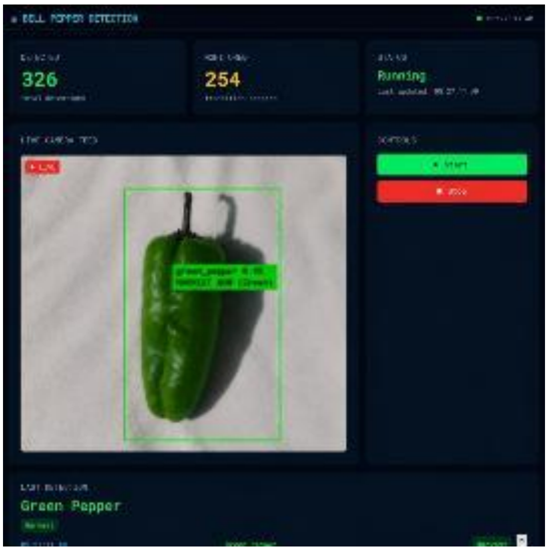


**Figure 8** Green Pepper Detection

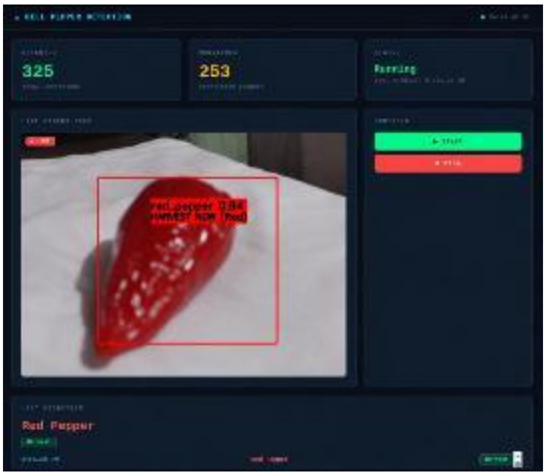


**Figure 9** Red Pepper Detection

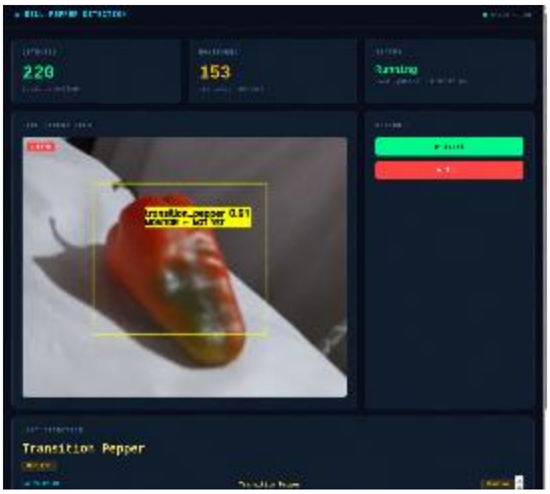


**Figure 10** Transition Pepper Detection

Figure 8-10 presents the real-time detection outputs of the deployed YOLOv8n model across the three bell pepper categories: transition pepper, green pepper, and red pepper. Each screenshot captures the model operating through a custom monitoring dashboard that displays a live camera feed alongside key inference metrics, including total detection count, inference count, and system status. Bounding boxes and confidence scores are rendered directly onto the live frame, allowing immediate visual verification of each classification. The dashboard also provides manual controls for starting and stopping the detection process, as well as a last-detected class indicator at the bottom of the interface, enabling users to track recognition events in sequence. The system successfully identified and localized each pepper class under varying lighting conditions and backgrounds, demonstrating that the YOLOv8n model maintains consistent accuracy and responsiveness within the developed interface, making it suitable for practical, real-time agricultural monitoring and classification applications.

## 4. Conclusion

This study was able to develop a working bell pepper (Sultan F1) detection and classification system that runs in real time using the YOLOv8n model deployed on a Raspberry Pi 5. Throughout the development process, the researchers found that a lightweight AI model, when properly trained and optimized, can still achieve strong and reliable results even on hardware with limited resources. The trained model performed well across all three pepper categories — green pepper, transition pepper, and red pepper — reaching high scores in precision, recall, and mAP50, which indicates that the system can accurately identify and classify each maturity stage. The integration of a custom monitoring dashboard also made the system more practical and easier to use, as it allowed users to view live detection results without needing a deep technical background. Overall, this project shows that AI-powered crop monitoring does not have to be expensive or complicated to be effective, and the researchers hope that this work can serve as a starting point for similar studies involving other crops or more advanced deployment setups in the future.

## Compliance with ethical standards

### *Acknowledgments*

The researchers would like to sincerely thank Jesus Reigns Christian College, particularly the Department of Information Technology, for the guidance and support given throughout this study. The researchers are also grateful for the platforms that made this project possible, including the cloud-based training environment that helped speed up the model training process, the dataset management platform used for image annotation, and the open-access journals. Lastly, the researchers would like to thank everyone who encouraged us during the development and completion of this work.

### *Disclosure of conflict of interest*

The researchers declare that there is no conflict of interest associated with this study. This research was conducted purely for academic purposes, and no financial support, sponsorship, or external influence was involved in the development, analysis, or reporting of the work presented in this paper.

### *Statement of ethical approval*

The researchers confirm that this study did not involve any human participants or animal subjects. The dataset used consisted solely of images of bell pepper (Sultan F1) specimens that were captured by the researchers themselves in a controlled indoor setting. No private, sensitive, or third-party data was collected without proper authorization. The researchers ensured that all aspects of this study were conducted honestly and responsibly, in keeping with the ethical standards expected in academic research.

## References

[1] Stein A. Digitalization in Agricultural Production. Cham: Springer; 2026. p. 155-172. doi:10.1007/978-3-032-09098-0_15.

[2] Chen YL, Hung KC, Zhang JY, Lin LW. Improved YOLOv8n Models for Object Detection in Remote Sensing Images. IEEE Access. 2025;13:97889-97899. doi:10.1109/ACCESS.2025.3574856.

[3] Khan Z, Shen Y, Liu H. Object Detection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions. Agriculture. 2025;15(13):1351. doi:10.3390/agriculture15131351.

[4] Woźniak M, Ijaz MF. Recent advances in big data, machine, and deep learning for precision agriculture. Frontiers in Plant Science. 2024;15:1367538.

[5] Mamat N, Othman MF, Abdulghafor R, Belhaouari SB, Mamat N, Hussein SFM. Advanced Technology in Agriculture Industry by Implementing Image Annotation Technique and Deep Learning Approach: A Review. Agriculture. 2022;12(7):1033.

[6] Mamalis M, Kalampokis E, Kalfas I, Tarabanis K. YOLO deep learning algorithm for object detection in agriculture: a review. J Agric Eng. 2024;55(4):1641. doi:10.4081/jae.2024.1641

[7] Tsouros DC, Bibi S, Sarigiannidis PG. Images and CNN applications in smart agriculture. Eur J Remote Sens. 2024;57(1):2352386. doi:10.1080/22797254.2024.2352386

[8] Kamilaris A, Prenafeta-Boldú FX. Computer vision in smart agriculture and precision farming: techniques and applications. Smart Agric Technol. 2024;8:100466. doi:10.1016/j.atech.2024.100466

[9] Sharma R, Patel N, Gupta A. YOLO-based deep learning framework for real-time multi-class plant health monitoring in precision agriculture. Sci Rep. 2025;15:29132. doi:10.1038/s41598-025-29132-w

[10] Nguyen T, Kim J, Park S. A review of CNN applications in smart agriculture using multimodal data. Computer Electron Agric. 2025;220:108768. doi:10.1016/j.compag.2025.108768