Developing AI-powered Image Processing Web Application for Plant Monitoring in Smart Farm

Kim Zuo, Do Jun Kwon, JongYoon Lim, Bruce A. MacDonald, and Ho Seok Ahn*
Centre for Automation and Robotic Engineering Science
The University of Auckland, New Zealand
{yzuo500, dkwo575}@aucklanduni.ac.nz, {jy.lim, b.macdonald, hs.ahn}@auckland.ac.nz

Abstract

The growing global food demand, coupled with climate change and resource scarcity, necessitates innovative solutions for efficient farmland management. This paper proposes a smart agriculture system utilising an AI-powered image processing web application to enhance crop management and reduce labour. Our system integrates computer vision AI models to monitor plant growth, including size, ripeness, and disease detection, alongside a web interface displaying plant status. This allows for informed decision-making by collecting, processing, and displaying real-time data on plant growth and environmental conditions. A user survey rated the system's usability and performance highly, with an average satisfaction of 4.7 out of 5, noting the interface as easy to navigate, beneficial, and visually appealing. By combining sensing technology and artificial intelligence, we show the potential of comprehensive crop monitoring solutions that improve agricultural productivity and sustainability. This approach leads to more efficient and data-driven agricultural practices, which can demonstrate excellent efficiency in the agricultural sector.

1 Introduction

The agricultural sector faces mounting challenges due to climate change, population growth, and resource scarcity, necessitating improved productivity and sustainability [Food and Agriculture Organization, 2017]. Artificial Intelligence (AI) has emerged as a promising solution to these challenges, offering ways to enhance crop management and optimize agricultural practices [Liakos *et al.*, 2018a].

Our research proposes an innovative smart agriculture system that leverages AI to improve crop management and streamline plant monitoring. By collecting, processing, and displaying real-time data on plant growth and environmental conditions, the system aims to support farmers and agricultural experts in making more informed decisions [Wolfert *et al.*, 2017].

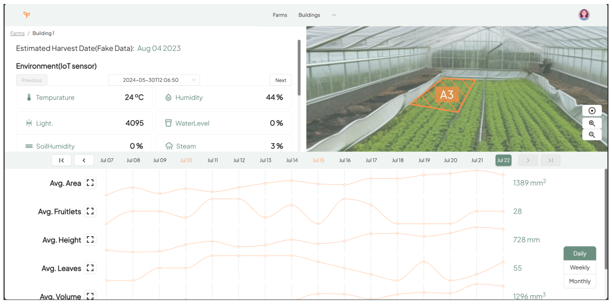
The key contributions of our research include:

1. AI-based monitoring system: We present an integrated system combining environmental variable detection, multi-model AI processing for tomato size and ripeness measurement, and real-time data collection and visualization.
2. Web Application for the smart farm: We developed a web application using Flask and created an intuitive React-based web interface, enabling efficient data transfer and user-friendly access to environmental variable data and analyzed information.
3. Real-world validation: We conducted tests in an actual tomato greenhouse environment, providing valuable insight into system effectiveness and identifying areas for future improvement. In addition, we analyzed what needs to be improved in the future by reflecting users' feedback on usability.
4. Open-Source Implementation: Our code is publicly accessible, contributing to the open-source community and facilitating further development and adaptation in smart agriculture. [1]

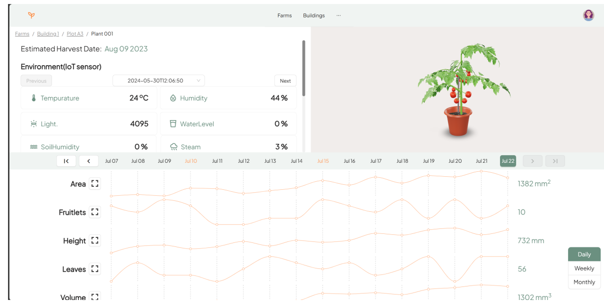
To address implementation challenges and verify the system's effectiveness, we collected real-world data in a tomato greenhouse. This practical application allowed us to explore the technology's feasibility and identify areas for future improvement. Our research contributes to the growing body of knowledge on smart agriculture applications and provides insights into the practical challenges and opportunities of implementing AI-driven crop monitoring solutions [Bacco *et al.*, 2018].

The user experience is a vital aspect of such systems, as effective adoption relies on intuitive and efficient interfaces. To evaluate the usability and performance of the web interface, we conducted a user survey, receiving overwhelmingly positive feedback. The interface was

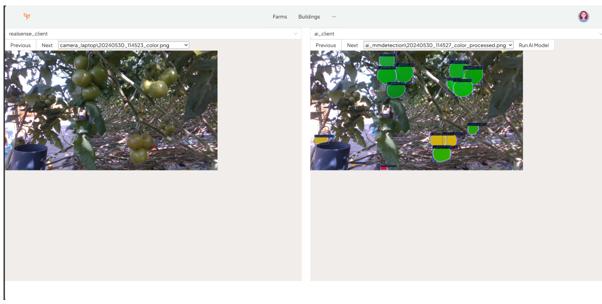
[1] https://github.com/dkwo575/P4P-61-Webpage



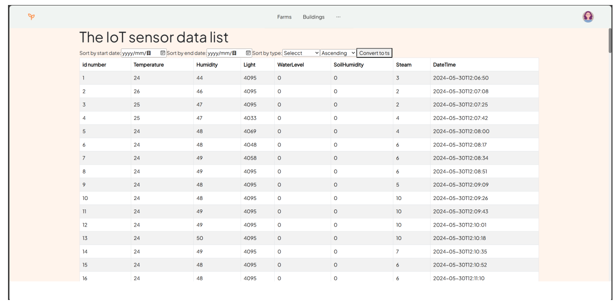
(a) Region Select page. Users can select buildings and areas to inspect.



(b) Plant Page. Users can view the environmental data of the plant.



(c) Image Page. Users can view the plant ripeness results.



(d) MySQL Database displayed in the Web Interface.

Figure 1: Overview of the Frontend Pages

rated an average of 4.7 out of 5, indicating a strong user experience across several dimensions, including ease of navigation, clarity of data visualization, and intuitive design. These results underscore the potential of our AI-based approach to enhance the efficiency of agricultural management while minimizing manual labour.

By combining sensing technologies with artificial intelligence, our system aims to create a comprehensive crop monitoring solution that can potentially lead to more efficient, data-driven agricultural practices. This research aligns with the broader goal of improving agricultural productivity and sustainability through technology, addressing the pressing need for innovative solutions in modern agriculture [Walter *et al.*, 2017].

In this paper, we describe related works in Section 2. We explain our plant growing monitoring system in Section 3 and show the results in Section 4. We conclude this paper in Section 5.

2 Related Works

2.1 AI-driven Web Platforms for Smart Farming

AI-driven web platforms are essential for smart farming, enabling seamless integration of data collection, storage, processing, and visualization [Faid *et al.*, 2021]. These platforms typically consist of a back-end database, AI models, APIs, and user-friendly interfaces that allow farmers to easily access and interpret complex agricultural data.

The iFarm system developed by Murakami is an example of a web-based cultivation and cost management system that supports efficient agricultural management by utilizing smartphones, cloud servers, and web browsers [Murakami, 2014]. In this iFarm system, farmers and managers can check various information about the farm using a website or smartphone application and upload work plans to the server based on the information. All data is stored in the cloud server, and other managers can check this data. Using this system, the work efficiency of farm staff has increased [O'Grady and O'Hare, 2017].

Bhat and their research team emphasized the importance of big data in smart agriculture and focused on how it can be used in agriculture. They said that big data analysis should be combined with massive data collection and storage, data analysis, AI, and predictive analysis [Bhat and Huang, 2021]. They said that the combination of big data and AI in the agricultural field could solve many of the current challenges facing agriculture and contribute to greatly increasing the quality and quantity of agricultural production [Osinga *et al.*, 2022]. They also said that AI and big data can be used to predict soil quality, diseases and pests, and crop harvesting times [Bhat and Huang, 2021].

2.2 AI-based Plant Growth Monitoring

Artificial intelligence (AI) has emerged as a powerful tool for analyzing the vast amounts of image data generated by cameras in smart agricultural environments [Eli-Chukwu, 2019]. AI technologies, particularly machine learning and computer vision, have been applied to various aspects of crop management, plant disease detection, yield estimation, and decision-making processes [Tian *et al.*, 2020].

JinhaJung developed an AI-based system for early detection of plant diseases using image analysis. Deep learning models trained on a large dataset of leaf images achieved 95 percentage accuracy in identifying common crop diseases. The system demonstrated the potential for rapid, automated disease detection, enabling timely intervention and reduced crop losses [Jung *et al.*, 2021]. In Haiyan and Yong's study, they designed an intelligent diagnosis system that helps non-skilled farmers detect plant diseases or nutritional deficiencies. They used Artificial Neural Networks (ANNs) to improve the accuracy of this system. This ANN-based system made the diagnosis system that was previously only available to experts available to non-skilled farmers and collected and provided visual symptoms using images so that non-skilled farmers could easily identify them. Their system analyzed crop symptoms into five parts (overall symptoms, root symptoms, leaf symptoms, fruit symptoms, and crop factors) and made a diagnosis based on them. As a result, the error rate was less than 8 percentage compared to field verification, and it was easy for non-skilled farmers to use [Song and He, 2005].

In Ramos' study, they developed a method to determine and estimate coffee yield by counting the number of coffee cherries using computer vision and AI. They counted coffee cherries into three categories: ripe, unharvestable, and intermediate. In addition, the weight and maturity rate of coffee cherries were estimated using this method. The purpose of this work was to obtain production data using the number of cherries and then allocate manpower and prepare machinery in advance based on the data. In addition, this was to provide information that maximizes economic benefits for coffee growers and makes it easier to plan agricultural operations [Ramos *et al.*, 2017].

In Konstantinos's study, he emphasized the application of machine learning in agricultural production. They said that in the agricultural sector, machine learning can be used to perform tasks such as yield prediction, weed detection, disease detection, and crop quality [Liakos *et al.*, 2018b]. Most of these tasks are performed by applying machine learning to data using sensor data and computer vision, and farm management systems are evolving into practical artificial intelligence systems, which can maximize the productivity and economic benefits of farmers and manage crops more efficiently [Tian *et al.*, 2020].

2.3 Environment Monitoring System

Environment sensor systems are now essential for smart farming, and IoT technology is already being used in most smart farming fields [Dagar *et al.*, 2018]. IoT sensor systems enable real-time monitoring of environmental variables and data collection [Sushanth and Sujatha, 2018]. These systems can consist of a network of sensors that measure various environmental conditions, soil properties, and plant health indicators [Jaiganesh *et al.*, 2017].

Rahman stated in his study that utilizing IoT in agriculture can increase efficiency, reduce costs, and improve the economic level of agriculture. He also stated that utilizing IoT technology can make farms smarter, more efficient, and more productive, which can effectively reduce time and financial resources by reducing manual work and increasing automation. According to their study, environmental monitoring is important in smart agriculture [Rahman *et al.*, 2023]. Environmental monitoring is one of the important factors in smart agriculture to maintain environmental conditions by using various sensors to monitor important environmental factors such as temperature, humidity, and soil moisture [Mico *et al.*, 2016].

Dolci emphasized that IoT technology can be used to increase the level of sophistication in agriculture without investing a lot of capital. In this study, Dolci stated that IoT technology using low-cost sensors such as humidity, temperature, light, and soil moisture can help farmers easily monitor environmental variables and achieve high results with low capital [Dolci, 2017]. These sensors provide farmers with important data to optimize irrigation, pest control, and overall crop management [Sushanth and Sujatha, 2018].

Wolfert introduced tomato indoor farm technology utilizing IoT technology. Developing various sensor and actuator-based systems can provide users with information related to crop growth, climate, and irrigation settings using environmental data from sensors, thereby improving crop quality. It also made this sensor data easily accessible to users through a web-based decision-making system [Jung *et al.*, 2022]. Therefore, it is possible to reduce resource use and increase productivity based on crop monitoring using IoT system technology [Wolfert and Isakhanyan, 2022].

3 Plant Growing Monitoring System

3.1 System Architecture

Figure 2 illustrates our Plant Growing Monitoring System Architecture, which consists of three key components:

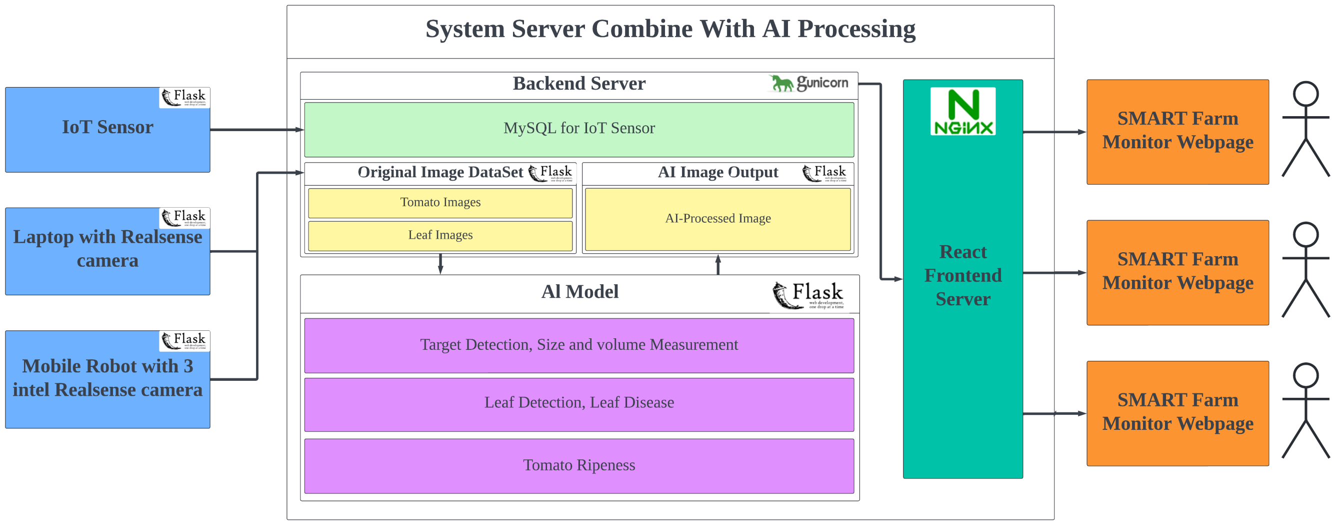


Figure 2: Overall System Architecture.

1. Input Devices (left):
   - IoT Sensor for environmental data collection
   - Laptop with RealSense camera for stationary image capture
   - Mobile Robot with 3 Intel RealSense cameras for dynamic image collection
2. AI Server (center):
   - Backend Server, which includes:
     - MySQL database for IoT Sensor data storage
     - Original Image Dataset for storing Tomato and Leaf images
     - AI Image Output for processed images
   - AI Model, performing:
     - Target Detection, Size and Volume Measurement
     - Leaf Detection and Disease Identification
     - Tomato Ripeness Analysis
3. User Interface (right):
   - SMART Farm Monitor Webpage, accessible by multiple users simultaneously

The client-side devices form the foundation of our data gathering process. IoT sensors deployed across the agricultural environment continuously collect vital environmental data such as temperature, humidity, and soil moisture. RealSense cameras, which can be mounted on autonomous robots or connected to portable devices, capture high-resolution images of crops. These devices serve as our system's eyes and ears, providing a comprehensive view of the agricultural landscape.

At the heart of our architecture lies the AI server, a powerful integration of backend services and artificial intelligence models. This server acts as the central nervous system of our setup, handling data management and analysis. Environmental data from IoT sensors is securely transmitted to a MySQL database within the server. Images from RealSense cameras are uploaded via a Flask application using HTTP protocol, ensuring reliable and secure data transfer. These images are stored in a dedicated dataset, ready for AI processing.

Our server system, initially developed and tested on local systems, is designed with flexibility in mind. While the current architecture shows the AI model integrated within the server, they can be easily deployed on separate machines if needed. This adaptability allows for efficient resource allocation, preventing overload on the main server and enabling scalable performance as the system grows. These AI models analyse the collected images, determining crop health, growth stages, and potential issues, with results stored in a designated server folder for easy access and review.

The user interface, developed using React, serves as the window into our system's insights. Displayed on the right side of our architecture diagram, this web application allows multiple users to simultaneously access and interact with the processed data in real-time. Users can view environmental trends, crop status updates, and AI-generated insights, all presented in an intuitive and user-friendly format.

This integrated and scalable architecture ensures a seamless flow of data from initial collection through analysis to final presentation. It enables efficient smart agricultural monitoring and informed decision-making, adaptable to farms of various sizes and complexities. By combining IoT technology, artificial intelligence, and user-centric design, our system paves the way for more sustainable and productive agricultural practices in an era of increasing global food demand.

3.2 Plant Growing Monitoring

Figure 3 illustrates the workflow of our plant growing monitoring method delineating into two primary sections:

1. Combined Backend (highlighted in amber): These components include image collection, image processing and data management functions.
2. Frontend System (highlighted in green): This section represents the user interface and client-side operations.

These components interact through two distinct communication channels:

1. HTTP Protocol (represented by red arrows): Use API calls, enabling request-response interactions for data retrieval and submission.
2. WebSocket Message Transfer (represented by blue arrows): Facilitates real-time, bidirectional communication, allowing for instant updates and live data streaming between the server and client.

Content Bellow is our System Workflow in detail:

1. **Image Capture and Upload:**
   - The Realsense Camera Client captures raw images.
   - The client uploads raw PNG images and NPY data files to the Backend Server (using HTTP protocol).
2. **Server Notification:**
   - Upon receiving the uploaded data, the Backend Server broadcasts a notification to all connected clients via WebSocket.
   - The notification includes information about the upload event from realsense_client.
3. **AI Client Data Request:**
   - The AI Processing Client (mmdetection) receives the WebSocket notification and sends a download request to the Backend Server.
4. **Data Transfer to AI Client:**
   - The Backend Server sends the PNG image and NPY data to the AI Client (using HTTP protocol).
5. **AI Processing and Result Upload:**
   - The AI Client processes the received PNG and NPY data using the mmdetection model.
   - After processing, it generates a new PNG image with detection results.
   - The AI Client uploads the processed image back to the Backend Server (using HTTP protocol).
6. **Web Application Display:**
   - The Backend Server sends the processed image to the Frontend Server.
   - The Frontend Server displays both the original and processed images on the Web application page for user viewing.

Paragraph bellow will describe our components in systems in detail.

On the left side of the diagram, the client script controls the RealSense camera and manages data upload and download. When a photo is captured, it is automatically sent to the server script via HTTP protocol. Upon receiving a new file, the server logs relevant information such as file format, uploading client ID to denote file type, and download URL. This information is then broadcast via WebSocket to all clients currently connected to the server.

When other clients receive notifications about files with IDs different from their own, they automatically download these files. Clients with image processing capabilities, such as the one depicted on the right side of the diagram, analyze the downloaded files using AI models and return processed images to the server.

To enhance system scalability and future-proof our design, we've implemented two crucial identifiers for each client Python script: `client_id` and `client_name`. These identifiers are transmitted to the server via HTTP protocol during file uploads, serving as references for file storage. This design also allows the server to communicate with specific clients using WebSocket, sending targeted messages to trigger various functions.

This identification system was initially conceived to prevent duplicate uploads and downloads from the same client, as our server supports both file upload and download functionalities. It ensures that clients upload their own files and download only the content they need. During development, we recognized the necessity of unique client IDs to prevent automatic downloading of self-uploaded files, particularly important when the server notifies all connected clients of new uploads via WebSocket.

This system enables efficient and targeted file distribution among multiple clients, significantly enhancing the functionality and scalability of our smart agricultural monitoring system. It provides a robust foundation for future expansions, such as incorporating additional AI models or adapting the system for other agricultural monitoring projects.

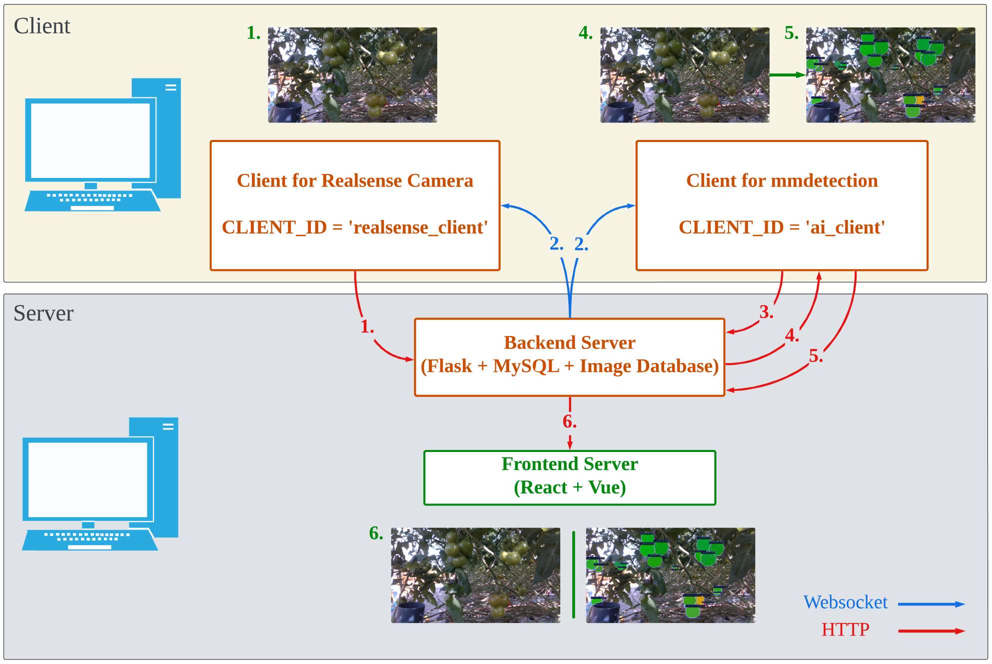


Figure 3: Plant growing monitoring Workflow.

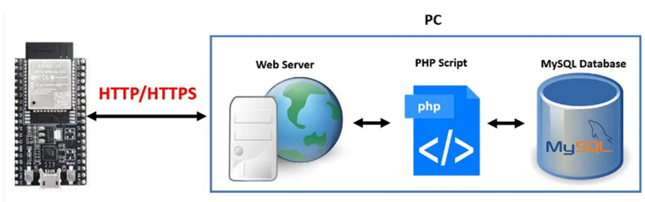


Figure 4: Sensor Workflow

3.3 Environment Monitoring

Our plant growth monitoring system includes a system that can monitor environmental variables. The environment monitoring system is designed to collect, transmit, store and visualize important environmental parameters that affect plant growth. By utilizing various sensor technologies and IoT technologies, our system provides detailed data in real-time for plant growth analysis and management.

First, the hardware of this system uses the ESP32 development board as the central processing unit. This microcontroller unit (MCU) has low power consumption, built-in WIFI function and the ability to interface with multiple sensors [Oleinikov *et al.*, 2023]. The excellent processing power of ESP32 enables real-time data collection and processing, and the built-in WiFi module facilitates data transmission to the database server.

Our system measures six sensor data: temperature, humidity, light, steam, soil humidity and water level using various sensors such as DHT11, steam sensor, Photoresistance, Soil Humidity sensor and water level sensor. Each sensor is connected to a microcontroller and the MCU is programmed to read data from the sensor regularly (every 30 seconds in the current configuration). Figure 4 shows how the collected data is stored in the MySQL database server. The collected data is transmitted to a local web server via HTTP protocol using WiFi. After that, the PHP script receives an HTTP POST request from the ESP32 and stores the transmitted data in a MySQL database, and each entry is timestamped for time analysis.

In order to facilitate data analysis and monitoring, the collected environment variables can be displayed through a web-based interface. The web interface was developed using React.js, and when the backend created with Flask makes an HTTP GET request to the MySQL server, the server displays the requested data on the web interface. All data is collected in real-time, and the collected environment variables can be viewed in a table or menu format for easy use by users. Users can select a date for data analysis and monitor environment variables anywhere with a mobile device.

3.4 Web Interface

Figure 1 shows different pages in our frontend web interface. We selected some of the more representative pages. The BuildingPage and PlantPage are quite similar; they display recent farm environment data in a clear manner using widgets, charts, and cartoon-style UI icons that retrieve data from the MySQL database.

In the IoTDataPage, users can more efficiently analyze and compare recent data. With the Sorting Function at the top of the page, they can quickly identify which environmental factor reached its peak on a given day. By selecting any environmental factor, users can arrange the data in ascending or descending order for easier visualization.

In the ImagePage, at the top of the columns on both sides, users can select different clients based on the client_id. In our server's directory, we have two main folder types, Original and Result, to differentiate between different images. Within these, folders are further organized by client_id, and within those, by client_name to create subfolders. This structure allows the images on both sides to be compared against each other, helping the user gain insights. By selecting the client_id, users can switch between different AI models, and when choosing files, they can refer to the client_name part of the file path (e.g., "camera_laptop" in the figure) to understand which client sent the file back to the server.

4 Experimental Results

To simulate file transmission in a real-world environment, we implemented a distributed setup as shown in the Figure 3. We separated different scripts to run on distinct devices, effectively emulating a realistic client-server architecture.

One device was designated to simulate the server, while another device emulated various clients interacting with the server through upload and download operations, as well as other corresponding actions. This configuration allowed us to test the system's functionality in a distributed environment, closely mimicking real-world conditions.

After ensuring that the entire system functioned exactly as intended in this distributed setup, we proceeded to integrate the whole system on a Linux machine using Gunicorn and Nginx. This additional step was taken to further test our server's performance and reliability in a Linux environment, which is commonly used for production deployments.

By implementing this two-phase testing approach - first with distributed devices and then with an integrated Linux setup - we were able to thoroughly validate our system's capabilities under various operational scenarios. This comprehensive testing strategy helps ensure that our smart agricultural monitoring system can perform robustly in real-world applications, whether in a distributed network or on a centralized Linux server.

To further enhance the realism of our tests, we utilized actual environmental data collected from a New Zealand hothouse using IoT sensors and a RealSense camera. This data served as input to our system, simulating real-world conditions and allowing us to evaluate the system's performance with authentic agricultural data. This approach provided valuable insights into how our system would function in a genuine smart farming environment.

4.1 Size Measurement Results

Figure 5d showcases the output of our size measurement AI model. When provided with an input image of a tomato, this model performs the following tasks:

- Automatically estimates the actual size of the tomato,
- Superimposes the size information directly onto the image,
- Generates a new image with the size annotation,
- Returns this processed image to the server.

This capability allows for non-invasive, automated size tracking of tomatoes throughout their growth cycle, providing valuable data on crop development without manual measurements.

4.2 Ripeness Measurement Results

Figure 5h illustrates the functionality of our ripeness assessment AI model. This model processes tomato images and performs the following actions:

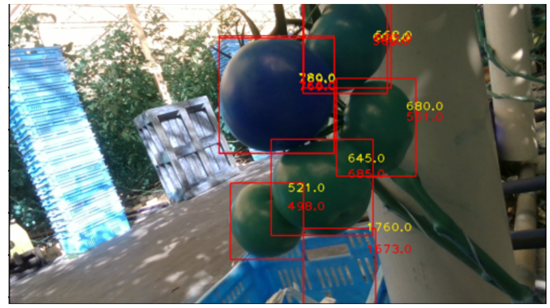
- Identifies and outlines the approximate position of each tomato in the image,
- Analyzes the visual characteristics of the tomatoes to assess their ripeness,
- Classifies the tomatoes according to their maturity levels,
- Annotates the image with ripeness information for each identified tomato,
- Returns the annotated image to the server.

This model enables automated tracking of tomato ripeness across multiple plants simultaneously, offering farmers a comprehensive view of crop maturity without the need for manual inspection.

4.3 Environment Sensing Results

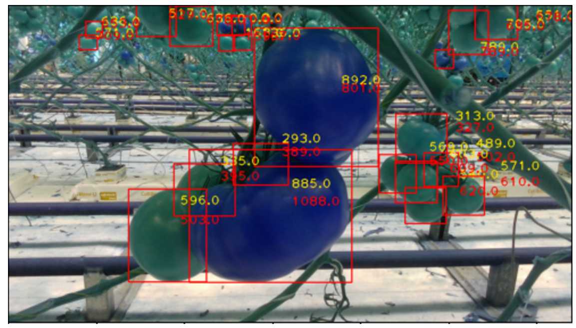
Figure 1a shows the environment sensor data result. The sensor data is stored in the MySQL database server. We visited an actual tomato farm and collected data from there. After collecting sensor data every 30 seconds, the Microcontroller Unit (MCU) sends the data to the MySQL server via WiFi. The data collection results showed that the average temperature of the farm was 24°C, and the humidity was between 40 and 50 percentage, which was very similar to the information provided by the farm manager. Also, because it was a greenhouse farm, the light was maintained at a constant value. The sensor data is very similar to reality and has excellent accuracy.

(a) Original Image for Size Measurement



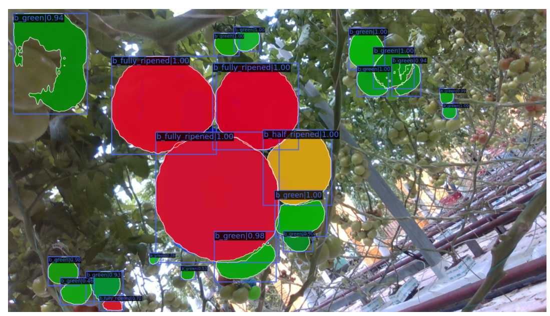
(b) Size Measurement Result

(c) Another Example of Original Image for Size Measurement



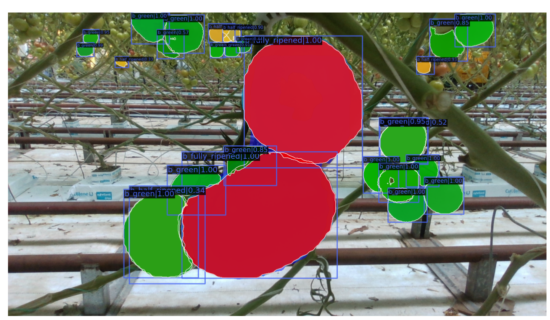
(d) Another Example of Size Measurement Result

(e) Original Image for Ripeness Measurement



(f) Ripeness Measurement Result

(g) Another example of Original Image for Ripeness Measurement



(h) Another example of Ripeness Measurement Result

Figure 5: Size and Ripeness Measurement Results

5 Conclusions

We introduce the process of building a Tomato Monitoring System from scratch, capable of being used in real-world environments. We utilize Python and Flask as the backend server to manage interactions between the server and different components within the system, such as robots and AI models, while React serves as the frontend to enable users to access the system from various devices.

By collecting real-world data and simulating farm or greenhouse conditions through launching servers and corresponding components on different devices, we demonstrate that our system is operable and usable in real-world environments. Additionally, with the help of Gunicorn and Nginx, the server can be hosted on Linux devices and communicate with devices on the local network, or even project the system onto public websites, allowing users to remotely monitor farm conditions in real-time from miles away.

By integrating various AI models into the system, this platform can also be adapted as a monitoring system for other crops. It can further evolve into a diversified network architecture capable of facilitating direct communication between different AI models and clients. This aspect reflects our long-term vision for future work during the design phase. We aim for this network architecture to serve as a foundation for building various AI systems or monitoring solutions. In the future, the system can also consider security measures such as encryption for file uploads and downloads to ensure data protection when the website is exposed to the internet, preventing unauthorized access to server contents.

Acknowledgments

This research was funded by the New Zealand Ministry for Business, Innovation and Employment (MBIE) on contract UOAX2116, Artificial Intelligence-based Smart Farming System, and funded by Ministry of Trade, Industry and Energy (MOTIE), Republic of Korea, Development of an Agricultural Robot Platform Capable of Continuously Harvesting more than 3 Fruits per minute and Controlling Multiple Transport Robots in an Outdoor Orchard Environment. Ho Seok Ahn* is the corresponding author.

References

[Bacco *et al.*, 2018] Manlio Bacco, Andrea Berton, Erina Ferro, Claudio Gennaro, Alberto Gotta, Stefania Matteoli, Fabio Paonessa, Massimiliano Ruggeri, Giuseppe Virone, and Alberto Zanella. Smart farming: Opportunities, challenges and technology enablers. In *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, pages 1–6, 2018.

[Bhat and Huang, 2021] Showkat Ahmad Bhat and Nen-Fu Huang. Big data and ai revolution in precision agriculture: Survey and challenges. *IEEE Access*, 9:110209–110222, 2021.

[Dagar *et al.*, 2018] Rahul Dagar, Subhranil Som, and Sunil Kumar Khatri. Smart farming–iot in agriculture. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1052–1056. IEEE, 2018.

[Dolci, 2017] Rob Dolci. Iot solutions for precision farming and food manufacturing: Artificial intelligence applications in digital food. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 384–385, 2017.

[Eli-Chukwu, 2019] Ngozi Clara Eli-Chukwu. Applications of artificial intelligence in agriculture: A review. *Engineering, Technology & Applied Science Research*, 9(4), 2019.

[Faid *et al.*, 2021] Amine Faid, Mohamed Sadik, and Essaid Sabir. An agile ai and iot-augmented smart farming: a cost-effective cognitive weather station. *Agriculture*, 12(1):35, 2021.

[Food and Agriculture Organization, 2017] Food and Agriculture Organization. *The future of food and agriculture*. Food & Agriculture Organization of the United Nations (FAO), Rome, Italy, June 2017.

[Jaiganesh *et al.*, 2017] S. Jaiganesh, K. Gunaseelan, and V. Ellappan. Iot agriculture to improve food and farming technology. In *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*, pages 260–266, 2017.

[Jung *et al.*, 2021] Jinha Jung, Murilo Maeda, Anjin Chang, Mahendra Bhandari, Akash Ashapure, and Juan Landivar-Bowles. The potential of remote sensing and artificial intelligence as tools to improve the resilience of agriculture production systems. *Current Opinion in Biotechnology*, 70:15–22, 2021. Food Biotechnology.Plant Biotechnology.

[Jung *et al.*, 2022] Jimin Jung, Jihyun Lee, and Hyemin Noh. Web-based data analysis service for smart farms. *KIPS Transactions on Software and Data Engineering*, 11(9):355–362, 2022.

[Liakos *et al.*, 2018a] Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8), 2018.

[Liakos *et al.*, 2018b] Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. Machine learning in agriculture: A review. *Sensors*, 18(8), 2018.

[Mico *et al.*, 2016] Onine M Mico, Paul Bryan M Santos, and Rionel B Caldo. Web-based smart farm data monitoring system: A prototype. *J. Eng. Comput. Stud.*, 3(3), 2016.

[Murakami, 2014] Yukikazu Murakami. ifarm: Development of web-based system of cultivation and cost management for agriculture. In *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, pages 624–627, 2014.

[O'Grady and O'Hare, 2017] Michael J O'Grady and Gregory MP O'Hare. Modelling the smart farm. *Information processing in agriculture*, 4(3):179–187, 2017.

[Oleinikov *et al.*, 2023] Nikolay N. Oleinikov, Anatoliy N. Kazak, Anna A. Dorofeeva, Nadezhda K. Boyarchuk, Daniil V. Gorobets, and Dmitry V. Nekhaychuk. Data collection with esp32 microcontroller for training neural networks. In *2023 Seminar on Information Computing and Processing (ICP)*, pages 227–229, 2023.

[Osinga *et al.*, 2022] Sjoukje A Osinga, Dilli Paudel, Spiros A Mouzakitis, and Ioannis N Athanasiadis. Big data in agriculture: Between opportunity and solution. *Agricultural Systems*, 195:103298, 2022.

[Rahman *et al.*, 2023] Mahbubur Rahman, Md Saidur Rahman Kohinoor, and Aftar Ahmad Sami. Enhancing poultry farm productivity using iot-based smart farming automation system. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6, 2023.

[Ramos *et al.*, 2017] P.J. Ramos, F.A. Prieto, E.C. Montoya, and C.E. Oliveros. Automatic fruit count on coffee branches using computer vision. *Computers and Electronics in Agriculture*, 137:9–22, 2017.

[Song and He, 2005] Haiyan Song and Yong He. Crop nutrition diagnosis expert system based on artificial neural networks. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 1, pages 357–362 vol.1, 2005.

[Sushanth and Sujatha, 2018] G Sushanth and S Sujatha. Iot based smart agriculture system. In *2018 international conference on wireless communications, signal processing and networking (WiSPNET)*, pages 1–4. IEEE, 2018.

[Tian *et al.*, 2020] Hongkun Tian, Tianhai Wang, Yadong Liu, Xi Qiao, and Yanzhou Li. Computer vision technology in agricultural automation—a review. *Information Processing in Agriculture*, 7(1):1–19, 2020.

[Walter *et al.*, 2017] Achim Walter, Robert Finger, Robert Huber, and Nina Buchmann. Smart farming is key to developing sustainable agriculture. *Proceedings of the National Academy of Sciences*, 114(24):6148–6150, 2017.

[Wolfert and Isakhanyan, 2022] Sjaak Wolfert and Gohar Isakhanyan. Sustainable agriculture by the internet of things – a practitioner's approach to monitor sustainability progress. *Computers and Electronics in Agriculture*, 200:107226, 2022.

[Wolfert *et al.*, 2017] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jeroen Bogaardt. Big data in smart farming – a review. *Agricultural Systems*, 153:69–80, 2017.