**Department of Electrical, Computer, and Software Engineering**

**Part IV Research Project**

Literature Review and Statement of Research Intent

Project Number: 61

Plant monitoring system to estimate harvesting timing.

Author: KimZuo

Project Partner]: Do Jun Kwon

Project Supervisor: Ho Seok Ahn

Date: 23/04/2024

**Declaration of Originality**

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

形状, 箭头

描述已自动生成

Name: Kim Zuo

**ABSTRACT:** The literature review aims to analyse the latest literature related to smart farms and the implementation of artificial intelligence in this context, with a specific focus on database structures for storing data in smart farms, as well as the deployment and functionalities of artificial intelligence and Internet of Things in the agricultural sector. By studying the methods adopted by other researchers, the objective is to identify the most effective strategies currently feasible for addressing our project's objectives on the way we stored the data for the whole system and method connecting IoT with artificial intelligence algorithms .

**Keywords: Agriculture, Tomato, Internet of Things, AI, Server, Smart Farming**

# Introduction

In 2022, ChatGPT emerged, showcasing the potential of AI technology, and echoing the advancements in artificial intelligence to a global audience. As time progresses, AI technology continues to evolve, with developments such as AI voice alteration, AI drawing, and AI song creation becoming more prevalent. Agriculture has consistently been regarded as a top priority for every nation due to its multifaceted impact on the economy, society, and the environment. By integrating technologies such as AI and Internet of Things (IoT) into agriculture, we anticipate a substantial increase in food productivity in the future, which serves as the general direction of our project's research.

In our project, we aim to leverage the resources accumulated by Uni in recent years, including various AI deep neural network (DNN) models and datasets, as well as a web page frontend developed from resources obtained last year for the smart farm system. Our goal is to integrate these resources into a functional system. To accomplish this, we will implement IoT sensors using the ESP32 IoT Control Smart Farm Starter Kit provided by Keyestudio. These sensors will simulate data input to the database, and we will utilize HTTP protocol requests to access the physical database created by program language and store this data for the entire system.

This literature review will examine how other researchers have approached the management of smart farm system structures and the integration of AI and IoT technologies. Specifically, we will analyze the latest literature related to smart farms and the implementation of artificial intelligence in this context, focusing on database structures for storing data in smart farms and the deployment and functionalities of artificial intelligence and IoT in the agricultural sector. By studying the methods adopted by other researchers, our objective is to identify the most effective strategies currently feasible for addressing our project's objectives regarding data storage for the entire system and the method of connecting IoT with artificial intelligence algorithms.

# Literature Review

What is a smart farming system specifically? A smart farming system harnesses advanced technologies like IoT sensors, data analytics, AI, and automation to revolutionize agricultural practices. By deploying sensors to monitor environmental conditions and collecting real-time data, farmers gain insights into crop health and resource usage. Precision agriculture techniques enable targeted interventions like precise irrigation and fertilization, maximizing yields while minimizing waste. Automation, including drones and robotic systems, streamlines tasks such as planting and harvesting, reducing reliance on manual labor. Remote monitoring and control allow farmers to oversee operations from anywhere, ensuring timely responses to challenges. Integrated data and decision support systems provide actionable insights, empowering farmers to make informed decisions for optimized farm management. Overall, smart farming systems optimize productivity, conserve resources, and promote sustainable agriculture for a more efficient and resilient food production system.

Rahman, Kohinoor, and Sami (2023) have developed a comprehensive smart farming automation system with a focus on enhancing productivity in poultry farming. Their system integrates various sensors such as the DHT1.1 temperature and humidity sensor, rain sensor, IR sensor, and water level sensor, calibrated to monitor critical factors like temperature and humidity within poultry farms, as well as to detect rainfall. Equipped with features like automatic food and water supply mechanisms, and automated curtain controls during rainfall, the system aims to reduce labour costs and improve productivity. Moreover, an alert system notifies personnel of extreme environmental issues or shortages in food and water storage, while a data-driven predictive model suggests future trends. Users can monitor the data related to the farm using a website and mobile application. This innovative approach to smart farming in poultry production underscores the potential for increased efficiency and sustainability in agricultural practices. Although Rahman, Kohinoor, and Sami (2023) concentrate on enhancing productivity in poultry farming rather than tomato farming as in our project, they demonstrate the embryo of a smart farm, showcasing its ability to reduce manual intervention and excel in big data management and monitoring design.

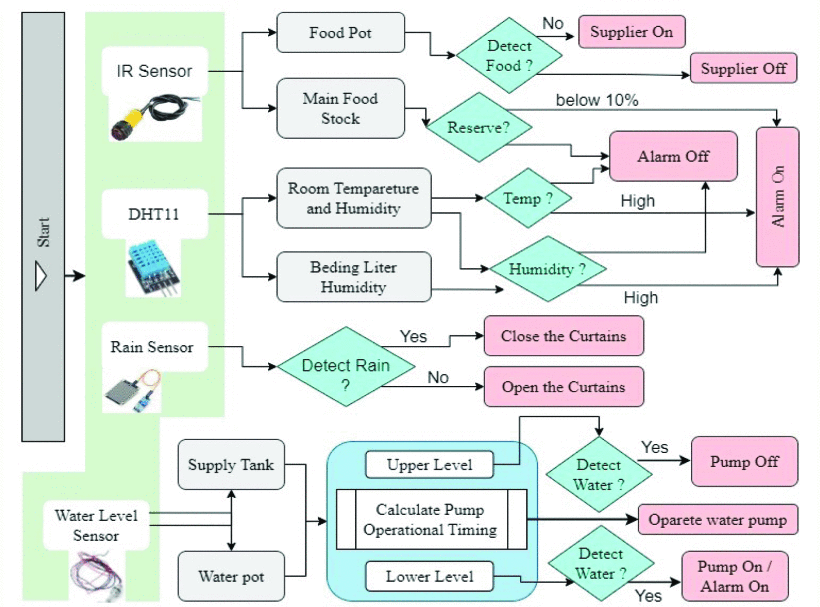


Figure 1 System Overview of IoT-based Automated Farm by Rahman, Kohinoor, and Sami (2023)

图示

描述已自动生成

Figure 2 Data-Driven Web Server by Rahman, Kohinoor, and Sami (2023)

Figure 1 provides a detailed overview of how the features of the smart farming system are realised through the utilisation of various sensors. It highlights the integration of sensor data with different functions within the system, showcasing the comprehensive approach taken by Rahman, Kohinoor, and Sami (2023) in designing their smart farming automation solution. This figure serves as a valuable reference for understanding the inner workings of the system and the interactions between its components.

Figure 2, on the other hand, simplifies the process involved in connecting an IoT sensor to a server and presenting the collected information on an application. The diagram clearly and concisely represents the data flow from the sensor to the server and finally to the application interface. It provides us with a communication method and database storage method that can be used in smart agricultural systems.

Together, the figures utilized in the architecture of the smart agricultural automation system developed by Rahman, Kohinoor, and Sami (2023) offer valuable insights into smart agricultural systems, providing inspiration for server and database creation. However, our project requires a deeper understanding of how AI can play a role in smart farming, particularly in analyzing data and identifying the unique characteristics of specific plants, such as tomatoes in our case.

Demilie, W. B. (2024) provides a detailed overview of recent advancements from 2010-2024 in plant disease detection and classification using machine learning (ML) and deep learning (DL) techniques. By conducting a comprehensive review of recent research publications, the paper compares the performance and efficiency of various technologies in identifying plant diseases. The research focuses on the accuracy, speed, and potential application of different techniques in practical agricultural production. In this report provide a detailed computer vision-based techniques for plant disease detection and classification in Figure 3.

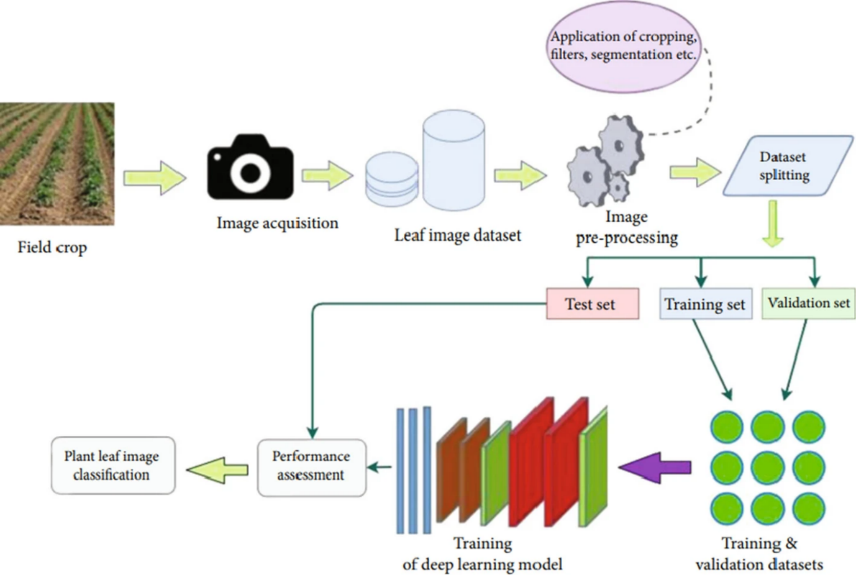


Figure 3 Computer vision-based techniques for plant disease detection and classification by Demilie, W. B. (2024)

This graph illustrates the process by which machine learning (ML) and deep learning (DL) techniques classify features contained within image datasets, highlighting the necessity for our database to store images and enable the transmission of specific images to an AI server or a local application within the server. In Demilie, W. B. (2024), various recommendations regarding DL and ML techniques are presented. The conclusion emphasizes that Convolutional Neural Networks (CNNs) are the optimal choice for classifying plant diseases due to their flexibility and feature extraction properties. I concur with this assertion because CNNs utilize convolutional layers and pooling layers to extract local features from images, reducing parameter count and computational complexity while improving model generalization. Additionally, CNNs are capable of learning hierarchical representations of data, enabling them to capture intricate patterns and structures within images effectively. This makes CNNs particularly well-suited for tasks requiring image analysis and classification, such as plant disease identification or tomato ripeness.

In the reports by Pangilinan, J. R., Legaspi, J., & Linsangan, N. (2022) and Nourbakhsh, A., Mehrabani, E. B., Faraji, S., Shirazi, F. A., & Hatefi, M. (2023), they detail the process of collecting real-world tomato image data and the resulting analysis. Pangilinan, J. R., Legaspi, J., & Linsangan, N. (2022) focus more on the recognition of individual tomatoes, assessing the accuracy of different CNN models to determine the most suitable model for predicting tomato ripeness. In contrast, Nourbakhsh, A., Mehrabani, E. B., Faraji, S., Shirazi, F. A., & Hatefi, M. (2023) simulate scenarios with multiple tomatoes present in an image, identifying the positions and ripeness of each tomato. Although our current primary goal is to integrate the entire system rather than create new models for our system, I believe these two articles can lay the groundwork for us if we need to analyze tomato ripeness and develop AI models in the future.

After analysing the concepts of smart farming and the potential applications of techniques such as IoT and AI within this domain, I want to introduce the work of Chukkapalli, S. S. L., et al. (2020), which provides a detailed analysis of the smart farming ecosystem and elucidates the key benefits of employing ontologies and AI systems within it. Their research not only highlights the significance of these technologies in enhancing agricultural management but also presents a novel possibility for the future of agricultural management.

In their introduction, they outline the pressing global challenge of food demand projected to surpass production by 2050, attributing this trend to agricultural challenges such as soil degradation and inadequate investment. They underscore the importance of adopting modern strategies like precision agriculture and integrating IoT technologies as crucial steps toward enhancing productivity. They develop two ontologies to support AI applications in a cooperative environment. The first ontology, named the member farm ontology, describes interactions that occur within the farm, while the second ontology, the cooperative agriculture ontology, details interactions within the cooperative ecosystem, such as how cooperative resources are shared between individual member farms. These ontologies utilize elements like Farm-Based Units (FBU), On-Board Units (OBU), Worker-Based Units (WBU), and Home-Based Units (HBU) to describe the units existing on the farm, including sensors, agricultural drones, harvesters, reapers, workers, and hubs, as well as Cooperative-Based Units (CBU), Cooperative Worker-Based Units (CWBU), Cooperative On-Board Units (COBU), and Cooperative Farm-Based Units (CFBU) to denote units within the cooperative. Their report provides detailed descriptions of when these elements exist on the farm, under what conditions the cooperative exists, and how they access and share data collected on the farm. This offers a macroscopic understanding of smart farming and suggests a new approach. If our system is to connect with various products from partners, once the basic server architecture is established, we will need to assign different security levels to these connections to strengthen the security of our system. The figure below, extracted from Chukkapalli, S. S. L., et al. (2020), illustrates how they view different farms as members of the system and the ways in which connections are established between them.

图形用户界面

描述已自动生成

Figure 4 Interactions in cooperative farming between member farms and the central co-op hub by Chukkapalli, S. S. L., et al. (2020)

# Project Scope

In our project, we aim to leverage the resources accumulated by Uni in recent years, including various AI deep neural network (DNN) models and datasets, as well as a web page frontend developed from resources obtained last year for the smart farm system. Our goal is to integrate these resources into a functional system. To accomplish this, we will implement IoT sensors using the ESP32 IoT Control Smart Farm Starter Kit provided by Keystudio. These sensors will simulate data input to the database, and we will utilise HTTP protocol requests to access the physical database created by program language and store this data for the entire system.

In the first semester, our plan is to divide the system into two main focus areas. On one side, we will concentrate on the IoT sensors part, utilising a smart farm kit provided by the supervisor. On the other side, we will focus on setting up the physical server with a database. Below is the project diagram we have created for this project:

日程表

描述已自动生成

Figure 5 Conceptual Project Design

In the Figure above we can clearly figure out which part is already done by previous years students (in Green) and which part is things we need to develop from zero (in Yellow). I will be working on creating a physical server with a database whereas my partner will be working on IOT sensors. While literature review, we already build up the ESP32 IoT Control Smart Farm Starter Kit during the semester break, I also using Flask, a lightweight and flexible web framework written in Python build up a workable server prototype.

# Research Objectives



Table above is the research objectives that we discussed for the project, in appendix there is a Gantt chart that include what we are done till the day we upload this literature review,

# Conclusions

Through this literature review, we have gained insights into the fundamental concepts of smart farming, including the integration of IoT sensors and AI support. We have explored the interconnectedness between smart farm IoT sensors and servers, as well as how AI datasets are leveraged by these servers. Additionally, by examining insights from other studies regarding smart ontologies, we have obtained a forward-looking perspective on the potential evolution of our own smart farming system.

# Acknowledgements

Thank the supervisor Gerard Hanson and the sponsoring company ‘Intelligent Automation’ for providing the support and guidelines to make this project successful.

Thank my partner Do Jun Kwon for the support of the project these few weeks, every weeks meeting section help me to keep on the things we done and understand what we will do further for that time.

Thank my supervisor, Ho Seok Ann, for his guidance and mentorship during our project meetings. His wealth of knowledge and insights have been invaluable in deepening my understanding of the project. The systematic approach of documenting our progress through slides every two weeks has proven highly effective, allowing us to review our accomplishments and address any overlooked aspects of our work.

# References

1. Rahman, M., Kohinoor, M. S. R., & Sami, A. A. (2023). Enhancing Poultry Farm Productivity Using IoT-Based Smart Farming Automation System. In Proceedings of the 26th International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). Cox's Bazar, Bangladesh. IEEE. <https://doi.org/10.1109/ICCIT60459.2023.10441525>
2. Demilie, W. B. (2024). Plant disease detection and classification techniques: A comparative study of the performances. Journal of Big Data, 11(5). <https://doi.org/10.1186/s40537-023-00863-9>
3. Pangilinan, J. R., Legaspi, J., & Linsangan, N. (2022). InceptionV3, ResNet50, and VGG19 performance comparison on tomato ripeness classification. In Proceedings of the 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) (pp. 619-624). Yogyakarta, Indonesia. IEEE. <https://doi.org/10.1109/ISRITI56927.2022.10052920>
4. Nourbakhsh, A., Mehrabani, E. B., Faraji, S., Shirazi, F. A., & Hatefi, M. (2023). Tomato ripeness evaluation and localization using Mask R-CNN and DBSCAN clustering. In Proceedings of the 11th RSI International Conference on Robotics and Mechatronics (ICRoM) (pp. 714-719). Tehran, Iran, Islamic Republic of. IEEE. <https://doi.org/10.1109/ICRoM60803.2023.10412430>
5. Chukkapalli, S. S. L., et al. (2020). Ontologies and artificial intelligence systems for the cooperative smart farming ecosystem. IEEE Access, 8, 164045-164064. <https://doi.org/10.1109/ACCESS.2020.3022763>

# Appendix A

图表

描述已自动生成

Figure 6 Gantt Chart Till 04/25/2024.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Variable Name | Variable Type |  |
| **data** |  |  |  |  | **data** | **List [dict {}, {}]** | **use to store all the information for this database** |
|  | id |  |  |  | id | int | use to store farm id |
|  | farmName |  |  |  | farmName | str | use to store farm name |
|  | **buildings** |  |  |  | **buildings** | **List [dict {}, {}]** | **store all infomation relate to a building in farm** |
|  |  | id |  |  | id | int | use to store buidling id |
|  |  | buildingName |  |  | buildingName | str | use to store buildings name |
|  |  | **events** |  |  | **events** | **List [dict {}, {}]** | **use to store event happend in building** |
|  |  |  | date |  | date | str | date of this event happend |
|  |  |  | text |  | text | str | what event happend |
|  |  | **environment** |  |  | **environment** | **List [dict {}, {}]** | **use to store the envrionement variable in different date** |
|  |  |  | date |  | date | str | date of these environment data record |
|  |  |  | temperature |  | temperature | int | temperature for that date |
|  |  |  | humidity |  | Humidity | int | fluorescents for that date |
|  |  |  | light |  | Light | int | Light for that date |
|  |  |  | WaterLevel |  | Water Level | int | Water level for that date |
|  |  |  | SoilHunidity |  | Soil Humidity | int | Soil humidity for that date |
|  |  |  | Steam |  | Steam | int | steam data for that date |
|  |  | **data** |  |  | **data** | **List [dict {}, {}]** | **use to store the data for all plants in the buidling** |
|  |  |  | date |  | date | str | date of these data record |
|  |  |  | area |  | area | int | area of plants take in total |
|  |  |  | fruitlets |  | fruitlets | int | fruitlets all plants in total |
|  |  |  | height |  | height | int | height for the highest plant |
|  |  |  | leaves |  | leaves | int | leaves in total |
|  |  |  | volume |  | volume | int | volument in total |
|  |  |  | width |  | width | int | width in total |
|  |  | **plots** |  |  | **plots** | **List [dict {}, {}]** | **use to store the data for each plant specific** |
|  |  |  | id |  | id | int | plant id |
|  |  |  | plotName |  | plotName | str | plant name |
|  |  |  | **data** |  | **data** | **List [dict {}, {}]** | **use to store the data for each plant in different date** |
|  |  |  |  | date | date | str | date of these data record |
|  |  |  |  | area | area | int | area of plants take |
|  |  |  |  | fruitlets | fruitlets | int | fruitlets this plant grow |
|  |  |  |  | height | height | int | height of this plant |
|  |  |  |  | leaves | leaves | int | leaves this plant have |
|  |  |  |  | volume | volume | int | volument take for this plant |
|  |  |  |  | width | width | int | wid take for this plant |

Table For Database Structure Base on IOT sensors

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Request in HTTP | Key Needed |
| **data** |  |  |  |  | **data** |  |  |
|  | id |  |  |  | id |  |  |
|  | farmName |  |  |  | farmName | @app.route('/args/update\_or\_add\_farm\_name', methods=['POST']) | farm\_id, new\_farm\_name |
|  | **buildings** |  |  |  | **buildings** |  |  |
|  |  | id |  |  | id |  |  |
|  |  | buildingName |  |  | buildingName | @app.route('/args/update\_or\_add\_building\_name', methods=['POST']) | farm\_id, building\_id, new\_building\_name |
|  |  | **events** |  |  | **events** |  |  |
|  |  |  | date |  | date |  |  |
|  |  |  | text |  | text | @app.route('/args/update\_or\_add\_event', methods=['POST']) | farm\_id, building\_id, event\_date, new\_text |
|  |  | **environment** |  |  | **environment** |  |  |
|  |  |  | date |  | date |  |  |
|  |  |  | temperature |  | temperature | @app.route('/args/update\_or\_add\_environment\_temperature', methods=['POST']) | farm\_id, building\_id, environment\_date, new\_environment\_temperature |
|  |  |  | fluorescents |  | Fluorescents | @app.route('/args/update\_or\_add\_environment\_fluorescent', methods=['POST']) | farm\_id, building\_id, environment\_date, new\_environment\_fluorescent |
|  |  |  | co2Concentration |  | Co2Concentration | @app.route('/args/update\_or\_add\_environment\_Co2Concentration', methods=['POST']) | farm\_id, building\_id, environment\_date, new\_environment\_Co2Concentration |
|  |  |  | irrigation |  | Irrigation | @app.route('/args/update\_or\_add\_environment\_irrigation', methods=['POST']) | farm\_id, building\_id, environment\_date, new\_environment\_irrigation |
|  |  | **data** |  |  | **data** |  |  |
|  |  |  | date |  | date |  |  |
|  |  |  | area |  | area | @app.route('/args/update\_or\_add\_data\_area', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_area |
|  |  |  | fruitlets |  | fruitlets | @app.route('/args/update\_or\_add\_data\_fruitlets', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_fruitlets |
|  |  |  | height |  | height | @app.route('/args/update\_or\_add\_data\_height', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_height |
|  |  |  | leaves |  | leaves | @app.route('/args/update\_or\_add\_data\_leaves', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_leaves |
|  |  |  | volume |  | volume | @app.route('/args/update\_or\_add\_data\_volume', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_volume |
|  |  |  | width |  | width | @app.route('/args/update\_or\_add\_data\_width', methods=['POST']) | farm\_id, building\_id, data\_date, new\_data\_width |
|  |  | **plots** |  |  | **plots** |  |  |
|  |  |  | id |  | id |  |  |
|  |  |  | plotName |  | plotName | @app.route('/args/update\_or\_add\_plot\_name', methods=['POST']) | farm\_id, building\_id, plot\_id, new\_plot\_name |
|  |  |  | **data** |  | **data** |  |  |
|  |  |  |  | date | date |  |  |
|  |  |  |  | area | area | @app.route('/args/update\_or\_add\_plot\_area', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_area |
|  |  |  |  | fruitlets | fruitlets | @app.route('/args/update\_or\_add\_plot\_fruitlets', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_fruitlets |
|  |  |  |  | height | height | @app.route('/args/update\_or\_add\_plot\_height', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_height |
|  |  |  |  | leaves | leaves | @app.route('/args/update\_or\_add\_plot\_leaves', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_leaves |
|  |  |  |  | volume | volume | @app.route('/args/update\_or\_add\_plot\_volume', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_volume |
|  |  |  |  | width | width | @app.route('/args/update\_or\_add\_plot\_width', methods=['POST']) | farm\_id, building\_id, plot\_id, plot\_date, new\_plot\_width |

Table For Database Structure Base on Fake Database used by frontend and method to post request to prototype server