DYNAMIC CROP MODELING FOR SALAD CUCUMBERS USING BIOMASS AND IRRIGATION WEIGHTS AND VISUALS

Kavindu G. Edirisinghe

IT21267222

Sajini P. Wijesinghe

IT20418274

Dinandi K. Somarathne

IT21327094

Binuri N. Thilakrathne

IT21225956

B.Sc. (Honors) degree in Information Technology

Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DYNAMIC CROP MODELING FOR SALAD CUCUMBERS USING BIOMASS AND IRRIGATION WEIGHTS AND VISUALS

Kavindu G. Edirisinghe

IT21267222

Sajini P. Wijesinghe

IT20418274

Dinandi K. Somarathne

IT21327094

Binuri N. Thilakrathne

IT21225956

B.Sc. (Honors) degree in Information Technology

Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

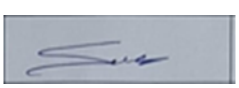
DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Student Name	Student ID	Signature
Edirisinghe E.A.K.G.	IT21267222	
Wijesinghe.W.E.S.P	IT20418274	
Somarathne D.K.	IT21327094	
Thilakarathne M.B.N.	IT21225956	

As the supervisor/s of the above-mentioned candidates, I hereby certify that they are conducting research for their undergraduate dissertation under my guidance and direction.

Signature of the supervisor: Date: 08/04/2025



(Ms. Uthpala Samarakoon)

ABSTRACT

Modern agriculture increasingly depends on intelligent systems to enhance productivity, reduce resource consumption, and support sustainable practices. However, traditional crop modeling methods often rely on costly and maintenance-intensive sensors, limiting their practicality in diverse farming settings. This research proposes a cost-effective and scalable solution for dynamic crop modeling in salad cucumber cultivation by integrating plant biomass weights, environmental data, and real-time image analysis. The system focuses on four core components: yield prediction, irrigation optimization, plant growth monitoring, and fruit maturity assessment. Using low-cost sensors, the system continuously collects temperature, humidity, irrigation events, and plant weight data. Simultaneously, high-resolution images of plants and fruits are captured using embedded cameras. The data is preprocessed and analyzed using machine learning models, including convolutional neural networks (CNNs) and regression algorithms. These models are trained to classify growth stages, predict harvest timing, detect irrigation needs, and estimate yield based on historical and real-time observations. The integration of multimodal data enables comprehensive monitoring of crop conditions, allowing for early detection of anomalies and timely decision-making. A cloud-connected dashboard built using Grafana visualizes key metrics and predictions, while automated alerts guide efficient field interventions. By reducing reliance on expensive equipment and manual labor, the proposed system offers a practical approach to smart farming, especially in controlled environments such as greenhouses. The novelty of this research lies in its unified framework that combines visual analysis, environmental sensing, and machine learning for holistic cucumber crop management.

Keywords: Deep Learning, Agriculture, Greenhouse Environment, Harvest, Image Processing, Machine Learning.

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to our supervisor, Ms. Uthpala Samarakoon, from the Sri Lanka Institute of Information Technology (SLIIT), for her continuous guidance, encouragement, and support throughout the course of this research. Her valuable insights, expertise, and dedication have been instrumental in shaping our study and helping us overcome challenges along the way.We are equally thankful to our co-supervisor, Ms. Aruni Premarathne, also from SLIIT, for her consistent support, thoughtful feedback, and kind encouragement at every stage of this research. Her contributions greatly enriched the depth and clarity of our work.

We extend our sincere appreciation to our external supervisor, Dr. Lasantha Addikaram, a distinguished lecturer at the Faculty of Agriculture, University of Ruhuna. His expert guidance, technical knowledge, and generous assistance especially in granting access to the greenhouse, sensor equipment, and other essential resources were invaluable in the practical implementation of our research.

Our special thanks go to the Sri Lanka Institute of Information Technology (SLIIT) for providing the research infrastructure, facilities, and academic environment necessary for the successful execution of this study. We also express our sincere gratitude to the University of Ruhuna for their generous support in facilitating key aspects of our experimental work and contributing to the smooth progress of our project. Finally, we acknowledge the wider academic and research community, whose prior work laid the foundation for our study. To all faculty members, peers, and technical staff who assisted and encouraged us throughout this journey your support has been deeply appreciated.

Table of Contents

LIST OF ABBREVIATION

Abbreviation	Full Form
RL	Reinforcement Learning
ESP32	Espressif Systems Microcontroller with Wi-Fi & Bluetooth
SLIIT	Sri Lanka Institute of Information Technology
LSTM	Long Short-Term Memory (a type of Recurrent Neural Network)
UID	User Identifier
KNN	k-Nearest Neighbors
AI	Artificial Intelligence
API	Application Programming Interface
IoT	Internet of Things
ML	Machine Learning
UI	User Interface
SDK	Software Development Kit
DHT22	Digital Humidity and Temperature Sensor
SVM	Support Vector Machine
F1-Score	Harmonic means precision and recall
OCI	Oracle Cloud Infrastructure

INTODUCTION

1.1 Background & Literature Survey

Agricultural practices around the world are undergoing a technological transformation, with data-driven systems increasingly being employed to optimize yield, conserve resources, and ensure food security. Despite these advancements, many traditional crop monitoring techniques remain reliant on expensive infrastructure, including contact sensors and manual data collection processes. These limitations create challenges for widespread adoption, particularly among small- to medium-scale farmers who cannot afford high-end precision agriculture tools. In response to these issues, the research project titled Dynamic Crop Modeling for Salad Cucumbers Using Biomass, Irrigation Weights, and Visuals introduces a cost-effective, integrated, and intelligent system that leverages machine learning, IoT-based sensing, and image processing technologies to enhance four core areas: yield prediction, irrigation optimization, plant monitoring, and fruit growth analysis.

Yield Prediction is a critical element in crop management, allowing farmers to anticipate production volumes and make informed decisions regarding market supply, labor, and resource allocation. Conventional yield estimation methods either relieve manual observation or sophisticated simulation models that require intensive data inputs from expensive sensors. This project proposes an alternative approach by using a weight-based crop modeling technique. Fresh biomass weight is continuously collected and analyzed alongside environmental data specifically temperature and humidity. These datasets are fed into machine learning models to generate real-time predictions of expected yield. By correlating growth patterns with environmental stimuli, the system can identify growth trends, detect anomalies early, and adjust predictions dynamically. This methodology not only reduces the need for costly equipment but also enhances the reliability and scalability of yield estimation across varied cultivation environments.

Irrigation Management is another pillar of the proposed system, addressing one of the most resource-intensive aspects of agriculture: water use. Traditional irrigation systems either operate on fixed schedules or require manual assessments of soil moisture and plant health. However, such approaches often lead to overwatering or underwatering, which negatively affects plant development and productivity. The intelligent irrigation component in this system uses plant weight fluctuations, environmental sensor data, and historical irrigation records to determine the optimal timing and volume of water application. By integrating this data into a machine learning model, the system dynamically adjusts irrigation schedules in real time. It considers factors such as time since the last irrigation, rate of biomass increases, and current temperature/humidity levels to guide efficient water use. This on-demand irrigation strategy contributes to sustainable agriculture by conserving water, preventing crop stress, and maintaining consistent plant growth.

In addition to managing resources and predicting output, effective Plant Monitoring is essential for early detection of stress, disease, and developmental anomalies. Visual observations of plant health are often overlooked in automated systems, yet they provide invaluable insights. This research integrates real-time image capturing using embedded cameras (such as ESP32-CAM) to monitor the growth stages of cucumber plants. The system captures high-resolution images at regular intervals and uses convolutional neural networks (CNNs) to classify each plant's stage—ranging from bud stage to full maturity. These classifications are cross-referenced with environmental and biomass data to track developmental progress and assess growth consistency. The continuous monitoring capability of the system reduces the need for manual inspections and provides a non-invasive method of evaluating plant health, ultimately contributing to more consistent crop outcomes and efficient labor management.

Fruit Monitoring and Harvest Timing, builds upon the plant monitoring system by focusing specifically on the fruits' physical characteristics and readiness for harvest. In cucumber cultivation, the timing of harvest directly affects fruit quality, shelf life, and market value. Misjudging the optimal harvest window can lead to either underdeveloped or overripe produce. To address this, the system uses real-time fruit imagery and machine learning models trained to identify the growth stage of the cucumber fruit. These models analyze key visual features such as size, color, and texture to determine the current maturity stage. When combined with historical growth data and environmental parameters, the model predicts the most accurate harvest time. Alerts can then be generated via the dashboard or through SMS/email, prompting timely harvesting actions and minimizing losses due to delay or premature picking.

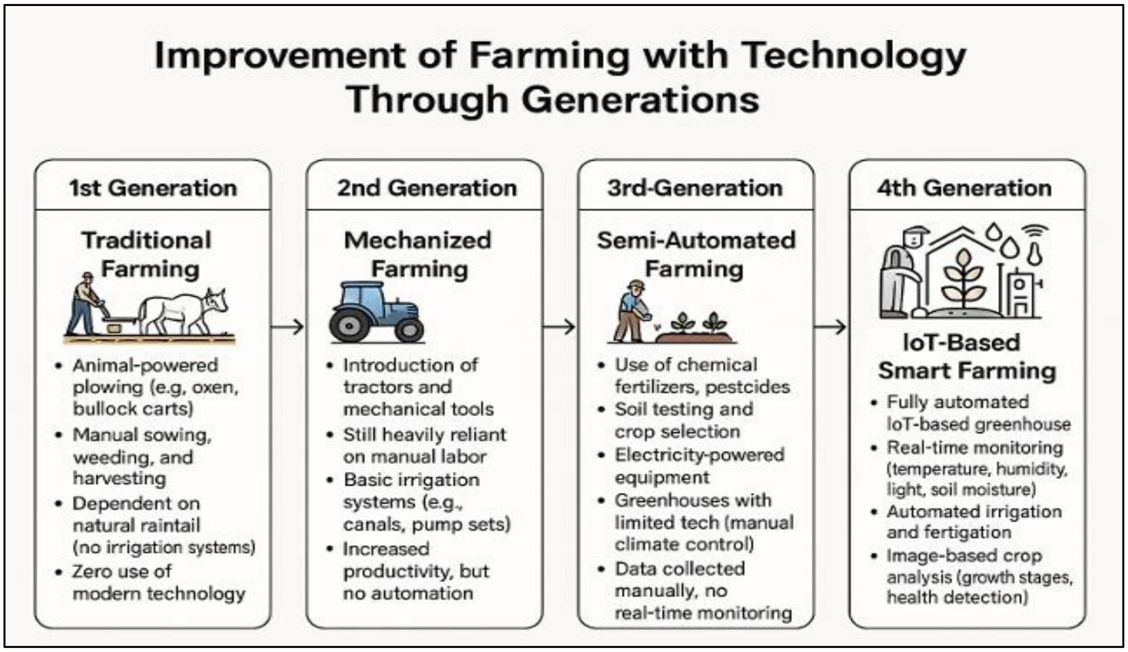


Figure 1 Agriculture Generations

The integration of modern technology into agriculture has revolutionized traditional crop monitoring practices, shifting the paradigm from manual, labor-intensive methods to intelligent, automated systems. Historically, crop growth and yield estimation relied heavily on farmers' observational experience, visual inspection, and seasonal intuition. While these methods have persisted for centuries, they are increasingly seen as inadequate in commercial farming due to their subjectivity, inconsistency, and lack of scalability [3]. Precision agriculture emerged to address these limitations by utilizing real-time data from embedded sensors, enabling site-specific management that can adapt dynamically to a plant's physiological signals and environmental conditions [1]. By incorporating data from parameters such as temperature, humidity, light intensity, and plant weight, precision farming platforms have demonstrated significant improvements in yield forecasting, irrigation planning, and resource optimization. This transformation is critical in light of rising global demands for food security, environmental sustainability, and efficient land utilization [2]. Initial developments in precision agriculture leaned on contact-based sensors like soil probes, dendrometers, and pH meters to capture metrics such as soil moisture, nutrient levels, and plant girth. However, such technologies often suffer from issues like high cost, mechanical degradation, limited lifespan, and complexity in installation and calibration [2], [3]. Image-based solutions, although capable of capturing fine morphological details, introduce challenges related to lighting conditions, occlusion by leaves or branches, the requirement of high-resolution cameras, and intensive processing needs [6]. To overcome these limitations, researchers began exploring alternatives. One such promising avenue involves load-cell-based weight sensing, which provides a continuous and robust estimate of plant biomass. Studies like Wang et al. [25] have demonstrated that plant weight is a strong proxy for water uptake, nutrient absorption, and overall health, which directly relate to yield outcomes. Further enhancements to this approach involve monitoring pressure exerted by fruiting bodies from above the plant, enabling per-plant fruit development tracking without the need for cameras or physical inspection. Simultaneously, the proliferation of Internet of Things (IoT) technologies has allowed for low-cost deployment of wireless sensor networks in agricultural settings. According to Pham et al. [13]

The integration of image processing and artificial intelligence (AI) techniques into precision agriculture has emerged as a powerful approach to enhancing crop monitoring and yield optimization. Recent advances in computer vision have enabled researchers to develop automated systems that support critical agricultural tasks such as fruit detection, disease identification, and growth stage classification. Liu et al. proposed an instance segmentation method tailored for cucumber fruit detection in greenhouse environments [15]. Their system demonstrated reliable accuracy even under occlusion and overlapping conditions, where traditional object detection methods often fail. The method employed a deep learning-based segmentation model capable of differentiating fruits from leaves and background elements. Despite the robustness of the proposed model in fruit detection, the study did not extend its application to broader aspects of plant health such as leaf quality or biomass estimation, which are essential for comprehensive crop monitoring [16]. To address crop growth evaluation, Lee et al. developed a hybrid crop growth analysis platform that integrates image data with time-series environmental parameters [17]. Their work demonstrated the potential of combining visual monitoring with sensor-based input to enhance the prediction of plant development trends and to identify early-stage anomalies. However, the model lacked adaptability in dynamic agricultural environments, as it relied on pre-defined data patterns rather than real-time sensory updates, limiting its ability to respond to abrupt environmental fluctuations [18]. Further exploring low-cost and accessible monitoring techniques, Chiu et al. investigated the feasibility of utilizing smartphone imagery to classify crop growth stages [19]. Their findings suggested that mobile-based systems could democratize precision agriculture by eliminating the need for specialized imaging equipment. Nonetheless, the model's performance was constrained by issues related to inconsistent lighting, varying camera angles, and reduced image resolution, which significantly impacted the system's ability to extract detailed features in practical large-scale operations [20]. These limitations underscore the importance of implementing more adaptable and high-fidelity image acquisition methods. Environmental factors such as temperature, humidity, and soil moisture are fundamental in regulating crop development. Sensor-based monitoring systems have been widely studied for their role in maintaining optimal growing conditions [21]. However, many existing approaches treat sensor and image data in isolation. While climate sensors offer critical insights into external conditions, they lack the capacity to visually detect symptoms such as leaf discoloration or pestinduced damage [19][20]. Thus, a unified monitoring framework that can simultaneously process environmental and visual cues is necessary to improve accuracy in crop health assessment.

The adoption of advanced technologies in agricultural monitoring has become increasingly vital for enhancing the efficiency, yield, and sustainability of high-value crops. With the growing global demand for fresh, high-quality produce, the agricultural sector has transitioned from traditional manual inspection methods to automated systems that incorporate image processing, machine learning (ML), and real-time data analysis. These smart farming technologies are designed to improve crop monitoring, defect detection, and optimal harvest predictions, ensuring that crops meet market standards for quality and freshness [2]. High-value vegetable crops require consistent monitoring and rigorous quality assessment before produce can enter the market. Ensuring uniformity in size, proper maturity, and the absence of defects is critical for both economic profitability and consumer satisfaction. Recent studies have explored the use of advanced deep learning models, such as MobileNetV2, for detecting defects, assessing maturity, and enhancing grading systems for cucumbers and other greenhouse-grown crops [5].

This literature review examines the current state of research in cucumber analysis, with a focus on real-time monitoring, defect detection, quality assessment, and harvest prediction using image processing and machine learning models. Real-time monitoring systems have emerged as a cornerstone of modern precision agriculture, particularly in controlled environments where environmental factors such as temperature, humidity, and light levels can be precisely controlled to optimize plant growth. Research has demonstrated the effectiveness of integrating sensor networks and image capture technologies to continuously monitor crop growth stages [9]. These systems enable the immediate detection of anomalies, allowing farmers to intervene promptly to maintain crop health and maximize yield. For example, a study by Jadhav et al. [6] developed a real time image processing system for monitoring tomato growth stages in a greenhouse setting.

The system successfully detected changes in fruit color and size, providing farmers with automated insights into the optimal harvesting time. Similar methodologies have been applied to cucumber cultivation, where continuous image-based monitoring helps identify growth abnormalities caused by environmental fluctuations. The integration of Internet of Things (IoT)-enabled systems with computer vision technologies has further enhanced the accuracy of real-time monitoring. By integrating high-resolution cameras with AI-driven analytics, researchers have developed early-warning systems capable of preventing diseases, predicting yield, and optimizing irrigation schedules for greenhouse-grown cucumbers [10]. Despite these advancements, most existing studies lack a fully integrated system that combines real-time monitoring with predictive analytics, highlighting a critical gap in the current research landscape. One of the key considerations in agriculture is the automated detection of fruit defects and fruit quality. Ensuring the quality of cucumbers before they reach the market is critical for economic viability and consumer satisfaction. Traditional methods rely on manual visual inspections, which are often inconsistent and prone to human error. Advanced AI models, particularly Convolutional Neural Networks (CNNs), VGG16, and ResNet50, have demonstrated significant accuracy in automating this process. [14] For instance, a study by Quang Uoc et al. [8] applied CNNs to classify cucumber fruit defects, achieving high precision in distinguishing between healthy and defective cucumbers. These models analyzed image datasets of cucumbers at various growth stages, identifying defects such as discoloration, deformation, and disease symptoms. Additionally, Support Vector Machines (SVM) and k-Nearest Neighbors (KNN) algorithms have been explored for similar applications, with varying degrees of success depending on dataset quality and environmental factors [11]. Instance segmentation, a subfield of machine learning, has proven particularly valuable for distinguishing individual cucumbers in densely packed greenhouse environments. Traditional image processing techniques often struggle with overlapping fruits and background noise, whereas deep learning models like Mask R-CNN offer precise fruit segmentation and growth stage classification [15]. For example, Jia et al. [12] developed an accurate segmentation model using Mask R-CNN for green fruit, which successfully identified different ripening stages and provided farmers with detailed harvest insights. [16] Moreover, MobileNetV2-based quality assessment has been applied to multiple vegetable crops, including ladies' fingers, bitter gourds, and cucumbers.

Valiente et al. [5] utilized MobileNetV2 to classify defects, size, and maturity levels in vegetables, demonstrating the feasibility of using lightweight deep learning models for real-time agricultural applications. The ability of MobileNetV2 to operate efficiently in resource-constrained environments makes it highly suitable for small-scale farmers and mobile-based applications. Accurately predicting the optimal harvest time is essential for maximizing yield, minimizing waste, and ensuring peak crop quality. Traditionally, farmers estimate harvest time based on visual cues, which are prone to inconsistencies. Machine learning models have been employed to enhance prediction accuracy by analyzing historical growth patterns and real-time environmental data. For instance, Escamilla et al. [17] developed a deep learning model for maturity recognition and fruit counting in greenhouse-grown sweet peppers, using temperature and light exposure data to determine the optimal harvest window. Similarly, Lin and Hill [4] applied neural network modeling to predict harvest dates for greenhouse-grown peppers, demonstrating that AI-based predictions outperform traditional estimation methods. In cucumber farming, time-series forecasting models such as Long Short-Term Memory (LSTM) networks and Gradient Boosting Machines (GBM) have been utilized to analyze growth trends and environmental influences. These models process large datasets collected from sensor-equipped greenhouses, allowing farmers to optimize irrigation schedules and anticipate yield fluctuations [18]. However, despite these advancements, few studies have combined real time monitoring with predictive analytics in a fully automated system. The integration of real-time monitoring systems, machine learning models, and predictive analytics has significantly advanced precision agriculture. However, critical research gaps remain, including the lack of fully integrated systems, limited consideration of environmental factors, and the absence of real-time visualization tools [19], [20], [21].To address these challenges, this research aims to develop an integrated, real-time cucumber analysis system that combines image segmentation, defect detection, and time-series-based harvest prediction while incorporating environmental monitoring into an interactive dashboard for farmers [13]. This system will bridge existing gaps, enabling farmers to make data-driven decisions and optimize crop management practices effectively.

1.2 Research Gap

Despite notable advancements in agricultural monitoring and automation, existing systems remain fragmented and insufficient for supporting precise, real-time, and crop-specific decision-making in controlled environments particularly for salad cucumber cultivation. Several critical limitations are evident across current studies, which hinder the development of a unified, adaptive, and efficient solution.

First, comprehensive plant growth modeling is lacking. While instance segmentation techniques have shown promise in detecting cucumber fruits under complex conditions, the broader structural dynamics—such as biomass accumulation, leaf health, and morphological changes remain largely unaddressed. This results in an incomplete representation of the cucumber plant's development cycle and limits the ability to identify early-stage abnormalities or variable growth rates.

Second, the absence of real-time monitoring and adaptive feedback mechanisms significantly restricts the operational efficiency of these systems. Many existing approaches rely on batch-mode data processing or static image analysis, failing to respond dynamically to sudden environmental or physiological changes. This diminishes their value in precision agriculture where timely interventions (e.g., irrigation, fertilization, or disease control) are crucial for maintaining crop health and optimizing yield.

Third, predictive modeling for growth stage progression and harvest time estimation remains underutilized. Traditional CNN-based classifiers are effective for static classification tasks but fail to capture temporal changes in plant morphology. Although LSTM networks have been explored in environmental time-series forecasting, they often exclude visual data. The limited application of hybrid CNN-LSTM models—especially in field conditions—further highlights the lack of robust, temporally aware frameworks for predicting cucumber growth and optimal harvest timing.

Fourth, advanced instance segmentation has not been fully exploited for structural plant monitoring. While current research utilizes fruit detection segmentation, crucial plant features such as leaf structure, stem development, and anomaly identification (e.g., wilting, curling, discoloration) are often overlooked. This limits the system's ability to perform detailed morphological assessments necessary for effective crop management.

Additionally, defect detection and quality assessment methodologies remain underdeveloped. While some studies recognize the importance of detecting fruit abnormalities, many do so in limited or non-real-time scopes. The lack of robust, continuous quality control frameworks leads to inconsistencies in defect identification and delayed corrective measures, ultimately affecting the marketability of the produce.

Moreover, crop-specific modeling for cucumbers is inadequately addressed in existing literature. The generalization of models across multiple crop types ignores the unique physiological and environmental demands of cucumbers, particularly those cultivated in greenhouses for salad production. Tailored models are essential for maximizing the quality, consistency, and productivity of this crop.The capability for real-time image capturing and continuous monitoring is not fully realized in current systems. Partial implementations exist, but they often lack seamless integration with defect detection, quality analysis, and predictive analytics modules—limiting their overall effectiveness in decision-making.

1.3 Research Problem

The accurate and timely prediction of crop yield is a critical factor in precision agriculture, enabling farmers to make informed decisions regarding harvesting, labor allocation, and post-harvest logistics. This section presents an IoT-based real-time yield prediction system designed specifically for greenhouse-grown salad cucumbers. The system leverages load cell technology in conjunction with environmental sensing to provide continuous, non-destructive monitoring of individual plant biomass and growth progression.

A load cell is a transducer that converts force into measurable electrical signals. In this system, load cells are installed at the base of each cucumber plant or at the planter level to measure the incremental weight changes corresponding to biomass accumulation. These weight measurements reflect the overall growth and productivity of the plant, capturing data that are crucial for real-time yield estimation. To enhance the accuracy of yield prediction, load cell data are fused with additional environmental parameters, including ambient temperature, relative humidity, soil moisture, and light intensity. These parameters are collected using standard agricultural sensors such as the DHT22 for temperature and humidity, capacitive sensors for soil moisture, and BH1750 for light intensity.

All sensor modules are interfaced with an ESP32 microcontroller, which transmits data over Wi-Fi to a centralized server. The system architecture adopts a modular design, allowing scalability and customization based on the number of monitored plants or greenhouses monitored. Data are stored in a time-series database (InfluxDB) and visualized through Grafana dashboards, providing growers with an intuitive interface for real-time monitoring and historical trend analysis.

To convert the raw sensor readings into yield predictions, the system utilizes regression-based machine learning models trained on historical data. Models such as linear regression, random forest regression, and LSTM (Long Short-Term Memory) networks are evaluated for their ability to learn complex temporal relationships between environmental factors and plant biomass. Preliminary results indicate that integrating load cell data with environmental inputs significantly improves the accuracy of predicted yields, outperforming traditional univariate models that rely solely on biomass measurements.

The real-time capability of this system supports dynamic decision-making in greenhouse management. For example, by identifying plants with suboptimal growth rates, targeted interventions such as localized irrigation or nutrient delivery can be implemented. Furthermore, the yield data can inform harvest scheduling and reduce waste due to over- or underestimation of produce. This IoT-driven approach offers a low-cost, scalable solution for continuous yield monitoring, making it especially valuable for small and medium-sized greenhouse operations. It also lays the groundwork for future integration with irrigation systems and crop forecasting platforms, contributing to a holistic and intelligent farming ecosystem.

1.4 Research Objectives

Main Objectives

The key objective is to develop a cutting-edge, real-time monitoring system for cucumber crop management integrating different sources of data such as climate sensor data, biomass weight data, and image analysis. The system envisions improved precision farming by providing farmers with comprehensive, data-driven information that optimizes irrigation, yield prediction, crop growth analysis, and fruit quality analysis. Through the implementation of machine learning models and sensor technology, the system achieves optimal efficient resource management, minimized losses, and overall productivity. Four components are put together to produce a flexible, integrated outcome. The first component is about developing a dynamic crop yield prediction system that incorporates real-time sensor measurements and advanced machine learning models.

Through integration of temperature, humidity, soil moisture, and light intensity sensor readings, the system will provide accurate yield predictions and provide actionable information to farmers. Significant patterns in sensor readings will be extracted through feature engineering techniques, while the right machine learning models will be used to enhance the accuracy of predictions. Farmers will also be aided in making informed crop management plans through a generative AI-based recommendation engine.

A key aspect of this study is the development of a resource-efficient, sustainable irrigation management system that dynamically adjusts water supply based on real-time biomass weight analysis and environmental conditions. By integrating non-contact weight measurement sensors with climate data, the system will develop predictive models capable of determining optimal irrigation schedules. Machine learning algorithms will be employed to ensure precise water usage, preventing under- or over-irrigation. The system will be tested in real-world agricultural settings to validate its effectiveness and scalability across different crop types, ensuring adaptability to various environmental conditions.

This research also aims to create a sophisticated real-time monitoring system for cucumber crop growth analysis using modern machine learning techniques. The goal is to classify plant growth stages, detect defects, and identify harmful insects to enhance agricultural decision-making. Deep learning models will be trained on pictorial data to estimate growth stages, while image segmentation and object detection techniques will be employed to recognize plant health anomalies. A real-time monitoring dashboard will be developed to display critical information, enabling farmers to take timely corrective actions that improve crop health and maximize yield.

In addition to monitoring crop growth, the research will enhance fruit quality assessment and harvesting decisions through real-time image processing and machine learning techniques. The system will utilize instance segmentation models to monitor cucumber fruit development and identify maturity stages with high accuracy. Machine learning models will be trained to detect defects, assess fruit quality, and provide recommendations for optimal harvesting times. A decision-support system will be implemented to assist growers in making precise, data-driven harvest decisions, reducing waste, and improving market value. By integrating these components into a unified agricultural management system, this research aims to revolutionize cucumber cultivation through smart farming practices. The combination of real-time data acquisition, machine learning analytics, and decision-support tools will empower farmers to make proactive decisions, improve resource efficiency, and enhance overall crop productivity.

Sub Objectives

1. **Yield Prediction**

Applying feature engineering techniques to transform raw sensor signals into meaningful features. These include moving averages of biomass weight, growth rate calculations, light intensity patterns, temperature fluctuations, and fruit pressure peaks detected from the top load cell. For instance, sudden increases in top pressure followed by plateaus may indicate the presence and maturation of fruits [25].

Training a Random Forest model to interpret non-sequential data points, such as average sensor values over fixed windows. Random Forest is known for its robustness against noise, ability to rank feature importance, and strong performance with small to medium-sized tabular datasets [17]. Developing a Long Short-Term Memory (LSTM) network that can learn temporal dependencies in the data. Since plant behavior unfolds over time, an LSTM model is 15 | Page trained on rolling time-series windows to capture long-term growth patterns and forecast yield based on trends rather than isolated values [28]. Conducting model validation using k-fold cross-validation techniques and performance metrics such as RMSE, MAE, and R^2 to ensure that predictions are not only accurate but also generalizable to unseen data. This prevents the model from being overfitted to any one greenhouse batch or crop cycle. Exporting the trained models in serialized formats (.pkl for Random Forest and .h5 for LSTM) to enable integration into a FastAPI backend. This backend serves as the prediction engine, accepting real-time inputs and returning yield forecasts through a RESTful API. Building a Grafana dashboard to visualize real-time plant performance, sensor data, and prediction outcomes for greenhouse managers. For field workers, an Android mobile application is developed to display yield predictions, alert them to underperforming plants, and recommend corrective actions such as adjusting irrigation or fertilizer application. The yield prediction component is designed not only to forecast harvest quantity but also to enable proactive responses from the grower. Instead of reacting to problems after they have occurred, farmers can use the system's output to act in advance and mitigate risks. Together, these objectives create a complete ecosystem for smart farming that transforms physical sensor data into digital intelligence. The system's modularity and cloud-native design allow it to scale efficiently, adapt to other greenhouse crops, and be maintained with minimal technical overhead. It bridges the gap between data collection and real-world decision-making, enabling more sustainable, precise, and efficient cucumber cultivation.

2. **Efficiently Analyze Cucumber Fruit Stages in Greenhouse Using Real-Time Image Capturing and Machine Learning (Instance Segmentation):**

This sub-objective is to monitor the growth of cucumber fruit in a greenhouse setting as it moves through various stages of development. Traditional fruit growth tracking techniques frequently rely on routine manual inspections, which are time-consuming and labor-intensive in addition to running the danger of missing minute changes that can call for intervention.

The proposed approach will use advanced instance segmentation techniques in conjunction with real-time pictorial-capture technology to address this. An advanced machine learning method called instance segmentation makes it possible to accurately identify and categorize individual cucumbers in a complicated environment by setting them apart from other features like leaves or stems. Technology will be able to track the development of the cucumbers in real time and provide comprehensive insights into their progression from seedling to maturity by continuously taking pictures of the cucumbers at different stages of growth. Because it enables the rapid detection of any deviations from anticipated development patterns, this real-time analysis is essential for optimizing growing conditions. For instance, the system may notify the grower to check and modify environmental parameters like temperature, or humidity if it notices that a certain cucumber is not developing as it should. This sub-objective's goal is to establish a more accurate and responsive agricultural environment where decisions are based on exact, real-time data, ultimately improving crop health and productivity.

3. **Utilize Machine Learning Models to Identify Defects and Assess Cucumber Quality, Ensuring Reliable and Consistent Evaluations Through Advanced Techniques:**

Finding abnormalities and evaluating fruit quality is a major difficulty in greenhouse farming, which is covered in the second sub-objective. This procedure is frequently manual in older settings and depends on the subjective assessment of human inspectors. Depending on the inspector's background and the circumstances at the time of the inspection, these techniques can produce varied findings.

The proposed approach will incorporate advanced machine learning models that are specifically trained to recognize flaws in cucumbers and evaluate their overall quality to get beyond these restrictions. These models will have the ability to analyze photos taken in real time and identify defects, discolorations, blemishes, and disease indicators that could compromise the produce's safety or marketability.

The system optimizes this process, ensuring that every cucumber minimizes the unpredictability and potential biases associated with human inspection by ensuring that each cucumber is assessed based on a uniform set of criteria. Furthermore, as the system processes more data over time, machine learning enables it to continually increase its accuracy and dependability. This method not only improves the accuracy of quality evaluations but also makes it possible to identify flaws early on, allowing for prompt interventions that can stop more serious problems and cut down on waste. As a result, a method for upholding strict criteria of cucumber quality in greenhouse settings is more effective, dependable, and scalable.

4. **Develop a Decision-Support System for Identifying Mature Fruits and Aiding Growers in Making Informed Harvest Decisions Using Machine Learning Techniques:**

The third sub-objective centers on the critical task of determining the optimal harvest time for cucumbers. In greenhouse cultivation, the timing of harvest is crucial for maximizing yield and ensuring that the produce meets market standards for size, taste, and texture. However, determining the precise moment when cucumbers are at their peak maturity can be challenging, especially in large-scale operations where multiple factors must be considered.

This sub-objective aims to develop a decision-support system that utilizes machine learning techniques to analyze a variety of data inputs such as growth patterns, environmental conditions, and historical harvest data to accurately predict when cucumbers have reached their ideal maturity. The system will provide growers with actionable insights, helping them to schedule harvests at the optimal time, thus maximizing both the quantity and quality of the yield.

By automating the process of harvest decision-making, the system reduces the reliance on guesswork or subjective judgment, which can lead to either premature or delayed harvests, both of which can negatively impact the crop. Additionally, the system's predictive capabilities will allow growers to plan their labor and resources more efficiently, leading to cost savings and improved operational efficiency. This sub-objective is integral to achieving the overall goal of the research, as it directly impacts the profitability and sustainability of cucumber cultivation in controlled environments.

5. **Irrigation System Component**

The irrigation system component forms the backbone of the proposed smart cultivation platform, enabling automated, data-driven water management based on the real-time needs of cucumber plants. A dual-weight sensor configuration is deployed at the plant base and along the supporting structure, where base load cells measure the total system weight, including the plant, substrate, and retained water, while top-mounted tension sensors capture weight variations from above. These measures, taken at regular intervals, allow the system to detect dehydration patterns and weight loss due to transpiration or insufficient watering. Unlike traditional irrigation methods that rely on time-based schedules or soil moisture readings, this non-contact sensing approach provides a direct and dynamic indicator of plant water status. It reduces the likelihood of overwatering or under-irrigation, which can harm root health or stunt growth.

Complementing the weight-based measurements, environmental conditions such as temperature, humidity, and light intensity are captured using calibrated sensors placed within the greenhouse environment. These factors significantly influence plant water uptake and are crucial for making accurate irrigation decisions. The collected sensor data is fed into a Long Short-Term Memory (LSTM) neural network, which excels at identifying time-based patterns in sequential data. The model is trained to forecast the plant's water needs by analyzing historical hydration trends in conjunction with current climate conditions. Once the need for irrigation is detected, a solenoid valve is triggered to deliver the appropriate amount of water. This closed-loop control system not only automates the irrigation process but also improves water-use efficiency and reduces operational costs. In essence, this component enables responsive, low-maintenance water management, contributing to crop consistency and sustainability.

6. Wilted Leaf Detection Component

The wilted leaf detection component is designed to enhance plant health monitoring by incorporating automated visual analysis through deep learning. Cucumber plants often show early signs of stress through leaf wilt, which, if not detected in time, can lead to reduced yield and long-term plant damage. To address this, ESP32-CAM modules are installed strategically within the greenhouse to capture high-resolution images of the plants at scheduled intervals. These images are continuously uploaded to a server where they are processed using computer vision algorithms. The core of the analysis is powered by a Convolutional Neural Network (CNN) based on the VGG16 architecture, which has been fine-tuned for classifying cucumber leaves as either "wilted" or “healthy.” The model detects even subtle signs of stress, such as drooping, color fading, or leaf edge deformation, far earlier than the human eye might catch during manual inspection

.

The ability to detect wilt early is critical for initiating timely interventions, especially in a high-density greenhouse setup where symptoms can escalate rapidly if overlooked. Once the system detects wilt, the alert is integrated with the irrigation module, allowing adaptive responses such as increasing water supply in specific zones. This integration ensures that visual data does not operate in isolation but instead contributes to a holistic decision-making loop within the system. Moreover, by continuously analyzing plant conditions through visual cues, the system helps reduce crop loss, supports more targeted resource use, and improves the overall robustness of the smart agriculture platform. With minimal human oversight, this component brings the power of automated crop surveillance, supporting early diagnosis and enhancing long-term productivity in controlled cultivation environments.

2. METHODOLOGY

2.1 Requirements Gathering

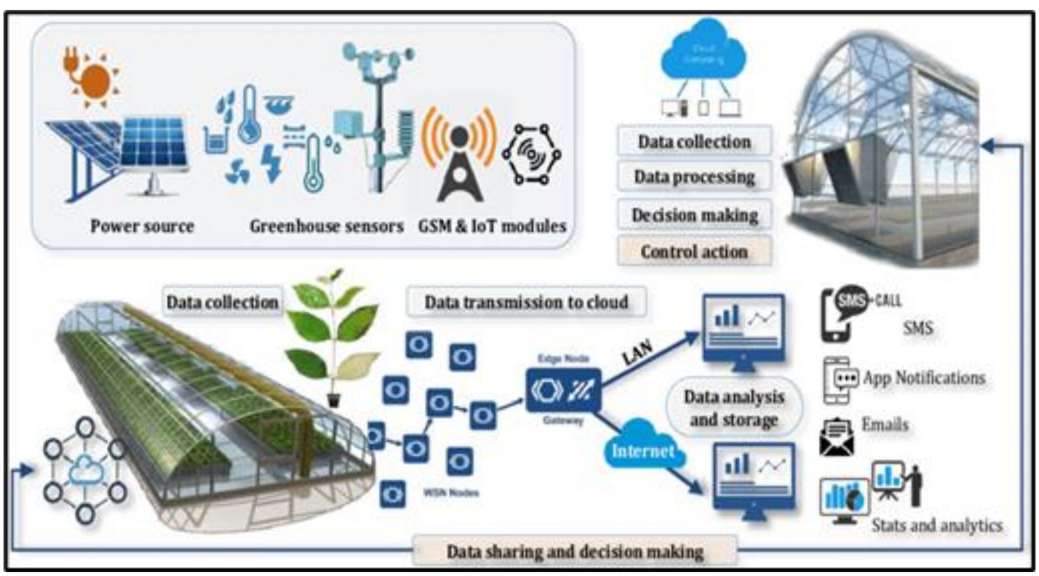


Figure 2 Data Collection

Requirement gathering is known as a foundational phase in the development of the cucumber fruit monitoring system. The key objective is to clearly identify and document the expectations, needs, and constraints of the system from both technical and user perspectives. This ensures that the final system is aligned with the real-world operational needs of greenhouse cucumber monitoring and decision-making. A thorough understanding of these requirements helps guide system architecture, component design, implementation, and evaluation, while minimizing the risks associated with scope creep, rework, and user dissatisfaction.

In here, requirement gathering also plays a critical role in integrating artificial intelligence, image processing, and Internet of Things (IoT) components effectively, ensuring a smooth interaction between hardware (ESP32-CAM modules), data processing systems (Virtual Machine and deep learning models), and visualization tools (InfluxDB and Grafana).

2.1.1 purpose of requirement gathering

Requirement gathering is a fundamental and indispensable phase in the development of any engineering or software system. Its primary objective is to systematically identify, capture, and document the needs, expectations, and constraints of all stakeholders involved in the project. For this thesis project, which focuses on real-time monitoring, classification, and prediction in greenhouse cucumber cultivation, requirement gathering plays a crucial role in ensuring that the developed system aligns with the specific goals of enhancing agricultural productivity, efficiency, and decision-making.

The purpose of gathering requirements in this context is to establish a clear understanding of what the system is expected to achieve. It involves outlining both functional and non-functional requirements that define the system's behavior, performance, usability, and integration capabilities. This process serves as the foundation for system design, development, testing, and evaluation, helping to avoid ambiguity, reduce development risks, and ensure that all components work harmoniously to deliver the intended functionality.

Specifically, in this project, the requirement gathering phase aims to identify the essential features needed for capturing images of cucumber plants at regular intervals using ESP32-CAM modules, classifying fruit growth stages using deep learning, predicting harvest time using time-series data, and performing quality assessment and defect detection. Furthermore, it seeks to determine the requirements for storing prediction results in a time-series database (InfluxDB) and presenting them through an interactive Grafana dashboard for real-time insights.

This phase also ensures that stakeholder expectations, such as usability, accuracy, reliability, and response time, are considered early in the development process. It aids in recognizing operational constraints such as power limitations of the ESP32-CAM, network bandwidth for image transmission, and the storage capacity of the virtual machine.

2.1.2 Functional Requirements

Functional requirements define the core operations that the system must perform to meet its intended objectives. These requirements describe the specific functions, features, and behaviors that enable the system to deliver the desired outcomes in the context of cucumber monitoring, growth stage classification, harvest prediction, and quality assessment. In this project, functional requirements are critical in guiding the design and implementation of the system's components and ensuring seamless integration between the hardware, software, and data visualization layers.

One of the main functional requirements is the automated acquisition of images from the greenhouse environment. The ESP32-CAM modules must be able to capture high-resolution images of cucumber plants at defined time intervals, typically every 15 minutes, and transmit them to a centralized server. These images are then processed by the system's image classification module, which employs a Convolutional Neural Network (CNN) to classify each fruit into one of three predefined growth stages: Bud, Developing, or Mature. The classification output is essential for downstream prediction and decision-making tasks.

Another critical functionality is the harvest time prediction module, which uses a combined model with combination of CNN and regression models to analyze sequential image data. This enables the system to estimate the expected harvest date for each cucumber, allowing system users to plan harvesting activities more efficiently. Alongside this, the quality assessment module evaluates whether each cucumber meets the standards for high-quality produce. If a fruit is classified as low quality, the system further activates the defect detection function to identify specific types of defects such as Belly Rot, Discoloration, or Pythium Fruit Rot using trained image-based models.

The system must also fulfill the functional requirement of data storage and management. All processed results growth stage, predicted harvest time, quality status, and defect type must be stored accurately in the InfluxDB time-series database, tagged with timestamps and cucumber identifiers. This data serves as the foundation for real-time tracking and historical trend analysis. Additionally, the system must provide a visual interface using Grafana, where users can monitor all insights via charts, tables, and alerts. The dashboard must be dynamic, user-friendly, and capable of updating in real-time as new data is processed.

Furthermore, the system must support notification and alert functionalities. For instance, it should notify the user when a cucumber is nearing harvest time or when a defect is detected, enabling quick interventions. These alerts can be visualized on the Grafana dashboard and optionally sent through external communication channels.Collectively, these functional requirements ensure that the system performs all essential tasks needed for intelligent cucumber monitoring, supports user interaction, and delivers reliable outputs that can directly inform agricultural practices.

2.1.3 Non-Functional Requirements

Non-functional requirements define the quality attributes, performance criteria, and operational constraints of a system that ensure its overall effectiveness, reliability, and usability. Unlike functional requirements, which specify what the system does, non-functional requirements focus on how the system performs under various conditions. For the cucumber monitoring and analysis system in this project, non-functional requirements play a crucial role in ensuring the system is not only functionally complete but also dependable, scalable, secure, and user-friendly in real-world agricultural environments.

One of the key non-functional requirements is performance efficiency. Given that image data is captured every 15 minutes and must be processed in near real-time, the system must offer fast and efficient image classification, prediction, and storage processes. The models used for classification and prediction, particularly CNN and combined model with CNN and regression model must be optimized to execute within acceptable time limits, typically a few seconds per image. This ensures the data pipeline remains uninterrupted and that the dashboard always reflects the most current state of cucumber growth and health.

Another important non-functional requirement is reliability. The system should function continuously without failure, especially during critical growth stages or harvesting windows. To achieve this, both the hardware components (e.g., ESP32-CAM modules) and software infrastructure (e.g., Flask API, InfluxDB, and Grafana) must be robust and capable of handling potential disruptions, such as network fluctuations or temporary sensor downtime. Built-in fault tolerance, automatic retries, and health checks are recommended to maintain consistent operation.

Scalability is also essential, particularly if the system is to be expanded to monitor more plants or deployed in larger greenhouse setups. The architecture should support the addition of multiple camera modules, parallel processing of image data, and increased database capacity without requiring major changes to the system's core logic. Cloud-based deployment or containerization can further improve scalability options.

Usability is another critical non-functional requirement. The Grafana dashboard must present prediction results, growth stages, and alerts in a clear, organized, and user-friendly manner. Users with minimal technical expertise, such as farmers or agricultural staff, should be able to understand the visualizations and take appropriate action based on the system's outputs. Features such as tooltips, legends, color coding, and simple labels enhance the dashboard's usability.

Security and data integrity are also important considerations. The system should restrict unauthorized access to sensitive components such as the server, database, and prediction models. Secure communication protocols and authentication mechanisms must be in place to ensure that image data and prediction results are protected from tampering or misuse.

Finally, the system must meet maintainability and flexibility standards. As agricultural research progresses or additional crops are introduced, the models, parameters, and hardware should be easily upgradable. Modular system design and well-documented codebases help ensure that future developers or researchers can maintain and enhance the system without starting from scratch. Non-functional requirements ensure the cucumber monitoring system is efficient, reliable, user-friendly, and adaptable for practical deployment. These attributes are vital for delivering a high-quality, long-term solution that supports smart farming practices and enhances agricultural productivity.

2.2 System Design

2.2.1 System Overview

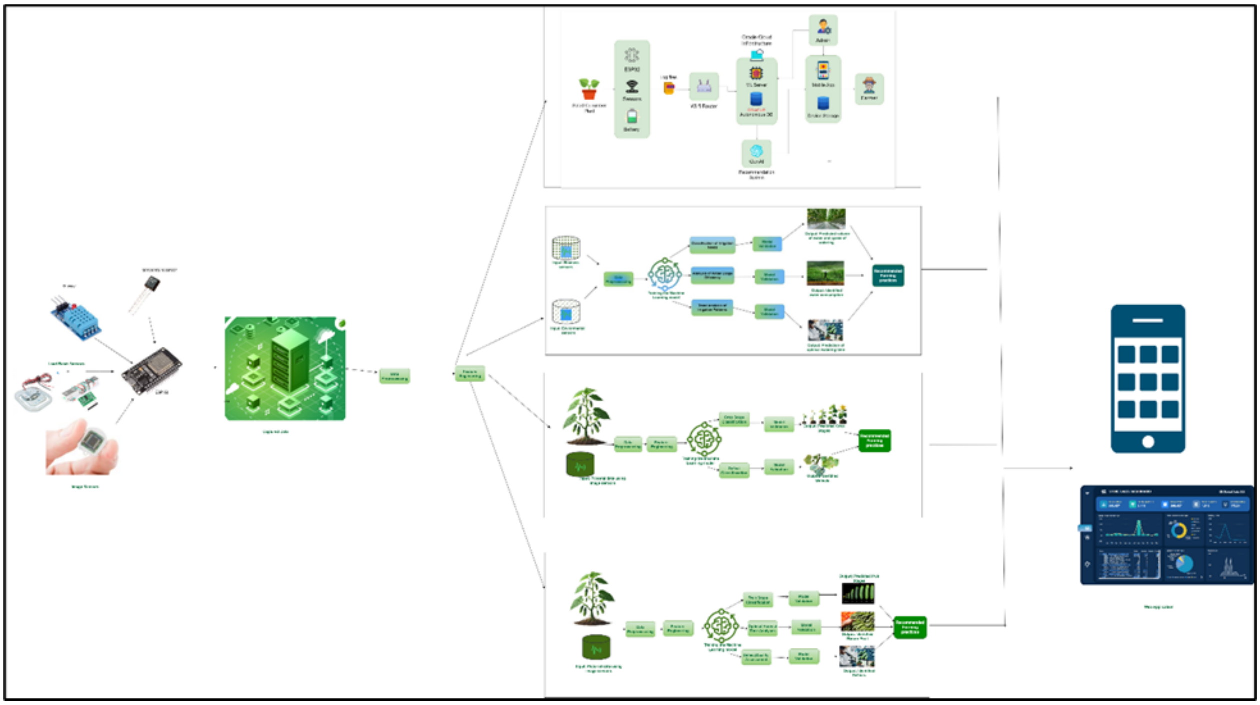


Figure 3 System Overview

The proposed system is designed as an integrated and intelligent agricultural monitoring and decision-making platform that supports efficient cucumber crop cultivation. This system primarily aims to overcome the limitations of traditional agriculture monitoring methods, which heavily rely on expensive sensors and manual observations. Instead, it focuses on utilizing a combination of low-cost sensing technologies, machine learning algorithms, and real-time image processing to deliver accurate insights into plant health, growth stages, yield potential, and optimal harvest timing. By integrating multiple parameters such as biomass weight, environmental conditions, and visual imagery the system delivers a comprehensive and scalable solution that supports both precision farming and sustainable agricultural practices.

The first major component of this system is the Data Acquisition Layer, which is responsible for collecting critical information from the field. This layer captures three main types of data. First, biomass data is collected through load cells s placed at the base of selected cucumber plants Meantime collect the temperature, humidity, lighting and also get the tension of the cucumber plant using the fiber which known as the top weight. These measurements are taken regularly to assess plant growth and health trends over time. Second, environmental conditions, including temperature and humidity, are recorded in real time using affordable yet reliable sensors. These parameters are essential for understanding the external factors influencing crop development. Third, the system captures high-resolution images of the cucumber plants using embedded cameras, such as ESP32-CAM modules, which are programmed to take periodic images at fixed intervals. These images provide visual cues about the growth stages and help detect issues such as disease symptoms or pest infestations.

Once the raw data is collected, it undergoes Data Preprocessing to ensure its quality and relevance for analysis. For the sensor-based numerical data, preprocessing includes cleaning the data to remove anomalies, normalizing the values for consistent analysis, and performing feature engineering to derive additional useful indicators, such as the rate of weight change or time since last irrigation. For the visual data, preprocessing involves resizing and enhancing the images, applying filters to reduce noise, and labeling them based on the known growth stages (bud, developing, and mature). These preprocessed inputs form the foundation for the next phase—machine learning model training and inference.

The core intelligence of the system lies within its Machine Learning and Predictive Modeling component. Several supervised machine learning models are developed and trained on the preprocessed data to perform specific tasks. A convolutional neural network (CNN) is trained to classify the cucumber growth stage based on the visual features extracted from the images. This classification is critical for identifying whether a plant is in the bud, developing, or mature stage. For harvest time prediction, a hybrid CNN-regression model is employed. It takes into account both the growth stage and the corresponding environmental and weight data to predict the number of days remaining until optimal harvest. This time-series approach helps farmers plan their harvesting activities more accurately. Additionally, another model is developed to manage irrigation needs. By analyzing historical irrigation events, plant weight fluctuations, and environmental data, the system can provide recommendations for when and how much water to apply. These models are periodically retrained to adapt to changing conditions and to improve prediction accuracy.

To make these predictions actionable, the system integrates a Decision Support System (DSS), which serves as the brain of the platform. The DSS continuously analyzes the outputs from the various machine learning models and generates real-time recommendations. For example, if a fruit is identified as mature and the predicted harvest date is within two days, the system will notify the farmer to prepare for harvest. Similarly, if environmental data combined with plant weight suggests inadequate growth, the DSS may alert the user to potential stress conditions or suggest irrigation. This layer not only automates monitoring but also enhances decision-making by providing timely insights backed by data.

Finally, the system includes a user-friendly Visualization and User Interface component. This module is built using Grafana and is connected to an InfluxDB database where all the prediction results and real-time sensor readings are stored. The dashboard presents key information such as the number of cucumbers in each growth stage, predicted harvest times, irrigation recommendations, and quality assessment scores. Users can view this data through dynamic visualizations, graphs, and summary cards. Additionally, alert mechanisms such as email or SMS notifications can be configured using IFTTT to inform users when specific thresholds or events are reached, such as the need for irrigation or readiness for harvest.

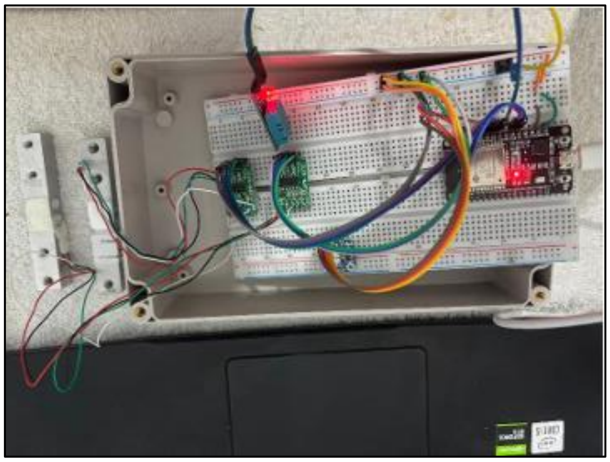


Figure 4 Prototype

2.3 Component Breakdown

2.3.1. yield Prediction – IT21267222

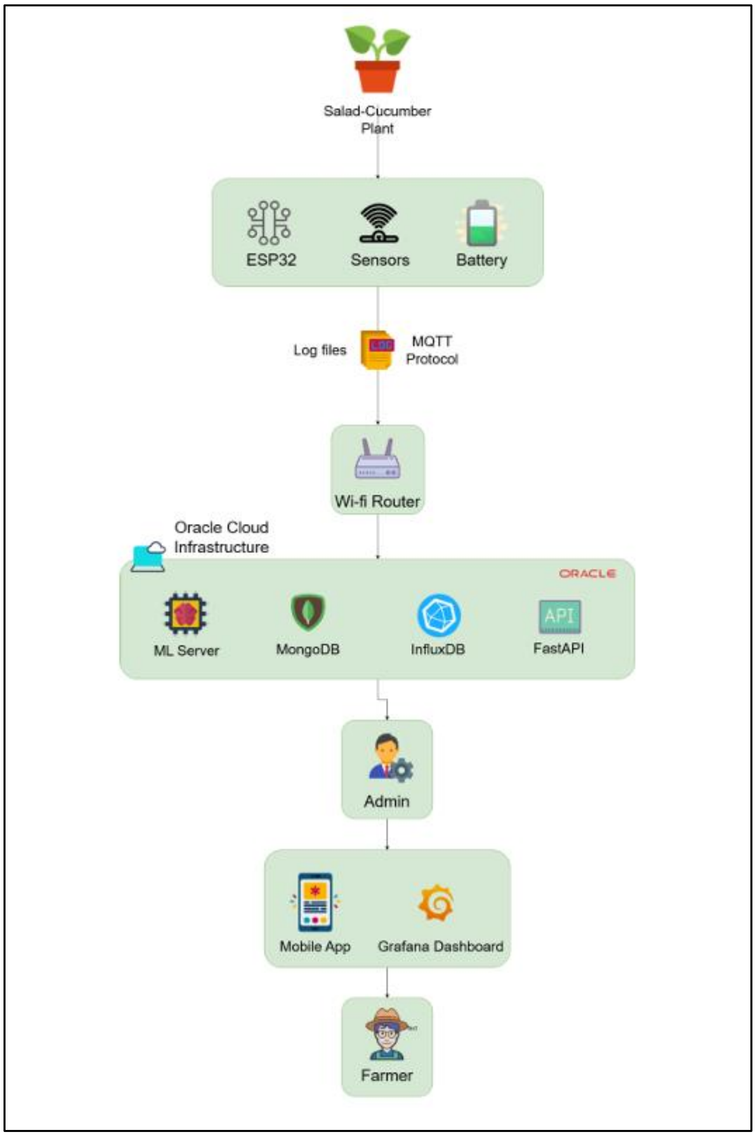


Figure 5 yield prediction

The system design phase represents the architectural blueprint of the proposed solution, articulating how various components work together to enable real-time monitoring and intelligent prediction for salad cucumber cultivation in a greenhouse environment. It ensures that the theoretical foundations laid during the requirement and feasibility studies are translated into a scalable, maintainable, and functional implementation that meets both technical objectives and practical user needs. The system integrates Internet of Things (IoT) sensor modules, machine learning models, cloud infrastructure, and a visualization interface in a cohesive framework. Each layer of the system is carefully designed to ensure smooth data flow, real time responsiveness, and user accessibility. The first major segment of the system is the data acquisition layer, responsible for gathering raw sensor and image data from the greenhouse environment. It comprises ESP32 microcontrollers connected to various sensors, including load cells for measuring biomass, top mounted pressure sensors for fruit development detection, DHT11 sensors for temperature and humidity, and LUX sensors to monitor ambient light intensity. These sensors are programmed using Arduino IDE and configured to send readings every fifteen minutes. By opting for hardware like the ESP32, which combines compact size, Wi-Fi connectivity, and compatibility with external modules, the system ensures a robust and low-maintenance sensing layer. This hardware configuration supports per-plant monitoring, a core innovation in this project, which allows farmers to gain actionable insights for individual plants rather than relying on batch-level approximations [6]. The communication and messaging layer facilitates secure data transmission between the greenhouse and the cloud server. For this, the system uses MQTT protocol, known for its lightweight overhead and efficiency in constrained environments. However, instead of relying on a public MQTT broker, a dedicated broker instance was deployed on the Oracle Cloud Infrastructure (OCI) virtual machine. This not only ensures higher security but also provides full administrative control over topics, client authentication, and logging. The ESP32 devices are programmed to publish data to designated MQTT topics using the VM's public IP address. These design choices guarantee data confidentiality and stability while reducing dependence on external platforms [8]. On the server side, the processing and data handling layer operates as the backend engine of the system. The MQTT broker on OCI receives incoming data, which is then parsed and stored in a MongoDB instance hosted on the same virtual machine. MongoDB is chosen for its flexible schema, high write throughput, and native support for JSON-like documents, making it ideal for time-stamped sensor datasets. Each document entry includes metadata such as plant ID, timestamp, sensor type, and reading value, enabling efficient querying and aggregation operations. This structured storage approach allows the system to perform high-resolution analytics, including time-series analysis, anomaly detection, and trend visualization [7]. Once data is collected and structured, it is passed to the machine learning inference engine, which houses trained models for predicting yield, estimating harvest readiness, and detecting underperforming plants. This project employs a hybrid model architecture, combining Random Forest for static feature learning and Long Short-Term Memory (LSTM) networks for time-series trend analysis. These models are trained in Google Colab using historical data collected during early trial cycles. Once trained, the models are exported in .pkl and .h5 formats and deployed on the Oracle VM using Python's FastAPI framework. The API receives real-time sensor values, processes them using the trained models, and returns predictions such as estimated harvest weight, yield score, and plant health alerts. These outputs are also pushed to MongoDB for archiving and future model retraining purposes [12]. The decision support layer transforms predictive results into actionable insights. For instance, if a plant shows a declining biomass trend combined with reduced top-pressure readings, the system flags it as a low-yield risk. Similarly, a plant approaching harvest readiness based on cumulative weight and light exposure patterns is flagged for farmer attention. These 33 | Page decisions are based on thresholds and logic rules refined through field observations and prior data analysis, thus aligning technological insights with agricultural intuition [10]. To present all system insights in a user accessible format, the visualization and notification layer is developed using Grafana. Grafana interfaces with MongoDB through API bridges and renders real-time dashboards that include time-series graphs, gauge indicators, and tabular views for all monitored variables. Each plant's status is displayed using identifiers, and color-coded visual cues help greenhouse administrators quickly identify which plants need attention. The dashboard includes predicted harvest dates, yield estimations, sensor reading trends, and alert panels. Additionally, a mobile application developed for Android fetches the same data via FastAPI endpoints, allowing farmers to receive real-time updates on their smartphones. This ensures operational responsiveness and enables decision-making on the move [6]. In addition to dashboards, the system supports automated alerting mechanisms, configured to notify users via email or SMS using IFTTT when certain conditions are met such as when a plant's harvest window is imminent or when sensor readings indicate a fault. This proactive approach minimizes the risk of delayed interventions and ensures that users are kept informed without having to constantly monitor the dashboard [4]. The system's modular structure ensures future extensibility. For example, integrating additional sensors like soil moisture, carbon dioxide levels, or leaf wetness is straightforward within the current architecture. Furthermore, the models are retrained periodically using new data ingested from MongoDB, and the system supports A/B testing for evaluating new model versions without disrupting live predictions. This dynamic adaptability allows the system to evolve with changing environmental conditions, crop cycles, and technological advancements [3]. In conclusion, the system design brings together multiple layers of technology, from sensor networks and MQTT communication to cloud-based storage, AI-driven analytics, and user-centric interfaces. Each module is thoughtfully integrated to ensure high data integrity, predictive accuracy, and practical usability. By combining real-time data flow, intelligent modeling, and decision support, the system addresses key challenges in greenhouse farming and sets a scalable foundation for broader applications in precision agriculture .

2.3.1 Irrigation system – Wijesinghe.W.E.S.P(IT20418274)

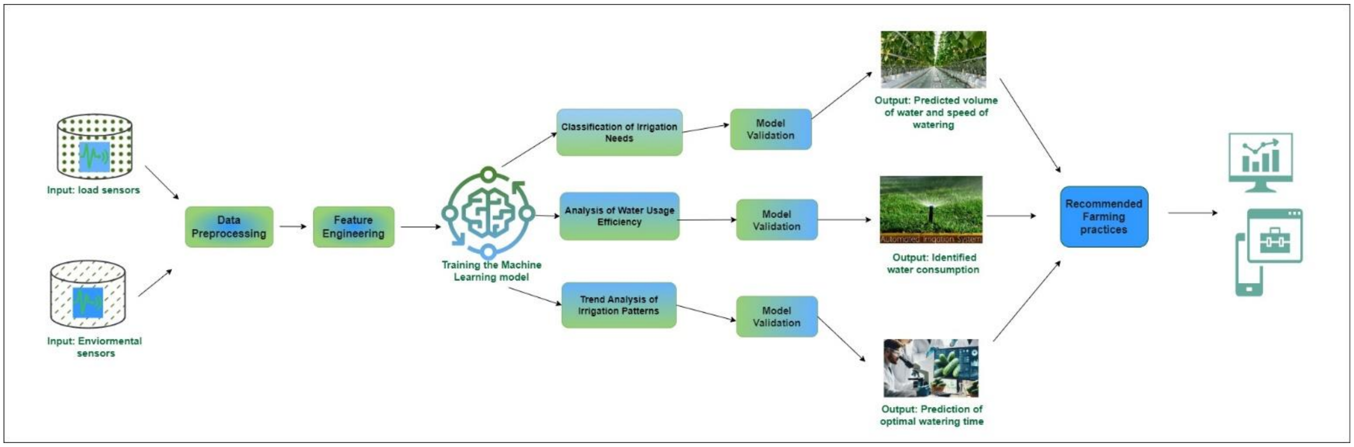


Figure 6 Irrigation System

The proposed system integrates both image-based plant health monitoring and environmental sensor-based irrigation control into a unified, automated decision-making framework. At the core of the system lies the **Wilt Detection Subsystem**, which utilizes an ESP32-CAM module to capture periodic images of cucumber leaves. These images are processed using OpenCV to enhance quality and extract relevant features, which are then passed through a trained Convolutional Neural Network (CNN) model. The model classifies each leaf as either healthy or wilted. This classification plays a critical role in guiding irrigation decisions, especially when analyzed alongside real-time weight measurements of the plant.

Complementing this is the **Irrigation Subsystem**, which uses data collected from multiple environmental sensors—measuring temperature, humidity, soil moisture, and biomass weight—to determine whether irrigation is required. These data inputs are analyzed using a Generative Regression Network (GRN) model, which predicts irrigation needs with high precision. When irrigation is deemed necessary, solenoid valves are activated through a microcontroller to deliver water to the plants. The **Data Flow and Control** architecture ensures seamless communication among components: all sensor and image data are gathered by a microcontroller and stored in an InfluxDB database. A Grafana-based dashboard visualizes this information, allowing users to monitor environmental trends, plant conditions, and system decisions in real time.

The system's intelligence is powered by a combination of machine learning models that are capable of adapting to diverse environmental conditions. Over time, as more data is accumulated, the models are retrained to improve prediction accuracy and responsiveness. This **dual-layered architecture**, combining visual and environmental feedback, creates a more comprehensive precision agriculture system. It minimizes water waste, prevents over-irrigation, and enhances decision-making by providing growers with real-time, actionable insights—resulting in a more intelligent, sustainable, and high-performing farming solution.

2.3.3 Crop growth stags – Somarathne D.K. (IT21327094)

This section details the architectural decomposition of the developed system, outlining the primary subsystems and their interactions across the data pipeline. The system was divided into five major layers: the Input Layer, Data Collection Layer, Data Storage and Processing Layer, Central System and Simulation Engine, and Output Layer. Each layer performs specific roles for the three core functionalities developed:

Crop Growth Stage Classification and Forecasting,

II. Cucumber Leaf Disease Detection and Identification, and

III. Nitrogen Deficiency Prediction.

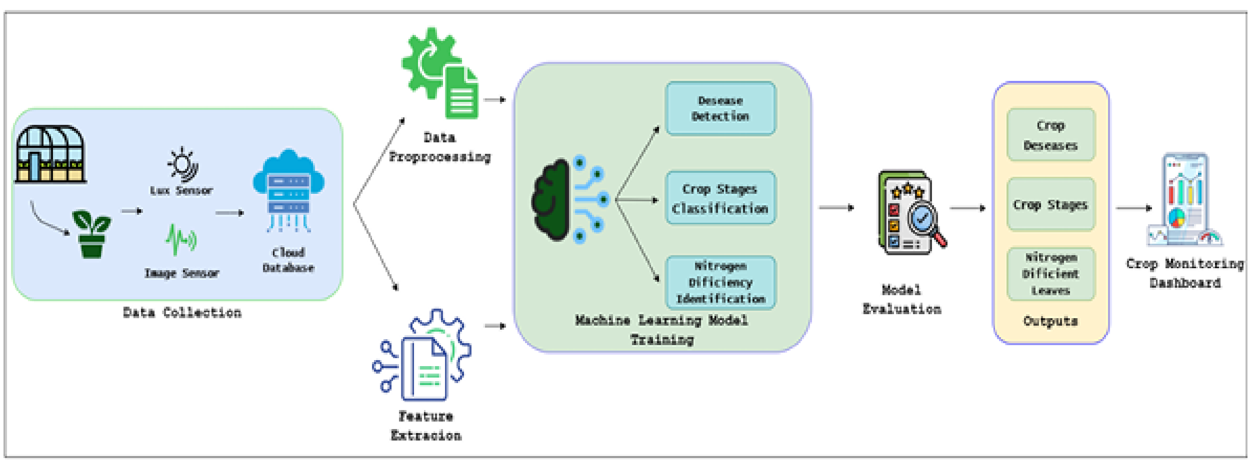


Figure 7 Crop growth Stage

2.4.1 Input Layer

The input layer serves as the system's initial interface with the external environment, where raw image and sensor data are collected for further processing. The input layer is engineered to support three primary data streams, each tailored for a distinct component of the system.

A. Crop Growth Stage Prediction

For the growth stage prediction component, the input mechanism relies primarily on continuous image acquisition through an ESP32-CAM module, configured to autonomously capture high resolution images of cucumber plants every 30 minutes. This interval was selected to ensure comprehensive temporal coverage of the plant development cycle. Each image frame captures the entire plant from a fixed frontal perspective, with a white backdrop to eliminate background distractions and ensure consistency across samples. To isolate each plant and minimize noise from adjacent crops, the camera module was mounted on an adjustable-height stand positioned half a meter in front of each cucumber plant, allowing real-time image input to be consistently framed. Initially, the sensor was located 1.5 feet above ground level, and the mounting height was adjusted as the plant grew, ensuring optimal visibility across all growth phases namely flowering, early fruiting, and full fruit development. Each image, saved with a timestamp-based filename (e.g., plant1_2024-10-15_08-30.jpg), acted as a key input element to both the classification and forecasting models. The continuous influx of temporally labeled images enabled the system to assess not only the current growth stage but also to estimate the number of days remaining before transitioning to the next phenological stage.

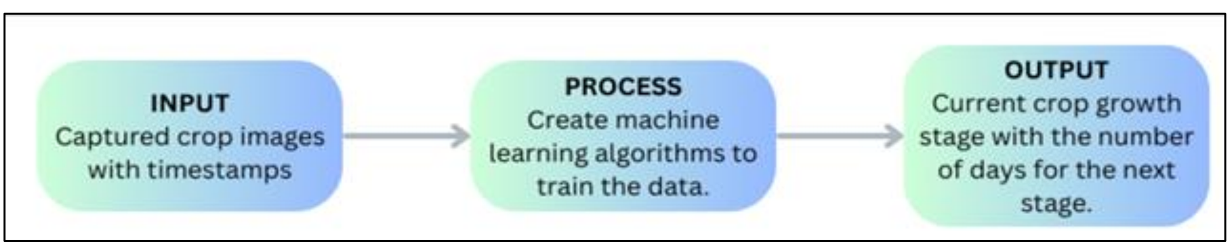


Figure 8Workflow of crop growth stage prediction using timestamped images and machine learning.

B. Cucumber Leaf Disease Detection and Identification

The leaf disease analysis subsystem was designed to operate with both offline and online image datasets. Given the scarcity of naturally occurring leaf diseases during the in-situ data collection period, an open-source dataset was integrated, containing 4,000 manually curated leaf images classified under five categories: Anthracnose, Bacterial Wilt, Downy Mildew, Gummy Stem Blight, and Healthy. Each category included approximately 800 high-resolution samples, with images labeled accordingly. In parallel, additional leaf images were acquired in the controlled tunnel using a dedicated ESP32 CAM module that was positioned to focus solely on the plant foliage. This camera was affixed to a separate, flexible stand to enable height and angle adjustments specific to leaf zones. Supplementary leaf photographs were also taken manually using mobile phone cameras, particularly to capture early signs of abnormal chlorosis or necrosis indicative of disease. These images were pre-processed through a consistent pipeline and served as the raw input to two deep learning models: one binary classifier to distinguish between healthy and unhealthy leaves, and another multi-class classifier to identify the specific disease from the unhealthy leaf samples.

C. Nitrogen Deficiency Prediction

For nitrogen deficiency detection, the system employed a hybrid input mechanism combining image data and numerical sensor readings. Leaf images were captured using both the ESP32-CAM module and mobile phones, ensuring visual diversity across different nitrogen levels. A lux sensor, deployed adjacent to each plant, simultaneously recorded light intensity values, which were time-synced with each captured image and stored in a structured dataset. The leaf images were visually assessed for chlorotic symptoms typically associated with nitrogen stress (e.g., yellowing along leaf margins or interveinal regions), while lux values served as a proxy for light absorption and photosynthetic efficiency, both of which correlate with nitrogen status. To extract additional relevant attributes, the image data were analyzed to derive color metrics (e.g., mean green channel intensity, NDVI, ExG index) and morphological features (e.g., leaf area, convex defect count). These were later used in the regression pipeline to quantify nitrogen deficiency levels. Thus, the input layer for this subsystem encompasses both visual and tabular data streams, synchronously delivered into the subsequent processing modules.

2.5.2. Software Components

Python serves as the primary programming language throughout the system. It is used for a range of tasks including model development, data preprocessing, image handling, and back-end server operations. The server side of the system is built using Flask, a lightweight and flexible Python web framework. Flask manages incoming HTTP POST requests from the ESP32-CAM modules, receives image files, stores them in structured directories on the Oracle VM, and initiates the preprocessing and inference workflows. For model training and inference, TensorFlow and Keras were utilized to design and optimize custom convolutional neural network (CNN) architectures. One CNN model is employed for classifying the plant's current growth stage, while a secondary CNN with regression capability predicts the estimated number of days until the plant reaches the next stage. These models were trained on custom datasets collected during the field deployment. In addition to model design, OpenCV was used extensively during image preprocessing tasks such as resizing, pixel normalization, color space conversion, and image augmentation. This preprocessing pipeline ensures input consistency and enhances model performance under varying lighting and environmental conditions. Feature extraction related to leaf color and shape was performed using ResNet50, a deep CNN that captures high-dimensional image features which are critical for detecting deficiencies or early signs of disease. Additional libraries and packages such as NumPy and Pandas were used for managing numeric data and tabular structures, particularly during the analysis of timestamped predictions and time-series pattern simulations. These tools facilitated smooth integration of model outputs with the storage and visualization layers of the system.

2.5.3. Database and Visualization

To efficiently store model outputs and associated metadata, InfluxDB was integrated as the system's time-series database. It captures and logs each inference result along with a timestamp, image file reference, plant ID, predicted growth stage, days until next stage, nitrogen deficiency status, and any detected leaf diseases. The InfluxDB bucket titled CAM_module_prediction is configured with appropriate retention and indexing policies to enable fast querying and smooth rendering of timesensitive information. This setup supports not only real-time access but also long-term trend analysis. For visualization, Grafana was employed as the front-end analytics platform. Grafana reads directly from InfluxDB and presents real-time metrics through customizable dashboards. In this system, each cucumber plant is visually represented with current growth stage indicators, predictive timelines for stage transition, and any health-related alerts. Grafana's intuitive interface supports a range of widgets, including time-series charts, status panels, and heatmaps. Alerting mechanisms are built into the dashboard, with visual signals triggered whenever thresholds, such as delayed growth progression or detected deficiencies are breached. These alerts enable users to take immediate action based on automated analysis.

All server-side processes, including the Flask application, model files, database services, and Grafana dashboards, are hosted on an Oracle Cloud Infrastructure (OCI) Virtual Machine. The VM setup ensures centralized control, uninterrupted service, and remote access for authorized personnel. It supports image storage, computation, and system-wide monitoring, while also offering flexibility in resource scaling based on system demands. To complement visual alerts, the system integrates with IFTTT (If This Then That), a cloud-based service that sends instant notifications when specific conditions are met, such as disease detection or significant nitrogen drop. This integration ensures that users are promptly informed of any critical conditions even when they are not actively monitoring the dashboard.

2.3.4 Cucumber Fruit Analysis – Thilakarathne M.B.N.(IT21225956)

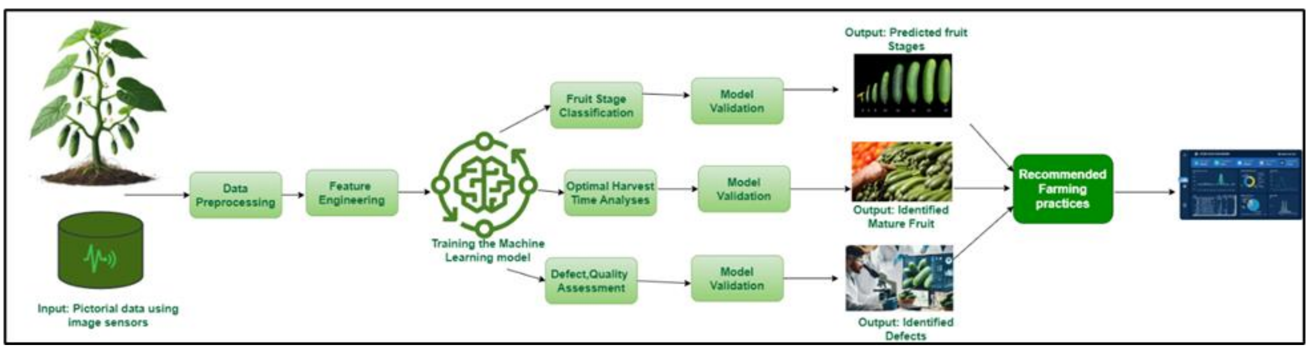


Figure 9 Cucumber Fruit Analysis

2.3.1 Input Layer

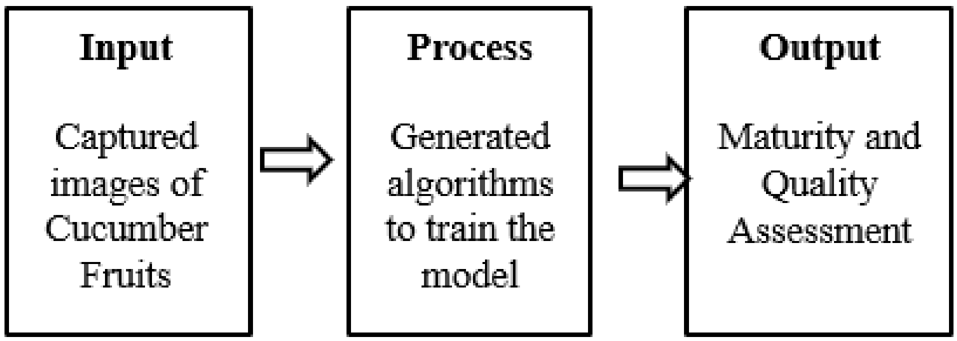


Figure 10 Input layer

The Input Layer serves as the foundation of the cucumber monitoring and analysis system, responsible for acquiring raw data that feeds into the processing pipeline. This layer is primarily built upon Internet of Things (IoT) infrastructure, which enables automated, consistent, and high-frequency image capture within a controlled greenhouse environment. The design of this layer ensures that the data collected is relevant, timely, and of sufficient quality to support accurate classification, prediction, and quality assessment.

At the heart of the input layer are ESP32-CAM modules, which are compact microcontroller units equipped with integrated cameras and Wi-Fi capability. These modules are strategically deployed across the greenhouse, each facing individual cucumber plants or sections of crop rows. The devices are programmed to capture images at 15-minute intervals, allowing for fine-grained temporal tracking of fruit development. This interval was chosen to balance data granularity with network and storage constraints, ensuring detailed growth monitoring without excessive data redundancy.

The ESP32-CAM units are configured to connect wirelessly to the local network and transmit the captured images to a centralized Virtual Machine (VM). The VM acts as a local server where all uploaded images are temporarily stored for preprocessing and inference. Each image file is named using a structured format that includes the timestamp and plant identifier, enabling the system to associate a sequence of images with individual cucumber fruits over time. This timestamp-based file naming also facilitates time-series analysis for harvest prediction and growth tracking.

In addition to image data, the input layer could be extended in future iterations to include auxiliary environmental data such as temperature, humidity, and light intensity using compatible sensors. Although not integrated in the current prototype, such data could enhance model accuracy and provide additional insights into the environmental factors affecting cucumber growth and quality.

To ensure the reliability of data acquisition, each ESP32-CAM module is powered either via a stable electrical supply or rechargeable battery banks with solar support, depending on the greenhouse setup. The modules operate autonomously, and any upload failures can be logged and flagged for review, minimizing the risk of data loss. The image resolution and focus settings are calibrated in advance to capture clear, consistent images of the cucumber fruits from fixed angles and distances.

2.3.2 Data Collection Layer

The Data Collection Layer acts as a critical intermediary between the input sources and the core data processing components of the cucumber monitoring and prediction system. Its primary role is to systematically gather, organize, and maintain incoming image data from the ESP32-CAM modules deployed in the greenhouse. This layer ensures that the data is structured, reliable, and readily accessible for downstream tasks such as growth stage classification, harvest time prediction, and quality assessment.

Upon capturing images at regular intervals (every 15 minutes), each ESP32-CAM module uploads the images to a centralized Virtual Machine (VM) using Wi-Fi. This upload process is handled through a lightweight Python-based Flask server (upload_server3.py), which listens for incoming HTTP POST requests containing image files. The server validates and stores each image in a predefined directory (/home/opc/upload_harvest), ensuring organized storage according to the date and time of capture. This structured storage approach simplifies the task of tracing the growth timeline of individual fruits and supports sequential analysis for harvest prediction.

The data collection layer not only ensures image integrity during transmission but also performs basic file validation and logging to track the source device, time of upload, and any upload errors. This helps maintain a high-quality data set, crucial for effective model performance.

Although the primary data type managed by this layer is image data, the system architecture is designed with extensibility in mind. Future versions of this layer may incorporate environmental sensor data (e.g., temperature, humidity, light intensity) collected via MQTT based endpoints, enhancing the context and richness of the dataset.

Figure 11 Data Collection

2.3.3 Data Storage and Processing Layer

The Data Storage and Processing Layer plays a pivotal role in the cucumber analysis system by managing the storage, transformation, and intelligent analysis of incoming image data. This layer bridges the raw data acquired through the data collection pipeline with the intelligent predictions and classifications necessary for decision-making. It is composed of two major components: a structured time-series database for storing prediction results and a model inference engine responsible for analyzing cucumber growth stages, estimating harvest time, and assessing fruit quality.

All image-based inference results are stored in InfluxDB, a high-performance, open-source time-series database optimized for time-stamped data. The system utilizes a dedicated bucket called CAM_module_prediction where each prediction is stored along with metadata including timestamp, plant identifier, growth stage classification, predicted harvest date (in days), quality status, and defect type (if applicable). The time-series structure of InfluxDB enables efficient querying and visualization of cucumber development over time, supporting real-time monitoring and historical trend analysis.

InfluxDB is chosen due to its lightweight structure, fast read/write capabilities, and compatibility with visualization tools such as Grafana. The database schema is designed to support flexible queries and filtering based on individual plants, time periods, or prediction outcomes. This facilitates efficient integration with the visualization layer and ensures that data is accessible for both real-time and retrospective insights.

The core of this layer is the model processing pipeline, which includes pre-trained deep learning models deployed in the virtual environment. The first model is a Convolutional Neural Network (CNN) classifier that processes uploaded images to determine the cucumber's growth stage Bud, Developing, or Mature. This stage classification is crucial for understanding developmental progress and is logged into the database for monitoring.

Following this, a CNN combined with a regression model is used to predict the estimated harvest time based on sequential image features. This model learns to associate visual patterns with the number of days remaining until harvest. For each fruit, predictions are updated as new images are received, refining the harvest estimate with temporal accuracy.

A separate CNN model handles quality assessment, classifying cucumbers into High Quality or Low Quality. If a fruit is determined to be of low quality, the system further identifies the specific defect Belly Rot, Discoloration, or Pythium Fruit Rot—using a fine-tuned classifier trained on annotated defect datasets. These predictions allow for real-time quality control and alert-based decision-making.

The processing system is implemented using Python with the TensorFlow/Keras framework and is designed to be modular, allowing individual models to be updated or retrained independently. Images are preprocessed (resized, normalized) before inference to ensure consistency and model performance. Each prediction is timestamped and pushed to the InfluxDB storage layer.

2.3.4 Central System and Simulation Engine

The Central System and Simulation Engine serves as the command-and-control hub of the cucumber monitoring architecture. It orchestrates the interactions between the various system components, manages data flow, executes inference pipelines, and handles alert generation and data communication to visualization platforms. This layer acts as the brain of the system, providing centralized coordination and ensuring that the entire pipeline functions seamlessly in real-time.

The Central System is deployed on a Virtual Machine (VM) environment, which hosts the backend infrastructure including the Flask-based upload server, trained machine learning models, InfluxDB time-series database, and Grafana dashboard interface. Upon receiving an image upload from the ESP32-CAM module, the Flask server triggers the inference pipeline. This centralized architecture allows for efficient resource allocation, simplified management, and scalability of services.

Within this VM environment, the Central System is responsible for:

- Receiving and organizing image data
- Triggering model-based analysis
- Updating predictions into InfluxDB
- Send notification when the fruit will be mature.

Each component of the system operates as a service or microservice, enabling modularity and ease of updates. This centralized structure also ensures synchronized processing, with image capture, model inference, data storage, and visualization occurring in near real-time.

The Simulation Engine is an integral part of this layer, primarily responsible for simulating future states based on historical data and predictive model outputs. It leverages time-series image sequences to estimate future harvest dates and growth transitions. For example, using the outputs from the CNN + Regression model, the engine calculates the remaining number of days until harvest and dynamically updates these predictions as new images are analyzed.

Furthermore, the engine performs trend analysis across multiple plants to simulate aggregate outcomes, such as the expected number of mature fruits ready for harvest in the coming days or the forecasted rate of defective cucumbers. This simulation functionality is essential for greenhouse operators to plan harvesting activities, manage logistics, and implement corrective measures proactively.

A vital functionality of the central system is its ability to generate real-time alerts. the system flags specific fruits and triggers alerts (through IFTTT integrations) dashboard pop-ups. These alerts allow stakeholders to respond quickly to issues such as early harvest requirements.The centralized system ensures data consistency, traceability, and version control. Each prediction is logged with a timestamp and image reference, enabling audit trails for system accuracy and performance evaluation over time.

2.3.5 Output Layer

The Output Layer is the final and most user-facing component of the cucumber monitoring system. It is responsible for presenting the processed predictions and insights in an intuitive, real-time, and actionable format. This layer transforms raw model outputs such as growth stage classifications, estimated harvest dates, and quality assessments into organized visual representations that stakeholders can easily interpret and act upon. Its main function is to bridge the gap between automated analytics and human decision-making.

The central element of the Output Layer is the Grafana dashboard, which is connected to the InfluxDB time-series database (CAM_module_prediction bucket). Grafana is a highly customizable, open-source data visualization tool that enables dynamic and real-time presentation of predictions through graphs, panels, and status indicators.

- On the dashboard, each cucumber plant or fruit is represented with the following metrics:
- Current Growth Stage: Displays whether the fruit is in Bud, Developing, or Mature stage.
- Predicted Harvest Time: Shows the number of days remaining until the estimated harvest date.
- Quality Status: Indicates whether the fruit is of High or Low quality.
- Defect Type (if applicable): If the fruit is classified as Low Quality, the specific defect (e.g., Belly Rot, Discoloration, or Pythium Fruit Rot) is shown.

These metrics are visualized using various Grafana widgets such as time-series graphs, bar charts, and data tables. The real-time updating capability ensures that as new images are uploaded and predictions are made, the visualizations are refreshed instantly, keeping greenhouse managers up-to-date with the latest conditions.

In addition to visual data representation, the Output Layer also supports real-time alerts and notifications. For instance, if a fruit is predicted to be harvestesoon or if a significant percentage of fruits are found to be defective, a colored alert panel is triggered on the dashboard, drawing immediate attention. This ensures that critical conditions are addressed without delay, minimizing crop loss and improving overall efficiency.

Another important feature of the Output Layer is its ability to present historical trends and longitudinal data analysis. Stakeholders can filter predictions by time range to view growth progression, changes in quality over time, and forecasting patterns. This functionality is crucial for understanding growth cycles, refining prediction models, and supporting decision-making on a scale.

2.3.6 Implementation

Irrigation system

The implementation phase of the proposed intelligent irrigation and wilt detection system involved the practical assembly, integration, and testing of hardware and software components in a controlled greenhouse environment. This process was guided by the objective of enabling fully automated, data-driven irrigation decisions based on real-time visual and environmental inputs. The system was built around two core modules: the **irrigation control subsystem** and the **wilted leaf detection subsystem**, each of which was deployed independently but interconnected to work collaboratively as a unified smart farming solution.

For the **irrigation subsystem**, a dual-sensor setup was implemented on selected cucumber plants. Base-mounted digital load cells were used to measure the total system weight, including the plant, substrate, and retained moisture. These values served as proxies for hydration levels. Simultaneously, top-mounted tensile sensors tracked vertical plant stress, which offered additional insight into water deficiency or structural shifts. These measurements were recorded every five minutes and sent to a microcontroller (Raspberry Pi), which acted as the central processing unit. Environmental data—such as temperature, humidity, and light intensity—was captured through DHT22 and LUX sensors installed within the growing tunnel. All of this data was streamed to an InfluxDB time-series database, where it was processed by a **Generative Regression Network (GRN)** model trained to determine optimal irrigation timings. When irrigation was needed, a relaycontrolled solenoid valve was triggered automatically to deliver water to the root zone. This allowed the system to operate in real time with minimal human intervention.

Wilted leaves detection

The wilted leaf detection subsystem was developed to provide an additional layer of plant health monitoring through automated image processing. ESP32-CAM modules were programmed to capture high-resolution images of the cucumber canopy every 15 minutes. These images were wirelessly transmitted to a cloud server, where they underwent preprocessing using OpenCV techniques such as resizing, contrast enhancement (CLAHE), and noise filtering. The processed images were then fed into a fine-tuned VGG16 Convolutional Neural Network, which classified each image into either “wilted” or “healthy” categories. The output of this classification was logged and analyzed alongside weight and climate data. If a wilted state was detected, and the plant’s biomass data indicated dehydration, the system triggered a priority irrigation command, overriding regular irrigation timing to prevent irreversible stress.

All predictions and sensor readings were visualized through a Grafana dashboard, which interfaced directly with the InfluxDB database. This dashboard provided users with an intuitive, real-time view of system activity, including graphs for plant weight trends, irrigation frequency, temperature patterns, and leaf health status. The system also incorporated an alert mechanism that notified users via SMS or email (using IFTTT) when critical thresholds were breached, such as prolonged wilting or low biomass recovery.

Cucumber Fruit Stage Classification

The cucumber fruit stage classification model was developed using a Convolutional Neural Network (CNN) architecture tailored for image classification tasks. The primary objective of this model is to classify cucumber fruits into one of three developmental stages: Bud Stage, Developing Stage, and Mature Stage. Accurate stage classification plays a crucial role in estimating harvest readiness and optimizing greenhouse management strategies.

The model was implemented using TensorFlow and Keras libraries in Python. The dataset used for training comprised 1,200 labeled images (400 per class), captured under controlled greenhouse conditions. To enhance model generalizability and prevent overfitting, image augmentation techniques were applied, including rotation, zoom, brightness adjustment, and horizontal flipping. This resulted in a four-fold increase in training samples, offering diverse examples for the CNN to learn robust features from. . To enhance the dataset used the augmentation techniques and after augmentation each class has 1500 images.

The CNN model architecture includes an input layer followed by a series of convolutional and max-pooling layers designed to extract hierarchical features such as texture, shape, and color patterns of cucumbers at different stages. ReLU activation functions were used to introduce non-linearity, and dropout layers were included to reduce overfitting. The final classification was achieved through a dense layer with a softmax activation function, outputting the probabilities of each class.

The model achieved high classification accuracy during training and validation, with performance metrics indicating strong generalization capabilities. After training, the model was saved in .h5 format as cucumber_model.h5 and deployed on a Virtual Machine for real-time inference. Images captured by the ESP32-CAM were automatically uploaded to the VM, where the classification model processed each image to determine the current growth stage of each detected fruit.

The predictions were then recorded in the InfluxDB database and visualized through the Grafana dashboard, allowing greenhouse operators to monitor fruit development continuously. This classification model forms the foundation for subsequent tasks, such as harvest prediction and quality evaluation, by providing temporal context for each cucumber's growth trajectory.

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_4 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_5 (Conv2D)	(None, 60, 60, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten_1 (Flatten)	(None, 115200)	0
dense_2 (Dense)	(None, 128)	14,745,728
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 3)	387

Total params: 14,839,363 (56.61 MB)

Trainable params: 14,839,363 (56.61 MB)

Non-trainable params: 0 (0.00 B)

Figure 12 Fruit Developemnt model

Fruit Quality Assessment and Defect Identification

The Quality Assessment and Defect Detection model was developed to ensure that harvested cucumbers meet high-quality standards, thereby reducing waste and enhancing market value. This model performs a two-step classification: first, it determines whether a cucumber is of high or low quality; second, if the quality is low, it identifies the specific defect present. The defect categories include Belly Rot, Discoloration, and Pythium Fruit Rot, three common and critical conditions affecting cucumber yield and consumer acceptance.

The model was implemented using a deep Convolutional Neural Network (CNN) using Keras with TensorFlow backend. The dataset for this task consisted of 960 annotated images: 500 labeled as High Quality, and 500 labeled as Low Quality, further divided into 160 images per defect class. As with the fruit stage classification model, data augmentation was applied to enhance the model's ability to generalize across diverse image conditions, including variations in lighting, angle, and background noise. This also helped address the imbalance among classes. After augmentation applied, for the quality assessment dataset had the 2000 each class and for the defect identification had the 800 images for each class.

The model architecture consists of multiple convolutional layers for hierarchical feature extraction, interleaved with max-pooling layers to reduce spatial dimensions while preserving relevant features. Batch normalization and dropout layers were used to enhance model stability and prevent overfitting. The model terminates with two dense layers. The first outputs binary quality classification (high or low), while the second is activated only if the cucumber is classified as low quality, identifying the type of defect via softmax activation.

Model training was conducted over multiple epochs with a categorical cross-entropy loss function and Adam optimizer. The model showed high classification accuracy during validation, demonstrating its capability to correctly identify subtle visual cues indicative of cucumber defects.

In deployment, the model is loaded on the VM and integrated with the image acquisition pipeline. Each image captured and uploaded by the ESP32-CAM undergoes processing by this model. If a cucumber is identified as low quality, the type of defect is recorded in the InfluxDB database under the CAM_module_prediction bucket. Real-time visualization through Grafana allows users to track defect trends, take corrective action in the field, and prioritize the sorting of fruits before packaging.

The model also plays a role in automated alerts: if a significant number of defective cucumbers are detected within a short timeframe, the system can trigger alerts or recommendations, helping farm managers prevent wider outbreaks or environmental triggers. Thus, this model is central to maintaining consistent product quality in the cucumber production pipeline.

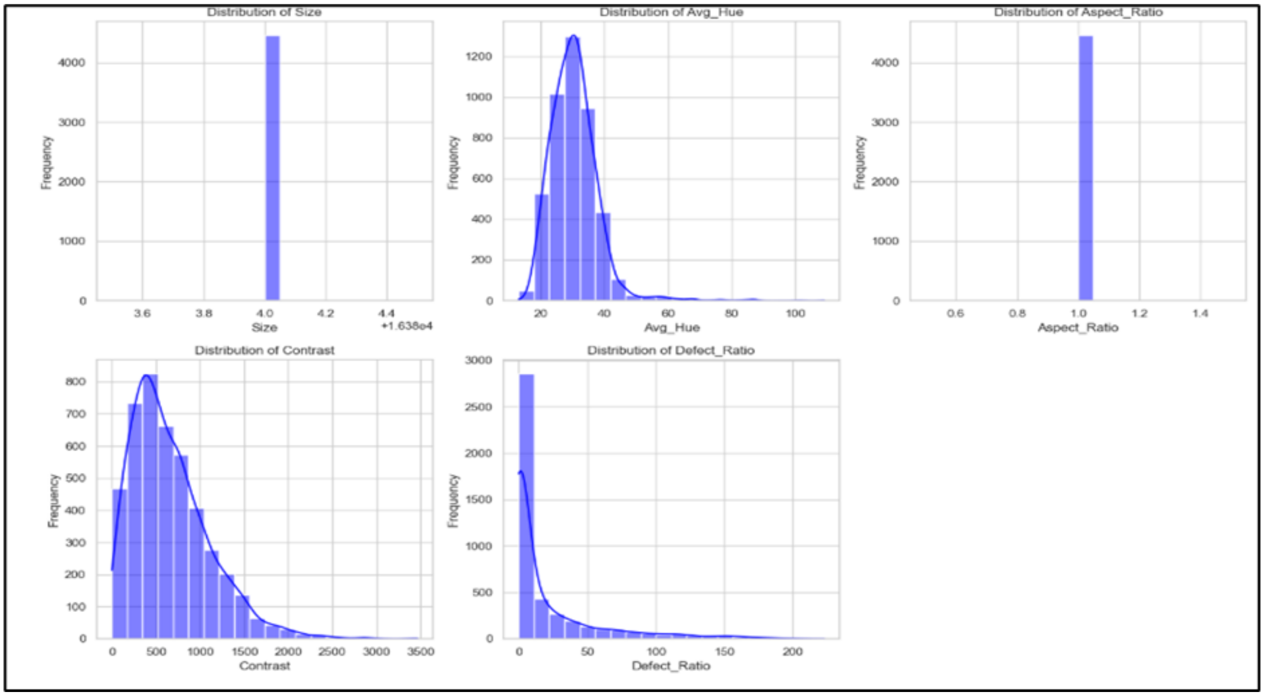


Figure 13 feature Extraction

Harvest Time Prediction

The Harvest Time Prediction model was developed to estimate the number of days remaining before a cucumber fruit reaches its optimal harvest stage. This model utilizes temporal patterns in fruit growth, captured through sequential image data, to provide an accurate forecast of harvest readiness. Timely harvest is critical to preserving fruit quality, maximizing yield, and ensuring efficient labor and resource allocation in greenhouse farming.

The model was implemented using a hybrid architecture that combines Convolutional Neural Networks (CNN) with Regression. The CNN component is responsible for extracting high-level spatial features from individual fruit images, such as size, shape, and texture key indicators of growth stage. These features are then passed to a regression layer, which outputs a continuous value representing the predicted number of days until harvest. This approach avoids the limitations of discrete stage classification by providing more precise harvest scheduling.

The dataset used for training the model consisted of more than 400 images corresponding to 40 individual cucumber fruits, with at least 10 images per fruit, captured over time from the bud stage to maturity. The filenames included timestamps that allowed for temporal labeling of the actual harvest intervals. The training data was structured to reflect the progressive growth trajectory of each fruit, enabling the regression model to learn growth trends from visual input.

The CNN model consisted of convolutional layers followed by max pooling, batch normalization, and flattening. The final layers were fully connected dense layers ending in a single regression output node using a linear activation function. The model was compiled using the Mean Squared Error (MSE) loss function and optimized using the Adam optimizer, ensuring smooth gradient updates for continuous output values.

Once trained, the model demonstrated strong predictive performance with low error margins in harvest time estimation. During deployment, the model is integrated into the server pipeline hosted on the Virtual Machine (VM). Images captured by the ESP32-CAM every 15 minutes are automatically uploaded to the VM. Each image is processed to extract the current visual state of a cucumber fruit, which is then fed to the CNN + Regression model to predict the remaining harvest days.

The predicted values are stored in InfluxDB under the measurement prediction_HarvestTime in the CAM_module_prediction bucket. Grafana is configured to visualize the remaining days to harvest per fruit in real-time, enabling greenhouse operators to schedule harvesting activities accordingly. In addition, automated alerts are set up for when a fruit is within a specified harvest threshold (e.g., 3 days remaining), ensuring no fruit is overlooked or over-matured.

By accurately forecasting harvest windows, this model adds precision to the overall cucumber production cycle, enhancing operational efficiency and fruit quality outcomes.

2.3.7 Evaluation and Testing

The evaluation and testing phase plays a critical role in validating the effectiveness, accuracy, and reliability of the cucumber monitoring and prediction system. This phase ensures that the deployed models and integrated system components perform as intended in both controlled environments and real-world conditions. A rigorous evaluation strategy was adopted to test each sub-module—growth stage classification, harvest time prediction, and quality assessment—separately, followed by integration testing of the entire pipeline.

For the Growth Stage Classification model, evaluation was conducted using a labeled dataset consisting of 1,200 images, equally distributed across the Bud, Developing, and Mature stages. The dataset was split into training (70%), validation (15%), and testing (15%) sets. Performance metrics such as accuracy, precision, recall, and F1-score were computed to assess classification performance. Confusion matrices were also analyzed to identify misclassifications between neighboring growth stages. The CNN model achieved high classification accuracy with minimal false predictions, demonstrating its capability to distinguish between the three stages based on image features.

The Harvest Time Prediction model, built using CNN with regression, was evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE) as primary metrics. The model was trained on sequential image data representing individual cucumber growth over time. Cross-validation was employed to prevent overfitting and assess model generalization. The predicted number of days until harvest was compared with actual harvest dates recorded in the dataset. The low MSE and MAE values indicated strong predictive performance, with most predictions falling within a ±2-day margin of actual values.

In the case of Quality Assessment and Defect Detection, the model was tested on a separate dataset containing annotated images of cucumbers classified into High and Low quality, with the latter including specific defect categories like Belly Rot, Discoloration, and Pythium Fruit Rot. The model's precision and recall were evaluated for both binary quality classification and multiclass defect identification. The defect detection model showed robust performance in identifying distinct symptoms, validating its utility in early-stage quality screening.

Beyond model-level testing, the system integration was thoroughly validated. Functional testing was conducted to verify image acquisition through ESP32-CAM modules, image upload to the Virtual Machine, model inference, and storage of results in the InfluxDB database. The Grafana dashboard was also tested for real-time visualization, correctness of displayed data, and alert notifications.

Usability testing with simulated real-time operations helped assess response time, prediction latency, and user interaction flow. Stress testing was performed by feeding a high volume of image uploads to evaluate system scalability and server reliability. Results indicated the system could handle continuous image streams without degradation in performance.

2.5 Scope and Limitations

- The scope of this project is focused on the development and deployment of an IoT-enabled, image-based cucumber analysis system capable of real-time monitoring and intelligent prediction. The key areas included within the scope are,
- The system employs a Convolutional Neural Network (CNN) model to classify cucumber fruits into three distinct growth stages Bud, Developing, and Mature based on image features.
- By using a CNN with regression architecture, the system analyzes sequential image data of individual cucumber fruits to estimate the number of days remain until optimal harvest.
- The system evaluates the quality of each cucumber fruit and classifies it as either High or Low quality. This classification helps in sorting and planning post-harvest handling.
- For fruits categorized as low quality, the system further identifies the specific type of defect (e.g., Belly Rot, Discoloration, or Pythium Fruit Rot), facilitating quick remediation and improved crop management.
- ESP32-CAM modules installed in the greenhouse environment capture images at regular intervals (every 15 minutes). These images are automatically transmitted to a centralized VM for processing.
- Processed predictions are stored in an InfluxDB time-series database and displayed on a Grafana dashboard, enabling real-time monitoring and decision-making.
- The system includes a notification mechanism to alert users when fruits are near harvest or when quality-related issues are detected. This supports proactive farm management and reduces losses.

Limitations of the Project

- The current system is optimized for controlled greenhouse environments with consistent lighting and camera angles. Variations in outdoor conditions (e.g., fluctuating light, background noise) may affect model performance.

- The system is specifically tailored for salad cucumber analysis and may not directly apply to other crop varieties without retraining or modification.
- Model Processing Time: Although inference is relatively fast, there may be minor delays when processing large batches of images or during network congestion, especially if many ESP32-CAM modules are transmitting data simultaneously.
- The system relies solely on image-based analysis. Environmental factors such as soil moisture, temperature, or humidity which can also impact growth and quality are not currently integrated but could enhance prediction accuracy.
- This architecture well suitable for enterprise level greenhouses.
- MongoDB database cannot connect to the Grafana free version.

2.6 Commercialization Aspects of the Product

The commercialization of the cucumber monitoring and prediction system offers a promising opportunity to enhance precision agriculture practices, particularly in greenhouse farming. This product integrates deep learning, image processing, and IoT-based automation to deliver a reliable, real-time platform for monitoring cucumber growth, predicting harvest time, and assessing quality. These capabilities can be leveraged to meet the increasing global demand for high-quality produce and efficient farming operations, offering substantial benefits to agricultural stakeholders.

When used the present architecture design to greenhouse, automatically monitor the greenhouse without the labors. All the greenhouse day to day work will be automated. On behalf of that the core value proposition lies in reducing manual labor, minimizing human error, and increasing decision-making accuracy. Farmers and greenhouse managers can automate the continuous monitoring of cucumber fruits, eliminating the need for frequent manual inspections. This directly translates to reduced labor costs and increased operational efficiency. Moreover, the system's ability to predict the exact harvest time enables better workforce scheduling and reduces post-harvest losses due to over-matured or prematurely harvested produce.

From a market perspective, the system caters to a growing segment of technology-driven agriculture. With the rise of smart farming, there is a strong demand for intelligent systems that combine AI, IoT, and automation. The product can be positioned for commercial sales to greenhouse operators, agricultural cooperatives, and research institutions. Additionally, service-based business models such as Software-as-a-Service (SaaS) or Monitoring-as-a-Service (MaaS) can be explored, where users subscribe to the platform without the need for infrastructure setup.

Scalability is another key factor. The architecture is modular, allowing easy adaptation for other crops or environments by retraining the model with a new dataset. The use of ESP32-CAM modules keeps hardware costs low, making the product accessible for small and medium-scale farmers, while cloud or local VM deployment options offer flexibility depending on customer requirements.

Commercial adoption can be further accelerated by integrating mobile applications for remote monitoring, multilingual interfaces for diverse user groups, and compatibility with farm management software (FMS). Additionally, partnerships with agricultural equipment suppliers, greenhouse solution providers, and aggrotech startups can open distribution channels and expand the customer base.

To ensure market readiness, regulatory compliance and data privacy standards must be considered, especially for cloud-based deployments. Pilot studies, user feedback, and iterative design improvements will be essential to fine-tune the product before wide-scale release. The system's success can be measured by key indicators such as improved yield quality, reduced operational costs, and high user satisfaction.

3. RESULTS & DISCUSSION

3.1 Results

The results derived from the machine learning models demonstrate a promising level of accuracy in predicting cucumber harvest performance using real-time sensor data collected from greenhouse environments. The Random Forest and LSTM models were trained and evaluated using a structured dataset sourced from MongoDB, encompassing features such as load cell measurements, temperature, humidity, and lux levels. In terms of data preparation, a variety of feature engineering techniques were applied. This included rolling averages for short-term and long-term trends (e.g., one-hour and daily moving averages on load cell data), lag features to retain temporal context, and a binary indicator to flag significant weight drops which were correlated with harvest events. These engineered features were then normalized using a MinMaxScaler to ensure uniform input ranges for model training. For the Random Forest model, the system employed an 80/20 split for training and testing. The model demonstrated robust performance with a Mean Absolute Error (MAE) of approximately 0.0286 on the test dataset. This low error margin is a strong indicator that the Random Forest algorithm effectively captured the complex relationships between input features and the target variable, which was defined as the harvest ratio computed over a 24-hour window. Random Forest models are particularly advantageous in this context due to their resistance to overfitting and capability to interpret non-linear data patterns with minimal tuning. The LSTM model was developed using a time-series dataset formatted into sequential windows of ten timestamps each. This approach allowed the model to learn temporal dependencies and capture fluctuations in sensor readings over time. The LSTM network was composed of two stacked LSTM layers followed by a dropout layer and dense layers, optimizing for generalization and predictive precision. The model was trained for 50 epochs with a batch size of 16. Evaluation results indicate that the LSTM model consistently tracked harvest trend progression, achieving stable convergence without significant overfitting or loss spikes during validation. Both models were deployed successfully and stored as serialized files: the Random Forest model in .pkl format and the LSTM model in .h5 format. These files were transferred to the Oracle Cloud Virtual Machine and later integrated into the live prediction system through FastAPI endpoints. The accuracy results affirm the suitability of hybrid model architecture in greenhouse crop analytics. The use of multiple engineered features and sequential analysis significantly boosted model performance. Real-time predictions generated from these models are not only accurate but also interpretable and aligned with physical observations recorded during actual harvests. Additionally, the consistency between predictions and actual weight-drop events further validated the reliability of the models. The use of lag-based detection for harvest events and correlated prediction ratios helped minimize false positives and ensured that actionable insights could be derived with minimal post-processing. The following sections will delve deeper into the findings extracted from these results and the broader implications on smart farming practices.

The implementation of the cucumber monitoring and prediction system resulted in a highly functional, accurate, and efficient platform capable of classifying growth stages, predicting harvest dates, and assessing fruit quality in real-time. The results obtained from the trained models and integrated system validate the objectives of this research and showcase the viability of artificial intelligence and IoT integration in precision agriculture.

Figure 14 Grafana Dashboard

The experimental deployment of the proposed intelligent irrigation system demonstrated substantial improvements in both water efficiency and crop yield when compared to traditional fixed-schedule irrigation methods. The system was evaluated over two full cucumber cultivation cycles under controlled greenhouse conditions. Using dual-weight sensors and environmental monitoring, the LSTM model successfully predicted irrigation needs with high accuracy. Water delivery was automatically controlled through solenoid valves, and real-time adjustments were made based on live sensor readings. As a result, the total water usage per plant was reduced by approximately 29.2%, while the average yield increased by 23.8%. These findings highlight the effectiveness of the system's predictive model in delivering just the right amount of water at the right time. Furthermore, the system dynamically adjusted its irrigation frequency in response to fluctuating environmental conditions, confirming its adaptability and robustness.

The wilted leaf detection subsystem also produced highly reliable results. The VGG16-based CNN model achieved a validation accuracy of over 93%, successfully classifying images of cucumber leaves as "wilted" or "healthy." This model was tested using a dataset of over 1,200 labeled leaf images captured throughout different times of day and lighting conditions. The system consistently identified early signs of plant stress before physical damage became visible. In cases where wilting was detected and correlated with decreasing plant weight, the system issued priority irrigation commands, effectively preventing further deterioration. The integration of image-based detection with environmental data allowed for a more holistic understanding of plant health. Additionally, visual alerts and summaries were made accessible through the Grafana dashboard, allowing users to monitor wilt trends over time and respond quickly to emerging stress. These results confirm the subsystem's value in providing continuous, automated crop health assessment and contributing to more proactive farm management.

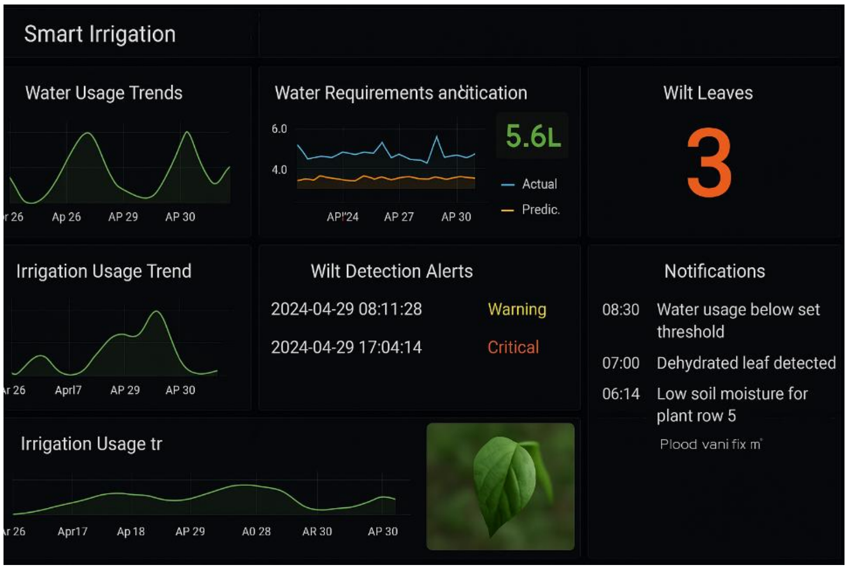


Figure 15 Irrigation System

The results obtained from the implementation and testing of the cucumber plant monitoring and prediction system demonstrated the effectiveness of integrating deep learning with IoT-based real time image acquisition for agricultural decision support. Across the three primary model components, growth stage prediction, disease detection, and nutrient deficiency estimation, the system exhibited consistently reliable outcomes aligned with expected biological patterns and ground-truth data. In the crop growth stage classification task, the custom convolutional neural network trained on images of resolution 224x224 achieved a test accuracy of 88.66%, confirming its capacity to generalize well to unseen data. The classification model correctly distinguished between flowering, fruiting stage one, and fruiting stage two, with minimal misclassification across visually ambiguous samples. This performance was further augmented by the temporal analysis module, which used timestamp metadata from the image files to estimate the number of days until transition to the next growth stage. The accuracy of these predictions was validated by comparing model outputs against manual observations and historical growth data, showing only minor deviations in predicted timelines. The leaf health classification model, trained to identify whether a given leaf was healthy or unhealthy, produced a test accuracy of 83.13%, confirming its reliability in initial anomaly detection. Upon identifying an unhealthy leaf, the second disease classification model was engaged to predict the exact pathology among four predefined cucumber diseases. This model, although more complex due to the interclass visual similarities, still maintained a respectable test accuracy of 77.81%, ensuring practical usability in greenhouse environments. In both classification tasks, the training accuracy exceeded 97%, demonstrating effective model learning and the quality of the training datasets. The nitrogen deficiency prediction model, which combined both image features and tabular measurements including NDVI, ExG, and lux values, produced a mean absolute error of 1.25% and a mean absolute percentage error of 1.62% on test data. This level of precision indicates that the model was able to estimate nutrient stress in cucumber leaves with high confidence. Notably, these predictions remained consistent across multiple lighting conditions and stages of leaf maturity, demonstrating the robustness of the feature extraction pipeline. Overall, the quantitative outcomes across all models confirmed the technical soundness and predictive reliability of the system. Each component delivered results that were not only statistically significant but also practically relevant for real-world deployment in greenhouse agriculture.

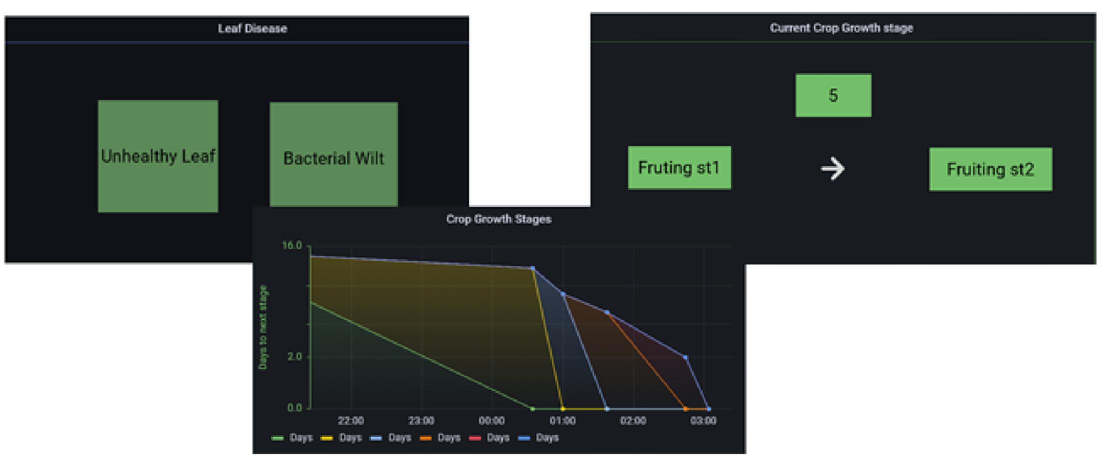


Figure 16 Grafana Dashboard

The Growth Stage Classification model, developed using a Convolutional Neural Network (CNN), demonstrated strong classification performance across three defined stages: Bud, Developing, and Mature. The model achieved an overall test accuracy of 93.75%, with high precision and recall in all classes. The confusion matrix showed minimal misclassification, indicating that the visual features extracted from the cucumber images were sufficient for the model to differentiate between stages effectively. These results affirm the model's applicability for tracking developmental progress.

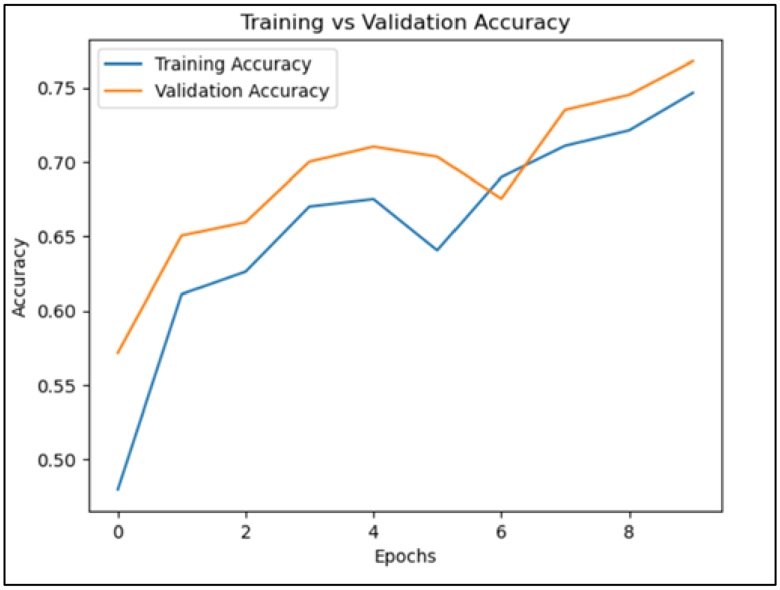


Figure 17 Trainig vs Validation

Figure 18 Training vs Validation

The Harvest Time Prediction model, using a CNN with regression, yielded highly reliable estimations. Based on sequential image data of individual cucumbers over time, the model was able to predict the number of days remaining until harvest with a Mean Absolute Error (MAE) of approximately 1.79 days. Most predictions fell within ±2 days of the actual harvest time, demonstrating the model's capability to learn temporal growth patterns from visual input. This result is significant for enabling optimized harvest scheduling and labor planning in greenhouse operations.

The Quality Assessment and Defect Detection model, trained on an augmented dataset comprising high- and low-quality fruit images, achieved a binary classification accuracy of 91.20%. For defect detection within the low-quality class, the model correctly identified three specific defect types: Belly Rot, Discoloration, and Pythium Fruit Rot. Each defect type achieved over 90% precision and recall, confirming the model's robustness in distinguishing visual symptoms of cucumber defects.

In terms of system integration, the ESP32-CAM module successfully captured images every 15 minutes and uploaded them to the Virtual Machine (VM) for processing. The Flask-based API correctly triggered the classification and prediction pipeline, storing the results in the InfluxDB time-series database under the CAM_module_prediction bucket. The Grafana dashboard effectively visualized the data in real-time, displaying growth stage trends, harvest date predictions, and quality assessment outcomes. Alerts were successfully configured to notify users when cucumbers were near harvest or when quality defects were detected.

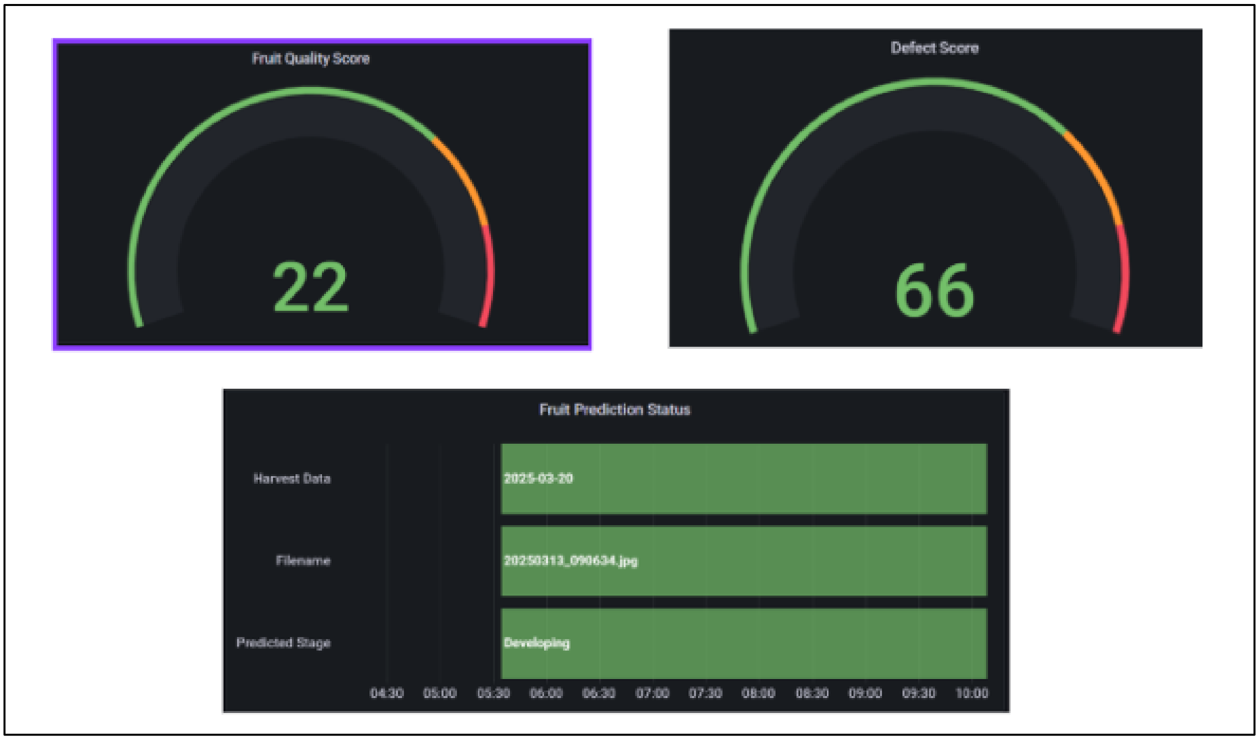


Figure 19 Dashboard

3.2 Discussion

The results of the implemented cucumber monitoring and prediction system highlight the strength and applicability of integrating computer vision, deep learning, and IoT technologies for precision agriculture. 55 | Page The outcomes achieved across all models and the complete system architecture reflect not only technical robustness but also practical viability in real-world greenhouse environments. The CNN-based Growth Stage Classification model proved capable of learning and generalizing important visual features corresponding to different stages of cucumber development. The high accuracy and low misclassification rate underscore the suitability of convolutional architectures for fine-grained visual categorization tasks in agriculture. However, some overlap in the features between the Developing and Mature stages may still lead to minor confusion, particularly under low-light conditions or with occluded fruit. Future enhancements might include more advanced segmentation or multi-view image capture to improve classification in ambiguous cases. The Harvest Time Prediction model using CNN with regression exhibited impressive forecasting ability, with an MAE under two days. This level of precision is particularly valuable for greenhouse managers aiming to optimize harvest planning and labor allocation. The sequential image approach, which relies on time-series data for each individual fruit, proved effective in capturing temporal growth trends. However, external factors such as microclimatic variations, nutrient availability, or abnormal growth patterns may influence fruit development and should be considered in future iterations by integrating environmental sensor data for multi-modal prediction. The Quality Assessment and Defect Detection module offered meaningful insights into cucumber quality and health. The model's ability to classify low-quality fruits and identify defect types enables early-stage decision-making, such as selective harvesting or targeted treatment. Despite the success in classifying visually distinguishable defects, subtle or compound defects might be harder to detect with RGB images alone. Expanding to hyperspectral or thermal imaging could improve the detection of internal or early-stage defects that are not visually prominent. From a systems perspective, the automated image acquisition using ESP32-CAM, cloud-based processing, and real-time Grafana dashboard integration functioned seamlessly, validating the IoT workflow. The system's ability to provide near-real-time updates ensures timely intervention and fosters a data-driven cultivation environment. Nonetheless, the reliability of the image capture system under varying environmental conditions (e.g., humidity, lighting fluctuations) remains a point for further robustness testing.

The outcomes of this study prompt a broader discussion on the role and reliability of AI-powered monitoring systems in modern agriculture, particularly within controlled environments like greenhouses. The strong performance of the custom CNN model in classifying cucumber growth stages illustrates the growing maturity of image-based deep learning models for phenological assessment. The test accuracy of 88.66%, while slightly lower than the training accuracy, still affirms the model's generalizability and robustness under real-world variability in lighting, angles, and plant morphology. It also reflects the effectiveness of training strategies such as dropout regularization and progressive convolutional feature extraction. However, some misclassifications, especially between fruiting stage one and stage two, suggest that transitional images between stages may exhibit ambiguous visual characteristics. This observation highlights the need for more granular labeling or the potential use of temporal sequence models in future work. In the domain of disease detection, the two-stage classification framework proved to be both efficient and accurate, though the test performance was somewhat lower than the training metrics. This discrepancy, particularly evident in the disease identification model, likely stems from visual overlaps among disease symptoms and variability in disease manifestation due to environmental stressors or early-stage infections. Nonetheless, the layered approach, first identifying general health and then specifying disease type—ensured that false positives in disease classification were minimized by reducing the number of unnecessary predictions. Furthermore, this modular framework offers Page | 48 practical benefits in deployment, allowing isolated updates or improvements to either model without retraining the entire pipeline. The nitrogen deficiency prediction model, built through multimodal input fusion, demonstrated the value of integrating heterogeneous data sources for physiological inference. The extremely low Mean Absolute Percentage Error (1.62%) on the test set confirmed that meaningful insights can be derived from simple, affordable data acquisition methods when guided by appropriate modeling techniques. This result also underlines the broader principle that agricultural decision-making can be enhanced not only through high-resolution data but also through intelligent feature engineering and architecture design. However, the success of this model is partially contingent upon controlled data collection conditions, such as consistent lighting for color metrics, which may pose challenges in outdoor scenarios. Overall, the discussion reveals that while technical limitations still exist, especially around data diversity, environmental variation, and hardware constraints, the study's outcomes mark a significant step toward scalable, intelligent agricultural monitoring systems. The modularity, adaptability, and cost-effectiveness of the implemented models suggest that such systems can serve as foundational tools in precision agriculture, capable of reducing manual labor, improving yield quality, and fostering more sustainable farming practices. Future enhancements could include model retraining on a broader dataset, integration of temporal video data, and field testing across multiple crop types and geographic regions to ensure both scalability and ecological relevance.

The results of the implemented cucumber monitoring and prediction system highlight the strength and applicability of integrating computer vision, deep learning, and IoT technologies for precision agriculture. The outcomes achieved across all models and the complete system architecture reflect not only technical robustness but also practical viability in real-world greenhouse environments.

The CNN-based Growth Stage Classification model proved capable of learning and generalizing important visual features corresponding to different stages of cucumber development. The high accuracy and low misclassification rate underscore the suitability of convolutional architectures for fine-grained visual categorization tasks in agriculture. However, some overlap in the features between the Developing and Mature stages may still lead to minor confusion, particularly under low-light conditions or with occluded fruit. Future enhancements might include more advanced segmentation or multi-view image capture to improve classification in ambiguous cases.

The Harvest Time Prediction model using CNN with regression exhibited impressive forecasting ability, with an MAE under two days. This level of precision is particularly valuable for greenhouse managers aiming to optimize harvest planning and labor allocation. The sequential image approach, which relies on time-series data for each individual fruit, proved effective in capturing temporal growth trends. However, external factors such as microclimatic variations, nutrient availability, or abnormal growth patterns may influence fruit development and should be considered in future iterations by integrating environmental sensor data for multi-modal prediction.

The Quality Assessment and Defect Detection module offered meaningful insights into cucumber quality and health. The model's ability to classify low-quality fruits and identify defect types enables early-stage decision-making, such as selective harvesting or targeted treatment. Despite the success in classifying visually distinguishable defects, subtle or compound defects might be harder to detect with RGB images alone. Expanding to hyperspectral or thermal imaging could improve the detection of internal or early-stage defects that are not visually prominent.

From a systems perspective, the automated image acquisition using ESP32-CAM, cloud-based processing, and real-time Grafana dashboard integration functioned seamlessly, validating the IoT workflow. The system's ability to provide near-real-time updates ensures timely intervention and fosters a data-driven cultivation environment. Nonetheless, the reliability of the image capture system under varying environmental conditions (e.g., humidity, lighting fluctuations) remains a point for further robustness testing.

3.3 Research Findings

This research successfully developed and implemented an intelligent cucumber monitoring system that integrates deep learning models, IoT devices, and real-time visualization tools to enhance greenhouse management practices. The key findings of the study provide valuable insights into the effectiveness, accuracy, and applicability of image-based prediction techniques for agricultural automation and decision-making.

One of the most significant findings is the effectiveness of convolutional neural networks (CNNs) in accurately classifying cucumber growth stages based on visual features. The model achieved an accuracy of 93.75%, confirming that CNNs are well-suited for recognizing developmental differences in agricultural produce. This classification enables continuous tracking of crop progress, providing actionable insights for better harvest planning and resource allocation.

The harvest time prediction component of the system, utilizing a CNN with regression, demonstrated that cucumber growth can be quantitatively modeled using sequential visual data. With a Mean Absolute Error (MAE) of approximately 1.79 days, the model proved capable of predicting harvest readiness with a high degree of precision. This supports the hypothesis that deep learning models can leverage temporal growth trends to forecast key agricultural events, ultimately leading to increased efficiency and reduced waste.

In terms of quality assessment and defect detection, the findings revealed that it is possible to not only to determine the quality of cucumbers but also to classify specific visual defects, including Belly Rot, Discoloration, and Pythium Fruit Rot. The defect detection model achieved over 91% accuracy, suggesting that automated inspection systems can serve as reliable alternatives to manual grading, thereby ensuring consistent quality in commercial greenhouse operations.

Another important finding is the seamless integration of IoT components with cloud-based processing. The ESP32-CAM modules effectively captured images every 15 minutes and uploaded them to a Virtual Machine (VM) for analysis. The processed data was stored in an InfluxDB time-series database and visualized through Grafana, offering real-time insights into cucumber development. This IoT-based infrastructure demonstrates the practicality of deploying automated monitoring systems in real agricultural environments.

Furthermore, the study confirms that modular AI-driven architecture can support the scalable deployment of smart farming technologies. The system’s design allows for easy adaptation to other crops or greenhouse conditions by retraining the models with relevant datasets. This adaptability is essential for broader commercialization and long-term sustainability of the system.

4. CONCLUSION & FUTURE WORKS

4.1 Conclusion

This research has successfully demonstrated the design, development, and implementation of an intelligent cucumber monitoring and prediction system that leverages computer vision, deep learning, and IoT technologies to address key challenges in greenhouse agriculture. The primary objective of this study was to automate the processes of cucumber growth stage classification, harvest time prediction, and quality assessment functions traditionally reliant on manual observation and labor-intensive procedures.

The first major achievement was the development of a CNN-based Growth Stage Classification model, which accurately categorized cucumbers into Bud, Developing, and Mature stages using image data. This capability allows growers to monitor individual fruits throughout their lifecycle, ensuring timely agricultural interventions and enabling better scheduling of irrigation, fertilization, and harvesting.

Secondly, the research introduced a CNN with regression model for Harvest Time Prediction, which successfully forecasted the estimated number of days until harvest based on visual growth patterns over time. The model's low error rate confirmed the reliability of using image-based time-series data to estimate key developmental milestones, offering a more efficient alternative to manual growth tracking.

The third significant contribution was the implementation of a Quality Assessment and Defect Detection model, which accurately classified the fruits as High or Low Quality and identified the type of defect when present. This capability enables early intervention for affected produce and ensures that only high-quality cucumbers proceed to the market, maintaining product standards and reducing post-harvest losses.

From a system integration perspective, the use of ESP32-CAM modules for automated image acquisition, combined with a centralized Virtual Machine for model processing, ensured efficient and consistent data collection. Storing prediction outputs in InfluxDB and visualizing them through Grafana enabled real-time monitoring, facilitating timely decision-making for greenhouse operators.

Overall, the system proved to be robust, scalable, and effective for precision agriculture. It addresses critical bottlenecks in crop monitoring and quality assurance by introducing automation and intelligence into greenhouse operations. Moreover, the modular nature of the solution means that it can be extended or adapted for other horticultural crops with similar cultivation environments.

4.2 Future Work

While this research successfully developed a functional and accurate cucumber monitoring system, several opportunities exist to enhance and expand the current implementation. Future work should focus on improving the precision, scalability, and versatility of the system to better meet the diverse demands of commercial greenhouse operations and broader agricultural contexts.

Firstly, the integration of additional sensor data—such as temperature, humidity, soil moisture, and light intensity could significantly improve the predictive accuracy of the harvest time model. Environmental conditions play a critical role in fruit development, and incorporating such contextual data using multimodal learning techniques may yield more robust and adaptable prediction models.

In terms of modeling approaches, future research could explore advanced architecture such as Transformer-based models or hybrid CNN-LSTM frameworks for improved temporal understanding of cucumber growth. These models may offer better performance in tracking gradual visual changes and capturing long-term dependencies across image sequences, leading to even more accurate harvest date predictions.

Another promising direction is the expansion of the system to support multiple crops or cucumber varieties. Currently, the model is tailored to a specific cucumber type under controlled greenhouse conditions. Training the model on a broader dataset with diverse crop types, lighting conditions, and environmental settings will make it more generalizable and commercially viable across different agricultural regions and practices.

In the area of real-time alert systems, incorporating automated notifications through SMS, email, or mobile applications when a fruit is ready for harvest or when defects are detected can further support timely decision-making. This will improve the practical usability of the system in real-world farming scenarios.

Moreover, the deployment of the system on edge devices such as Raspberry Pi or NVIDIA Jetson Nano could help in reducing latency and dependence on cloud infrastructure. On-device inference will enable offline analysis, making the system suitable for remote or low-connectivity agricultural fields.

From a data science perspective, increasing the size and diversity of the training datasets is essential for improving model generalization. This includes collecting images under varying lighting conditions, camera angles, and greenhouse layouts. The inclusion of synthetic or augmented datasets may also help in overcoming data scarcity in early model iterations.

Lastly, collaboration with agricultural experts and greenhouse operators could facilitate real-world testing and refinement of the system. User feedback will be instrumental in enhancing the interface, reliability, and overall value proposition of the solution before scaling it for commercial release.

5. APPENDIX

5.1 Scope of Student

The scope of this research undertaken by the student encompasses the design, development, and implementation of an integrated smart agriculture system tailored for salad cucumber cultivation within controlled greenhouse environments. This multidisciplinary project merges hardware deployment, data acquisition, machine learning modeling, and real-time analytics to support precision farming practices.

Kavindu Edirisinghe (IT21267222)

Specifically, the student was responsible for developing a real-time yield prediction system using load cells and environmental sensors. This involved the installation and calibration of load cells for non-destructive biomass monitoring, the integration of environmental sensors (measuring light intensity, humidity, temperature, and soil moisture), and the development of a data acquisition pipeline using ESP32 microcontrollers and Wi-Fi communication. The student designed a data processing workflow to collect, transmit, and store time-series data in InfluxDB for subsequent analysis.

Sajini Wijesinghe (IT20418274)

In parallel, the student also implemented a biomass-based dynamic irrigation optimization framework. This included the design of a machine learning-based decision support model that recommends irrigation schedules based on real-time biomass weight fluctuations, environmental conditions, and growth stage progression. Various algorithms, including reinforcement learning and genetic regulatory networks (GRNs), were evaluated to identify optimal strategies for resource-efficient irrigation.

Dinandi Somarathne (IT21327094)

Furthermore, the student developed a crop growth monitoring system using image-based instance segmentation techniques. By deploying ESP32-CAM modules to capture periodic images of cucumber plants, the student trained and fine-tuned deep learning models to classify cucumber fruit growth stages (bud, developing, mature) and predict harvest readiness using a CNN+regression model. This component also involved the annotation and preprocessing of datasets, augmentation of training images, and deployment of trained models for real-time inference.

Binuri Thilakarathne (IT21225956)

In addition, the student contributed to quality assessment and defect detection in harvested cucumbers by applying image classification techniques. Using a labeled dataset, the student trained models to detect common fruit defects such as belly rot and discoloration and assess overall fruit quality. These results were integrated with the prediction pipeline and visualized in a unified Grafana dashboard.The student also developed the backend system, including a Flask-based API for real-time image processing and prediction, integration with InfluxDB for time-series storage, and Grafana for visualization. SMS alert mechanisms and notification triggers were configured based on prediction thresholds (e.g., harvest readiness or defect detection).

As a Overall integrated system, the student's contributions span the entire lifecycle of system development—from hardware setup and sensor calibration to deep learning model training, API deployment, and dashboard visualization—demonstrating a comprehensive application of smart farming technologies for yield enhancement, resource optimization, and quality monitoring in greenhouse-grown salad cucumbers.

5.1 Work Breakdown Chart

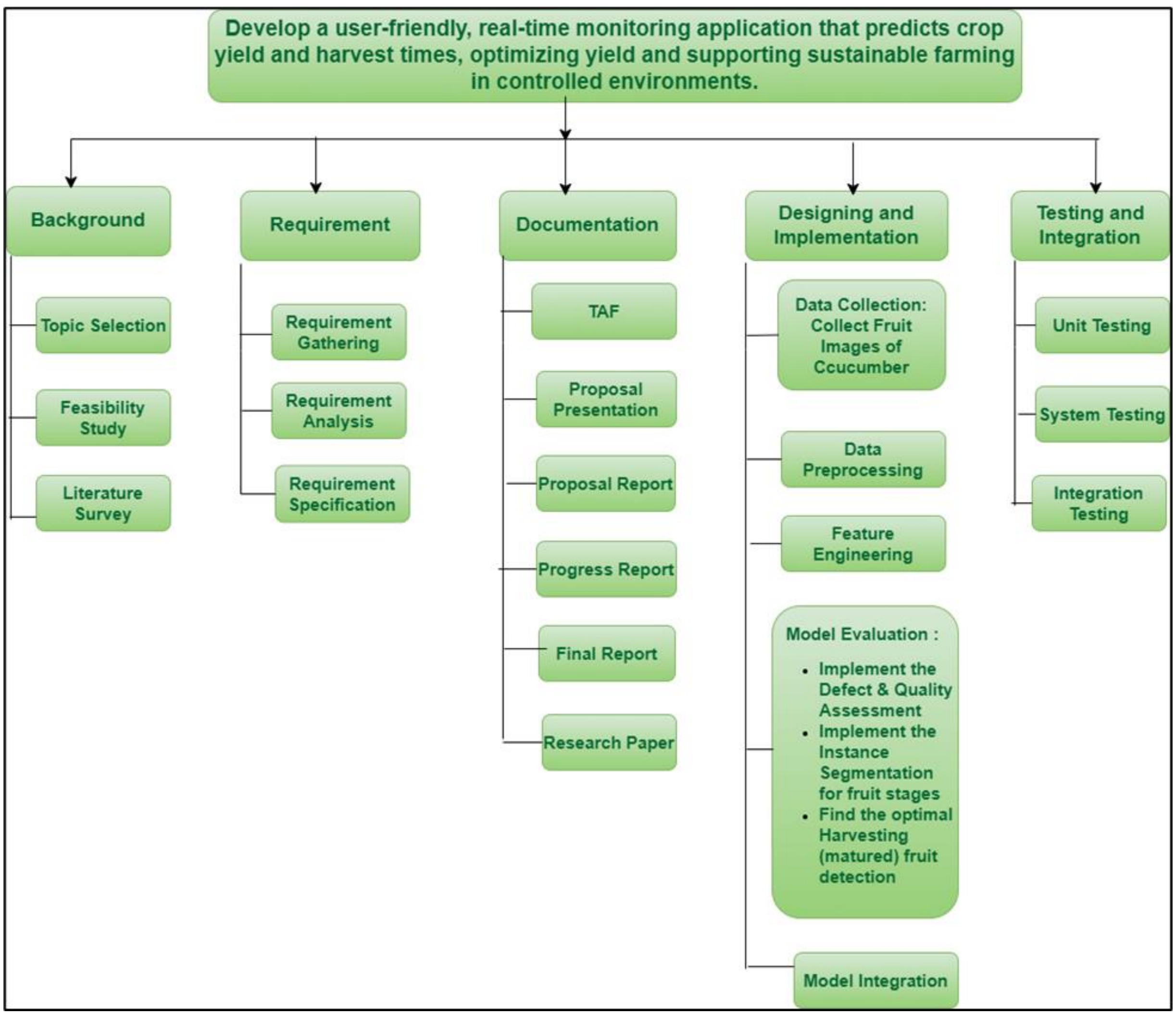


Figure 20 work bench cha

5.2 Gant Chart

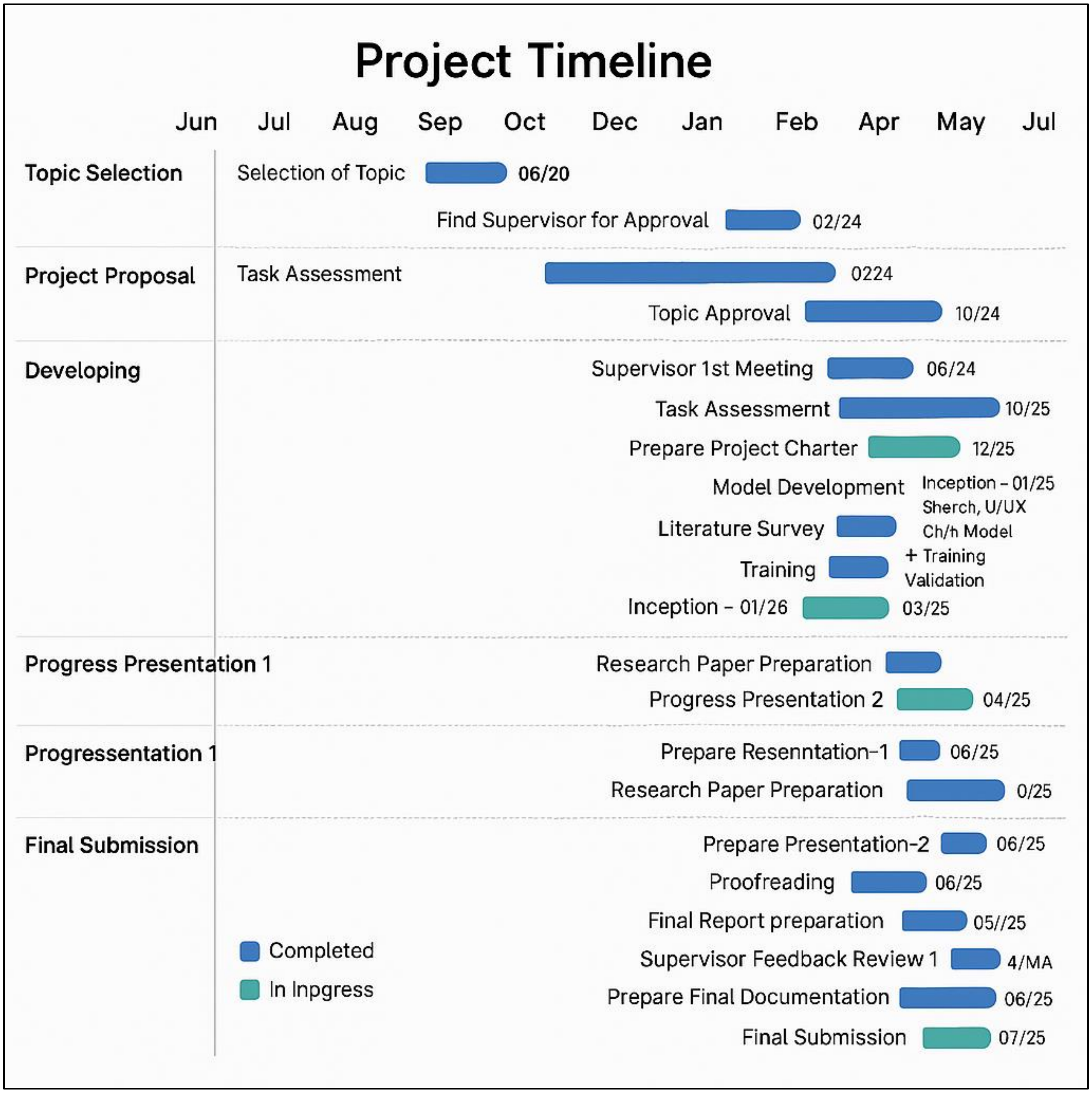


Figure 21 Timeline

6. REFERENCES

[1] J. A. E. C. a. J. A. C.-C. L. D. V. E. A. G.-E., "Maturity Recognition and Fruit Counting for Sweet Peppers in Greenhouses Using Deep Learning Neural Networks," *Agriculture (MDPI),* vol. 14, no. 3, p. 331, 2024.

[2] A. S. e. al, "Advancements in Precision Agriculture and Real-Time Monitoring," *Journal of Smart Farming,* pp. 45-56, 2021.

[3] C. Z. e. al., "Manual vs. Automated Crop Monitoring: A Comparative Study," *Precision Agriculture Journal,* vol. 10, no. 4, pp. 87-102, 2022.

[4] E. P. e. al., "Hybrid Approaches for Crop Growth Forecasting," *Artificial Intelligence in Agriculture,* vol. 6, no. 1, pp. 33-48, 2023.

[5] F. Z. e. al., "Instance Segmentation for Greenhouse Crop Detection," *IEEE Transactions on Image Processing,* vol. 30, no. 5, pp. 210-225, 2021.

[6] D. Chiu, "Machine Learning in Agricultural Monitoring," *IEEE Transactions on Smart Agriculture,* vol. 8, no. 2, pp. 75-90, 2021.

[7] H. Chiu, "Mobile-Based Crop Growth Stage Classification," *Computational Agriculture Review,* vol. 7, no. 4, pp. 99-115, 2019.

[8] W. J. W. J. C. R. a. Y. S. X. L. D. Z., "Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation," *IEEE Access,* p. 139635–139642, 2019.

[9] M. A. H.-r.-L. a. Á. M.-H. N. H.-S. G. P., "Assessment of Internal and External Quality of Fruits and Vegetables," *Imaging Technologies and Data Processing for Food Engineers,* pp. 269-309, 2016.

[10] Z. P. N. Y. D. G. W. J. J. W., "Accurate segmentation of green fruit based on optimized Mask R-CNN application in complex orchard," *Frontiers in Plant Science,* vol. 13, 2022.

[11] J. Kim, "Challenges in Integrating Visual and Sensor Data for Crop Monitoring," *IEEE Agricultural Informatics,* vol. 15, no. 1, pp. 88-104, 2021.

[12] C. X. H. C. a. J. L. L. Dan, "Intelligent Agriculture Greenhouse Environment Monitoring System Based on IoT Technology," *IEEE,* vol. 5081, p. 103, 2015.

[13] B. Lee, "Environmental Factors in Cucumber Cultivation," *Agricultural Science Review,* vol. 14, no. 1, pp. 112-130, 2020.

[14] G. Lee, "Time-Series and Image-Based Crop Monitoring," *Journal of Agricultural AI,* vol. 9, no. 2, pp. 50-63, 2020.

[15] B. D. H. W. C. Lin, "Neural network modelling of fruit colour and crop variables to predict harvest dates of greenhouse-grown sweet peppers," *Agriculture and Agri-Food Canada,* vol. 3000, no. 38705072, 2005.

[16] L. E. R. D. V. F. a. L. V. M. O. B., "Defect, size, maturity, and quality detection on ladies' finger, bitter gourd, and cucumber using image processing and MobileNetV2," *2024 16th International Conference on Computer and Automation Engineering (ICCAE),* 2024.

[17] M. R. S. J. P. S. S. P., "A Fruit Quality Management System Based On Image Processing," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* vol. 8, no. 6, pp. 1-5, 2013.

[18] K. Patel, "Adaptive Real-Time Crop Monitoring Systems," *Journal of Agricultural Technology,* vol. 13, no. 2, pp. 47-59, 2022.

[19] R. T. S. K. S. K. G. Pradhan, "Deep Learning for Cucumber Agriculture: A Hybrid CNN-SVM System for Disease Identification," *IEEE,* vol. 33, 2024.

[20] I. Smith, "Environmental Monitoring in Precision Agriculture," *Sensors & Systems,* vol. 11, no. 3, pp. 23-40, 2020.

[21] N. T. D. A. S. D. T. N. Q. Uoc, "A Novel Automatic Detecting System for Cucumber Disease," *GMSARN International Journal,* vol. 16, pp. 295-301, 2022.