

Article

Not peer-reviewed version

Cloud Computing Framework for Space Farming Data Analysis

Adrian Genevie Janairo , [Ronnie Concepcion II](#) , [Arvin Fernando](#) , [Marielet Guillermo](#) *

Posted Date: 3 January 2025

doi: 10.20944/preprints202501.0120.v1

Keywords: cloud computing; computer vision; crop germination; CubeSat; Internet of Space Things; microgravity; space farming



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Cloud Computing Framework for Space Farming Data Analysis

Adrian Genevie Janairo, Ronnie Concepcion II, Arvin Fernando and Marielet Guillermo *

Gokongwei College of Engineering, De La Salle University, Manila 1004, Philippines

* Correspondence: marielet.guillermo@dlsu.edu.ph

Abstract: The study presents a system framework by which cloud resources are utilized to analyze crop germination status in a 2U CubeSat. The research aims to address the onboard computing constraints in nanosatellite missions to boost space agricultural practices. Through the ESP-NOW technology, communications between ESP-32 modules were established. The corresponding sensor readings and image data were securely streamed through AWS IoT to an ESP-NOW receiver and Roboflow. Real-time plant growth predictor monitoring was implemented through the web application provisioned at the receiver end. On the other hand, sprouts on germination bed were determined through the custom-trained Roboflow computer vision model. The feasibility of remote data computational analysis and monitoring for a 2U CubeSat, given its minute form factor, was successfully demonstrated through the proposed cloud framework. The germination detection model resulted to an mAP, precision, and recall of 99.5%, 99.9%, and 100.0% respectively. The temperature, humidity, heat index, LED and Fogger states, and bed sprouts data were shown in real-time through a web dashboard. With this use case, immediate actions can be done accordingly when abnormalities occur. The scalability nature of the framework allows adaptation to various crops to support sustainable agricultural activities in extreme environments such as space farming.

Keywords: cloud computing; computer vision; crop germination; CubeSat; Internet of Space Things; microgravity; space farming

1. Introduction

The rapid expansion of space exploration into new areas has created a growing need for new technologies and methods that can support human life in space. In the past, space exploration was mostly controlled by big government agencies like NASA and the European Space Agency (ESA). However, this is changing. The development of small, affordable satellites called CubeSats has played a big role in this shift. CubeSats are small, typically measuring 10x10x10 cm, and its simple and modular design makes it accessible to a wide range of groups, including universities, private companies, and even smaller countries. This has opened up space research to more people, allowing a wider variety of participants to contribute to space science and exploration [1].

One of the most promising applications of CubeSats lies in astrobiology, specifically the study of plant biology in microgravity. As humanity looks towards long-term space missions, including potential inhabitation on the Moon, Mars, and beyond, understanding the biological responses of plants in space environments is critical. Plants are fundamental to life support systems in space, providing oxygen, food, and psychological benefits to astronauts [2]. However, microgravity has been shown to influence various physiological processes in plants, including seed germination, growth, and reproduction. Research conducted on the International Space Station (ISS) has demonstrated that while seeds can germinate in microgravity, the absence of gravity affects the orientation, root growth, and overall morphology of the plants, raising questions about how best to support plant life in space [3]. To advance our understanding of seed germination in microgravity, it is essential to develop sophisticated experimental platforms that can replicate the environmental

conditions of space while providing the necessary support for biological studies. CubeSats, with their small size and modular design, offer an ideal platform for such experiments [4,5]. However, its payload standard attributes constrain onboard computing and storage functions. Thus, restraining precise and real-time control of environmental factors such as temperature, humidity, and light to sustain plant growth. To address this, there is a need to establish a framework that will minimize data handling tasks onboard while maximizing the satellite data that can be obtained for crop germination nanosatellite mission. Figure 1 illustrates the exploded view of a standard CubeSat subsystems [6] and its communications system structure to Earth through the ground stations [7]. Ground stations are physical infrastructures housing the necessary communications subsystems for transmitting, receiving, and decoding signals. Base or controller facilities are often stationed to perform more specific tasks according to their primary designation such as security, network, and weather monitoring.

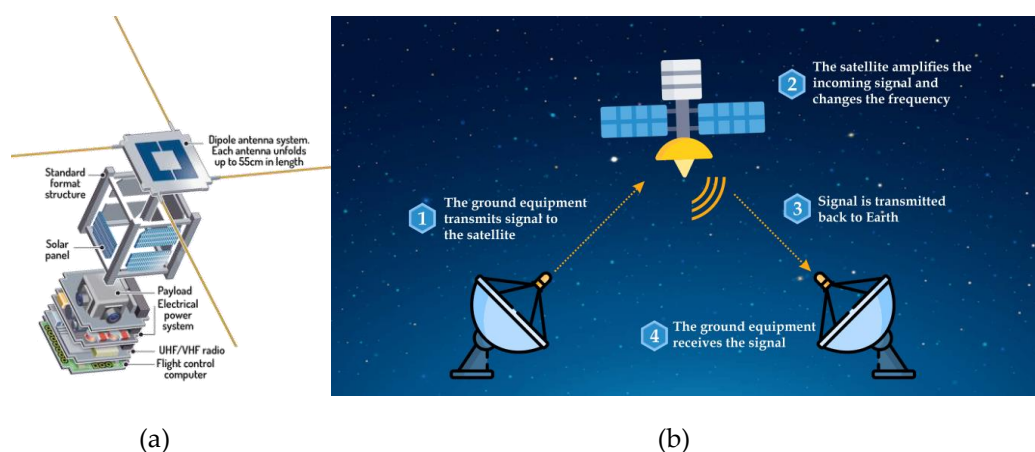


Figure 1. CubeSat (a) physical subsystems and (b) data communications to ground stations.

In this study, tasks specific to crop growth status and predictors condition monitoring are proposed to be neither on the payload nor the base stations, but on the cyberspace. Effectively, implementing a space-to-edge-to-cloud solution framework with provision to a potential Internet of Space Things application. This will be discussed further in the next section. The ARCHER (Agricultural CubeSat for Horticulture Experiments and Research) as shown in Figure 2 was used as the CubeSat for the experimental setup.



Figure 2. ARCHER (Agricultural CubeSat for Horticulture Experiments and Research) version 1.

Each specific objective listed below contributes strategically to bridge terrestrial and space-based monitoring systems for the reinforcement of the agricultural sector:

- Development of a webserver as a monitoring hub
- Establishment of edge-to-cloud communications using AWS IoT Core
- Data collection and transmission from 2U CubeSat using the ESP-NOW technology

- Development and deployment of a machine vision model for germination detection using Roboflow

Furthermore, the extension of the proposed framework to space-based applications underscores its potential to contribute to interdisciplinary research areas, including sustainable precision agriculture, environmental monitoring, and space science. By combining advanced technological solutions with practical utility, the study represents a significant contribution to the evolving landscape of smart monitoring systems.

1.1. Review of Related Studies

The convergence of Internet of Things (IoT) technologies with agricultural and space research has driven significant advancements in real-time monitoring and data collection. In agriculture, IoT systems enable precise tracking of environmental parameters such as temperature, humidity, and soil conditions, improving resource management and crop productivity [8]. Similarly, in space research, IoT technologies have become indispensable for managing experiments under the constraints of harsh conditions, limited access, and the need for real-time data transmission [9]. Innovations such as low-cost microcontrollers, cloud-based platforms, and machine learning models have expanded the potential for automation, accessibility, and data-driven insights [10–12]. This review explores the application of IoT systems in agricultural and space-based research, with a focus on CubeSat missions and the development of solutions for studying seed germination in microgravity. In agricultural settings, the adoption of low-cost and versatile microcontrollers, such as ESP32, has facilitated the development of robust, real-time monitoring systems [13]. Studies have shown that these systems efficiently collect and transmit environmental data, making them ideal for use in remote or resource-constrained environments [14,15]. Furthermore, the integration of remote access solutions like NGROK has expanded the accessibility of such systems, allowing users to retrieve data from any location with minimal infrastructure requirements [16]. The application of IoT in space science further underscores its versatility. CubeSat missions often rely on IoT-based systems to monitor internal conditions such as temperature, humidity, and light, ensuring that experiments onboard remain within operational parameters [17]. These systems also support the collection of biological data, such as plant growth metrics, enabling detailed analysis of processes like seed germination under microgravity conditions [18]. Cloud-based platforms such as AWS IoT have become instrumental in managing the large volumes of data generated by IoT devices [19]. These platforms provide secure storage, scalable data processing, and user-friendly real-time visualization and analysis dashboards. This capability benefits agricultural and space applications, where consistent data monitoring and secure storage are essential [20,21].

The development of CubeSats for space-based research presents significant opportunities and notable challenges, particularly in studying biological processes like seed germination in microgravity [22]. CubeSats, small, modular satellites that typically measure 10x10x10 cm per unit, offer a unique platform for conducting scientific experiments in space [23]. Their cost-effective, scalable design has opened doors for a wider range of organizations, including universities and private companies, to contribute to space research [23]. However, the limitations of CubeSat platforms must be carefully considered when it comes to biological research, such as studying plant growth and seed germination in microgravity. Microgravity significantly affects plant development, including seed germination, root growth, and overall morphology [24]. Without gravity, plants struggle to orient themselves, leading to altered root growth and nutrient and water distribution changes. The challenge is to design a system that can manage environmental factors such as temperature, humidity, and light and even simulate gravity or microgravity conditions. The 2U CubeSat, twice the size of the standard 1U CubeSat, offers more space for research payloads [25]. However, it is still limited in terms of overall volume and power capacity, making it difficult to incorporate comprehensive environmental controls and sensors. Despite these constraints, CubeSats remain valuable due to their small size, low cost, and ability to perform targeted experiments in space [26]. Yet, the harsh space environment—characterized by radiation, extreme temperatures, and

vacuum conditions—poses additional challenges to payload design [27]. These factors require that CubeSats be designed with durable materials and efficient power systems to ensure their functionality during long-duration missions. Advanced monitoring systems and power management techniques are critical to overcoming these limitations. For instance, integrating sensors for real-time data collection allows researchers to monitor plants' growth conditions during experiments [28]. With effective power management, it is possible to optimize the use of CubeSat resources and maintain the payload's functionality. These technologies enable continuous data collection despite the limited payload size and power. Understanding how plants behave in microgravity is essential for long-term space missions and technical considerations [24]. The insights gained from seed germination studies in microgravity can inform strategies for supporting space agriculture and ensuring food sustainability during extended space missions [29]. Furthermore, these findings can enhance agricultural practices on Earth, particularly in extreme or resource-limited environments.

Machine learning and vision models are vital in improving data analysis in CubeSat-based biological research [30]. These technologies can automate the analysis of large datasets generated from plant growth monitoring, making it easier to identify trends and anomalies that would otherwise be difficult to detect. Machine learning models can be trained to recognize subtle changes in plant health, while vision models can provide real-time visual assessments of seed germination and growth [31,32]. This approach increases the efficiency of data processing and enables more accurate and actionable insights. Machine learning (ML) and computer vision (CV) models, combined with cloud-based platforms like Roboflow, are increasingly used for crop germination detection and classification [33]. Roboflow enables the development of custom object detection models by training on labeled image datasets, allowing researchers to monitor and classify plant growth stages, including seed germination, using visual data. This is especially beneficial for space applications, where traditional monitoring methods are impractical, such as with CubeSat payloads in microgravity environments [34]. Roboflow's platform supports image analysis techniques like object detection and image segmentation, ideal for identifying subtle differences in germination stages [35]. By integrating this with CubeSat technology, real-time monitoring of plant growth in space becomes possible, providing essential data on seed behavior in microgravity. The cloud-based system offers remote monitoring and real-time feedback, allowing researchers to analyze plant health remotely [36]. As new images are collected, the detection models can continuously improve through adaptive learning, enhancing accuracy over time [37]. Additionally, the scalability of cloud-based platforms allows for easy application across different crops or growth conditions, making them versatile for space-based agriculture.

In conclusion, this review underscores the critical role of integrating IoT technologies, cloud-based platforms, and machine learning in overcoming the challenges inherent in space farming research. CubeSat missions, constrained by limitations in size, power, and accessibility, demand innovative approaches to facilitate real-time monitoring and analysis of biological experiments. By leveraging tools such as AWS IoT, NGROK, and Roboflow, researchers can implement scalable and efficient systems for automated germination detection and environmental control. These advancements not only enhance the functionality of CubeSat payloads for studying seed germination in microgravity but also contribute to the broader field of sustainable agricultural practices in extreme environments. Collectively, this research provides a robust framework for addressing current gaps in space-based biological studies, offering valuable insights for future explorations in space farming.

1.2. Proposed Framework for Space Farming Data Analysis

To achieve the objective of performing advanced crop growth monitoring on nanosatellites considering the payload constraints, the framework illustrated in Figure 3 is proposed. The framework is composed of four segments represented by a green dashed-line boxes with blue box identifier labeled: CubeSat, ground station, cloud services, and end user. These are end blocks utilizing different technologies with different purposes interconnected through IoT running on the cloud platform. The corresponding technology per block is marked in red box. The first two blocks

use ESP-NOW while the remaining blocks use HTTP. The orange boxes denote the functions of these blocks.

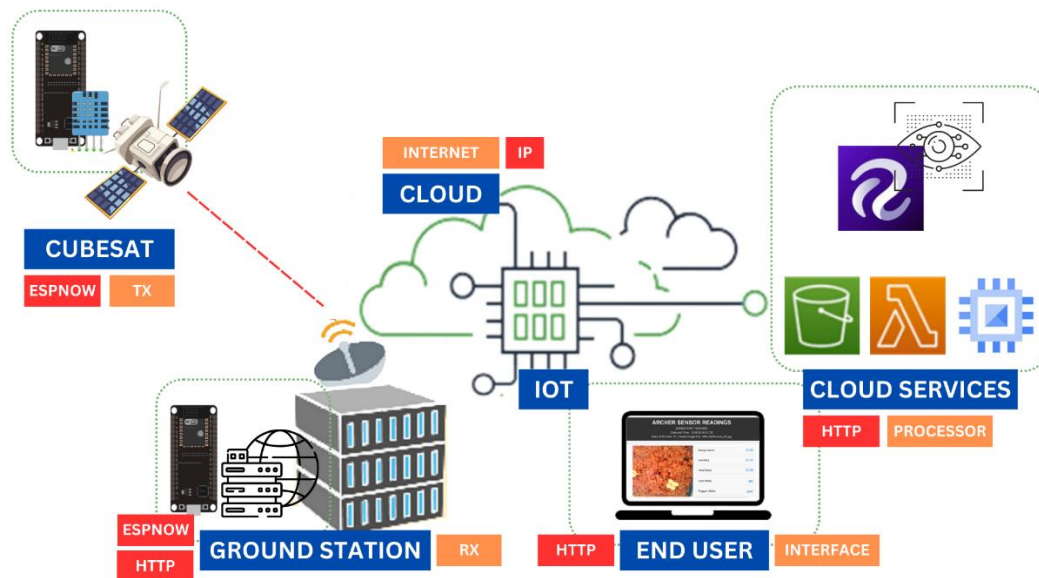


Figure 3. Space Farming Data Analysis using Cloud Computing Framework.

CubeSat block covers the nanosatellite mission standard subsystems such as imaging, power, and communications. The components specific to space farming are further explained in Chapter 2. The block does the transmission function wherein data captured through the onboard sensors are sent to the receiver end through the satellite dish at the ground station. The data is decoded in a way that the receiver will understand on a transport layer using ESP-NOW protocol. The ground station block covers the receiving antenna and its communications subsystems to convert RF signals into data format readable in the data link layer. For simplicity purposes, the block was labeled ground, but it also signifies other stations such as base, access gateway, or customer premise which physically situated on the ground to perform various functions. The block does the receiving and preliminary decoding functions. It uses the ESP-NOW protocol for basic sensor reading and image store-and-forward tasks and http protocol for in-house web server that will allow connection with http client requests from the end user block.

The data from the ground station block is passed onto the cloud where the IoT technology resides. It does internetwork or interconnectivity of devices from all end blocks using IP protocol. Depending on the data type, TCP or UDP can be utilized. Cloud services block does the storing and computing functions. It also does application programming interfacing between different cloud service providers and applications. In this case, the Roboflow which houses the computer vision model-related tasks as discussed further in the next Chapter. End user block does the http request to the web server to access and to visualize the crop germination monitoring application. Both the cloud service and end user blocks have two-way communication with the IoT block.

The proposed framework can be further reconfigured to advance the nanosatellite mission further into an IoST (Internet of Space Things) structure such that multiple CubeSat with different missions can be deployed in space communicate with each other. In essence, converting these to active satellites with the ability to do decision making and control onboard. See Figure 4 for the illustration.

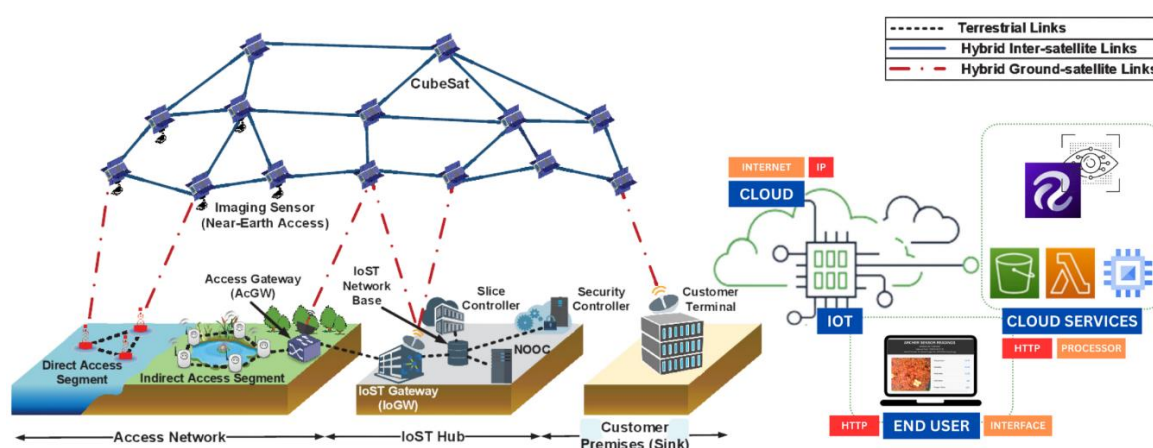


Figure 4. Proposed Framework Expanded to IoST architecture.

The first two blocks in the initial framework were now replaced with the IoST framework as introduced in an existing study [38] which explores bringing the cyber physical system into the space to realize true global connectivity. In this hybrid scenario, multiple CubeSats can be deployed either in homogenous setup performing similar mission [39,40] and serve as a repeater or in heterogenous form [41] designated with unique missions and act with a server-client relationship in space.

The study addresses significant gaps in space-based agricultural research, particularly the need for real-time remote monitoring systems for seed germination experiments in CubeSat missions. By developing an integrated system using AWS IoT for data collection, storage, and remote access, the research ensures efficient and accessible monitoring of space-based plant experiments. Additionally, it advances the field by incorporating deep learning and cloud-based platforms like Roboflow for automated germination detection, providing a scalable solution for space research. The study also contributes to the design of a 2U CubeSat capable of supporting plant growth experiments in microgravity, highlighting the importance of environmental control in CubeSat payloads. By combining cloud-based image detection and automated biological analysis, this research fills critical gaps in space farming, offering a practical and sustainable framework for future experiments in space-based biological research.

2. Materials and Methods

This chapter details the development of an ESP32-based system for efficient data collection, monitoring, and transmission in a controlled agricultural setting. Leveraging ESPNOW protocol for device communication and advanced libraries for data handling, the system integrates sensor data acquisition, real-time monitoring, and cloud-based access via AWS IoT Core. Additionally, it incorporates cloud-based machine vision modeling [42] for detecting crop germination using images captured by the ESP32-Cam, enabling automated analysis and insights. Each component ensures reliability, scalability, and precision, supporting agricultural applications in a CubeSat environment through seamless data management and accessibility.

2.1. Data Collection Using the ESP32-Based System Simulation

Using two ESP32-based microcontrollers, data collection from the DHT11 sensor with the ESP32 and the ESP32-Cam are temporarily stored in the microSD memory card inserted in the ESP32-Cam microcontroller. The process is illustrated in Figure 5. This uses the ESPNOW technology where the ESP32 is used to broadcast a structure of data to the ESP32-Cam over a local router. This simulates the data transfer from the 1st two blocks in the proposed framework.

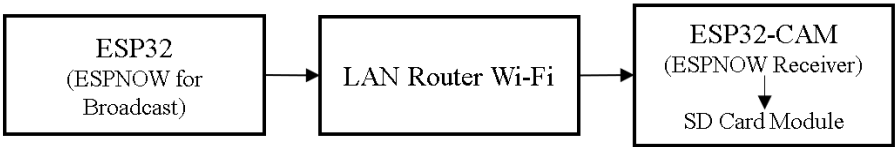


Figure 5. Process of data transfer using ESPNOW Technology.

The ESP32 microcontroller was used to be the main clock of the whole system by internally setting the time with the method of using the header, <ESP32Time.h>. This method doesn’t use internet connection in accessing the real-time and date but starts with the given date in the Arduino code. In practice, the date input is the same date of the starting day of data collection and the time is set to 00:00:00 in hr:min:sec format. The next data in the ESP32 is the data from the DHT11. This includes temperature, humidity, heat index in units of degrees Celsius (°C), percentage (%) of relative humidity, and degrees Celsius (°C), respectively. This is made possible using the header, <DHT.h>. Since the ESP32 also controls the lighting system and irrigation system of the CubeSat, the last two information to be transferred are the status of the LED light, on or off, and the status of the pump if it is on or off also. Figure 6 shows the corresponding code snippet.

```
// Receiver MAC Address
uint8_t broadcastAddress[] = {0x--, 0x--, 0x--, 0x--, 0x--, 0x--};

// Define data structure
typedef struct struct_message {
    String Date;
    String Time;
    float Temperature;
    float Humidity;
    float Heat_Index;
    String LED_State;
    String WaterPump_State;
} struct_message;

//Create structured data object
struct_message TransmitterData;
```

Figure 6. Arduino Code Snippet for Initializations of Data Structure and Transmitting Data via ESP-NOW.

This study effectively leverages ESP-NOW for wireless communication between ESP32 devices. By initializing ESP-NOW, defining a data structure, and establishing a peer-to-peer connection, the code enables the transmission of sensor data, LED control commands, and Fogger state updates. ESP-NOW’s simplicity and low-latency characteristics make it suitable for this application. The code efficiently transmits data without the complexity over the local network via Wi-Fi technology. After this, the data structure as in Figure 7 is received at the ESP32-Cam, which captures the images of the current situation of the plant bed in the ARCHER CubeSat in JPEG format and named according to the data number. In addition, information such as data number was also generated in the ESP32-Cam and was added in the data structure.

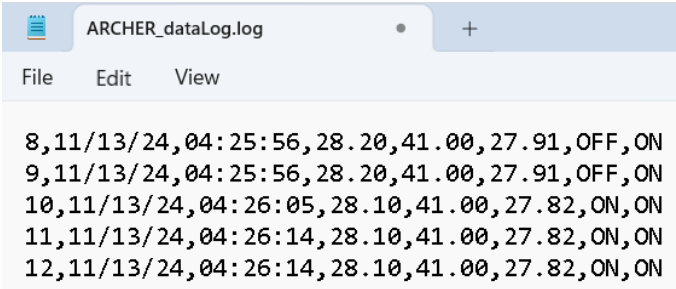


Figure 7. Data Structure in the SD Card with filename ARCHER_dataLog.log.

Using the built-in SD card module in the ESP32-Cam, the numerical and text data were each saved as a string in a comma-separated values (CSV) format. Each row corresponds to a single reading and includes the following columns: ID, Date, Time, Temperature, Humidity, Heat Index, LED State, and Fogger State. The use of an SD card for data saving is an integral part of the methodology in this study, ensuring secure and accessible storage of sensor and image data. This approach is designed to facilitate the systematic collection, organization, and retrieval of information critical to evaluating crop health and environmental parameters. The process begins with SD card initialization, which is carried out using the SD_MMC library. Headers such as <Arduino.h>, <FS.h>, and <SD_MMC.h> enable core functionality for handling data storage operations, while <Arduino_JSON.h> was utilized for formatting data into JSON strings when needed. Each entry in the data file is uniquely identified using an ID and timestamp, ensuring traceability and chronological organization. For the image data, high-resolution images captured by the hyperspectral or thermal imaging system are stored separately on the SD card in a compressed format, such as JPEG. The files are dynamically named using a combination of a unique identifier and the timestamp of capture. These filenames are cross-referenced in the sensor data logs to maintain a cohesive relationship between the numerical data and corresponding visual or spectral information.

2.2. Development of the Asynchronous Local Webserver on ESP32

The development of the ESP32-based web server integrates advanced functionalities such as live data updates as shown in Figure 8. It includes a live image upload feature, enabled by the ESP32-CAM module, which allows real-time monitoring through camera feeds. Images are dynamically captured and displayed on the web interface, offering users the ability to visually monitor the germination environment. Complementing this, the server provides live sensor readings, showcasing critical environmental parameters such as temperature, humidity, and heat index. This real-time data aids in making informed decisions based on the monitored conditions. Additionally, the inclusion of a live date and time update feature ensures temporal accuracy in all data interactions that the web server tracks and displays, which is crucial for time-sensitive applications. This is accompanied by the current data ID and image filename saved on the microSD card, providing transparency and ease of access to historical data. This feature simplifies data management and retrieval, especially in scenarios where large datasets are generated over extended periods. The entire system was developed using PlatformIO within Visual Studio Code, leveraging the Arduino framework for the ESP32. The key libraries used are in Table 1 with their corresponding function.

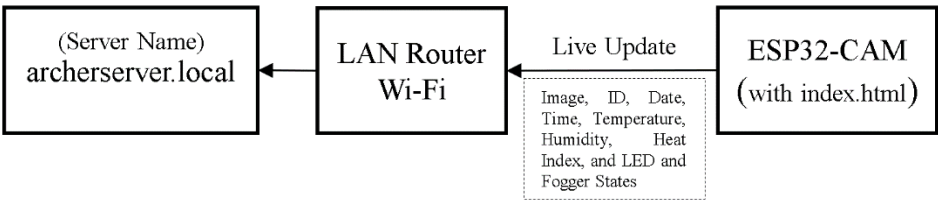


Figure 8. Framework of Asynchronous Live Updates in the local web server using WebSocket.

Table 1. Key Libraries used in Web Server Setup.

Library Header	Function
WiFi.h	Provides functions to connect the ESP32 to a Wi-Fi network, enabling network communication.
WebServer.h	Allows the ESP32 to act as a web server, handling HTTP requests and serving web pages and files.
WebSocketsServer.h	Enables real-time, bidirectional communication between the ESP32 and the client via WebSockets.

FS.h	Provides file system functionality for managing files on the ESP32's internal flash memory.
LittleFS.h	A lightweight file system is used for storing files (e.g., images) on the ESP32's internal storage.
ArduinoJson.h	Used to format and parse JSON data for easy transmission between the ESP32 and the client.

To manage the web server's interface, an index.html file was designed and uploaded to the ESP32-CAM as a filesystem image. This process utilized the LittleFS file system, with the LittleFS.h library enabling the ESP32 to access and serve the uploaded web page. After which, the ESP32 microcontroller facilitates both Wi-Fi-based web server functionality and device-to-device communication via ESP-NOW. The use of WebSocket technology within the web server allows for real-time, bidirectional communication, enabling instantaneous updates of sensor readings, image feeds, and system states without requiring repeated HTTP requests. This reduces latency and optimizes bandwidth, making the system suitable for applications demanding high responsiveness.

Numeric and string data, such as sensor readings, current date and time, data IDs, and image filenames, are sent to the client in structured JSON format. The client-side JavaScript listens for WebSocket messages and dynamically updates the HTML elements using the Document Object Model (DOM). Captured images from the ESP32-CAM are stored with unique filenames on the microSD card or the ESP32's memory. The images are then encoded in Base64 format and made accessible via WebSocket connections, enabling remote monitoring and analysis. The index.html file dynamically requests the latest image, which is served over HTTP and displayed on the web page. To avoid caching issues, the image URL includes a timestamp, ensuring the browser fetches the most recent capture. The image filename is updated through WebSocket alongside other data.

2.3. Establishment of Data Monitoring and Saving using AWS IoT

This part of the methodology aims to connect the ESP32-Cam microcontroller to Amazon AWS IoT Core via the MQTT protocol, which provides a structured approach to integrating IoT devices with cloud services. This allows device integration with cloud-based solutions for remote monitoring outside the local network, enables data transmission, and facilitates bidirectional communication securely and efficiently.

The first step involves configuring the AWS IoT Core to recognize the ESP32-Cam as a device, referred to as a "Thing." This representation serves as the bridge between the hardware and cloud services. By generating the required certificates and private keys for SSL/TLS encryption, the process ensures the creation of a secure environment where data integrity and secure communication between the ESP32-Cam and AWS IoT Core are established. After which, the setup involves defining MQTT topics that act as channels for data exchange. In the Arduino code in PlatformIO, two important libraries are included, which are <PubSubClient.h> for MQTT and <ArduinoJSON.h> for data formatting. This also uses the library <WiFi.h> to connect to the internet. After connecting to the internet, the next step is configuring the MQTT client with AWS IoT Core endpoint details and the generated security credentials in a created file header named, <secrets.h>. Functions for publishing sensor data and subscribing to topics are implemented to enable bidirectional communication. The inclusion of SSL/TLS settings ensures secure data transmission. Once the sketch is uploaded to the ESP32-Cam, the serial monitor in VS Code will confirm if the device's serial output of successful Wi-Fi and MQTT connections. Next is the validation in the AWS IoT Console, where the MQTT Test Client is used to verify that the ESP32-Cam can receive data messages by subscribing to a topic, "dlsuARCHER/pub." Once subscribed to the topic, the MQTT test client will receive the data from the CubeSat in JSON format such as variables named data_id, date_time, image_id, temperature, humidity, heat_index, ledState, and foggerState. Publishing real-time sensor data to AWS IoT Core enables remote access, even outside the local network, to the data and monitoring that the CubeSat is still working.

2.4. Development of Roboflow Model for Germination Detection

The development of a Roboflow-based model for detecting seed germination represents an innovative approach to advancing the monitoring system inside a CubeSat. See Figure 9 for the actual germination bed setup in 2U CubeSat. The model enabled efficient and non-invasive method for real-time monitoring of germination, with the objective of counting the number of germinated seeds as an input for quantifying the growth rate.



Figure 9. Germinating bed for the ARCHER CubeSat with the ESP32-CAM placed on top and the DHT11 sensor at the back.

Images from the ARCHER’s SD card mounted in the ESP32-Cam were retrieved and used as the image data for annotation. Manual annotation was implemented in the Roboflow application for placing bounding boxes on the germinated seeds in the plant bed as shown in Figure 10.

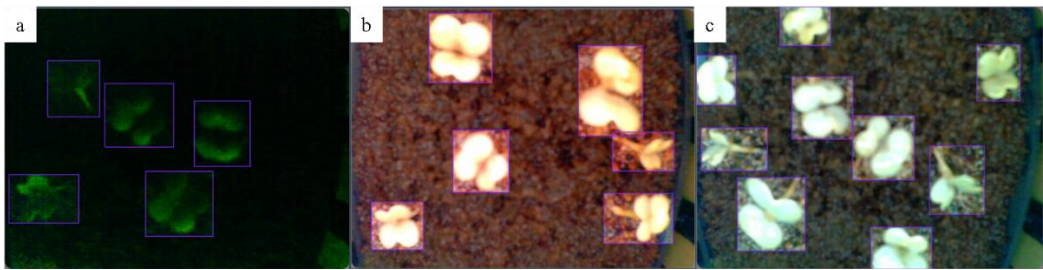


Figure 10. Manual annotation performed in Roboflow where germinated crops are labeled as “Seedling” with three different light conditions such as (a) no light, and two different levels of exposure as in (b) and (c).

The image dataset contains 95 images that were annotated. The germinated crops were labeled as “seedling.” After this, preprocessing and augmentations were performed, which generated more versions of the original image dataset, resulting in 162 images. The list of preprocessing and augmentations are listed in Table 2.

Table 2. Preprocessing and Augmentation Parameters for the Germination Detection Model.

Category	Parameter	Details
Preprocessing	Auto-Orient	Applied
	Resize	Stretch to 640x640
Augmentations	Outputs per Training Example	2
	Grayscale	Apply to 100% of images

Saturation	Between -72% and +72%
Brightness	Between -38% and +38%
Blur	Up to 2.1px
Noise	Up to 1.6% of pixels

The 162 images were divided into train, validation, and test datasets by 83%, 12%, and 6%, respectively. The ARCHER germination detection model was trained using the Roboflow 3.0 object detection framework to achieve efficient and accurate germination stage detection. The selected checkpoint, COCO, provides a robust foundation, as it is pre-trained on a large, diverse dataset of objects. This was evaluated with the evaluation metrics such as mean average precision (mAP), precision, and recall. The mAP metric provides a comprehensive measure of the model’s ability to correctly identify germination stages across varying levels of overlap between predicted and actual bounding boxes, ensuring both localization and classification accuracy. Precision highlights the proportion of correctly identified germinating seeds out of all positive predictions, minimizing false positives and ensuring reliability in practical applications. Conversely, recall measures the proportion of actual germinating seeds correctly detected, reflecting the model’s sensitivity and ability to minimize false negatives. In addition, training graphs such as mAP, box loss, class loss, and object loss provided insights into the model’s performance during the training process. The mAP curve tracks the model’s accuracy in detecting and classifying germination stages across training epochs, with a steady upward trend indicating improved learning and convergence. The box loss graph evaluates the precision of bounding box predictions, with a declining trend reflecting enhanced localization of germinating seeds. Similarly, the class loss graph measures errors in classification, and its reduction signifies improved differentiation between germination stages and other objects. Lastly, the object loss graph monitors the model’s confidence in identifying objects, with decreasing values showcasing growing reliability in distinguishing germinating seeds from background noise.

3. Results and Discussions

This chapter analyzes the outcomes of implementing the crop germination detection and remote monitoring system within the 2U CubeSat using the proposed framework. It presents data from various methodologies, including local web server performance, AWS IoT Core effectiveness for data management and remote monitoring, and deep learning model evaluation for seedling detection. This analysis emphasizes the significance of agricultural innovation and space research, demonstrating how advanced technologies can assist in real-time monitoring [43] and data collection.

3.1. Local Webserver Based on ESP32

After building the Arduino code in PlatformIO using VS Code to transfer information data, sensor data, and image data to update dynamically the local web server and uploading the HTML file as an image to the ESP32-CAM system, the local server can be accessed using the server name “archerserver.local” to be typed as a browser URL. The capability to use a unique DNS was enabled by adding a multicast DNS (mDNS) protocol in the Arduino code. This removes the need to manually access and type the dynamic IP address that usually changes as you restart the router where the ESP32 devices are connected, enabling zero-configuration networking.

Using the WebSocket communication protocol, dynamic updates from ESP32-CAM to the web server were successfully established. This protocol enabled real-time, bidirectional communication between a client, a web browser, and a server over a single, persistent TCP connection going to the microcontroller. This surpasses the capability of a traditional HTTP request-response model where both parties need to refresh their connections just to send and receive messages simultaneously. This provided the capability to have real-time remote data updates such as date, time, data ID number, filename of the latest saved image, sensor readings for temperature, humidity and heat index, LED and fogger states, and an image of the current situation of the germination bed. The image transfer

through the WebSocket protocol was made possible by encoding the image file in Base64 format. The web server automatically decodes the Base64 format to display the image. An example output of the web application is presented in Figure 11.

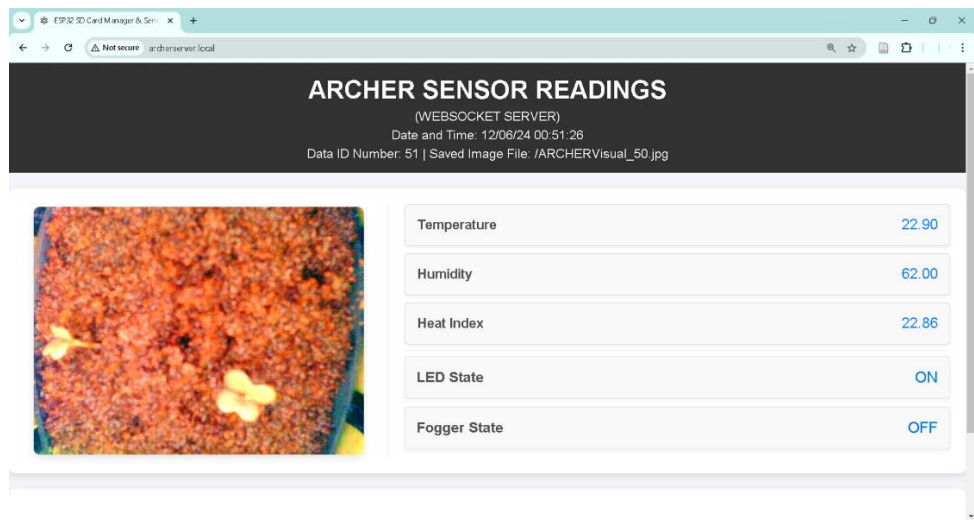


Figure 11. The web application for remote monitoring of the ARCHER CubeSat.

3.2. Secure Remote Monitoring Using AWS IoT

The establishment of data monitoring and saving using AWS IoT Core was effectively implemented by enabling a comprehensive framework for remote monitoring and data management through the ESP32-CAM. The system facilitated bidirectional communication by setting up MQTT topics for publishing sensor data, such as temperature and humidity, while also subscribing to relevant topics for receiving updates from the cloud. This capability ensures that users can both send and receive critical information in real-time. However, the feature of sending back data or commands to the microcontroller still has no use with the current configuration. The objective of data management and accessibility is also met, as real-time sensor data is stored in the cloud, making it readily accessible for monitoring and analysis from any location. A snippet of the publishing of the topic and a sample of received data in JSON format are exhibited in Figure 12.

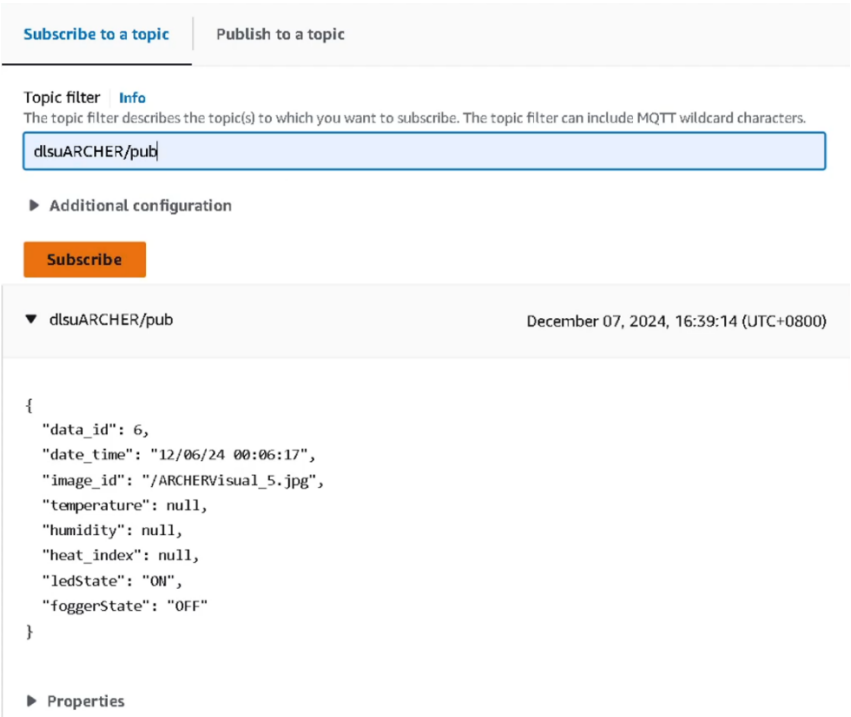


Figure 12. The AWS IoT Core console where a user subscribes to the “dlsuARCHER/pub” MQTT topic, displaying real-time sensor data in JSON format.

By connecting the device to AWS IoT Core via the MQTT protocol, users can achieve remote monitoring and data transmission, allowing for secure and efficient data transfer beyond local networks. This is complemented by the secure device integration objective, where the ESP32-Cam is configured as a “Thing” in AWS IoT Core, utilizing SSL/TLS encryption to ensure that all communications maintain data integrity and security.

3.3. Performance and Evaluation of the Roboflow 3.0 Model for Germination Detection

The evaluation of the Roboflow model for crop germination detection begins with analyzing the training graphs, which offer valuable insights into the model’s learning process. These graphs track the progression of key loss metrics and demonstrate how well the model improves over time.

The mAP graph shown in Figure 13 (a) shows a steady increase over training epochs, indicating the model’s growing accuracy in detecting germinating seeds. The stabilization of the mAP curve at a high value during training indicates that the model has effectively converged, successfully distinguishing between germinated and non-germinated seeds. Further training likely yields minimal improvements. The model achieves an impressive mAP of 99.5%, which is a strong indication of its ability to correctly predict germinations across multiple overlap thresholds between predicted and actual bounding boxes.

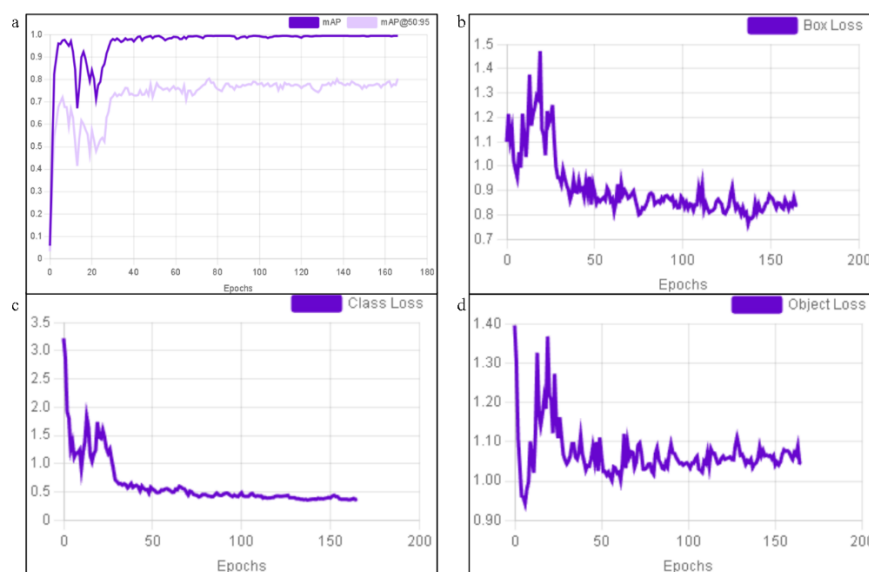


Figure 13. Training graphs for the Roboflow 3.0 seeding detection model: (a) mAP Graph, (b) Box Loss Graph, (c) Class Loss Graph, (d) Object Loss Graph.

The box loss graph in Figure 13 (b) exhibits fluctuations in the early stages of training, showing that the model is refining its ability to localize germinating seeds by adjusting the bounding boxes. The fluctuations reflect the model's exploration of optimal bounding box placements. The decline indicates that the model is progressively improving, ensuring that the detected bounding boxes closely match the actual seed locations.

The class loss graph shown in Figure 13 (c) exhibited a rapid decrease in the early epochs, suggesting that the model quickly learns to detect the germinated seeds. This sharp reduction in class loss indicates that the model is efficiently learning the classification task. The stable low value nearing the end implies that the model has reached a strong understanding of the class distinctions.

The object loss graph presented in Figure 13 (d) shows initial fluctuations as the model learns to detect germinated seeds and adjusts its predictions. These fluctuations occur as the model refines its understanding of localization and classification. As training progressed, the object loss decreased steadily, signaling that the model was becoming increasingly proficient in detecting seedlings.

After the training process, validation, and testing, the Roboflow model is evaluated by analyzing key performance metrics such as [44] as mean average precision (mAP), precision, and recall. These metrics are instrumental in understanding the resulting model's efficacy in detecting and classifying germinated seeds accurately. The model achieves an outstanding mAP of 99.5%, reflecting its high detection accuracy across various overlap thresholds between predicted and actual bounding boxes. This is further supported by a precision of 99.9%, which indicates a minimal false positive rate, ensuring that almost every prediction of germination is correct. The model's recall of 100.0% highlights its perfect sensitivity, capturing every instance of seed germination without missing any true positives. These metrics establish the model's robustness and reliability in accurately identifying germinated seeds, providing a highly accurate tool for crop germination detection. This is demonstrated in Figure 14, which is one of the images used in the testing phase of the objection detection model. In using the model for detection with a video of the images, the confidence threshold of the detection was set at 70% to accurately distinguish the germinated seeds. While the overlap threshold was maintained at the default level of 50%.

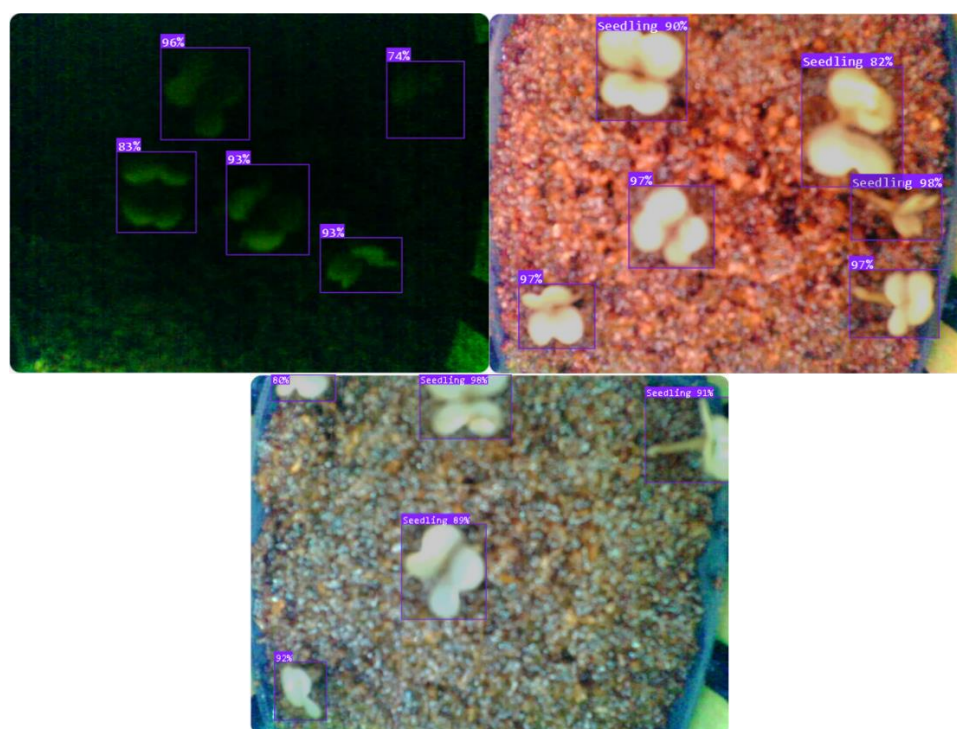


Figure 14. Detection of seedlings using the developed Roboflow 3.0 model exhibiting a confidence level of detection from 74% up to 98%.

4. Conclusion and Recommendations

In conclusion, this study successfully developed a cloud-based crop germination detection and real-time remote monitoring system tailored for a 2U CubeSat environment following the proposed framework. By integrating IoT technologies, deep learning, and cloud platforms such as AWS IoT and Roboflow, the research addresses critical challenges in space agricultural practices, particularly the need for efficient monitoring solutions [45–47]. The implementation of an ESP32-based system for data collection and the potential for deploying a machine vision model for automated germination detection demonstrated significant advancements in real-time monitoring capabilities. These innovations not only enhance the functionality of CubeSat missions, on-Earth simulations, or in space but also contribute to the broader field of sustainable agriculture in extreme environments. Overall, this research highlights the potential of combining advanced technologies to facilitate the growth of plants in microgravity, ultimately supporting long-term human habitation in space.

Given the promising results, future research should focus on expanding the system's capabilities, such as integrating additional sensors, and implementing using the proposed hybrid framework to advance nanosatellite mission into IoST architecture. Moreover, further optimization and actual deployment of the machine vision model could enhance its adaptability to different crop stages, types, and environmental conditions, thereby increasing its applicability across various space missions. It is also recommended to explore the integration of autonomous data collection and decision-making systems, enabling greater operational efficiency in remote or resource-limited settings. Lastly, conducting field tests in simulated microgravity environments on Earth could provide additional insights into the system's performance and reliability in space-like conditions, further solidifying its potential for space farming applications.

Author Contributions: Conceptualization, Adrian Janairo and Marielet Guillermo; methodology, software, visualization, writing—original draft preparation, Adrian Janairo; resources, supervision, Ronnie Concepcion and Marielet Guillermo; writing—review and editing, formal analysis, funding acquisition, Arvin Fernando and Marielet Guillermo. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by De La Salle University through its Science Foundation publication development grant.

Data Availability Statement: Data can be requested from the corresponding author as needed.

Acknowledgments: The authors wish to thank the Department of Manufacturing Engineering and Management and the Department of Mechanical Engineering of De La Salle University for the administrative and technical support with the use of laboratory equipment and spaces. Special thanks to Engr. Valencia for the help in the design of CubeSat used in the study. Lastly, the peer reviewers and the editorial board are very much appreciated for their significant contribution in the improvement of this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. M. Swartwout, "The first one hundred cubesats: A statistical look," *Journal of small Satellites*, vol. 2, no. 2, pp. 213–233, 2013.
2. P. Zabel, M. Bamsey, D. Schubert, and M. Tajmar, "Review and analysis of over 40 years of space plant growth systems," *Life sciences in space research*, vol. 10, pp. 1–16, 2016.
3. A.-L. Paul, C. E. Amalfitano, and R. J. Ferl, "Plant growth strategies are remodeled by spaceflight," *BMC Plant Biol*, vol. 12, no. 1, p. 232, Dec. 2012, doi: 10.1186/1471-2229-12-232.
4. J. Puig-Suari, C. Turner, and W. Ahlgren, "Development of the standard CubeSat deployer and a CubeSat class PicoSatellite," in 2001 IEEE aerospace conference proceedings (Cat. No. 01TH8542), IEEE, 2001, pp. 1–347. Accessed: Dec. 02, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/931726/>
5. D. J. Barnhart, T. Vladimirova, and M. N. Sweeting, "Very-Small-Satellite Design for Distributed Space Missions," *Journal of Spacecraft and Rockets*, vol. 44, no. 6, pp. 1294–1306, Nov. 2007, doi: 10.2514/1.28678.
6. Jiaolong Zhang, Jingao Su, Chao Wang, Yiqian Sun, Modular design and structural optimization of CubeSat separation mechanism, *Acta Astronautica*, Volume 225, 2024, Pages 758-767, ISSN 0094-5765, <https://doi.org/10.1016/j.actaastro.2024.09.067>.
7. Sarat Chandra Nagavarapu, Laveneishyan B. Mogan, Amal Chandran, Daniel E. Hastings, CubeSats for space debris removal from LEO: Prototype design of a robotic arm-based deorbiter CubeSat, *Advances in Space Research*, 2024, ISSN 0273-1177, <https://doi.org/10.1016/j.asr.2024.08.009>.
8. A. Z. Babar and O. B. Akan, "Sustainable and Precision Agriculture with the Internet of Everything (IoE)," Apr. 13, 2024, arXiv: arXiv:2404.06341. doi: 10.48550/arXiv.2404.06341.
9. V. Bhanumathi and K. Kalaivanan, "The Role of Geospatial Technology with IoT for Precision Agriculture," in *Cloud Computing for Geospatial Big Data Analytics*, vol. 49, H. Das, R. K. Barik, H. Dubey, and D. S. Roy, Eds., in *Studies in Big Data*, vol. 49, Cham: Springer International Publishing, 2019, pp. 225–250. doi: 10.1007/978-3-030-03359-0_11.
10. L. Capogrosso, F. Cunico, D. S. Cheng, F. Fummi, and M. Cristani, "A machine learning-oriented survey on tiny machine learning," *IEEE Access*, 2024, Accessed: Dec. 07, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10433185/>
11. J. Kua, S. W. Loke, C. Arora, N. Fernando, and C. Ranaweera, "Internet of things in space: a review of opportunities and challenges from satellite-aided computing to digitally-enhanced space living," *Sensors*, vol. 21, no. 23, p. 8117, 2021.
12. S. T. Arzo et al., "Essential technologies and concepts for massive space exploration: Challenges and opportunities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 1, pp. 3–29, 2022.
13. D. Witczak and S. Szymoniak, "Review of Monitoring and Control Systems Based on Internet of Things," *Applied Sciences*, vol. 14, no. 19, p. 8943, 2024.
14. P. Singh and R. Krishnamurthi, "IoT-based real-time object detection system for crop protection and agriculture field security," *J Real-Time Image Proc*, vol. 21, no. 4, p. 106, Aug. 2024, doi: 10.1007/s11554-024-01488-8.
15. J. Miao, "A Fog-Enabled Microservice-Based Multi-Sensor IoT System for Smart Agriculture," PhD Thesis, University of Colorado at Boulder, 2024. Accessed: Dec. 07, 2024. [Online]. Available:

- <https://search.proquest.com/openview/de1fb724683f0eafe46c21b732eca54d/1?pq-origsite=gscholar&cbl=18750&diss=y>
16. B. R. Babu, P. M. A. Khan, S. Vishnu, and K. L. Raju, "Design and Implementation of an IoT-Enabled Remote Surveillance Rover for Versatile Applications," in 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), IEEE, 2022, pp. 1–6. Accessed: Dec. 07, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10119249/>
 17. J. Kua, S. W. Loke, C. Arora, N. Fernando, and C. Ranaweera, "Internet of things in space: a review of opportunities and challenges from satellite-aided computing to digitally-enhanced space living," *Sensors*, vol. 21, no. 23, p. 8117, 2021.
 18. K.-K. Phan, "Development of a Plant Growth and Health Monitoring System Using Imaging and Sensor Array Information for CubeSat Applications," Master's Thesis, University of South Florida, 2024. Accessed: Dec. 07, 2024. [Online]. Available: <https://search.proquest.com/openview/5551d8f931b1b3a264cf9e3462547b6d/1?pq-origsite=gscholar&cbl=18750&diss=y>
 19. G. Fortino, A. Guerrieri, P. Pace, C. Savaglio, and G. Spezzano, "Iot platforms and security: An analysis of the leading industrial/commercial solutions," *Sensors*, vol. 22, no. 6, p. 2196, 2022.
 20. O. Debauche, S. Mahmoudi, P. Manneback, and F. Lebeau, "Cloud and distributed architectures for data management in agriculture 4.0: Review and future trends," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7494–7514, 2022.
 21. W. Liu, M. Wu, G. Wan, and M. Xu, "Digital Twin of Space Environment: Development, Challenges, Applications, and Future Outlook," *Remote Sensing*, vol. 16, no. 16, p. 3023, 2024.
 22. T. Shymanovich and J. Z. Kiss, "Conducting Plant Experiments in Space and on the Moon," in *Plant Gravitropism*, vol. 2368, E. B. Blancaflor, Ed., in *Methods in Molecular Biology*, vol. 2368, New York, NY: Springer US, 2022, pp. 165–198. doi: 10.1007/978-1-0716-1677-2_12.
 23. N. J. Bonafede Jr, "Low-Cost Reaction Wheel Design for CubeSat Applications," Master's Thesis, California Polytechnic State University, 2020. Accessed: Dec. 07, 2024. [Online]. Available: <https://search.proquest.com/openview/2432317f023a1312d5582f7cd6365c0b/1?pq-origsite=gscholar&cbl=18750&diss=y>
 24. M. Sathasivam, R. Hosamani, and B. K. Swamy, "Plant responses to real and simulated microgravity," *Life Sciences in Space Research*, vol. 28, pp. 74–86, 2021.
 25. A. Poghosyan and A. Golkar, "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions," *Progress in Aerospace Sciences*, vol. 88, pp. 59–83, 2017.
 26. A. Poghosyan and A. Golkar, "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions," *Progress in Aerospace Sciences*, vol. 88, pp. 59–83, 2017.
 27. F. Arneodo, A. Di Giovanni, and P. Marpu, "A review of requirements for gamma radiation detection in space using cubesats," *Applied Sciences*, vol. 11, no. 6, p. 2659, 2021.
 28. P. Marzioli et al., "CultCube: Experiments in autonomous in-orbit cultivation on-board a 12-Units CubeSat platform," *Life Sciences in Space Research*, vol. 25, pp. 42–52, 2020.
 29. P. Carillo, B. Morrone, G. M. Fusco, S. De Pascale, and Y. Rouphael, "Challenges for a sustainable food production system on board of the international space station: A technical review," *Agronomy*, vol. 10, no. 5, p. 687, 2020.
 30. K. Johansen, M. G. Ziliani, R. Houborg, T. E. Franz, and M. F. McCabe, "CubeSat constellations provide enhanced crop phenology and digital agricultural insights using daily leaf area index retrievals," *Scientific reports*, vol. 12, no. 1, p. 5244, 2022.
 31. K. G. Falk et al., "Computer vision and machine learning enabled soybean root phenotyping pipeline," *Plant Methods*, vol. 16, no. 1, p. 5, Dec. 2020, doi: 10.1186/s13007-019-0550-5.
 32. V. G. Dhanya et al., "Deep learning based computer vision approaches for smart agricultural applications," *Artificial Intelligence in Agriculture*, vol. 6, pp. 211–229, 2022.
 33. F. Fuentes-Peñailillo, G. Carrasco Silva, R. Pérez Guzmán, I. Burgos, and F. Ewertz, "Automating seedling counts in horticulture using computer vision and AI," *Horticulturae*, vol. 9, no. 10, p. 1134, 2023.

34. T. Mahendrakar, R. T. White, M. Tiwari, and M. Wilde, "Unknown Non-Cooperative Spacecraft Characterization with Lightweight Convolutional Neural Networks," *Journal of Aerospace Information Systems*, vol. 21, no. 5, pp. 455–460, May 2024, doi: 10.2514/1.I011343.
35. S. Lange, "UTILIZING MACHINE LEARNING ENSEMBLES FOR PHENOTYPIC TRAIT ANALYSIS IN PLANT MONITORING." 2024. Accessed: Dec. 07, 2024. [Online]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1867296>
36. N. N. Thilakarathne, M. S. A. Bakar, P. E. Abas, and H. Yassin, "Towards making the fields talks: A real-time cloud enabled iot crop management platform for smart agriculture," *Frontiers in Plant Science*, vol. 13, p. 1030168, 2023.
37. A. Paul, R. Machavaram, D. Kumar, and H. Nagar, "Smart solutions for capsicum Harvesting: Unleashing the power of YOLO for Detection, Segmentation, growth stage Classification, Counting, and real-time mobile identification," *Computers and Electronics in Agriculture*, vol. 219, p. 108832, 2024.
38. I. F. Akyildiz and A. Kak, "The Internet of Space Things/CubeSats," in *IEEE Network*, vol. 33, no. 5, pp. 212–218, Sept.-Oct. 2019, doi: 10.1109/MNET.2019.1800445.
39. A. Fernando, L. Lim, A. Bandala, R. Vicerra, E. Dadios, M. Guillermo, and R. Naguib, "Simulated vs Actual Application of Symbiotic Model on Six Wheel Modular Multi-Agent System for Linear Traversal Mission," *J. Adv. Comput. Intell. Inform.*, Vol.28 No.1, pp. 12–20, 2024.
40. Fernando, Arvin H., et al. "Load Pushing Capacity Analysis of Individual and Multi-Cooperative Mobile Robot through Symbiotic Application." *International Journal of Mechanical Engineering and Robotics Research* 13.2 (2024)
41. R. A. R. Bedruz, J. Martin Z. Maningo, A. H. Fernando, A. A. Bandala, R. R. P. Vicerra and E. P. Dadios, "Dynamic Peloton Formation Configuration Algorithm of Swarm Robots for Aerodynamic Effects Optimization," 2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA), Daejeon, Korea (South), 2019, pp.264–267, doi: 10.1109/RITAPP.2019.8932871.
42. J. M. Paule, J. R. Roca, K. M. Subia, T. J. T. Tiong, M. Guillermo and D. de Veas-Abuan, "Integration of AWS and Roboflow Mask R-CNN Model for a Fully Cloud-Based Image Segmentation Platform," 2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Coron, Palawan, Philippines, 2023, pp. 1–6, doi: 10.1109/HNICEM60674.2023.10589105.
43. A. R. A. Pascua, M. Rivera, M. Guillermo, A. Bandala and E. Sybingco, "Face Recognition and Identification Using Successive Subspace Learning for Human Resource Utilization Assessment," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 1375–1380, doi: 10.1109/ICTC55196.2022.9952691.
44. A. H. Fernando, A. B. Maglaya and A. T. Ubando, "Optimization of an algae ball mill grinder using artificial neural network," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 3752–3756, doi:10.1109/TENCON.2016.7848762.
45. Agoo, J.; Lanuza, R.J.; Lee, J.; Rivera, P.A.; Velasco, N.O.; Guillermo, M.; Fernando, A. Geographic Information System-Based Framework for Sustainable Small and Medium-Sized Enterprise Logistics Operations. *ISPRS Int. J. Geo-Inf.* 2025, 14, 1. <https://doi.org/10.3390/ijgi14010001>
46. Amante, K., Ho, L., Lay, A., Tungol, J., Maglaya, A., & Fernando, A. (2021, March). Design, fabrication, and testing of an automated machine for the processing of dried water hyacinth stalks for handicrafts. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1109, No. 1, p. 012008). IOP Publishing.
47. Fernando, A., and L. GanLim. "Velocity analysis of a six wheel modular mobile robot using MATLAB-Simulink." *IOP Conference Series: Materials Science and Engineering*. Vol. 1109. No. 1. IOP Publishing, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.