SMART GREENHOUSE MEASUREMENT AND MONITORING SYSTEM BASED ON IOT AND COMPUTER VISION

HOANG THI TU UYEN*, NGUYEN HOANH, NGO THANH QUYEN, LAM BINH MINH, NGUYEN TRUNG NHAN
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City
**Coresponding Author: hoangthituuyen@iuh.edu.vn*
DOIs: https://www.doi.org/ 10.46242/jstiuh.v83i5. 6189

Abstract. This paper presents a smart greenhouse measurement and monitoring system integrating IoT and computer vision for non-invasive plant growth observation and remote supervision. The proposed architecture combines environmental sensors, STM32-based data acquisition and control modules, a Raspberry Pi 4 with a USB camera, a lightweight deep learning-based stem segmentation module, Firebase Realtime Database, and a web application for monitoring and control. In the computer vision module, tomato plant images are processed directly on the Raspberry Pi to segment stem regions and estimate plant height from calibrated image measurements, enabling edge-based phenotypic monitoring without destructive sampling or manual measurement. Experimental results on a tomato cultivation prototype show that the system can collect temperature, air humidity, soil moisture, and light intensity data in real time while providing automated plant height estimation from RGB images. During the reported period, temperature ranged from 27.8°C to 38.8°C, air humidity from 40.2% to 77.0%, soil moisture from 52.0% to 82.4%, and plant height increased from 65.8 cm to 77.0 cm. The segmentation results demonstrate the feasibility of extracting stem-level structural information under practical greenhouse conditions, including complex background, leaf occlusion, and non-uniform illumination. The prototype confirms the potential of a low-cost smart greenhouse monitoring system and provides a practical basis for future expansion toward richer agronomic indicators, such as stem width, leaf area, canopy coverage, growth rate estimation, and automated environmental control.
Keywords. computer vision, greenhouse monitoring, IoT, smart agriculture, lightweight deep learning model

1. INTRODUCTION

In the context of the strong development of precision agriculture and digital agriculture, greenhouse growing technology is not only particularly significant in areas with harsh climate conditions and environmental pollution, but also serves as a solution to address one of humanity's greatest challenges: how to provide more sustainably grown food sources while still increasing production output. Thanks to the ability to precisely control the cultivation environment and monitor crop growth status, smart greenhouses can operate automatically, reducing dependence on human experience and manual operations, increasing continuous monitoring capability, and thereby improving crop yields.

In recent years, both globally and in Vietnam, there have been numerous studies implementing the most modern digital technologies such as IoT and computer vision into smart greenhouses, aiming to develop precision agriculture, help optimize productivity, minimize risks from nature, ensure food security, and so on. Paper [1] develops an environmental monitoring system for greenhouses based on an IoT platform. The system includes multiple sensors to measure environmental parameters such as temperature, air humidity, soil moisture, pH level, and radiation. These environmental parameters are collected and initially processed by the LinkIt Smart 7688 Duo microcontroller, after which the information is transmitted to the Thinger.io platform, where data is processed, analyzed, and displayed in real time in graphical form. Users are granted access to this platform and can observe the collected data from any internet-connected electronic device. Paper [2] presents a smart greenhouse monitoring and control system in India, where the collected environmental parameters include soil moisture, temperature, air humidity, and light intensity. The actuators used to modify the environment are a water pump, water sprinkler, fan, and light bulb respectively. The system uses GSM technology to transmit collected data to the user's mobile phone. Users can monitor and control temperature, irrigation, and light intensity changes via their mobile phones. Soheli et al. [3] developed a smart greenhouse system based on fuzzy inference and IoT. The system uses real-time sensors

© 2026 Industrial University of Ho Chi Minh City

to collect data on temperature, air humidity, sunlight, and soil moisture. These data serve as input variables for a fuzzy controller, which performs inference and sends control signals to fans, heaters, and spray pumps to adjust temperature and humidity to optimal levels for crops. The system is implemented using the Arduino ATmega8233 microcontroller. Users can monitor greenhouse information through a website and mobile devices. Reference [4] presents a real-time smart greenhouse system consisting of multiple wireless sensor nodes, wireless actuator nodes, and a website to receive data from sensor nodes, process the information, and send commands to actuator nodes to control greenhouse parameters according to crop growth requirements. The wireless sensor nodes include multiple sensors measuring parameters such as temperature, air humidity, light intensity, soil temperature, soil moisture, and air pressure. These sensors are connected to an Arduino MKR ENV Shield board; data collected from the sensors is processed by the MKR Arduino microcontroller and sent to a Gateway via the LoRaWAN wireless network. The data is then transmitted from the Gateway to the web server via a WiFi network. Similarly, commands from the website are sent to the Gateway via WiFi, and the Gateway forwards them to actuator nodes via the LoRaWAN network. The actuator nodes are implemented using the Arduino MKR WAN1310 board, connected to multiple intermediate relays to control actuators such as fans, pump motors, and lights.

Along with the development of Artificial Intelligence (AI), computer vision and machine learning are increasingly being applied in agriculture. Computer vision techniques, combined with high-quality image acquisition using remote cameras, create non-contact measurement solutions in agriculture. This transforms agricultural operations, opening up a series of new applications in the agricultural sector, thereby saving production costs and increasing productivity. In the field of crop growth analysis and monitoring, with advances in computer vision and deep learning, many new solutions have been used to determine the overall health status of crops. These solutions can identify plant diseases, water stress, and nutrient deficiencies, enabling timely detection of plant condition or disease status, thereby preventing losses caused by the spread of plant disease, ultimately leading to improved crop quality. The study in [5] identifies the status of crops through changes in leaf color and temperature. Similarly, in reference [6], Xu et al. identified symptoms related to nutrient deficiencies through leaf color and texture in soilless cultivation environments, where plants are commonly nutrient-deficient. Image analysis for detecting plant nutrient deficiency through leaf samples has been evaluated as effective, achieving an accuracy of 82.5%. Hughes and Salathe [7], Singh et al.[8] created the PlantVillage and PlantDoc databases of healthy and diseased plant leaves, as well as plant diseases. In addition, there are many other studies using computer vision and deep learning to locate disease spots and provide accurate disease information [9], [10], [11] as well as to identify water stress and nutrient deficiency[12], [13]. Also in the field of plant condition analysis and crop growth, many studies use computer vision and deep learning to analyze plant phenotypes. Plant phenotypic analysis helps agronomists determine the optimal growing environment and appropriate nutrient concentrations for crops, and detect and conserve valuable genetic varieties, etc [14]. Plant phenotype includes many parameters such as root morphology, biomass, leaf characteristics, and fruit characteristics. Previously, manual measurement and phenotypic analysis (using rulers, weighing balances) was very labor-intensive, expensive, and time-consuming. Automated, non-invasive, rapid, high-accuracy quantification based on computer vision has become a core area attracting a great deal of current research. Arend et al. [15], Seren et al.[16] constructed database systems to efficiently store, manage, and retrieve large volumes of plant phenotypic data. In references [17], [18], [19], the authors utilized pattern recognition and computer vision, combined with data mining methods and machine learning algorithms, to investigate, analyze, or evaluate non-invasive image-based phenotypic data.

Along with the development of Agriculture 4.0, the integration of the IoT and AI has revolutionized agriculture with through continuous environmental monitoring. Recent studies have emphasized the role of IoT and AI in optimizing irrigation and fertilization strategies, mitigating adverse environmental effects on crops, and improving resource use efficiency [20], [21], [22].

The objective of this paper is to apply emerging technologies, including IoT, computer vision, and deep learning, to support the development of smart agriculture in Vietnam. This paper designs an intelligent greenhouse monitoring and control system based on an IoT platform, with particular emphasis on an embedded computer vision module for non-invasive and automated plant growth measurement. Environmental parameters, including temperature, air humidity, soil moisture, and light intensity, are collected through the sensor system, while plant phenotypic information, particularly stem regions and plant

height, is extracted from RGB images using a lightweight deep learning-based segmentation model deployed on a Raspberry Pi. The collected environmental data and image-based growth measurements are transmitted to a database and displayed through a web application or mobile device, enabling users to remotely monitor both greenhouse conditions and plant development. Compared with previous studies [1]–[4], which mainly focus on IoT-based collection of greenhouse environmental parameters, the proposed system integrates edge-based computer vision to perform non-invasive plant phenotypic measurement for evaluating crop growth and development. Furthermore, while existing studies combining IoT and computer vision have investigated tomato fruit diameter for growth rate analysis [20], tomato ripening stage identification [21], or leaf coverage monitoring [22], this paper focuses on stem-level segmentation and calibrated plant height estimation as direct indicators of tomato growth. By processing images locally on the Raspberry Pi, the proposed approach reduces dependence on cloud computation, improves response time, and enhances the feasibility of low-cost deployment in practical greenhouse environments.

The remainder of this paper is organized as follows: Section 2 details the system design and implementation; Section 3 discusses the experimental results and discussion; and Section 4 concludes the paper.

2. SYSTEM DESIGN AND IMPLEMENTATION

2.1 Overall Architecture

This paper develops a greenhouse simulation model with the dimensions shown in Figure 1. The system architecture consists of four main functional layers: i) the STM32 Slave data acquisition module, ii) the STM32 Master central control module, iii) the computer vision module comprising a Raspberry Pi 4 and a camera, and iv) the storage and display layer including Firebase, and a web app.

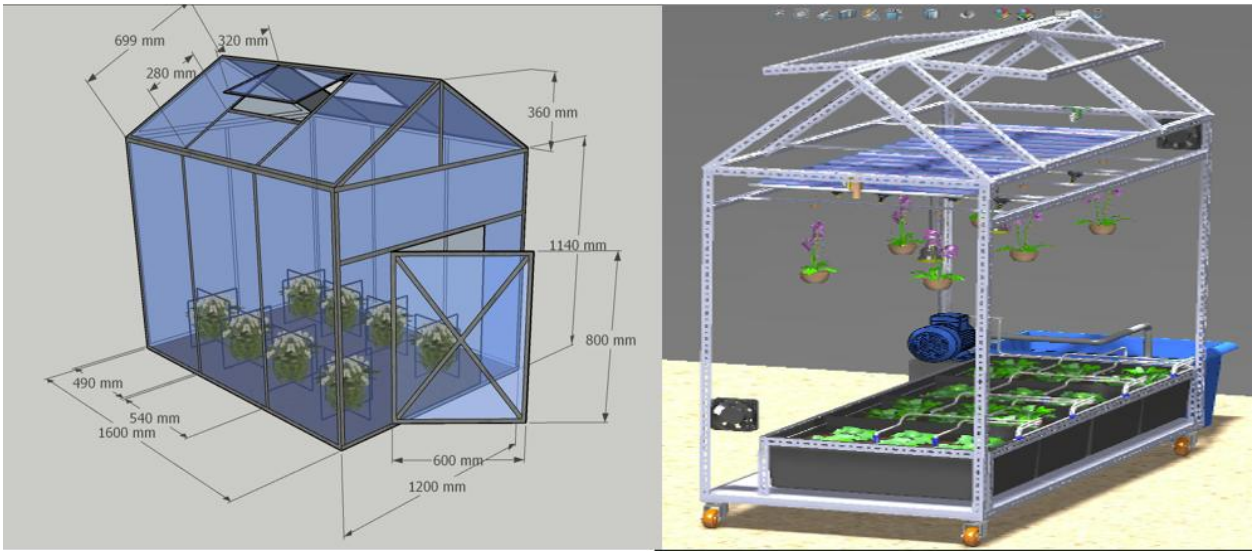


Figure 1. Simulation model of greenhouse and the dimensions

Figure 2 illustrates the block diagram and workflow of the system. The environmental parameters collected by sensors and processed by the STM32 Slave microcontroller include temperature, air humidity, soil moisture, and light intensity; these are crucial input variables reflecting the operational status of the greenhouse. The collected and processed environmental parameters are then transmitted by the STM32 Slave to the STM32 Master via the RS485 communication standard.

In the computer vision module, images of the plants are automatically captured by the camera and transmitted to the Raspberry Pi via a USB port. The Raspberry Pi analyzes the captured images using the proposed StemLite-Seg algorithm, a lightweight deep learning-based stem segmentation model designed for edge deployment. The model extracts the visible stem region from RGB images, after which post-processing and pixel-to-centimeter calibration are applied to estimate plant height. The extracted phenotypic data are then sent to the STM32 Master via the UART communication protocol.

The STM32 Master central control module performs the following tasks: i) receiving greenhouse environmental parameter data from the STM32 Slave; ii) receiving plant phenotypic data from the Raspberry Pi and subsequently transmitting the data to the Firebase platform via a GSM SIM 4G module. The use of the SIM 4G module enables the smart greenhouse to transmit information to Firebase from any location. This facilitates internet connectivity for greenhouses situated far from residential areas, where internet access and Wi-Fi availability are limited; iii) receiving control commands, such as setpoint thresholds for humidity, temperature, and light intensity, from operators via a website or mobile devices; iv) controlling actuators such as pump motors, fan motors, and lights to ensure that the greenhouse environmental parameters remain within the operator's required range.

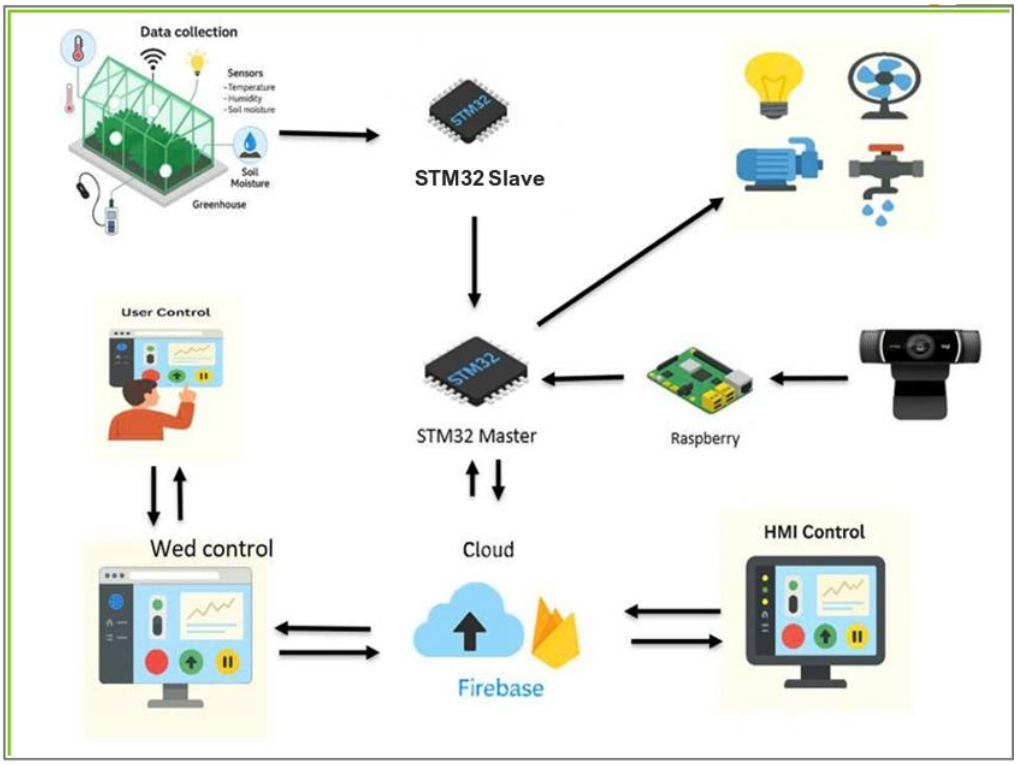


Figure 2. Block diagram and workflow of the system

The data storage and display module is implemented using the Firebase platform, where the Realtime Database is utilized to store data transmitted from the STM32 Master. Firebase Authentication is employed for user identity verification via email and password, while Cloud Functions enable the system to automatically save and delete data when the Realtime Database memory is full. A web application is developed on this platform to allow greenhouse operators to monitor environmental parameters and control settings and greenhouse equipment from any location using devices such as smartphones or computers. Additionally, the system can communicate and display data on an HMI at any location via the ESP32 microcontroller.

2.2 Hardware Modules and Communication

The STM32 Slave module is designed to sample sensor data at multiple locations within the greenhouse, package the data, and transmit it to a central controller. This module incorporated wireless communication standards such as LoraWan and NRF; however, under experimental conditions at IUH, where significant electromagnetic interference from industrial machinery is present, wireless transmission was subject to considerable signal interference, leading to inaccurate results. The research team therefore selected the RS-485 communication standard for data transmission from the STM32 Slave to the STM32 Master.

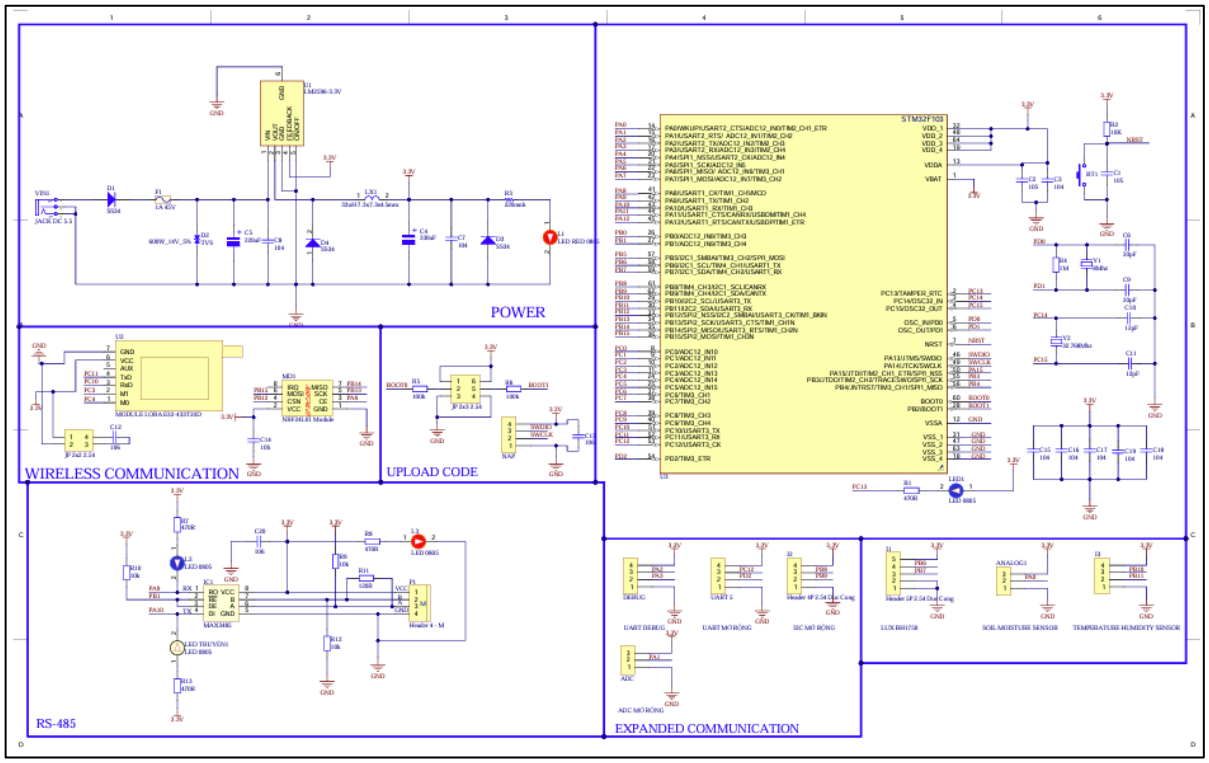


Figure 3. Schematic diagram of the STM32 Slave Module



Figure 4. Board of STM32 Slave Module

The STM32 Master central control module integrates multiple functional components: (i) a communication module providing flexible connectivity options suited to operating conditions and operator requirements, including wireless communication technologies such as an nRF radio transceiver, a LoRaWAN wireless network, and a SIM 4G module, as well as wired communication interfaces such as RS-485, UART, and LAN; (ii) an intermediate relay module for controlling actuators such as pumps, fans, and lighting; and (iii) a manual control panel, enabling operators to directly intervene in the system during emergency situations.

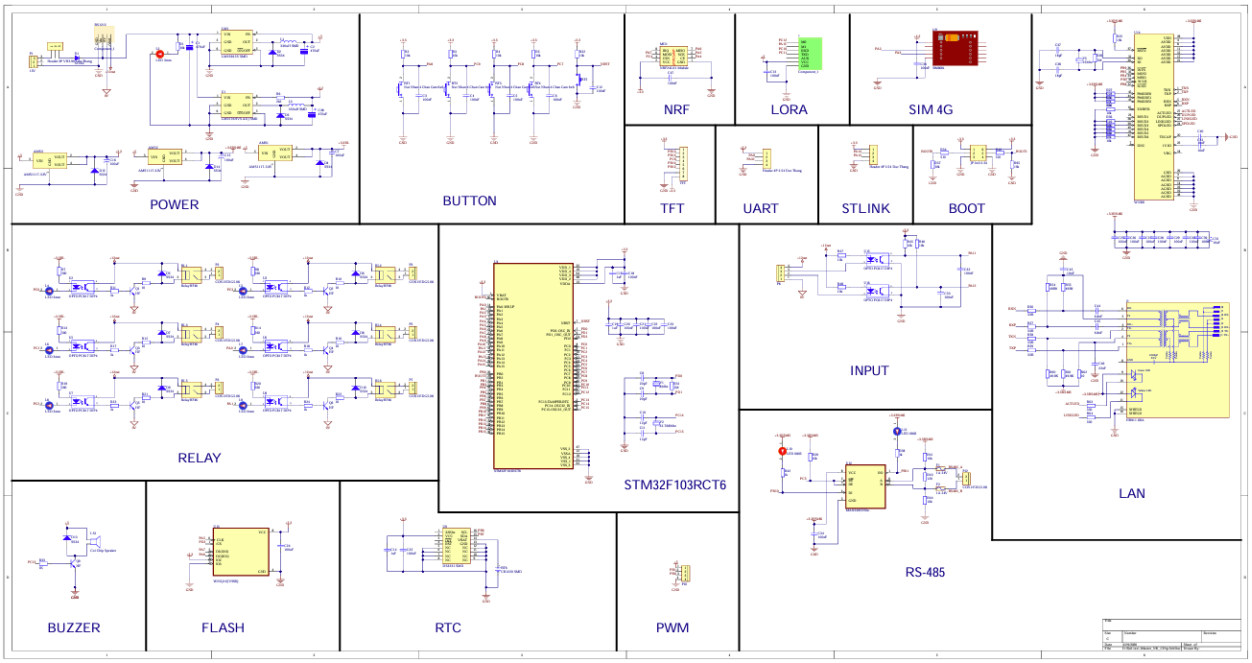


Figure 5. Schematic diagram of the STM32 Master Module

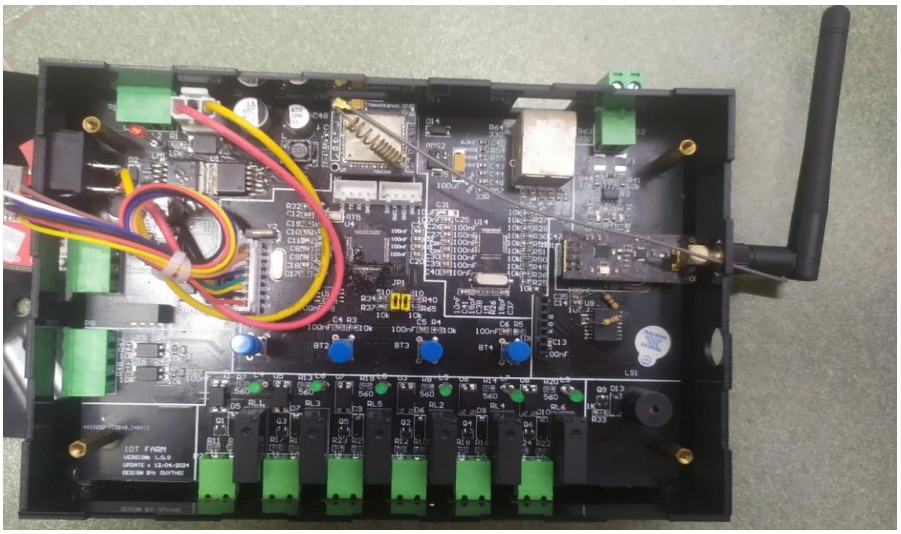


Figure 6. Board of the STM32 Master Module

The computer vision and camera module employs a Logitech C922 Pro HD camera. Images captured by the camera are transmitted to a Raspberry Pi 4 via the USB interface, where image analysis is performed using the proposed lightweight deep learning-based stem segmentation model. The model extracts the visible stem region from RGB images, and the segmented mask is subsequently post-processed to estimate plant height through calibrated pixel-to-centimeter conversion. The extracted phenotypic data are then forwarded to the STM32 Master via the UART interface, as illustrated in Figure 7.

Figure 8 illustrates the control cabinet of the greenhouse model. Both the control cabinet and the greenhouse model are located at X5 – IUH, spaced less than 3 meters apart between the two units.

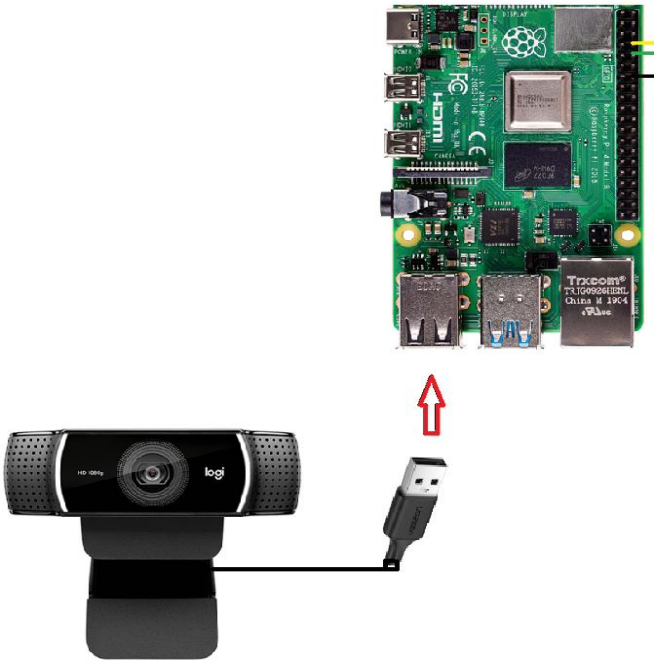


Figure 7. Computer Vision and Camera Module



Figure 8. Exterior and interior views of the control cabinet

2.3 Computer Vision Module

The computer vision module is designed to provide non-invasive plant growth measurements from greenhouse images. In the proposed system, a Logitech C922 Pro HD camera is used as the image acquisition device, while a Raspberry Pi 4 Model B serves as the edge computing unit. The captured images are transmitted to the Raspberry Pi through the USB interface and processed locally to extract plant phenotypic information, including stem height, stem region, and plant count. The extracted results are then

formatted as lightweight JSON messages and sent to the STM32 Master controller via the UART interface for subsequent synchronization with Firebase and visualization on the monitoring platform.

To satisfy the computational constraints of embedded deployment, this study proposes a lightweight deep learning model, referred to as StemLite-Seg, for tomato stem segmentation and growth measurement. Instead of employing a general-purpose object detector only to locate plants, the proposed model directly segments the visible stem region. This design is more suitable for plant growth monitoring because the stem mask provides pixel-level structural information from which height, inclination, and temporal growth trends can be estimated more reliably. The overall pipeline consists of four stages: image acquisition, lightweight stem segmentation, morphological measurement, and data transmission.

Given an input RGB image $I \in R^{H\times W\times 3}$, the image is first resized to a fixed resolution suitable for Raspberry Pi inference, and its pixel values are normalized. The proposed segmentation network then predicts a binary stem probability map:

$$M = f_\theta(I) \tag{1}$$

where $f_\theta(.)$ denotes the proposed lightweight segmentation model and $M \in [0,1]^{H\times W}$ represents the probability of each pixel belonging to the stem class. A thresholding operation is subsequently applied to obtain the binary stem mask:

$$\widehat{M}(x,y) = \begin{cases} 1, & M(x,y) \geq \tau \\ 0, & M(x,y) < \tau \end{cases} \tag{2}$$

where τ is the segmentation threshold. The largest connected stem component is then retained to suppress small false-positive regions caused by leaves, soil, shadows, or greenhouse background objects.

The proposed StemLite-Seg architecture follows an encoder-decoder design [23] optimized for edge inference. The detailed architecture of StemLite-Seg is presented in Figure 9. Each captured frame is first restricted to the calibrated plant region of interest and letterbox-resized to 320×320 pixels while preserving its aspect ratio. The input is normalized to [0,1] before being passed to the network. StemLite-Seg adopts a four-stage encoder and a three-stage decoder with an overall encoder output stride of 16. This relatively small downsampling factor is selected because tomato stems occupy only a narrow portion of the image and may disappear when the feature maps are excessively downsampled.

The encoder starts with a 3×3 convolution with stride 2 and 16 output channels. It is followed by four feature extraction stages producing feature maps at 1/2, 1/4, 1/8, and 1/16 of the input resolution. Except for the first convolution, the encoder is constructed from depthwise separable convolution (DSConv) blocks [24]. Each DSConv block consists of a 3×3 depthwise convolution followed by a 1×1 pointwise convolution. Batch normalization and ReLU6 activation are applied after each convolution. For an input with C_{in} channels and C_{out} output channels, this block reduces the convolutional cost from $9C_{in}C_{out}$ to $9C_{in} + C_{in}C_{out}$, making it suitable for CPU-based inference on the Raspberry Pi 4.

The bottleneck produces a 20×20×96 representation containing high-level stem semantics. In each decoder stage, the feature map is bilinearly upsampled by a factor of two and concatenated with the encoder feature map having the same spatial resolution. A 1×1 convolution first compresses the concatenated channels, after which a DSConv block performs spatial refinement. The three decoder stages generate feature maps of 40×40×64, 80×80×32, and 160×160×16, respectively. These skip connections preserve fine spatial cues that are important for recovering thin stem structures and partially occluded boundaries.

To improve the current vision module, two specific enhancements are introduced. First, A Stem-Oriented Attention Module (SOAM) is applied to the last decoder feature map. Given $F \in R^{H_f\times W_f\times C_f}$, two parallel depthwise asymmetric convolutions with kernels of 7×1 and 1×7 are used to aggregate vertical continuity and horizontal contrast, respectively. Their outputs are concatenated and projected to a one-channel spatial attention map using a 1×1 convolution and a sigmoid function. The attention-enhanced feature is computed as:

$$A_v = \sigma(Conv_{1\times 1}[DWConv_{7\times 1}(F); DWConv_{1\times 7}(F)]) \, , \, F' = F \odot A_v \tag{3}$$

where [;] denotes channel-wise concatenation, $\sigma(.)$ is the sigmoid function, and the single-channel attention map A_v is broadcast along the channel dimension. The residual formulation retains the original decoder information while strengthening spatially continuous responses associated with elongated stems.

Second, an Edge-Guided Refinement Branch is added to improve mask boundary quality. The Edge-Guided Refinement Branch receives the shallow 160×160×16 encoder feature map because this feature retains stronger boundary information than the bottleneck representation. A DSConv block reduces it to eight channels, and a 1×1 convolution predicts a one-channel edge probability map. This edge map is concatenated with F', followed by a 3×3 DSConv refinement block with 16 output channels. The refined feature is then upsampled to the input resolution and passed through a 1×1 convolution and a sigmoid function to produce the final 320×320 stem probability map. Before morphological measurement, the predicted mask is mapped back to the original calibrated region of interest so that image resizing does not alter the pixel-to-centimeter relationship.

During training, the network is optimized using a combined segmentation loss:

$$L = L_{seg} + \lambda L_{edge} \tag{4}$$

where L_{seg} is the binary segmentation loss, L_{edge} is the auxiliary edge loss, and λ is a weighting coefficient. The segmentation loss is implemented as a combination of binary cross-entropy and Dice loss to handle the class imbalance between the small stem region and the dominant background area:

$$L_{seg} = L_{BCE} + L_{Dice} \tag{5}$$

After segmentation, morphological processing is applied to convert the predicted stem mask into measurable growth indicators. The top and bottom points of the stem mask are identified along the vertical axis, and the stem height in pixels is calculated as:

$$h_{px} = y_{bottom} - y_{top} \tag{6}$$

To convert the pixel measurement into a real-world value, a calibration factor is estimated using a reference object with known height placed in the image plane:

$$h_{cm} = h_{px} \times s \tag{7}$$

where s denotes the pixel-to-centimeter conversion factor. When multiple plants are present, connected component analysis is used to separate individual stem regions, and the height of each plant is computed independently.

To improve temporal stability, the system further applies a lightweight confidence-based smoothing strategy. If the segmentation confidence of the current frame is lower than a predefined threshold, the measurement is not immediately updated. Instead, the previous valid value is retained or updated using an exponential moving average:

$$H_t = \alpha h_t + (1 - \alpha) H_{t-1} \tag{8}$$

where h_t is the current measured height, H_t is the smoothed height at time t, and α is the smoothing coefficient. This step reduces abnormal fluctuations caused by motion blur, lighting changes, partial occlusion, or temporary segmentation errors.

The output of the computer vision module is encoded in JSON format to simplify communication with the STM32 Master controller. This format is suitable for UART transmission and can be directly parsed by the STM32 Master before being uploaded to Firebase. By performing image processing and segmentation locally on the Raspberry Pi, the proposed module reduces network dependence, protects the system from communication interruptions, and enables near real-time plant growth monitoring at the greenhouse edge.

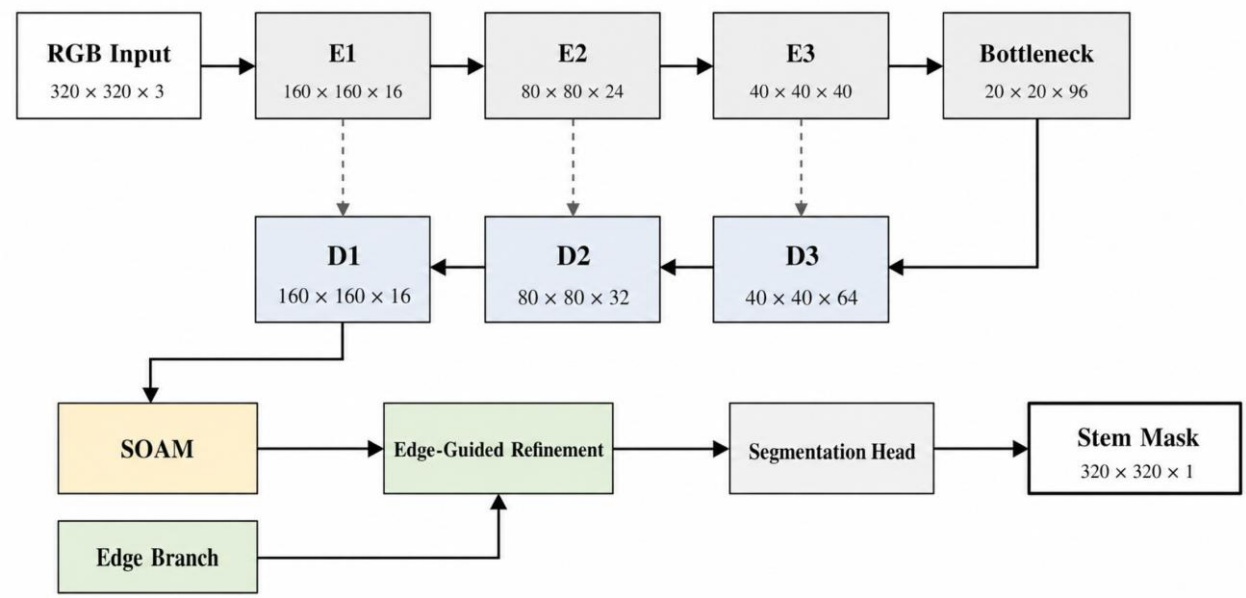


Figure 9. Detailed architecture of the proposed StemLite-Seg network

3. EXPERIMENTAL RESULTS AND DISCUSSION

3.1 Greenhouse Environmental Data

The proposed system was evaluated on a tomato-growing greenhouse model, with data recorded four times daily (at 10:00, 13:00, 17:00, and 21:00) over the period from December 6 to 11, 2025, using the following sensors and actuator activation thresholds:

The SHT30 Humidity and Temperature Sensor is used to measure ambient air temperature and relative humidity. For relative humidity measurement, the sensor has a measurement range of 0 to 100% RH, a typical accuracy of ±2% RH within the range of 10 to 90% RH, a 14-bit resolution, and a response time $\tau_{63\%}$of 8 seconds. For temperature measurement, the sensor has a measurement range of −40 to 125°C, a typical accuracy of ±0.3°C within the range of 0 to 65°C, and a typical resolution of 0.015°C. The sensor uses the I2C communication protocol and is equipped with a 1 m cable. The air temperature threshold for activating the ventilation fan is set to 27°C.

The IP65 Capacitive Soil Moisture Sensor is used to measure soil moisture. Its technical specifications include a supply voltage of 3.3 to 5.5 VDC, an analog output voltage of 0 to 3 VDC, and a cable length of 1.5 m. The irrigation pump is activated when the soil moisture level decreases to 57% and is deactivated when the soil moisture level reaches 80%.

The I2C IP68 Waterproof Ambient Light Sensor is used to measure ambient light intensity. Its technical specifications are as follows: detection range of 1 to 65,535 lx, accuracy of 1.2 lx, I2C communication interface, operating temperature range of −40 to 85°C, and a cable length of 1 m. The light intensity threshold for activating the lighting system is set to 140 lx.

Table 1 summarizes the key statistics for experimental period. It can be observed that the sensors provided sampling interval data that was successfully synchronized to the storage system and consistently displayed on the monitoring interface.

Table 1. Statistical summary of greenhouse environmental data for period December 6-11, 2025.

Parameter	Minimum Value	Maximum Value	Average Value
Temperature (°C)	27.8	38.8	31.61
Air Humidity (%)	40.2	77.0	57.48
Soil moisture (%)	52.0	82.4	68.99

Light Intensity (lux)	92	14.832.7	5.550.58

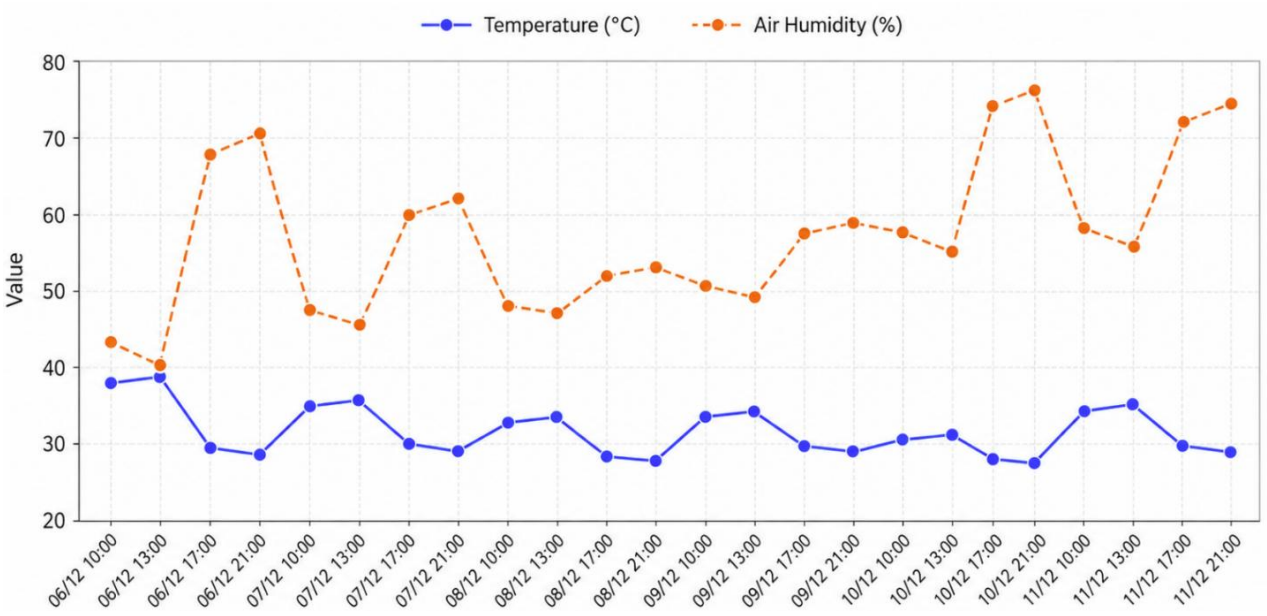


Figure 10. Temperature and air humidity variation of greenhouse

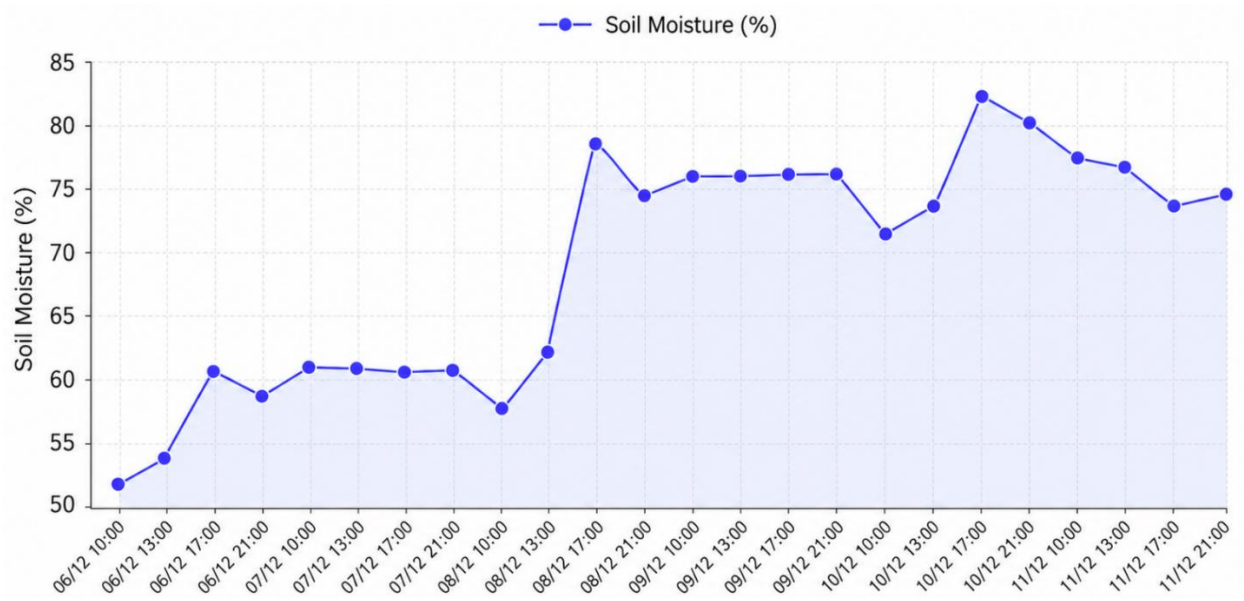


Figure 11. Soil moisture variation of greenhouse

Figures 10-12 present the operational data of the greenhouse from December 6, 2025 to December 11, 2025. The results indicate that the temperature inside the greenhouse fluctuated over a relatively wide range, from 27.8 to 38.8°C; air humidity varied between 40.2 and 74.5%; soil moisture was maintained within the range of 52.0–82.4%; and light intensity exhibited significant variation across different times of day. These variations accurately reflect the characteristic dynamics of greenhouse cultivation environments and demonstrate that the system is capable of capturing both short-term trends and rapid transient changes.

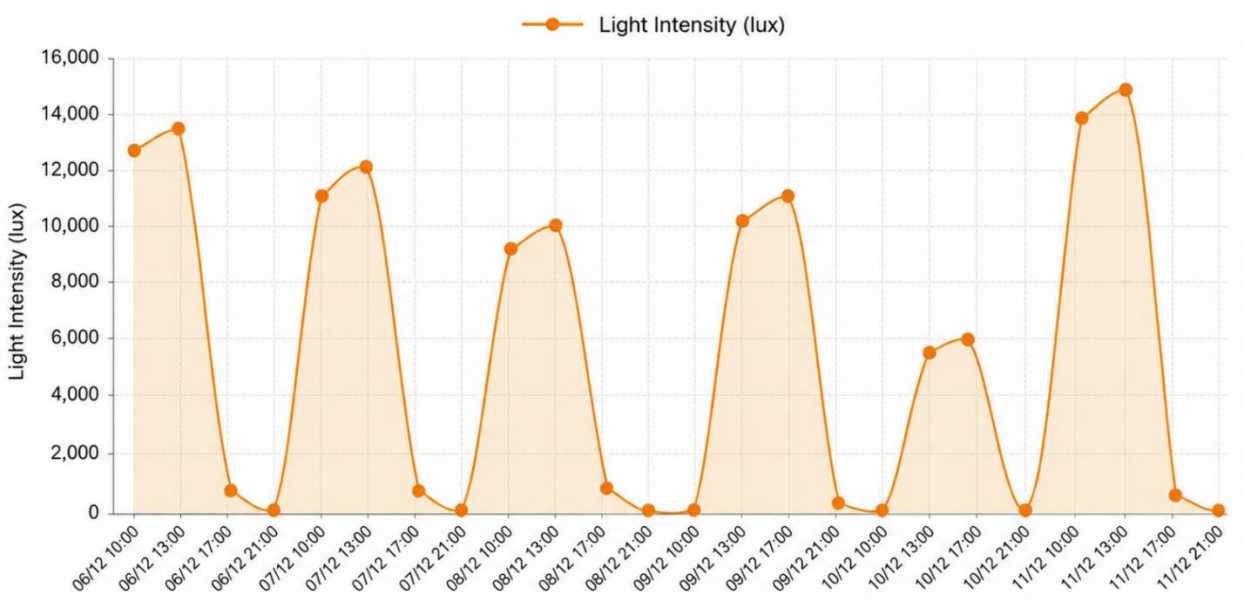


Figure 12. Light intensity variation of greenhouse

From an operational standpoint, sensor data transmission from the STM32 Slave to the STM32 Master via RS-485 was stable, and image data processed on the Raspberry Pi was consistently transferred via UART prior to synchronization with Firebase. This establishes the foundation for an integrated monitoring loop combining environmental data and computer vision data within a unified platform.

3.2 Image Dataset and Evaluation Protocol

Tomato stem images were acquired using the Logitech C922 Pro HD camera installed in the experimental greenhouse. For plant height analysis, only the image samples collected at 10:00 and 17:00 each day were selected, over the period from December 6 to December 11, 2025, resulting in 12 acquisition sessions. The camera was maintained at a fixed position and viewing distance throughout the experiment. Images were captured at an original resolution of 1920×1080 pixels under naturally varying greenhouse conditions, including changes in illumination, leaf-stem occlusion, shadows, stem thickness, and background complexity. After removing severely blurred and highly redundant frames, the resulting dataset contained 720 RGB images collected from 10 tomato plants. Each plant was represented by approximately 72 images distributed across the 12 acquisition sessions. The dataset included both clearly visible stems and challenging cases in which the stem was partially obscured by leaves or exhibited low contrast against soil, supporting structures, pots, and other greenhouse objects.

Each visible stem was manually delineated at the pixel level using the Computer Vision Annotation Tool (CVAT). The annotation included the main visible stem from its defined base to the highest observable stem point, while leaves, petioles, supporting strings, soil, and background structures were assigned to the background class. All masks were reviewed by a second annotator to correct disconnected stem regions and ambiguous boundaries caused by leaf-stem overlap. The annotated masks were stored as binary images having the same spatial resolution as the corresponding RGB images. To prevent images of the same plant from appearing in both the training and test subsets, the dataset was divided at the plant level. Images from six plants were used for training, images from two plants were used for validation, and images from the remaining two plants were reserved for testing. This produced 432 training images, 144 validation images, and 144 test images, corresponding to a ratio of 60%, 20%, and 20%, respectively. Thus, the test set contained plants that were not observed during training, providing a more realistic assessment of model generalization than a random frame-level split.

All images and masks were letterbox-resized to 320×320 pixels. Bilinear interpolation was used for RGB images, whereas nearest-neighbor interpolation was used for binary masks to preserve discrete class labels. During training, data augmentation included horizontal flipping with a probability of 0.5, random rotation within $\pm 10^0$, brightness adjustment within ±20%, contrast adjustment within ±15%, and Gaussian blur with a probability of 0.2. Vertical flipping was not applied because it would violate the natural vertical

orientation of the tomato stems. StemLite-Seg was trained for a maximum of 150 epochs using the AdamW optimizer with an initial learning rate of 1×10^{-3}, a weight decay of 1×10^{-4}, and a batch size of 16. A cosine annealing strategy reduced the learning rate to 1×10^{-6}. Training was terminated when the validation Dice score did not improve for 20 consecutive epochs. The edge-loss coefficient in Eq. (4) was set to λ=0.2. The checkpoint obtaining the highest validation Dice score was selected for final evaluation, and the test subset was not used for model selection or hyperparameter tuning.

Segmentation performance was evaluated using precision, recall, intersection over union (IoU), Dice score, and boundary F1-score (BF1). Precision measures the proportion of predicted stem pixels that correspond to the annotated stem, whereas recall measures the proportion of annotated stem pixels recovered by the model. IoU and Dice evaluate the region overlap between the predicted and ground-truth masks. Because tomato stems are thin structures for which small boundary displacements can affect morphological measurements, BF1 was additionally computed using a boundary tolerance of two pixels at the 320×320 evaluation resolution. All metrics were calculated for each test image and then macro-averaged over the 144-image test subset, preventing images with larger stem regions from dominating the reported results. The segmentation threshold τ in Eq. (2) was set to 0.5 for all evaluated models.

3.3 Plant Growth Monitoring Results

The quantitative stem segmentation results are reported in Table 2. StemLite-Seg was compared with the conventional U-Net and a MobileNetV2-U-Net baseline. U-Net represents a commonly used encoder–decoder segmentation architecture, whereas MobileNetV2-U-Net provides a more computationally efficient baseline based on a lightweight backbone. All models were trained and evaluated using the same plant-level data split, input resolution, augmentation procedure, and model-selection criterion. Two ablation experiments were further conducted to evaluate the individual contributions of the Stem-Oriented Attention Module and the Edge-Guided Refinement Branch. In the first variant, SOAM was removed and its input was passed directly to the refinement stage. In the second variant, the edge-prediction branch and its auxiliary loss were removed, and the final decoder feature was used directly for segmentation refinement. Except for the investigated component, all remaining architectural and training settings were kept unchanged.

As shown in Table 2, the conventional U-Net achieved an IoU of 79.74% and a Dice score of 88.73%. Although its segmentation performance was reasonable, the model required 7.760 million parameters and 15.67 GMACs, making it less suitable for direct deployment on the Raspberry Pi. Replacing the conventional encoder with MobileNetV2 increased the IoU and Dice scores to 81.96% and 90.09%, respectively, while considerably reducing the computational requirements. Nevertheless, MobileNetV2-U-Net still contained approximately 2.410 million parameters and required 1.29 GMACs per image. The complete StemLite-Seg achieved a precision of 93.62%, recall of 92.08%, IoU of 86.63%, Dice score of 92.84%, and BF1 score of 89.57%. Compared with MobileNetV2-U-Net, the proposed model improved the Dice score by 2.75 percentage points and the BF1 score by 4.36 percentage points. Simultaneously, StemLite-Seg used approximately 43 times fewer parameters and 8.6 times fewer multiply-accumulate operations. These results demonstrate that the proposed task-specific design provides a more favorable accuracy-complexity trade-off for embedded tomato stem segmentation.

Removing SOAM reduced recall from 92.08% to 89.64% and BF1 from 89.57% to 86.37%. The larger reduction in recall indicates that the directional attention mechanism contributes to recovering weak or partially occluded stem regions. Without SOAM, the model was more likely to produce discontinuities where leaves crossed the stem or where the stem exhibited low contrast against the greenhouse background. Removing the Edge-Guided Refinement Branch reduced IoU from 86.63% to 84.67% and BF1 from 89.57% to 84.96%. The 4.61-percentage-point reduction in BF1 was larger than the corresponding reduction in region-level Dice score, indicating that the edge branch primarily improves boundary localization rather than only increasing the overall number of correctly classified pixels. This property is important because small boundary errors in narrow stem regions may propagate to the subsequent estimation of stem height and width.

Figure 13 presents representative segmentation outputs obtained from tomato plant images captured inside the greenhouse. The predicted masks are overlaid on the original images to visualize the regions identified by the proposed lightweight segmentation model. As observed, the model is able to localize the main stem regions under practical greenhouse conditions, including non-uniform illumination, complex

background, overlapping leaves, and partial occlusion. These results demonstrate that the proposed vision module can extract meaningful plant structural information from ordinary RGB images without requiring destructive sampling or manual measurement. The segmentation results in Figure 13 also indicate that the model can distinguish elongated stem structures from surrounding leaves, soil, pots, and background objects. This is important for growth monitoring because stem height estimation is highly dependent on the accurate extraction of the visible stem region. Although some small discontinuities and boundary variations may occur due to leaf occlusion and image blur, the predicted masks remain sufficiently consistent for estimating plant height. Therefore, the proposed segmentation-based approach provides a practical basis for automated phenotypic measurement in a low-cost greenhouse monitoring system.

Table 2. Quantitative comparison and ablation study on the held-out tomato stem test set.

Method	**Parameters (M)**	**GMACs**	**Precision (%)**	**Recall (%)**	**IoU (%)**	**Dice (%)**	**BF1 (%)**
U-Net	7.760	15.67	90.12	87.38	79.74	88.73	83.02
MobileNetV2-U-Net	2.410	1.29	91.47	88.76	81.96	90.09	85.21
StemLite-Seg without SOAM	0.0558	0.1436	92.38	89.64	83.47	91.00	86.37
StemLite-Seg without edge branch	0.0558	0.1423	92.91	90.52	84.67	91.70	84.96
Complete StemLite-Seg	0.0561	0.1501	93.62	92.08	86.63	92.84	89.57

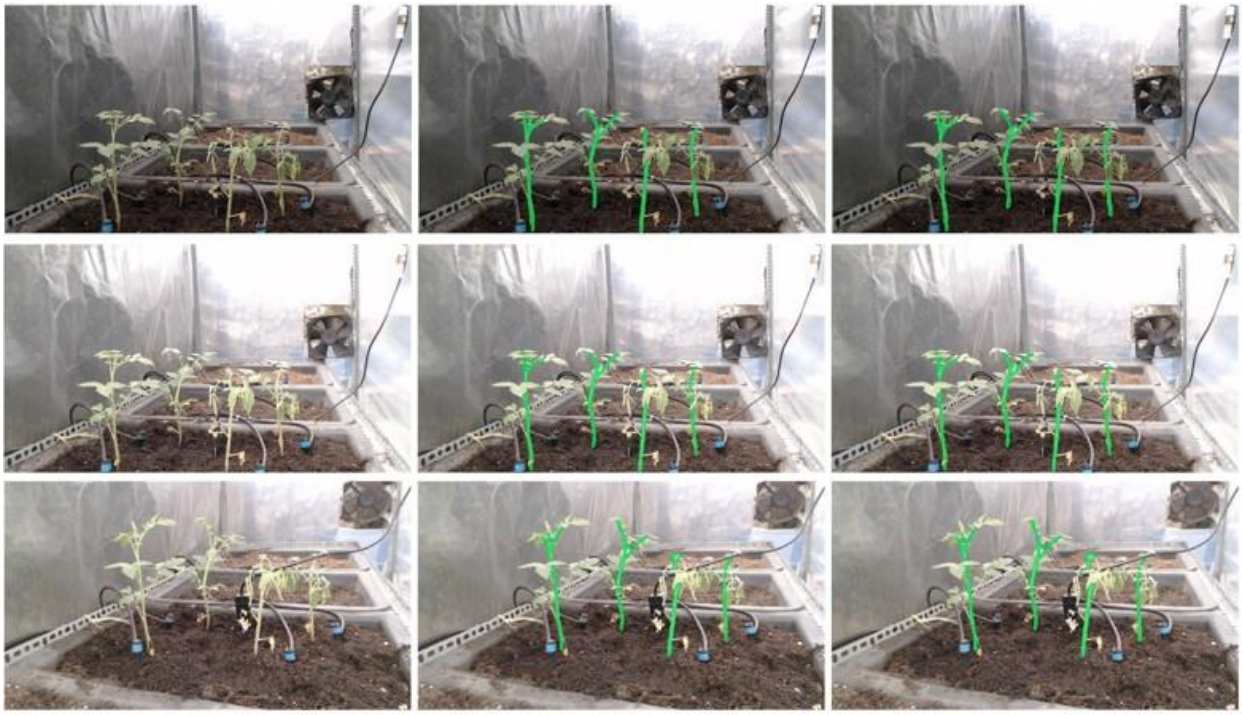


Figure 13. Representative stem segmentation results under different greenhouse conditions. From left to right: input image, annotated ground-truth mask, StemLite-Seg prediction

After obtaining the segmented stem masks, post-processing was applied to remove small noisy regions and retain the main connected stem components. The plant height was then estimated based on the vertical distance between the uppermost and lowermost points of the segmented stem region. A pixel-to-centimeter calibration factor was used to convert image-based measurements into real-world height values. The resulting plant height data were transmitted from the Raspberry Pi to the STM32 Master controller and synchronized with the monitoring platform.

Figure 14 shows the variation in tomato plant height during the experimental period from December 6 to December 11, 2025. The plant height increased from approximately 65.8 cm at the beginning of the experiment to approximately 77.0 cm at the end of the monitoring period. This corresponds to an overall increase of about 11.2 cm, indicating a clear upward growth trend. The height curve shows a gradual increase over time, with no abnormal decrease or large fluctuation, suggesting that the proposed computer vision module produced stable measurements across consecutive observations. The measured height trend reflects the growth progression of tomato plants under the monitored greenhouse conditions. In particular, the continuous increase in plant height indicates that the environmental conditions recorded during the experiment, including temperature, air humidity, soil moisture, and light intensity, were generally suitable for tomato development. Combining these visual growth measurements with environmental sensor data allows agronomists and greenhouse operators to evaluate plant responses to cultivation conditions more effectively.

Overall, the results confirm the feasibility of using an edge-deployed lightweight segmentation model for non-invasive plant growth monitoring. The proposed computer vision module can provide both qualitative visual evidence through segmentation masks and quantitative growth indicators through plant height estimation. These results demonstrate the potential of the developed system for practical greenhouse applications and provide a foundation for future extensions, such as stem width estimation, leaf area measurement, canopy coverage analysis, and growth rate prediction.

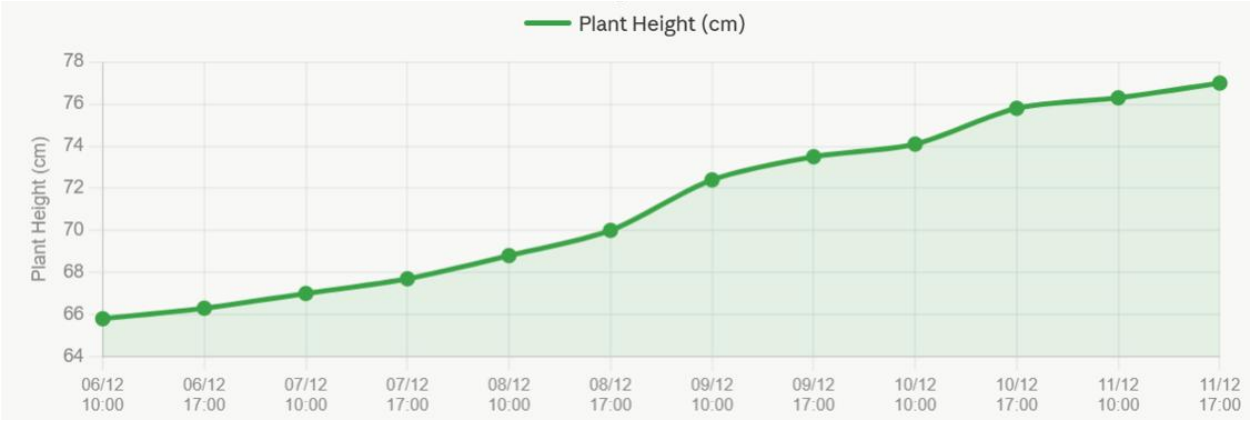


Figure 14. Plant height variation in greenhouse

4. CONCLUSION

This paper has presented the design of a smart greenhouse measurement and monitoring system based on IoT and computer vision. The system enables the collection of environmental parameters, including temperature, air humidity, soil moisture, and light intensity, through a sensor network, while plant morphological information, particularly the stem region and plant height, is extracted from RGB images using a lightweight deep learning-based segmentation model deployed on a Raspberry Pi. By processing images locally on the Raspberry Pi, the proposed approach reduces dependence on cloud computing, improves response time, and enhances the feasibility of low-cost deployment in real greenhouse environments.

Experimental results demonstrate that the system operates stably at the prototype level, supporting both on-site and remote control and monitoring, thereby fulfilling the objectives of a university-level research project focused on mastering IoT, computer vision, and machine learning technologies in smart agriculture. The prototype confirms the potential of a low-cost smart greenhouse system and provides a practical foundation for future scaling toward smart greenhouses with richer agronomic indicators, such as stem width, leaf area, canopy coverage, growth rate estimation, and control of additional environmental parameters such as CO_2 concentration and soil NPK levels.

ACKNOWLEDGMENT

This research is supported by Industrial University of Ho Chi Minh City (IUH) under grant number 72/HD-DHCN, dated day 22 month 12 year 2025. The authors gratefully acknowledge the support of IUH for providing the resources and favorable conditions to carry out this research.

REFERENCES

[1] E. Collado, E. Valdés, A. García, and Y. Sáez, "Design and implementation of a low-cost IoT-based agroclimatic monitoring system for greenhouses," *AIMS Electronics and Electrical Engineering*, vol. 5, no. 4, pp. 251–283, 2021, doi: 10.3934/electreng.2021014.

[2] A. Kumar, V. Singh, S. Kumar, S. P. Jaiswal, and V. S. Bhadoria, "IoT enabled system to monitor and control greenhouse," in *Materials Today: Proceedings*, Elsevier Ltd, 2020, pp. 3137–3141. doi: 10.1016/j.matpr.2020.11.040.

[3] S. J. Soheli, N. Jahan, M. B. Hossain, A. Adhikary, A. R. Khan, and M. Wahiduzzaman, "Smart Greenhouse Monitoring System Using Internet of Things and Artificial Intelligence," *Wirel. Pers. Commun.*, vol. 124, no. 4, pp. 3603–3634, Jun. 2022, doi: 10.1007/s11277-022-09528-x.

[4] D. I. Săcăleanu, M. G. Matache, Ștefan G. Roșu, B. C. Florea, I. P. Manciu, and L. A. Perișoară, "IoT-Enhanced Decision Support System for Real-Time Greenhouse Microclimate Monitoring and Control," *Technologies (Basel).*, vol. 12, no. 11, Nov. 2024, doi: 10.3390/technologies12110230.

[5] H. E. Nilsson, "Remote sensing and image analysis in plant pathology," *Canadian Journal of Plant Pathology*, vol. 17, no. 2, pp. 154–166, 1995, doi: 10.1080/07060669509500707.

[6] G. Xu, F. Zhang, S. G. Shah, Y. Ye, and H. Mao, "Use of leaf color images to identify nitrogen and potassium deficient tomatoes," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1584–1590, Aug. 2011, doi: 10.1016/j.patrec.2011.04.020.

[7] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[8] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A Dataset for Visual Plant Disease Detection," Nov. 2019, doi: 10.1145/3371158.3371196.

[9] J. Liu and X. Wang, "Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network," *Front. Plant Sci.*, vol. 11, Jun. 2020, doi: 10.3389/fpls.2020.00898.

[10] C. H. Son, "Leaf spot attention networks based on spot feature encoding for leaf disease identification and detection," *Applied Sciences (Switzerland)*, vol. 11, no. 17, Sep. 2021, doi: 10.3390/app11177960.

[11] M. P. Mathew and T. Y. Mahesh, "Leaf-based disease detection in bell pepper plant using YOLO v5," *Signal Image Video Process.*, vol. 16, pp. 841–847, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:244231228

[12] A. Abdalla, H. Cen, L. Wan, K. Mehmood, and Y. He, "Nutrient Status Diagnosis of Infield Oilseed Rape via Deep Learning-Enabled Dynamic Model," *IEEE Trans. Industr. Inform.*, vol. 17, no. 6, pp. 4379–4389, Jun. 2021, doi: 10.1109/TII.2020.3009736.

[13] P. Jahagirdar and S. V. Budihal, "Framework to Detect NPK Deficiency in Maize Plants Using CNN," in *Advances in Intelligent Systems and Computing*, 2021. doi: 10.1007/978-981-15-6353-9_33.

[14] Z. Li, R. Guo, M. Li, Y. Chen, and G. Li, "A review of computer vision technologies for plant phenotyping," 2020. doi: 10.1016/j.compag.2020.105672.

[15] D. Arend, A. Junker, U. Scholz, D. Schüler, J. Wylie, and M. Lange, "PGP repository: A Plant phenomics and genomics data publication infrastructure," *Database*, vol. 2016, 2016, doi: 10.1093/database/baw033.

[16] Ü. Seren *et al.*, "AraPheno: A public database for Arabidopsis thaliana phenotypes," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D1054–D1059, Jan. 2017, doi: 10.1093/nar/gkw986.

[17] P. C. Sen, M. Hajra, and M. Ghosh, *Emerging Technology in Modelling and Graphics*, vol. 937. 2020.

[18] Y. Zhou *et al.*, "An automatic non-invasive classification for plant phenotyping by MRI images: An application for quality control on cauliflower at primary meristem stage," *Comput. Electron. Agric.*, vol. 187, Aug. 2021, doi: 10.1016/j.compag.2021.106303.

[19] M. F. Anshori *et al.*, "An overview of image-based phenotyping as an adaptive 4.0 technology for studying plant abiotic stress: A bibliometric and literature review," Nov. 01, 2023, *Elsevier Ltd*. doi: 10.1016/j.heliyon.2023.e21650.

[20] Y. Li, T. Wang, H. Zu, S. Liu, and L. Zu, "Explainable AI-driven interpretation of environmental drivers of tomato fruit expansion in smart greenhouses using IoT sensing," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-24800-3.

[21] A. Saxena *et al.*, "Deep learning-driven IoT solution for smart tomato farming," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-15615-3.

[22] C. Kang, X. Mu, A. N. Seffrin, F. Di Gioia, and L. He, "A Recursive Segmentation Model for Bok Choy Growth Monitoring with Internet of Things (IoT) Technology in 2 Controlled Environment Agriculture," 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169924012572

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[24] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 1800–1807.

HỆ THỐNG ĐO LƯỜNG VÀ GIÁM SÁT NHÀ KÍNH THÔNG MINH DỰA TRÊN IOT VÀ THỊ GIÁC MÁY TÍNH

HOÀNG THỊ TÚ UYÊN*, NGUYỄN HOÀNH, NGÔ THÀNH QUYỀN, LÂM BÌNH MINH, NGUYỄN TRUNG NHÂN

Khoa Công Nghệ Điện, Trường Đại học Công nghiệp Thành phố Hồ Chí Minh

** Tác giả liên hệ: hoangthituuyen@iuh.edu.vn*

Tóm tắt. Bài báo này trình bày thiết kế của hệ thống đo lường và giám sát nhà kính thông minh tích hợp IoT và thị giác máy tính để giám sát sự phát triển của cây trồng một cách không xâm lấn và từ xa. Hệ thống bao gồm các cảm biến môi trường, các mô-đun điều khiển và thu thập dữ liệu dựa trên STM32, máy tính Raspberry Pi 4, camera USB, mô-đun phân đoạn thân cây dựa trên học sâu nhẹ, cơ sở dữ liệu thời gian thực Firebase và một ứng dụng web để giám sát và điều khiển từ xa. Trong mô-đun thị giác máy tính, hình ảnh cây cà chua được xử lý trực tiếp trên Raspberry Pi để phân đoạn các vùng thân cây và ước tính chiều cao cây từ các phép đo hình ảnh đã hiệu chuẩn, cho phép giám sát kiểu hình dựa trên thiết bị biên mà không cần lấy mẫu bằng cách đo thủ công. Kết quả thực nghiệm của hệ thống trên cây trồng cà chua cho thấy hệ thống có thể thu thập dữ liệu nhiệt độ, độ ẩm không khí, độ ẩm đất và cường độ ánh sáng trong thời gian thực đồng thời cung cấp ước tính chiều cao cây tự động từ hình ảnh RGB. Trong thời gian báo cáo, nhiệt độ dao động từ 27,8°C đến 38,8°C, độ ẩm không khí từ 40,2% đến 77,0%, độ ẩm đất từ 52,0% đến 82,4% và chiều cao cây tăng từ 65,8 cm đến 77,0 cm. Kết quả phân đoạn chứng minh tính khả thi của việc trích xuất thông tin cấu trúc ở cấp độ thân cây trong điều kiện nhà kính thực tế, bao gồm cả nền phức tạp, sự che khuất của lá và ánh sáng không đồng đều. Những kết quả này khẳng định tiềm năng của một hệ thống giám sát điều khiển nhà kính thông minh chi phí thấp, cung cấp cơ sở thực tiễn cho việc mở rộng hệ thống trong tương lai hướng tới các chỉ số nông học phong phú hơn, chẳng hạn như chiều rộng thân cây, diện tích lá, độ che phủ tán cây, ước tính tốc độ tăng trưởng và điều khiển môi trường sống của cây trồng tự động.

Từ khóa: Thị giác máy tính, Giám sát nhà kính, IoT, Nông nghiệp thông minh, Mô hình học sâu dung lượng nhẹ.

Received on August 24 - 2026
Revised on October 08 - 2026