Sensors and Materials, Vol. 37, No. 11 (2025) 5099–5122
MYU Tokyo

S & M 4235

Remote Assessment of Size, Ripeness, and Count of Fruits in Orchard Greenhouse

Min-Chie Chiu,[1*] Shih-Ming Cho,[2] Long-Jyi Yeh,[1]
Ming-Guo Her,[1] Yun-Cheng Lan,[1] and Tian-Syung Lan[3]

[1]Department of Mechanical and Materials Engineering, Tatung University,
No.40, Sec. 3, Zhongshan N. Rd., Taipei City 10452, Taiwan
[2]Department of Computer Science and Engineering, Tatung University,
No.40, Sec. 3, Zhongshan N. Rd., Taipei City 10452, Taiwan
[3]School of Artificial Intelligence, Guangzhou Huashang College,
No. 1, Huashang Road, Lihu Street, Zengcheng District, Guangzhou, Guangdong 511300, China

(Received August 31, 2025; accepted November 5, 2025)

Keywords: LoRa, greenhouse system, image identification, OpenCV, Jetson Nano

We developed and validated a remote fruit recognition system designed for greenhouse orchards, integrating image processing with sensor-based monitoring and long-range (LoRa) wireless communication. Utilizing the open source computer vision library (OpenCV) and Jetson Nano, the system identifies fruit quantity, ripeness, and size from high-resolution images captured by the Pi Camera V2.1. The recognition algorithm incorporates hue-saturation-value color space conversion, morphological operations, and advanced parameters such as bounding box area, aspect ratio, and intersection over union to enhance accuracy. Validation was conducted using a scaled-down greenhouse simulation and kumquat tree images. The modified system achieved recognition accuracies of 90% for fruit count, 100% for unripe and half ripe fruits, 89% for fully ripe fruits, and 81 and 88% for large and small fruits, respectively. These results demonstrate significant improvements over the initial system and confirm the feasibility of integrating sensor technologies with image recognition for real-time agricultural monitoring. The system's compatibility with LoRa communication enables deployment in network-deficient environments, offering a scalable solution for precision agriculture.

1. Introduction

Owing to global carbon emissions leading to Earth's warming and the resultant drastic climate changes causing significant agricultural losses in crops and orchards,[1,2] coupled with escalating issues of water resources, employing greenhouses for crop and orchard protection has become inevitable.[3–5] Furthermore, ensuring water resources for agricultural needs is paramount, with increasing research on water resource recycling. Chiu *et al.* developed a water circulation system in the greenhouse for aquatic feeding.[6]

*Corresponding author: e-mail: mcchiu@gm.ttu.edu.tw
https://doi.org/10.18494/SAM5915

ISSN 0914-4935
https://myukk.org/

To enhance labor efficiency and comprehensive care, remote monitoring systems are required for greenhouse farming. Cheng and colleagues utilized network monitoring techniques to ensure farm safety, theft prevention, and security.[7,8] However, these require synchronizing information for timely notifications and actions at the control center.

Traditional wireless communication for remote farms requires the deployment of a wireless network and reliable communication between network access points and endpoints at farms using Bluetooth, Zigbee, and Wi-Fi.[9] Under typical conditions, Bluetooth supports communication within 10 m, Zigbee covers distances from 10 to 100 m, and Wi-Fi extends up to 300 m.[10] Considering the substantial distance between farm systems and farmers' residences, Wi-Fi is commonly adopted to establish connectivity between both ends.

Establishing a wireless network requires networking equipment at both the near end (farm side) and the far end (user side), functioning as data relay stations. The near-end equipment serves as the server, whereas the far-end equipment operates as the client. On the software side, each endpoint incorporates a human–machine interface (HMI) to facilitate the transmission of network status and data. Communication between the two ends is generally achieved using the transmission control protocol/internet protocol and the user datagram protocol.

However, many farms are located in remote areas where wireless communication is hindered by geographical constraints, such as Taiwan's mountainous terrain. Owing to the limited availability of wireless network infrastructure, long range (LoRa) has been widely used as an alternative to conventional wireless technologies for connecting farms with remote users. Implementing LoRa requires hardware modules at both the near end and the far end. LoRa wide area network (LoRaWAN), a protocol for wide-area networking, supports transmission distances from 3 km in dense urban environments and 10 km in rural areas.[11] Software-based data transmission and reception can be achieved through programming or HMIs.

For remote regions lacking network infrastructure, Huang *et al.* integrated LoRa-based systems with greenhouses for soil-based cultivation.[12] Their system collected data on air temperature and humidity, soil moisture, rainfall, light intensity, and magnetic switch status. These data were transmitted via LoRa modules to a remote PC, where a C# interface was used to visualize and store the data in a Microsoft Excel file for subsequent analysis using large data techniques.[12]

Despite advancements in monitoring greenhouse farming, considerations for greenhouse design and water resource conservation are lacking, relying on existing network infrastructure. Therefore, it is necessary to enhance the yields of orchards and promote water conservation under network-deficient conditions. For the crop growth monitoring of greenhouses, Chiu introduced Internet Protocol cameras for visual recognition.[13] However, this method is labor-intensive and dependent on network availability. Therefore, Zhang *et al.* developed the second-generation tomato posture detection algorithm (TPMv2), integrating image processing with intelligent recognition to estimate fruit quantity.[14] Wan NurazwinSyazwani *et al.* employed unmanned aerial vehicles and machine learning to identify pineapple crown locations,[15] whereas Su *et al.* introduced circular detection methods for fruit identification.[16] However, remote control mechanisms for fruit image recognition have not been applied in remote farming.

Therefore, we applied fruit image recognition in remote farming to reduce labor demands. Owing to the absence of wireless networks in most agricultural environments, LoRa was

employed for long-distance data transmission in this study. However, LoRa's limited bandwidth necessitates on-site image processing. Therefore, we employed a high-performance Jetson Nano for image recognition. The Jetson Nano is a dedicated edge computing device, enabling computationally intensive tasks, such as the current open source computer vision library (OpenCV) base's image recognition. Jetson Nano's small form factor, relatively low power consumption, and ability to operate autonomously on the edge perfectly meet the requirements of a scalable, remote precision agriculture solution. The recognition results are transmitted to a remote monitoring center via LoRa. For image analysis, OpenCV and JetsonHacks are employed.[17] PyCharm and Anaconda3 were utilized to support image recognition software development.

The developed fruit image recognition system in this study can be integrated into existing greenhouses. The system quantifies the count, size, and ripeness of fruits and enables farmers to access the results for remote monitoring. The system with LoRa-based communication and image recognition ability in the orchard exemplifies the application of sensor technologies and proves the applicability of remote sensing and image recognition to agricultural monitoring.

2. Image Recognition Process

2.1 Greenhouse system

The greenhouse of an orchard comprises near-end equipment and different devices (Fig. 1). The ESP-32S microcontroller is used as the central component of this setup for the transmission and reception of signals between sensors and actuators. The sensors used include DHT11 for temperature and humidity measurements, the capacitive soil moisture sensor for soil moisture detection, the TEMT-6000 for light intensity monitoring, and a magnetic switch used for access control. Irrigation pumps, misting systems, convection fans, and shading curtain motors are controlled using actuators. The Pi Camera V2.1 with an integrated GPU is employed to capture fruit images for high-speed image recognition. Image recognition is performed by Jetson Nano, which processes corresponding algorithms and transmits the results to the ESP-32S microcontroller. The data collected and fruit status recognition results are transmitted using LoRa wireless technology.

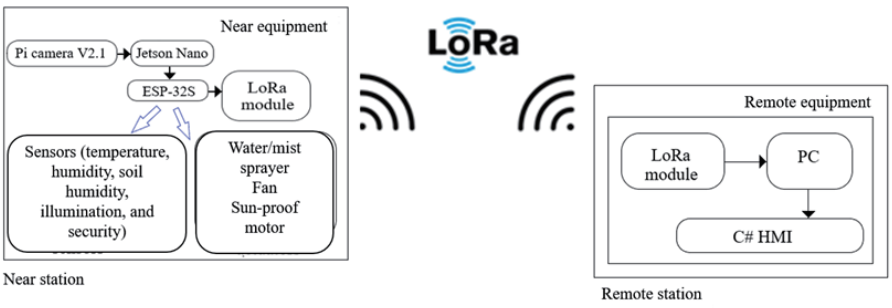

Fig. 1. (Color online) Diagram of near and far ends.

Since the ESP-32S board and standard laptop antennas are not appropriate for LoRa connectivity, dedicated wireless modules were adopted at the near and far ends to enable data transmission. The LoRa wireless communication module and a HMI were developed in C# (Fig. 2). The HMI displays information related to fruit growth and greenhouse environmental conditions, as shown in Fig. 2, enabling farmers to monitor and record the data effectively.

Jetson Nano and Pi Camera are integrated into a track-mounted vehicle used in the greenhouse. The mobile imaging unit operated autonomously on a timed schedule or was remotely controlled to capture and store images. Users configure recording cycles or schedule periodic image recognition tasks without waiting for the camera to reposition or complete image capture before initiating analysis. The use of the imaging device is depicted in Fig. 3.

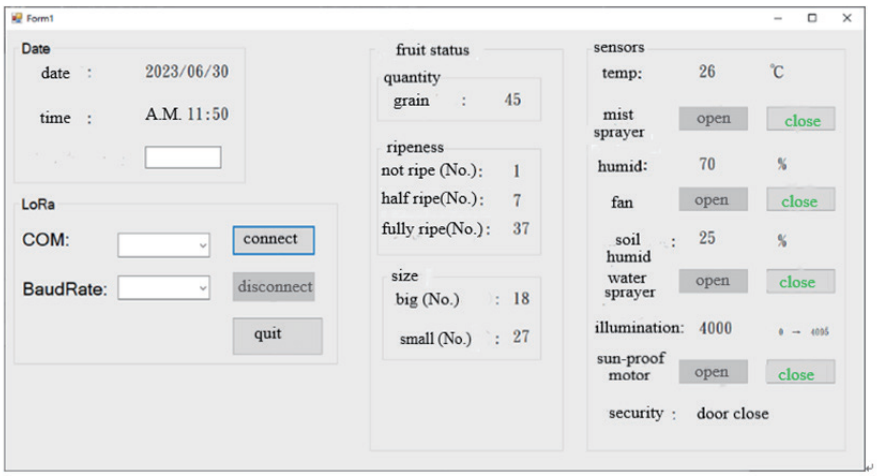

Fig. 2. (Color online) HMI developed in C#.

Fig. 3. (Color online) Imaging device in greenhouse.

2.2 Operation process

The HMI of the developed system presents data collected from the near-end system (Fig. 4). Through the HMI, users configure LoRa-related parameters in the area shown on the display. Upon activating the connection using the designated button on the interface, fruit status information is presented (Area B in Fig. 4), while environmental data are updated and displayed in Area C. These data are stored in a Microsoft Excel file for subsequent analysis.

2.3 Image processing

Fruit images are acquired using the Pi Camera V2.1 in a wide view and a high resolution, providing details for image recognition. Jetson Nano is responsible for image recognition, leveraging its high-performance capabilities to accelerate processing. As illustrated in Fig. 5, the image recognition workflow begins with importing fruit images from storage. These images are converted into binary-formatted images for further processing. The system performs calculations for the recognition of fruit quantity, size, and ripeness. Fruit size is estimated on the basis of the measured distance between the Pi camera and the principles of linear proportionality.

2.3.1 Fruit identification

The smart orchard system was tested using a fruit-bearing kumquat tree (*Citrus japonica*). The images of the tree were captured and transmitted to Jetson Nano using the Pi camera. The recognition workflow is illustrated in Fig. 6.

To optimize the computational efficiency and reduce the processing load, the resolution of the original images (3280 × 2464 pixels) is downscaled to 1280 × 720 pixels. In the developed system, the initial processing step involves OpenCV's cvtColor function to convert image data from RGB color space to the hue-saturation-value (HSV) model.

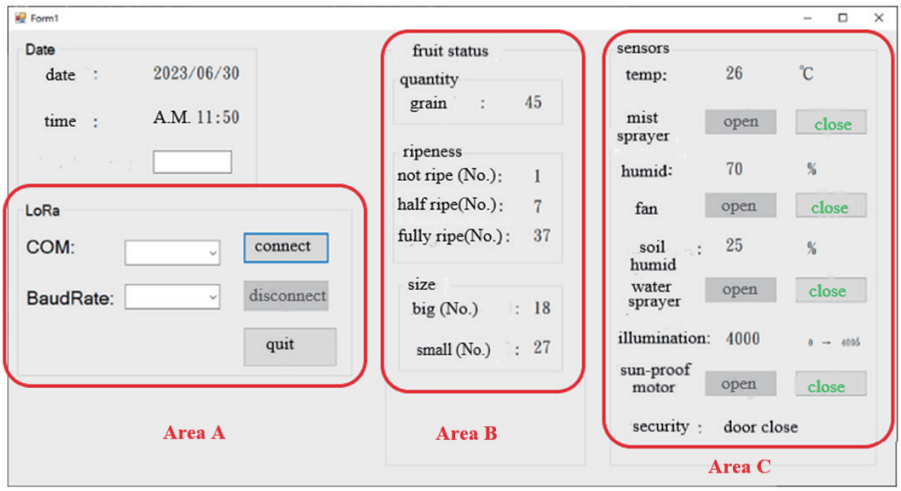

Fig. 4. (Color online) Remote monitoring of HMI written in C#.

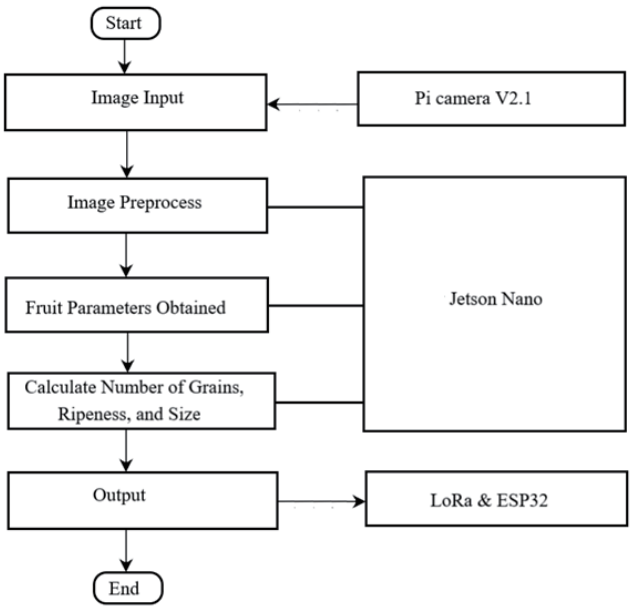

Fig. 5. Image recognition process.

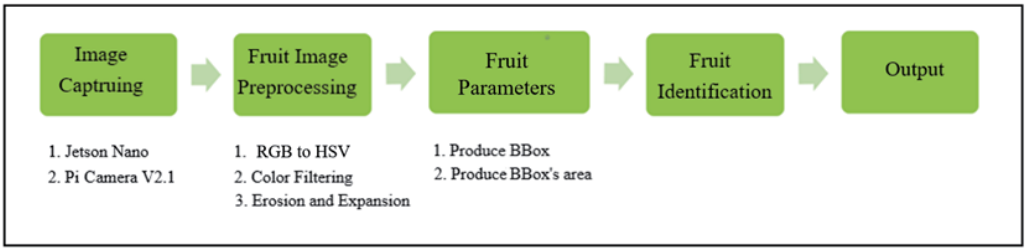

Fig. 6. (Color online) Image generation process of developed system in this study.

The RGB color space is commonly employed for color classification. However, it lacks perceptual intuitiveness, rendering it less effective for interpreting and describing color characteristics from a human visual perspective. In contrast, the HSV color space closely aligns with human color perception, making it appropriate for filtering irrelevant color information. Therefore, we adopted the HSV color space for image recognition. To isolate relevant color regions, the OpenCV function inRange was applied to exclude pixels outside the target color range, thereby preserving the color segments corresponding to the detected fruits. Subsequently, combined with morphological operations such as closing, the image masking function, bitwise_and, was used to generate the final image data containing distinct fruit masks (Fig. 7).

The HSV color space is advantageous in image processing tasks such as color range selection (e.g., skin tones, red, or blue), color detection, and color histogram extraction, as the HSV model uses an inverted cone in which black is located at the bottom vertex, whereas white is positioned at the center of the top circular base. In the HSV color space, *H* presents the color (e.g., red,

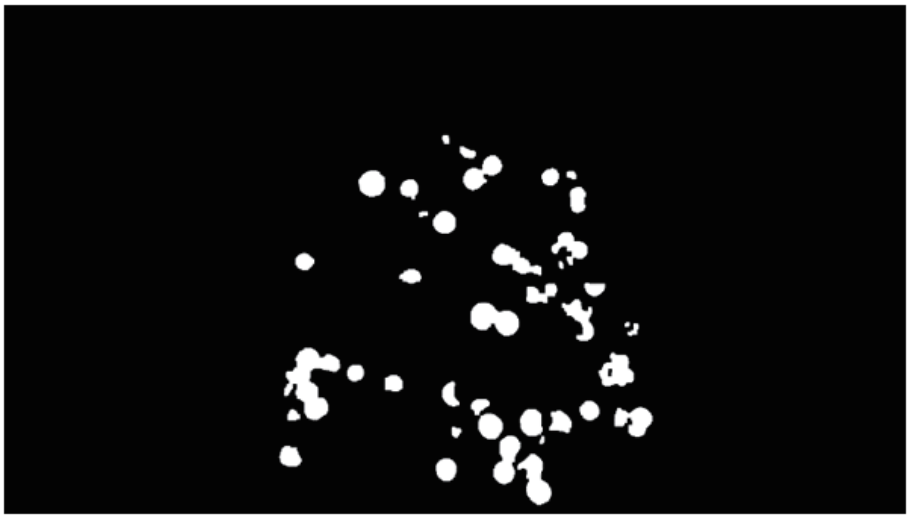

Fig. 7. Binary images of fruits generated using HSV images for color filtering (fruits are shown in white and all other areas in black).

orange, or green) using a value from 0 to 170, S presents the purity of the color, or the amount of gray present using a value from 0 to 255, and V presents the color's intensity or brightness using a value from 0 to 255. In the HSV model derived from the RGB system, H, S, and V are calculated as follows.

$$R' = R / 255 \tag{1}$$

$$G' = G / 255 \tag{2}$$

$$B' = B / 255 \tag{3}$$

$$C_{max} = \max(R',G',B') \tag{4}$$

$$C_{min} = \min(R',G',B') \tag{5}$$

$$\Delta = C_{max} - C_{min} \tag{6}$$

$$H = \begin{cases} 0^{\circ} & \Delta = 0 \\ 60^{\circ} x \left(\dfrac{G' - B'}{\Delta} \bmod 6 \right), & C_{max} = R' \\ 60^{\circ} x \left(\dfrac{B' - R'}{\Delta} + 2 \right), & C_{max} = G' \\ 60^{\circ} x \left(\dfrac{R' - G'}{\Delta} + 4 \right), & C_{max} = B' \end{cases} \tag{7}$$

$$S = \begin{cases} 0, & C_{max} = 0 \\ \dfrac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases} \tag{8}$$

$$V = C_{max} \tag{9}$$

Here, C_{max} and C_{min} are the maximum and minimum normalized components, respectively, and Δ is the chroma range (the difference between the maximum and minimum normalized components). By this method, Fig. 7 was converted into Fig. 8, as an outcome of the system developed.

2.3.2 Counting fruits

To count identified fruits, edge detection is performed using OpenCV's findContours function, which returns an array of coordinate points outlining each contour. The contourArea function is then applied to calculate the area of each detected contour, and the boundingRect is used to generate rectangular bounding boxes (BBoxes) for subsequent analysis. To determine the number of fruits, BBoxes are drawn around contours using the boundingRect. The number of fruits per BBox is estimated on the basis of the area of each box. A scoring system was implemented to score for each count, and the total number of fruits was obtained by summing these scores. For example, a total of 31 BBoxes of varying dimensions were detected in Fig. 9. Eighteen boxes contained a single fruit, six contained two fruits, and seven contained more than two fruits. In cases with multiple fruits per BBox, additional image processing was conducted to improve recognition accuracy.

Fig. 8. (Color online) Fruit image converted from RGB to HSV color space.

Fig. 9. (Color online) Recognition results of image with BBoxes.

2.3.3 Fruit ripeness

Fruit ripeness is assessed by analyzing the HSV values of each BBox. The dominant color blocks within BBoxes were regarded as the fruit's surface color. In this study, ripe fruits were identified with an average HSV value of (18, 150, 100). Ripeness was classified into three categories: fully ripe, half ripe, and unripe. As indicated in Fig. 10, the HSV ranges for each category were determined through subjective visual assessment as follows.

- Fully ripe: (12, 94, 85) to (27, 176, 139)
- Half ripe: (4, 154, 144) to (21, 214, 225)
- Unripe: (16, 87, 45) to (28, 127, 95)

Each fruit is assigned a ripeness label on the basis of the estimated HSV value. The system counted the number of fruits in each category and selected the category with the highest count as the final ripeness. This result was transmitted through the ESP32-S module, completing the ripeness identification process.

2.3.4 Fruit size

The system determines fruit size by calculating the area of the BBox. On the basis of the computed area, fruits are categorized into small fruits [diameter < 26 mm (approximately 750 pixels)] and large fruits [diameter > 32 mm (approximately 750 pixels)]. After classification, the total count of fruits is computed and transmitted to the remote PC.

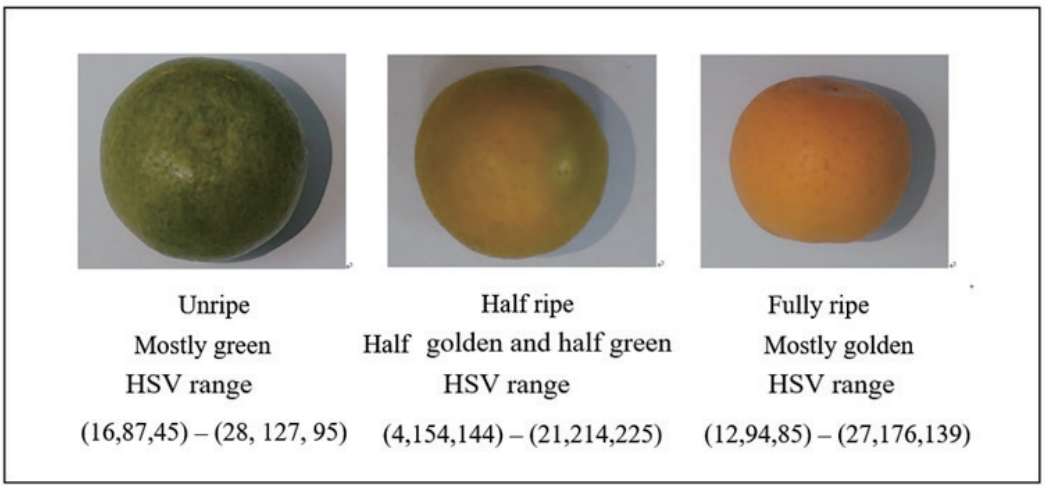

Fig. 10. (Color online) Three stages of fruit ripeness defined in this study.

3. Results of Initilly Developed System

3.1 Testing results

The manual recognition was conducted by three experienced kumquat tree farmers. Each farmer had a minimum of ten years of experience cultivating and commercially grading the specific variety of fruits used in the study. Their extensive practical knowledge served as the standard for defining ripeness stages. The integrated results of the number, ripeness, and size of fruits showed an average accuracy for each category as follows: 66% for number, 39% for ripeness, and 69% for size (Table 1). Such inaccuracies were caused by discrepancies between the estimated and actual sizes and incorrect assessments of ripeness. Therefore, we optimized the system to enhance its accuracy in these areas.

On the basis of the initial results, five fruits of high reference value were identified (Table 2). These fruits were selected using a one-fruit-per-box method, in contrast to other BBoxes that contained multiple fruits. The selected fruits presented minimal interference, indicating that the one-fruit-per-box method was accurate in real measurement.

The developed system recognized ripeness using the HSV threshold values for unripe, half ripe, and fully ripe fruits (Table 3). However, the results indicate that these thresholds lack sufficient precision. Therefore, in developing the second-generation fruit recognition system, these HSV thresholds will be refined on the basis of the observed data.

On the basis of the initial results, five scenarios in which recognition errors occurred in BBox-based fruit images were constructed as follows (Fig. 11).

- Kumquat fruits are segmented into smaller sections owing to obstruction by branches and leaves.
- Multiple fruits are positioned vertically in overlapping arrangements.
- Non-fruit elements, such as tree leaves, are mistakenly included within the BBox.
- Fruits are presented on the tree but not captured within the mask.
- Fruit portions are obscured by surrounding foliage.

Table 1
Results of initially developed system.

Parameter	Degree	Recognition by initially developed system	Manual recognition	Accuracy of system recognition (%)
Count	Total number	35	53	66
Ripeness	Unripe	0	2	0
	Half ripe	4	9	44
	Fully ripe	31	42	74
Size	Large fruits	13	22	59
	Small fruits	22	31	79

Table 2
Representative BBox.

BBox number	Count	Ripeness	Size
0	1	Fully ripe	Small
20	1	Fully ripe	Small
23	1	Fully ripe	Large
25	1	Fully ripe	Small
27	1	Fully ripe	Large

Table 3
HSV threshold values for ripeness.

Ripeness	Value
Upper limit of unripe	(28, 127, 95)
Lower limit of unripe	(16, 87, 45)
Upper limit of half ripe	(21, 214, 225)
Lower limit of half ripe	(4, 154, 144)
Upper limit of fully ripe	(27, 176, 139)
Lower limit of fully ripe	(12, 94, 85)

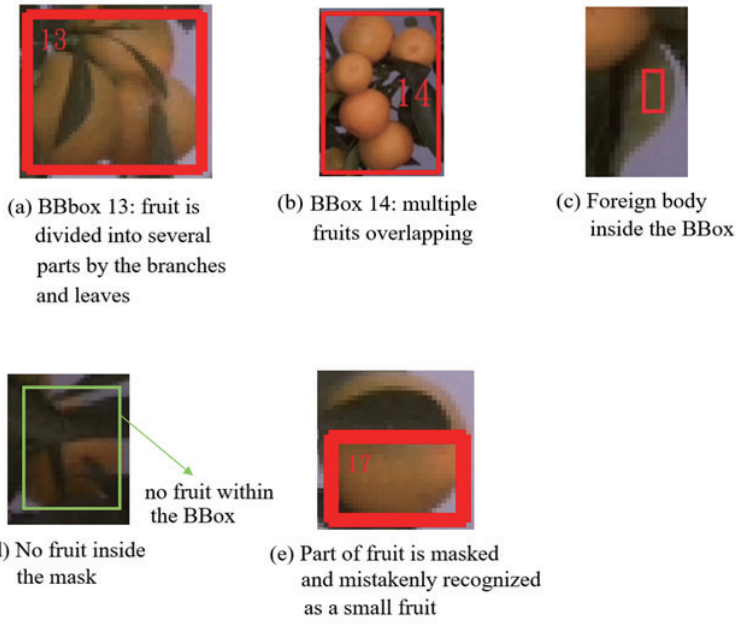

Fig. 11. (Color online) Scenarios of identification error occurrences.

3.2 System modification

On the basis of the error occurrence scenarios and their underlying causes (Table 4), we refined the advanced image processing method. An analysis of the reasons for these five types of scenario is outlined in Table 5.

Modifications to the recognition algorithm are detailed in Fig. 12. We integrated Gaussian blurring and parameters, including intersection over union (IoU), BBox position, BBox aspect ratio, and fruit area ratio.

In the modified system, the OpenCV GaussianBlur function was employed to smooth fruit images and suppress noise to enhance the effectiveness of subsequent edge detection. The preprocessing steps had a similar structure to the initially developed system: conversion to the HSV color space, the filtering of nontarget colors, and the execution of morphological operations using the morphologyEx function with the MORPH_OPEN parameter. These preprocessing steps were constructed to yield binary images of the fruit masks, as shown in Figs. 13 and 14.

To accurately assess fruit ripeness, the modified system adopted additional parameters from the fruit masks for classification and statistical evaluation, including BBox area, BBox position, fruit area ratio, BBox aspect ratio, and IoU.

Table 4
Identified scenarios of error occurrences.

Scenario	Description	Number	Ripeness	Size
(a)	Fruit is divided into several parts by the branches and leaves	1	Half ripe	Large fruit
(b)	Multiple fruits overlapping	1	Half ripe	Large fruit
(c)	Foreign body inside the BBox	0	0	0
(d)	No fruit within the mask	0	0	0
(e)	Part of fruit is masked	1	Half ripe	Small fruit

Table 5
Analysis of the status of the corresponding number.

Item	Condition Analysis
(a)	After pretreatment, it is identified as a fruit
(b)	The fruits grow on the same branch and tend to overlap
(c)	The color of the leaf itself is similar to that of the fruit, and it was mistakenly selected by BBox
(d)	One of the fruits was heavily obscured and unrecognizable
(e)	It should actually be a large fruit, but some of it is partially obscured, so it is identified as a small fruit

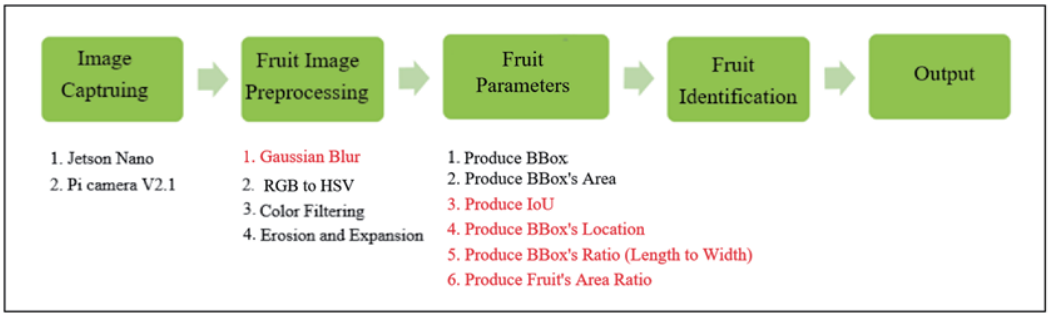

Fig. 12. (Color online) Workflow of modified fruit identification system.

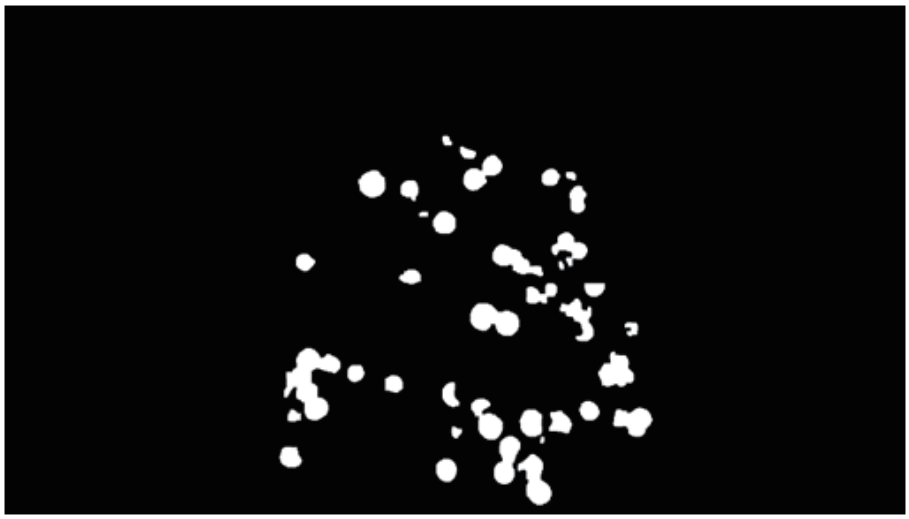

Fig. 13. HSV image for color filtering to generate new binary image.

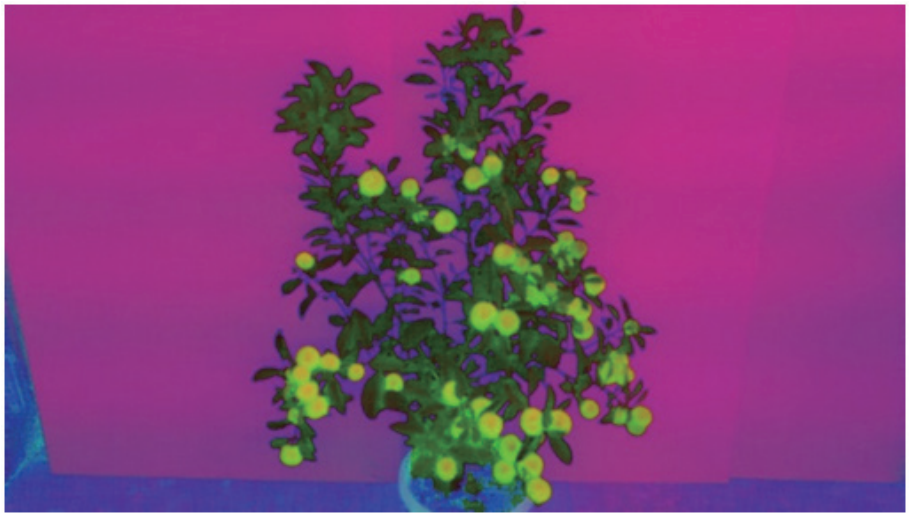

Fig. 14. (Color online) Image converted from RGB to HSV color space.

- BBox: a rectangular region used in object detection to localize fruits within the image (Fig. 15)
- BBoxarea: the total pixel area enclosed by the BBox after recognition
- BBoxposition: the coordinates of the upper-left corner of the BBox, expressed as (x, y), along with its width and height, indicating the box's location and dimensions within the image
- Fruit area ratio: the proportion of the fruit mask area relative to the total BBox area, used to assess how much of the box is occupied by fruit
- BBox aspect ratio: the ratio of the longer side to the shorter side of the BBox, which helps determine its orientation (horizontal or vertical) and infer fruit overlap direction (Fig. 16)

Fig. 15. (Color online) BBox identification of kumquat tree using modified system.

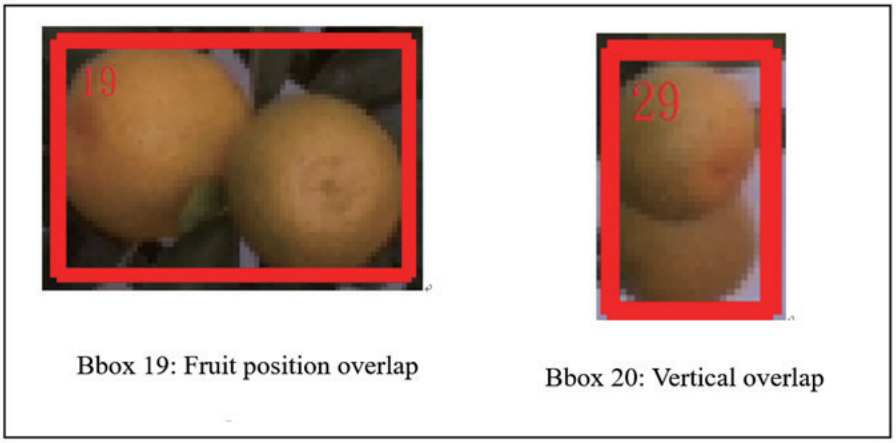

Fig. 16. (Color online) Estimated orientation of fruits based on BBox aspect ratio.

- IoU: a metric used to evaluate the overlap between two BBoxes. It is calculated by dividing the area of the intersection by the area of the union. An IoU value exceeding 90% indicates a duplicate selection of the same fruit (Fig. 17).

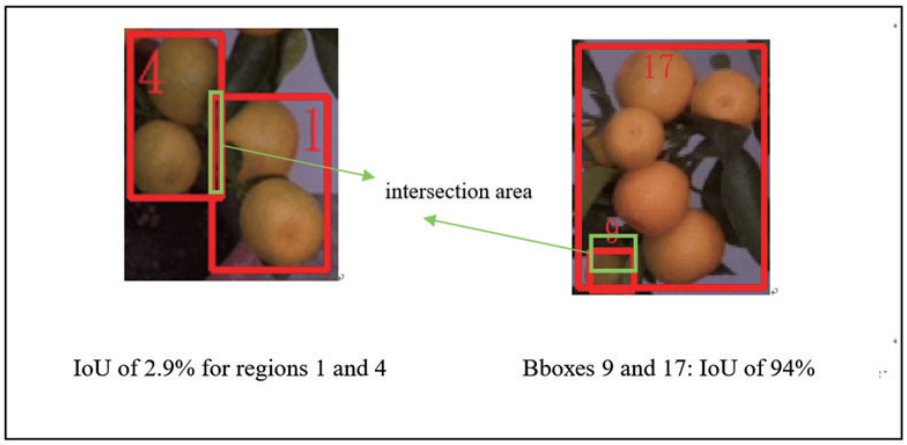

Fig. 17. (Color online) IoU to measure the overlap between two BBoxes.

3.2.1 Fruit counting

With a modified recognition process and decision-making criteria, the number of BBoxes was manually counted, excluding those with minimal area as non-fruit frames. BBoxes with an IoU exceeding 90% were considered duplicates. In such cases, the smaller BBox was discarded, and each BBox underwent an intersection assessment using IoU to eliminate duplicate detections. BBoxes were classified on the basis of fruit area proportion: those with less than 70% fruit coverage were assigned to Class B, representing single fruits, whereas those exceeding 70% were assigned to Class A as potential multiple-fruit regions. The BBoxes of Class A underwent additional image processing to improve recognition accuracy. The BBox aspect ratio was used to determine the direction of fruit overlap, using OpenCV's Erode function along the longer axis of the box. Following erosion, the findContours function was applied to Class A BBoxes to detect fruit edges and count individual fruits. The total fruit count was obtained from Classes A and B.

3.2.2 Fruit ripeness

The modified system retained the three-stage maturity classification: unripe, half ripe, and fully ripe. Each fruit was identified using its corresponding BBox coordinates, which were applied for the fruit mask to isolate the fruit region and exclude background elements. HSV color values in the masked region were analyzed to determine ripeness. The classification ranges were defined as follows.

- Fully ripe: (17, 100, 84) to (22, 170, 130)
- Half ripe: (9, 160, 150) to (16, 208, 216)
- Unripe: (21, 93, 55) to (23, 121, 86)

Figure 18 presents examples of fruit recognition results. Each fruit was labeled according to its ripeness. The system aggregated these labels, and the most frequently occurring category was selected as the final output.

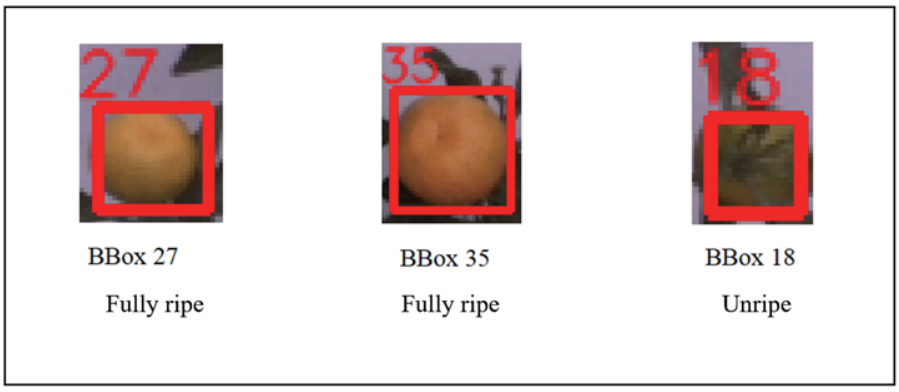

Fig. 18. (Color online) Ripeness recognition of fruit.

3.2.3 Fruit size identification

Fruit size recognition was performed following the fruit count. Fruits were categorized into small and large classes. Size was determined on the basis of the fruit area derived from the fruit area ratio. Seven hundred pixels was used as a threshold for classification: fruit images exceeding the threshold were labeled large, whereas those below the threshold were labeled small. Figure 19 shows the recognition results.

The performance of the modified system demonstrated accuracies of 80% for fruit count, and 50% (unripe), 50% (half ripe), and 93% (fully ripe) for fruit ripeness, and 72% (large) and 87% (small) for fruit size. The results in Tables 6 and 7 confirm that improvements in image preprocessing significantly enhance recognition accuracy, validating the effectiveness of the proposed refinements.

4. Results of Modified System

4.1 System implementation and validation

We modified a fruit recognition system using OpenCV based on the recognition results of the initially developed system. The modified system integrated image preprocessing, parameter extraction for fruit recognition, and growth status determination for field applications. The system architecture includes the image recognition algorithm and a network communication module. For validation, the modified recognition algorithm was developed. Jetson Nano and the Pi Camera V2.1 were also used as the core processing unit for recognition and fruit image capturing. The system's performance was evaluated using this image, confirming its recognition accuracy.

Accuracy in fruit count, ripeness, and size was calculated. Ripeness was categorized into unripe, half ripe, and fully ripe, while size was categorized into small and large. Ground truth data were used to calculate recognition accuracy. To validate the recognition capability of the modified system, a scaled-down greenhouse orchard simulation setup was constructed (Fig. 20).

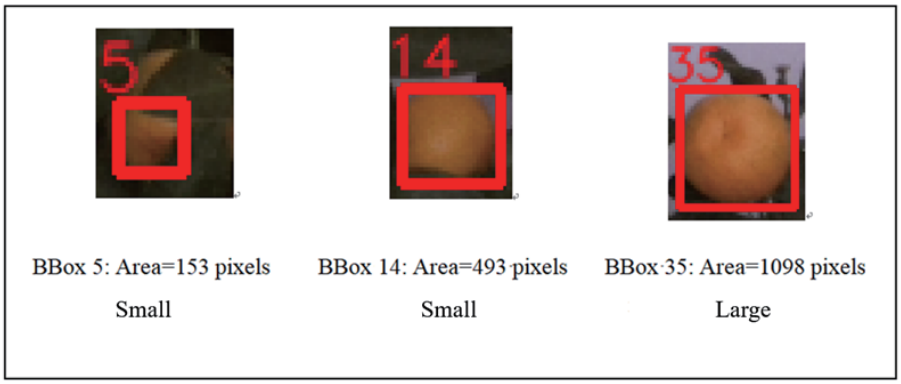

Fig. 19. (Color online) Size recognition.

Table 6
Recognition accuracy of modified system.

Parameter	Degree	Recognition by modified system	Manual recognition	Accuracy of system recognition (%)
Count	Total number	45	56	80
Ripeness	Unripe fruits	1	2	50
	Half-ripe fruits	7	14	50
	Fully ripe fruits	37	40	93
Size	Large fruits	18	25	72
	Small fruits	27	31	87

Table 7
Accuracy comparison between initial and modified systems.

	Accuracy of initially developed system (%)	Accuracy of modified system (%)	Improvement rate (%)
Quantity	66	80	14
Unripe	0	50	50
Half ripe	44	50	6
Fully ripe	74	93	19
Large fruit	59	72	13
Small fruit	71	87	16

The system wirelessly transmits data to the remote PC, and the results are displayed on HMI (Fig. 21). The system workflow is illustrated in Fig. 22, with subsystem control workflows detailed in Figs. 23 and 24. Figure 23 outlines the workflow of the environmental and safety monitoring, while Fig. 24 presents the fruit image recognition process. The workflow of the remote control center system (e.g., farmer's residence) is shown in Fig. 25, whereas the hardware circuitry is depicted in Fig. 26. The recognition results are presented in Fig. 27.

4.2 Discussion

The recognition results are summarized in Table 8. The system achieved accuracies of 90% for fruit quantity; 100% for unripe fruits, 100% for half ripe fruits, and 89% for fully ripe fruits in the ripeness classification; and 81% for large fruits and 88% for small fruits in the size

Fig. 20. (Color online) Scaled-down greenhouse orchard used for validation of modified system.

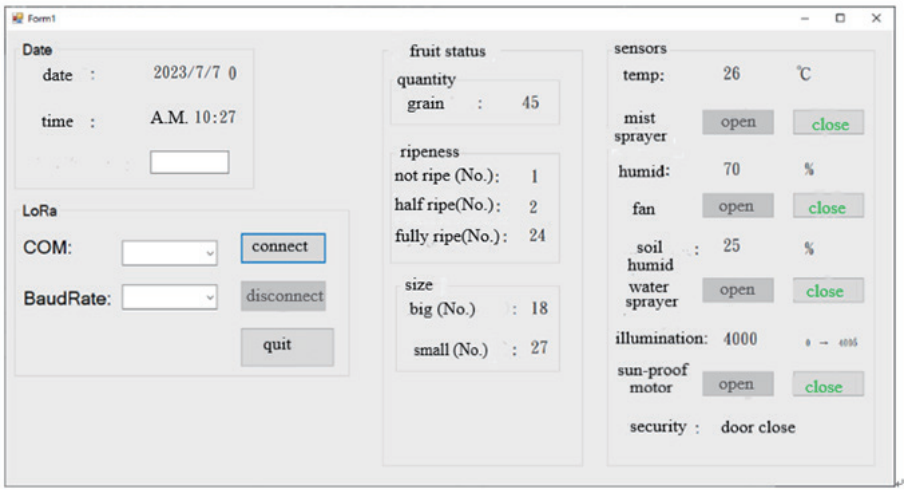

Fig. 21. (Color online) HMI in C# on remote monitoring center.

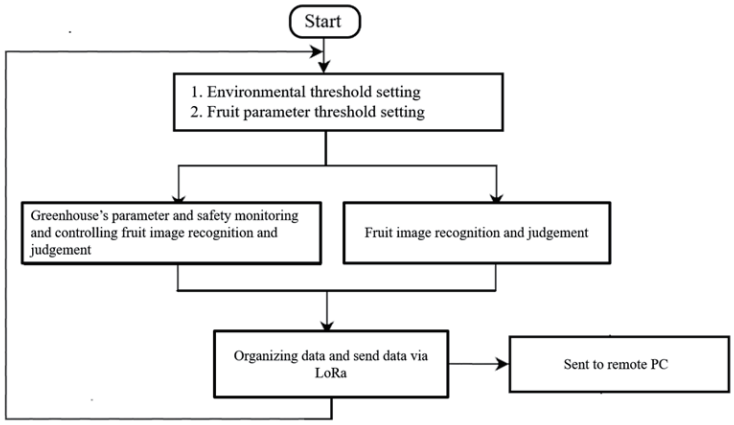

Fig. 22. Workflow of fruit recognition and data transmission of modified system.

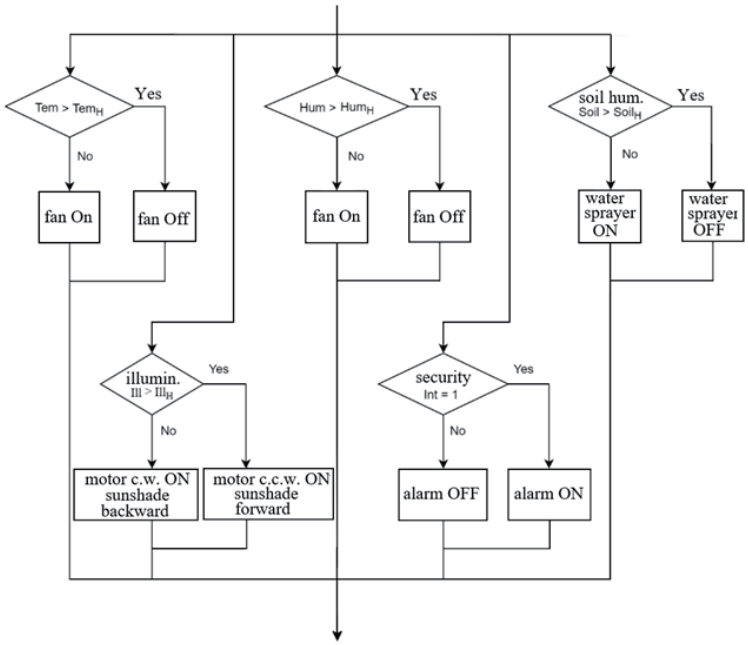

Fig. 23. Workflow of environmental and safety monitoring of modified system.

classification. Exceptional cases were analyzed using BBoxes with irregular conditions (Fig. 28). Compared with the initially developed system, the modified model improved its recognition ability with enhanced accuracy. As shown in Fig. 28, four BBox scenarios for the kumquat tree were examined on the basis of the results of the modified system.

In Fig. 28(a), Box 19 presents composite conditions where two fruits are correctly identified, confirming the effectiveness of enhanced erosion. Box 27 in Fig. 28(b) contains a non-fruit object, which the system excludes owing to its small BBox area, identifying it as noise. Prefiltering such noise can improve the accuracy in such cases. Box 8 in Fig. 28(c) includes a

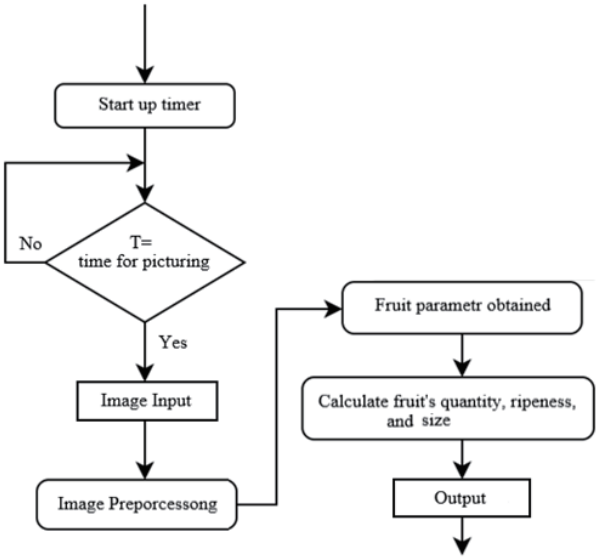

Fig. 24. Fruit image recognition process.

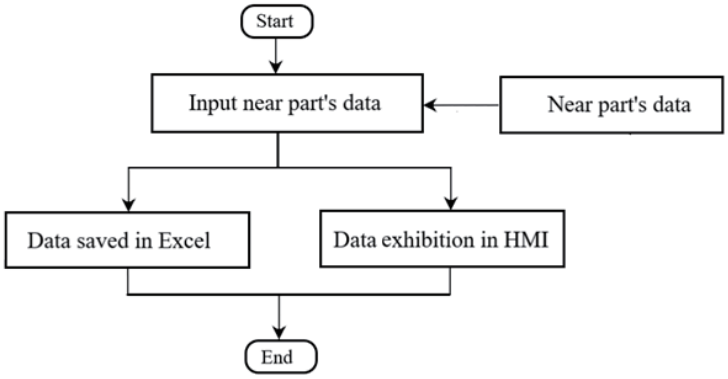

Fig. 25. Workflow of remote monitoring system.

fruit not captured in the mask, resulting in exclusion from analysis. Box 20 in Fig. 28(d) shows a partially obscured fruit, misclassified as small. Despite using the fruit area and radius as criteria, the system cannot accurately recognize the fruit.

We explored optimal communication methods for greenhouse orchard environments under time and resource constraints. While wireless network communication enables remote monitoring, it requires internet connectivity, which is often unavailable in orchard settings. Given the limited infrastructure and large coverage areas, LoRa wireless communication was selected for its LoRa capabilities, despite its limited data transmission capacity. To address this limitation, fruit recognition was performed in the greenhouse, and the results were transmitted remotely. OpenCV-based computer vision was adopted for its efficiency and suitability under constrained conditions. The modified system achieved a 90% accuracy, validating its effectiveness.

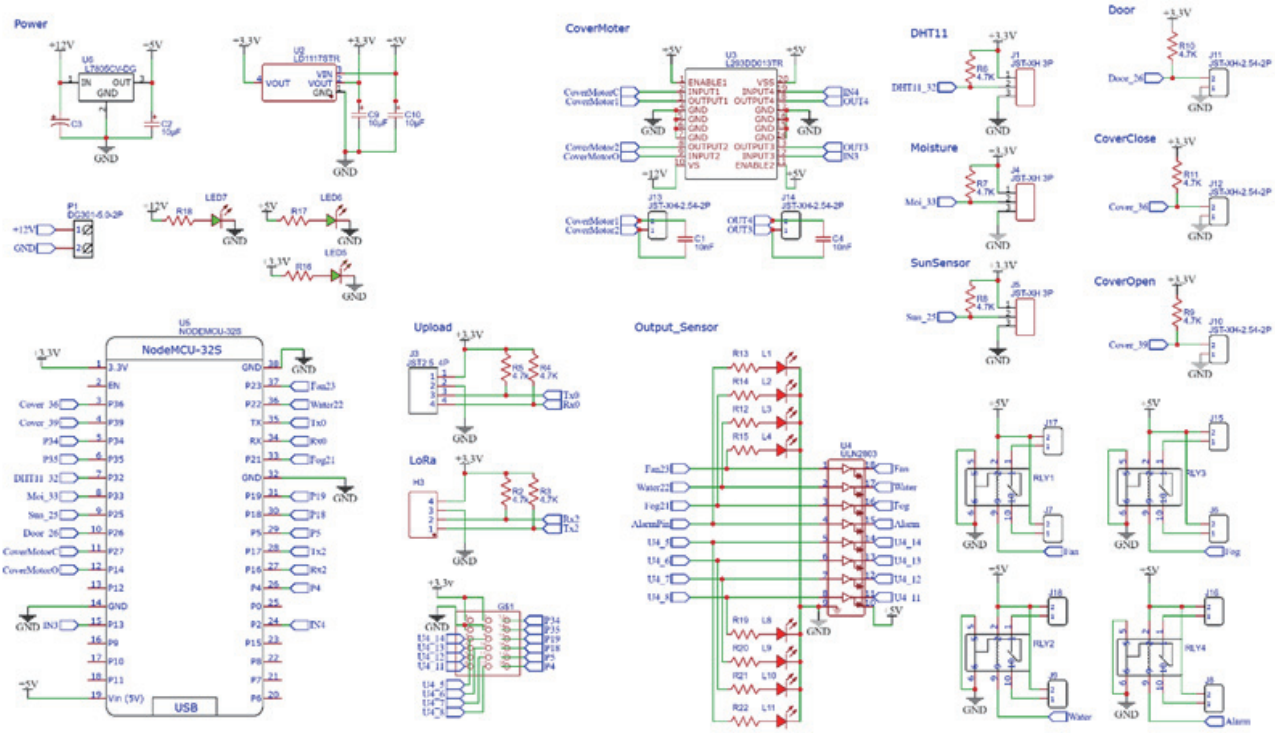

Fig. 26. (Color online) Hardware circuitry of modified system.

Fig. 27. (Color online) Fruit recognition of modified system in this study.

Table 8
Recognition accuracy of modified system.

Parameter	Degree	Recognition by modified system	Manual recognition	Accuracy of system recognition (%)
Count	Total number	27	30	90
Ripeness	Unripe fruits	1	1	100
	Half ripe fruits	2	2	100
	Fully ripe fruits	24	27	89
Size	Large fruits	17	21	81
	Small fruits	10	9	88

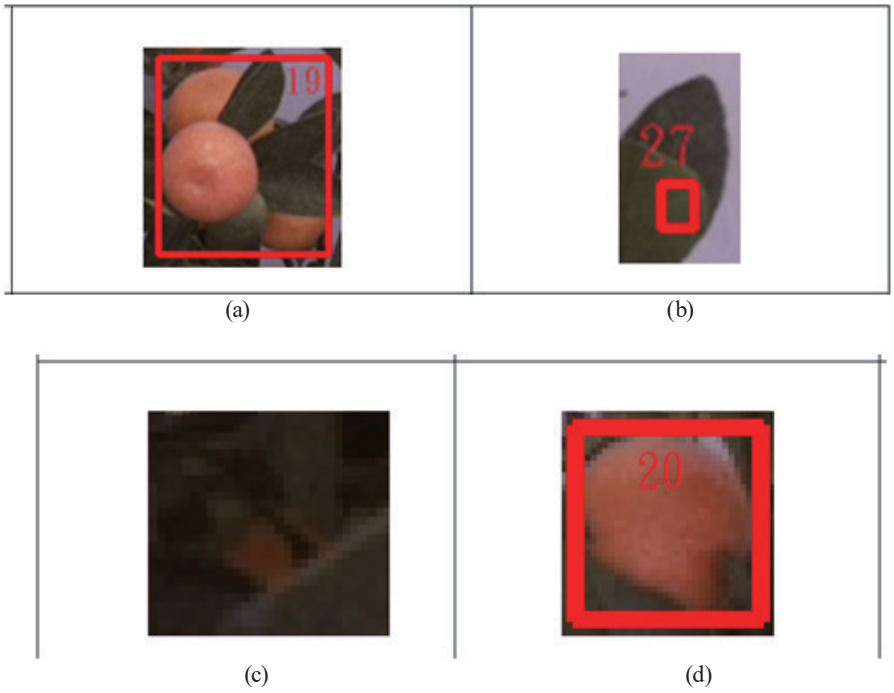

Fig. 28. (Color online) BBoxes with different conditions: (a) fruits with branches and leaves and overlapped (Bbox 19), (b) fruits vertically overlapped (Bbox 27), (c) no fruit appearance (Bbox 8), and (d) fruit partly obscured by branches and leaves (Bbox 20).

The modified system assists farmers in monitoring crop growth, estimating annual yields, and determining optimal harvest times. The results can be used for marketing planning by segmenting fruit based on quantity, ripeness, and size, thereby optimizing pricing strategies and increasing profitability.

A CNN-based model using You Only Look Once (YOLO) v3 developed by Muresan and Oltean, presented a 75% accuracy.[18] However, their system did not assess fruit quantity, ripeness, or size. Luo *et al.* developed software based on histogram matching, edge detection algorithms, Otsu thresholding, and clustering for grading harvested fruits, with classification accuracies of 94.58% (peaches), 88.23% (lemons), 70% (apples), and 93.33% (tomatoes).[19] Their system focused on post-harvest grading and did not estimate yield or ripeness in situ. Yang *et al.* addressed challenges such as occlusion and environmental variability by developing the efficient lightweight object detector, achieving an 87.4% accuracy.[20] Their model improved detection accuracy but did not support the simultaneous assessment of quantity, ripeness, and size.

In contrast with such previous results, the system developed in this study enables the remote, real-time identification of fruit quantity, ripeness, and size within orchard trees, with experimental accuracy ranging from 81 to 100%, surpassing the models. The recognition results of the modified system presented feasibility and operational simplicity. Through the validation by farmers in real farming, the deployability and applicability of the modified system can be significantly improved.

5. Conclusions

The fruit recognition system developed in this study integrates sensor technology, image processing, and LoRa wireless communication to address the challenges of remote agricultural monitoring. By leveraging Jetson Nano for on-site image recognition and transmitting results via LoRa, the system overcomes bandwidth limitations common in rural orchards. Quantitative validation revealed substantial improvements in recognition accuracy: fruit count (90%), ripeness classification (100% for unripe and half ripe and 89% for fully ripe), and size identification (81% for large and 88% for small). These improvements were achieved through refined preprocessing techniques and the incorporation of key parameters such as BBox metrics and IoU. The system's integration with environmental sensors (e.g., DHT11, soil moisture, and light intensity sensors) further supports comprehensive greenhouse management. This research highlights the potential of sensor-driven image recognition systems to enhance yield estimation, harvest planning, and market segmentation, thereby improving agricultural productivity and optimizing resource efficiency. In the future, we plan to test the modified system for field deployment and usability with farmers to ensure practical adoption and scalability.

Acknowledgments

This research was supported by Tatung University (B112-M03-020).

References

1 M. Rosegrant, X. Cai, and S. A. Cline: World Food Prize:.https://www.worldfoodprize.org/documents/filelibrary/images/borlaug_dialogue/2002/transcripts/rosegrant_transcript_F2A98817596C1.pdf (accessed October 2025).
2 M. Rosegrant, S. Msangi, C. Ringler, T. B. Sulser, T. Zhu, and S. A. Cline: International Food Policy Research Institute (2008) 1. https://www.researchgate.net/profile/Siwa-Msangi/publication/267995611_International_Model_for_Policy_Analysis_of_Agricultural_Commodities_and_Trade_IMPACT_Model_Description/links/550d6daa0cf2ac2905a6a384/International-Model-for-Policy-Analysis-of-Agricultural-Commodities-and-Trade-IMPACT-Model-Description.pdf
3 W. Baudoin, E. Baeza, M. Teitel, and M. Kacira: FAO (2013) 1. https://www.researchgate.net/publication/260984593_Good_Agricultural_Practices_for_greenhouse_vegetable_crops_Principles_for_Mediterranean_climate_areas
4 J. I. Montero, N. Castilla, D. R. Gutierrez, and F. Bretones: Acta Hort. (1985) 227. https://www.actahort.org/books/170/170_26.htm
5 J. C. López, A. Baille, S. Bonachela, M. M. González-Real, and J. Pérez-Parra: Span. J. Agric. Res. **4** (2006) 289. https://sjar.revistas.csic.es/index.php/sjar/article/view/206
6 M. C. Chiu, W. M. Yan, S. A. Bhat, and N. F. Huang: J. Agric. Food Res. (2002) 1. https://www.sciencedirect.com/science/article/pii/S2666154322000904

7 B. Cheng, M. C. Chiu, and C. M. Chiu: MATEC Web Conf. (2018) 1. https://www.matec-conferences.org/articles/matecconf/pdf/2018/44/matecconf_icpmmt2018_00038.pdf.
8 T. J. Chan, M. C. Chiu, H. C. Cheng, L. J. Yeh, and W. C. Haung: IOP Conf. Ser. Mater. Sci. Eng. (2019) 012006. https://iopscience.iop.org/article/10.1088/1757-899X/644/1/012006
9 G. Ou, Y. Chen, Y. Han, Y. Sun, S. Zheng, and R. Ma: Agriculture **15** (2025) 467. https://doi.org/10.3390/agriculture15050467
10 J. Huang, S. Chai, N. Yang, and L. Liu: AISR (2017) 1. https://www.atlantis-press.com/proceedings/caai-17/25881194
11 A. Augustin, J. Yi, T. Clausen, and W. M. Townsley: Sensors **16** (2016) 1466. https://www.mdpi.com/1424-8220/16/9/1466
12 W. C. Huang, M. C. Chiu, L. J. Yeh, and C. M. Chiu: J. Phys. **2020** (2021) 012006. https://iopscience.iop.org/article/10.1088/1742-6596/2020/1/012006/meta
13 M. C. Chiu: J. Appl. Sci. **10** (2010) 1944. https://scialert.net/fulltext/?doi=jas.2010.1944.1950
14 F. Zhang, J. Gao, C. Song, H. Zhou, K. Zou, J. Xie, T. Yuan, and J. Zhang: Comput. Electron. Agric. **210** (2023) 107878. https://dl.acm.org/doi/abs/10.1016/j.compag.2023.107878
15 R. Wan NurazwinSyazwani, H. M. Asraf, M. A. MegatSyahirul Amin, and K.A. Nur Dalila: Alex. Eng. J. **61** (2022) 1265. https://www.sciencedirect.com/science/article/pii/S111001682100418X
16 Y. Su, X. Zhang, B. Cuan, Y. Liu, and Z. Wang: Pattern Recognit. **97** (2020) 107022. https://www.sciencedirect.com/science/article/abs/pii/S0031320319303255
17 JetsonHacks: https://jetsonhacks.com/ (accessed October 2025).
18 H. Muresan and M. Oltean: Acta Univ. Sapientiae Inf. (2018) 26. https://ieeexplore.ieee.org/document/9921050
19 D. Luo, R. Luo, J. Cheng, and X. Liu: Front. Plant Sci. **20** (2024) 1415095. https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2024.1415095/full
20 X. Yang, W. Zhao, Y. Wang, W. Q. Yan, and Y. Li: Sci. Rep. **14** (2024) 26086. https://doi.org/10.1038/s41598-024-76662-w