

Article

Simple and Affordable Vision-Based Detection of Seedling Deficiencies to Relieve Labor Shortages in Small-Scale Cruciferous Nurseries

Po-Jui Su ¹, Tse-Min Chen ¹ and Jung-Jeng Su ^{2,3,4,*} ¹ Department of Bio-Industrial Mechatronics Engineering, National Chung Hsing University, Taichung 402202, Taiwan² Department of Animal Science and Technology, National Taiwan University, Taipei 106032, Taiwan³ Bioenergy Research Center, College of Bio-Resources and Agriculture, National Taiwan University, Taipei 106319, Taiwan⁴ Agricultural Net-Zero Carbon Technology and Management Innovation Research Center, College of Bio-Resources and Agriculture, National Taiwan University, Taipei 106319, Taiwan* Correspondence: jjsu@ntu.edu.tw

Abstract

Labor shortages in seedling nurseries, particularly in manual inspection and replanting, hinder operational efficiency despite advancements in automation. This study aims to develop a cost-effective, GPU-free machine vision system to automate the detection of deficient seedlings in plug trays, specifically for small-scale nursery operations. The proposed Deficiency Detection and Replanting Positioning (DDRP) machine integrates low-cost components including an Intel RealSense Depth Camera D435, Raspberry Pi 4B, stepper motors, and a programmable logic controller (PLC). It utilizes OpenCV's Haar cascade algorithm, HSV color space conversion, and Otsu thresholding to enable real-time image processing without GPU acceleration. The proposed Deficiency Detection and Replanting Positioning (DDRP) machine integrates low-cost components including an Intel RealSense Depth Camera D435, Raspberry Pi 4B, stepper motors, and a programmable logic controller (PLC). It utilizes OpenCV's Haar cascade algorithm, HSV color space conversion, and Otsu thresholding to enable real-time image processing without GPU acceleration. Under controlled laboratory conditions, the DDRP-Machine achieved high detection accuracy (96.0–98.7%) and precision rates (82.14–83.78%). Benchmarking against deep-learning models such as YOLOv5x and Mask R-CNN showed comparable performance, while requiring only one-third to one-fifth of the cost and avoiding complex infrastructure. The Batch Detection (BD) mode significantly reduced processing time compared to Continuous Detection (CD), enhancing real-time applicability. The DDRP-Machine demonstrates strong potential to improve seedling inspection efficiency and reduce labor dependency in nursery operations. Its modular design and minimal hardware requirements make it a practical and scalable solution for resource-limited environments. This study offers a viable pathway for small-scale farms to adopt intelligent automation without the financial burden of high-end AI systems. Future enhancements, adaptive lighting and self-learning capabilities, will further improve field robustness and including broaden its applicability across diverse nursery conditions.

Keywords: cruciferous vegetable; deficient plug seedling; machine learning; GPU-free; image processing; maneuver control



Academic Editors: Tao Wu, Qingting Liu and Zhigang Zhang

Received: 12 September 2025

Revised: 9 October 2025

Accepted: 23 October 2025

Published: 25 October 2025

Citation: Su, P.-J.; Chen, T.-M.; Su, J.-J. Simple and Affordable Vision-Based Detection of Seedling Deficiencies to Relieve Labor Shortages in Small-Scale Cruciferous Nurseries. *Agriculture* **2025**, *15*, 2227. <https://doi.org/10.3390/agriculture15212227>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agriculture is increasingly adopting automation to address labor shortages, yet seedling nursery farms remain dependent on manual oversight in managing mis-planted plug trays, creating inefficiencies in the agricultural supply chain in Taiwan [1]. While automation supports sowing, seedling growth, and environmental control, the lack of an integrated real-time detection and correction system for deficient seedlings presents a scientific gap in precision agriculture and smart farming technologies. Current contract farming practices rely on manual verification, leading to inconsistencies in seedling quality and delays in farm operations. A major challenge in seedling nurseries is their inability to dynamically adjust production based on real-time demand forecasting and government agricultural planning data. Without a data-driven system to analyze predicted harvests and cultivation trends, nurseries struggle to optimize seedling supply, leading to overproduction or shortages that affect market stability and pricing. This study bridges the gap by proposing an automated vision-based system to detect and reposition deficient seedlings, reducing reliance on manual labor while enhancing accuracy. By integrating low-cost image processing techniques, this approach improves seedling management efficiency and lays the foundation for intelligent production systems in nurseries, ensuring precision planting and adaptive seedling distribution.

A 2019 survey highlighted the growing reliance on automation in Taiwan's vegetable seedling industry, with many nurseries integrating mechanical or automated equipment to improve efficiency [2]. Despite these advancements, the industry still faces challenges related to labor-intensive tasks such as manually filling missing seedlings or removing double seedlings, leading to significant labor pressure within the crucial early germination period. Several studies have explored machine vision solutions to address this gap, yet the choice of methodologies and their limitations vary significantly.

The Wen et al. [3] study addresses the issue of missing or unhealthy vegetable plug seedlings during cultivation, packaging, and transportation, which negatively impacts transplanting efficiency and seedling survival rates. To mitigate these challenges, researchers developed an automated seedling selection system integrated with machine vision and adaptive control algorithms. The system identifies plug seedlings based on morphological features, eliminates weak or defective seedlings via a PLC-controlled nozzle, and supplements missing seedlings to maintain a continuous supply on the conveyor belt. An adaptive fuzzy PID algorithm ensures precise seedling positioning and delivery. Field experiments using 30-day-old pepper seedlings demonstrated high identification success rates (96.99–98.84%) and significantly improved robust seedling rates (up to 16.07% higher) across varying extraction speeds (60–100 plants/min). The results of Wen et al. confirm the system's effectiveness in enhancing seedling quality and transplanting reliability, offering a promising solution for precision agriculture and automated nursery operations [3].

However, most research has developed machine vision systems based on Intel RealSense technology or the State Space Model (SSM) for close-range seedling monitoring [4–8]. These systems employ advanced 3D segmentation techniques, such as point cloud clustering and depth-based masking enable accurate morphological analysis and facilitate tasks like automated transplanting. However, their systems relied on high-performance computing tools such as MATLAB R2023a and ImageJ with Java 8, with high cost, which offer robust image processing capabilities but are computationally demanding and impractical for real-time applications in commercial farms. Similarly, Fu et al. [7] implemented threshold segmentation and morphological processing for leafy vegetable seedlings, yet their method required extensive image preprocessing, making it difficult to scale for nursery operations with high seedling turnover. A more modern approach was introduced by Yan et al. [9], who utilized YOLOv5x, a deep-learning model capable of real-time de-

tection and replanting coordination for missing tomato seedlings. Their system achieved 92.84% detection accuracy with a 13.475 s average detection time per tray, demonstrating superior precision compared to traditional image processing techniques. While deep-learning models like YOLOv5x provide state-of-the-art detection capabilities, they require high computational power, often necessitating GPUs or cloud processing, which can be cost-prohibitive for small-scale nursery farms seeking affordable automation.

In contrast, our study employs OpenCV's Haar Cascade classifier, an older yet resource-efficient approach that enables real-time seedling detection on a Raspberry Pi without requiring high-performance computing hardware. While Haar Cascade models generally fall short of deep learning alternatives in terms of detection accuracy, they offer a cost-effective solution that balances affordability and automation feasibility. The trade-off between deep learning models like YOLOv5x and traditional image processing methods lies in computational demand versus accessibility, while Yan et al.'s system achieves higher accuracy, our approach ensures low-cost, scalable deployment, making automation attainable for small-scale commercial nurseries with limited infrastructure investment [9].

This study directly addresses the gap between high-performance seedling detection and affordability by proposing a hybrid solution that integrates low-power deep learning with edge computing. By leveraging lightweight convolutional networks, the system maintains high accuracy without demanding costly AI hardware. Unlike previous approaches that depend on specialized equipment and software, our method enables traditional seedling nurseries to adopt intelligent, automated vision systems that are both scalable and economically viable for small-scale farms. Using a common depth camera as the image source, the study tracks image recognition to calibrate the unsuccessful germination of seedlings. A Raspberry Pi and a programmable logic controller (PLC) form the control core, using the RS-485 communication protocol to establish mutual signal flow and transmit the number of seedlings and the coordinates of the target cell grid. The system controls the positioning of the X-Y stepper motor on the replanting rack, achieving the goal of automatic replanting pre-positioning. This aims to stabilize the upstream high-quality seedling supply capacity in the current agricultural business model.

2. Materials and Methods

2.1. Structure of the Deficiency Detection and Replanting Positioning (DDRP) Machine

The DDRP machine system for detecting deficient seedlings is an automatic machine with four independent wheels, including two front wheels with in-wheel stepper motors, allowing smooth forward and backward movement with responsive maneuverability. The main structure is a cube shape (55 cm × 55 cm × 62 cm, LWH) constructed from aluminum extrusion (Figure 1). All sides are covered with 5 mm thick black foam boards to block all visible light, except for the bottom side (Figure 1). A plug tray is placed under the DDRP system for deficiency detection and replanting positioning tests (Figure 2).

On the top side, there are two cameras for different purposes: a depth camera (RealSense Depth Camera D435i, Intel Co., Santa Clara, CA, USA) for image recognition and a surveillance camera (W200, HP Inc., Palo Alto, CA, USA) for monitoring the entire area. As an alternative to expensive laser rangefinders and network cameras, the Azure Kinect DK depth camera was available before the Intel® RealSense™ depth camera. In terms of size, weight, and price, the Intel® RealSense™ camera is an excellent choice for users needing to identify image and distance information. A depth camera is a lens module with an RGB lens and uses two sets of infrared rangefinders to obtain distance values, with some models adding a six-axis accelerometer module for increased functionality. Initially, distance calculation required using the lens proportion, but with the infrared ranging module, the

starting position and angle of measurement are consistent, reducing measurement error and making it easier to capture target distance information.

In the proposed methodological framework, the integration of two distinct camera modules, a depth camera (Intel® RealSense™ D435i) and a surveillance camera (HP W200), offers a balanced solution in terms of cost, performance, and system compatibility. The depth camera serves as the primary sensor for image recognition and distance measurement, while the surveillance camera provides wide-area monitoring to ensure operational oversight. Compared to earlier alternatives such as the Azure Kinect DK and high-cost laser rangefinders, the Intel® RealSense™ D435i presents a cost-effective and compact option. Its lightweight design and affordability make it particularly suitable for small-scale agricultural systems. Functionally, the RealSense™ camera combines an RGB lens with dual infrared rangefinders, enabling accurate depth sensing. The inclusion of a six-axis accelerometer in some models further enhances spatial awareness and system responsiveness. From a performance standpoint, the infrared-based distance measurement ensures consistent starting positions and angles, significantly reducing error margins compared to traditional lens-proportion methods. This consistency improves the reliability of seedling detection and spatial mapping. Moreover, the modular nature of these components allows seamless integration with edge computing platforms, enhancing compatibility and scalability across various deployment environments.

Three LED strips (5050 white LED, 500 mm length, 5 V, China) are attached to the top and front sides to illuminate the plug trays and aid image recognition. Two stepper motors provide high torque for the front wheels, driving the machine's movement (stepper motor, 24 V, YH57BYGH51-402A, Zhejiang Yuhui Electronics, Yueqing City, China). Additionally, two stepper motors are applied to the linear x/y -axis actuator for processing replantation positions, including a stepper motor (3.7 V, 17HS1352-P4130, Shenzhen Rtelligent Technology Co., Ltd., Shenzhen, China) for X-axis movement and another stepper motor (24 V, 17PM-KA39B, Shenzhen Rtelligent Technology Co., Ltd., Shenzhen, China) for Y-axis movement (Figure 2).

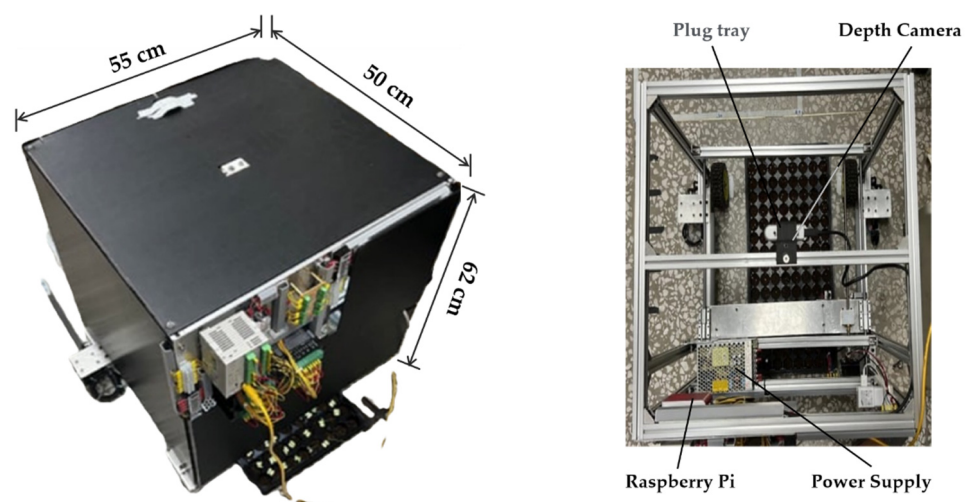


Figure 1. Photo of the machinery structure of the DDRP system with size (Left) and the top view of the machine (Right).

All electronic control devices and wiring are installed on the rear side, including a Raspberry Pi 4B (Raspberry Pi Ltd., Cambridge, UK), a Programmable Logic Controller (PLC) (DVP-285V, Delta Electronics, Inc., Taipei, Taiwan), a communication adaptor (CH340 chip USB to TS485 converter adaptor, Shenzhen, China), and a stepper motor driver (TB6600, Sysmotor, Sys Tech. Co., Ltd., Dongguan, China) (Figure 2). The Raspberry Pi 4B is

connected to the DVP-285V PLC through a USB to RS-485 converter adaptor. Additionally, a red laser target designator (RLTD1, Bulcomers KS Ltd., Sofia, Bulgaria) with a red laser beam is used to point out deficiency plantings on the 128-cell plug trays (60 cm × 30 cm, LW, DD128, Wen-Kang Plastic Inc., Ltd., Nantou, Taiwan).

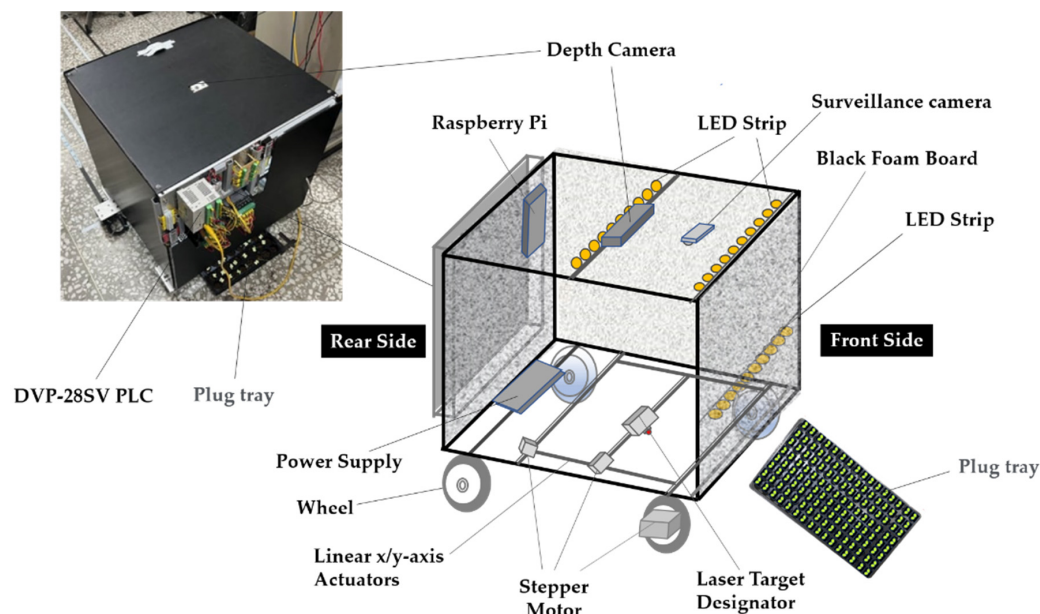


Figure 2. Sketch of the machinery structure of the DDRP system with brief indexes.

2.2. Design of the Image Control System on Raspberry PI for Deficiency Detection

2.2.1. Image Processing Suite

In image deep learning, three main algorithms are commonly used: R-CNN series (R-CNN, Fast R-CNN, Faster R-CNN), Single Shot Detector (SSD), and YOLO (You Only Look Once) [10]. Although R-CNN was the earliest algorithm and delivers highly accurate results, it has a significant drawback: its slow speed. To address this issue, YOLO and OpenCV were developed. YOLO is a convolutional neural network that can predict multiple box positions and categories simultaneously, achieving end-to-end object detection and recognition, with its key advantage being speed. OpenCV (Open Source Computer Vision Library) is a cross-platform computer vision library initially developed by Intel and is available for free in commercial and academic applications.

Considering cost factors, OpenCV is entirely open-source and free, eliminating licensing fees or subscription costs. It operates efficiently on standard CPUs without requiring specialized hardware, and training custom models with OpenCV demands significantly fewer computational resources compared to deep learning frameworks. In contrast, YOLO implementation incurs higher costs due to the necessity of expensive hardware (such as GPUs or TPUs) for training and inference. Additionally, training a customized YOLO model is computationally intensive because of its extensive parameters.

While Tiny-YOLO and MobileNet SSD are designed for resource-limited devices, their implementation requires model training, dataset refinement, and specialized hardware acceleration, adding complexity and increasing deployment costs. The primary objective of this study is to develop an affordable, real-time detection system that nurseries can adopt with minimal investment. By leveraging the Haar cascade combined with optimized image pre-processing, we achieved reliable detection while significantly reducing hardware and computational requirements. Therefore, OpenCV is selected for this study instead of YOLO.

2.2.2. Processing of the RGB Image to HSV Color Space with Grayscale Processing and the Otsu Thresholding Methodology

The RGB image control and processing system relies on a depth camera and a Raspberry Pi 4B single-board computer to analyze image data efficiently. Two distinct image-processing strategies have been implemented in this study (Figure 3):

(1) Process #A (Grayscale and Otsu thresholding, GO): Captures an RGB image of the plug tray → Converts the image to grayscale → Applies Otsu thresholding to segment the image → Using photomask processing to refine detection → Performs image recognition using OpenCV's Haar cascade algorithm to identify targets → Extracts pixel coordinates of the detected object → Sends coordinate data via Modbus to the PLC control system for further processing.

(2) Process #B (HSV, Grayscale, and Otsu thresholding, HGO):

Captures an RGB image of the plug tray → Applies HSV conversion, followed by grayscale processing → Uses Otsu thresholding on both the HSV-converted grayscale image and the directly processed grayscale image → Merges the outputs from both paths before proceeding with photomask processing → Utilizes OpenCV's Haar cascade algorithm for image recognition → Extracts pixel coordinates of the identified object → Transmits this data via Modbus to the PLC control system for precise execution.

The Raspberry Pi was used as the central processing unit for RGB image processing. The RGB image stream from the depth camera was fed into the system, with depth distance data retained temporarily for secondary verification if necessary. Initially, the RGB images were converted to grayscale and processed using the Otsu thresholding method [11] (Figure 4). This method automatically selects an optimal threshold based on pixel values, minimizing intra-class variance within the image. Proposed by Nobuyuki Otsu [11], it maximizes inter-class variance because the squared distance between any two values remains constant [12]. The method's ability to find a suitable threshold under varying lighting conditions makes it ideal for this study, as external environmental light sources significantly impact image processing and threshold selection.

To mitigate the influence of irrelevant image areas, photomasks were applied around the perimeter, reducing the impact of color variations from unrelated objects (such as the floor) on the binarization results (Figure 4). The relevant programs were implemented using Python's OpenCV 4.5.5 package. However, even after direct binarization, there was still excessive noise. Therefore, morphological operations were applied to clean up the results. It was found that performing two mathematical morphology operations, opening and closing, yielded satisfactory outputs. The opening operation involved erosion followed by dilation processing, while the closing operation involved dilation followed by erosion processing.

Dilation is a fundamental operation in mathematical morphology, initially developed for binary images but now extended to grayscale images and complete lattices. The dilation operation typically employs a structuring element to probe and expand the shapes present in the input image. Erosion, the other fundamental operation in morphological image processing, typically employs a structuring element to probe and reduce the shapes present in the input image.

In the experiment, the addition of LED strips (5050 white LED strips) improved stability and allowed us to obtain more consistent parameters (Figure 2). To mitigate the impact of external environmental changes during movement, light-blocking foam boards were installed above and around the test bench. Initially, when using a direct approach of turning off external light sources, the Otsu thresholding method still resulted in the loss of many fine details due to the wide color domain. To address this, the approach was modified by segmenting problematic color regions using photomasks and creating

separate image sources for subsequent binarization. Finally, the binarization results from these segmented images were combined to retain more feature details.

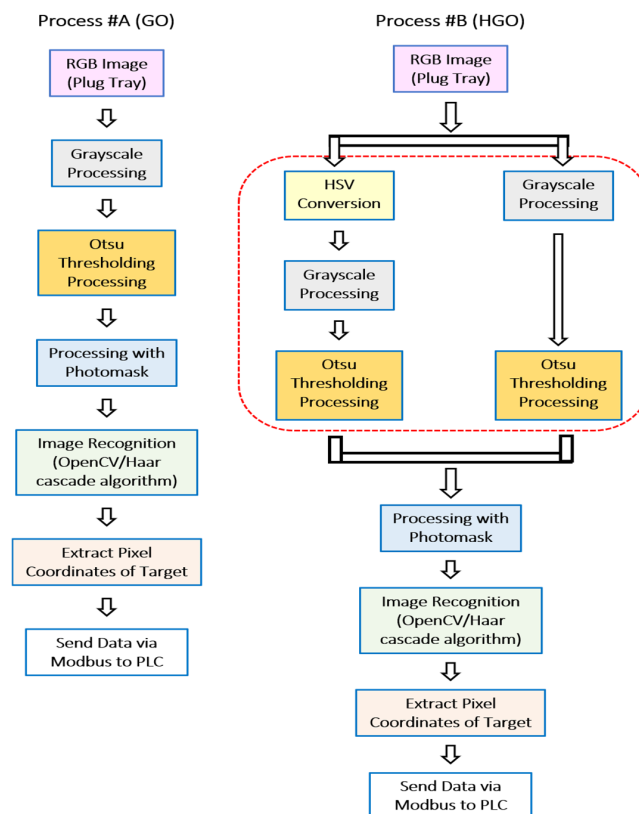


Figure 3. Flowchart of Process #A (Grayscale and Otsu thresholding, GO) and Process #B (HSV, Grayscale, and Otsu thresholding, HGO) of the image processing system. Red frame indicates both image treatment of Process #B before processed with photomask.

To enhance the RGB images of the plug trays and seedlings, the images were first converted to HSV color space, then processed with grayscale and Otsu thresholding techniques, followed by erosion and opening operations using the `cv2.bitwise_` and `command` from the OpenCV 4.5.5 package. All processed images were further refined with photomasks for image recognition in machine learning, identifying the pixel coordinates of the targets, such as the empty cells in the plug trays, on the Raspberry Pi. Due to the differing coordinate systems between the Raspberry Pi and the PLC modules, all target pixel coordinates were converted to a 0–255 scale. These rescaled coordinates were then transmitted to the PLC module via the Modbus protocol. The PLC module utilized the received pixel coordinates from the Raspberry Pi to detect and pinpoint the empty cells in the plug trays and control the machine’s movement (Figure 5).

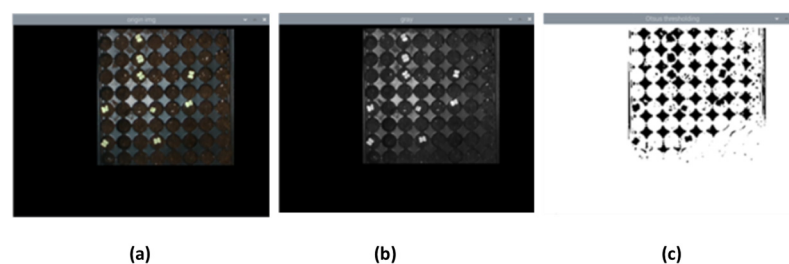


Figure 4. Comparison of image processing results: (a) after applying the photomask, (b) grayscale conversion, and (c) Otsu thresholding processing.

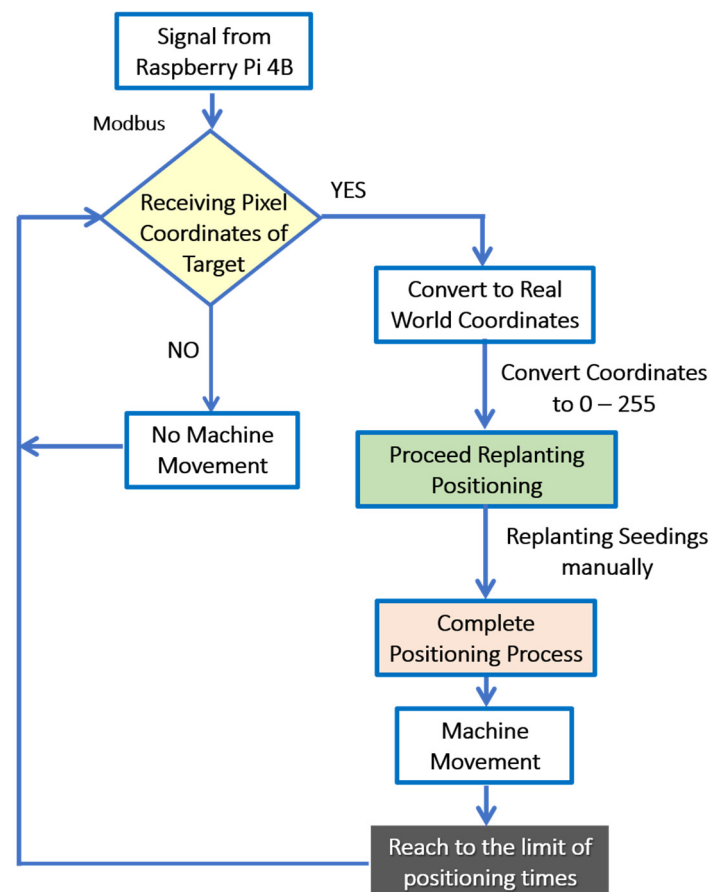


Figure 5. Flowchart of PLC control system.

2.2.3. Machinery Image Reorganization

There are various methods for real-time recognition of missing plant cells in seedling plug trays. While deep learning has recently gained popularity, earlier machine learning approaches can also accomplish this task. The goal of this research is to identify missing plants. During the hardware introduction, we discussed how the choice of algorithm impacts hardware selection. For example, using a neural network-based deep learning method for real-time recognition requires a display chip with high computing capabilities and a graphics card with a high Compute Unified Device Architecture (CUDA) number for real-time computing. Alternatively, if portability and compact size are essential, one must optimize for machine learning algorithms that impose less hardware burden. In this study, we utilize OpenCV's Haar cascade algorithm for the image recognition of empty cells in each plant tray.

2.2.4. Haar Cascade Algorithm

Introduced by Ali et al. [13], the Haar Cascade classifier detects object features and remains effective for real-time recognition thanks to hardware advancements. This machine-learning algorithm, included in OpenCV's sample programs, is designed for object detection in images and videos. Training the algorithm requires placing images of objects to be identified in a folder of positive samples, while targets that should not be identified go into a folder of negative samples, which are used as backgrounds. Mixing the target with negative samples can cause training failure. In this experiment, the method was used to identify empty cells in seedling plug trays, with images binarized into white and black parts, making the empty tray's shape characteristics fixed. The process for training image recognition included seeding the target samples into fixed-size images, collecting

pre-processed samples, and using random internet images for negative samples. Training feature files in the Haar Cascade classifier uses a special training mode in OpenCV. The classification algorithm, derived from the AdaBoost algorithm and the Probably Approximately Correct (PAC) model, involves screening rectangular features to construct a weak classifier. These classifiers, combined with suitable weight parameters, form a strong classifier. Weak classifiers need only exceed 50% accuracy to be used. Finally, multiple strong classifiers are combined into a Cascade Classifier [13].

During the recognition process (Figure 6, modified from Figure 1 of [14]), the screen to be identified is scanned from the top left corner based on the user-defined box in the program. Figure 6 shows a sequential decision-making process for evaluating image blocks using a cascade of classifiers. The goal is to filter and process only the most relevant image data for advanced analysis. This type of architecture is commonly used in machine vision and image processing systems to: (1) Reduce computational load by filtering out irrelevant data early. (2) Improve accuracy by applying increasingly strict criteria. (3) Ensure only high-quality or relevant image blocks undergo intensive processing. The more pre-classifiers pass, the higher the probability of detecting the target. The same block may be scanned multiple times, and the program will mark duplicates as successful detections. The program parameters are designed based on the set number of passes [13].

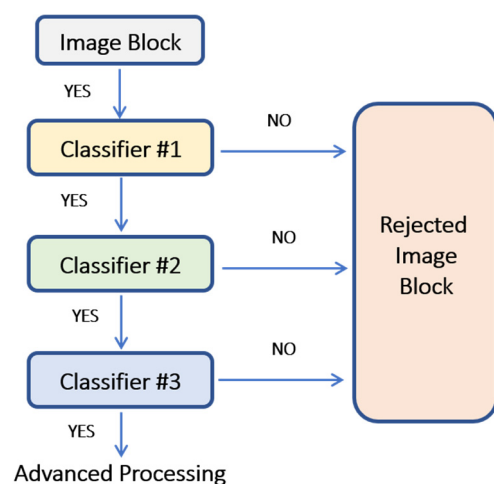


Figure 6. A sequential decision-making process for evaluating image blocks using a cascade of classifiers.

Our study prioritizes low-cost automation for small-scale nursery farms, where the adoption of computationally expensive deep-learning approaches presents financial and hardware constraints. While Tiny-YOLO and MobileNet SSD are optimized for edge devices like the Raspberry Pi, they still require GPU acceleration or TPU support for optimal performance, which many small-scale commercial nurseries lack the infrastructure to support. The Haar cascade classifier, though dated, remains computationally lightweight, enabling real-time defect detection without excessive processing power. This trade-off ensures affordability and accessibility, addressing the economic constraints of smaller farms that may not be able to afford deep-learning-based hardware.

2.2.5. Target Coordinate Transmission Section

The original display result is presented as $[x, y, w, h]$, where x and y are the starting pixels of the successfully detected image block, and w and h are the width and height of the image block (Figure 7). The representative value can be obtained by extracting the original value and post-processing it. For transmission hardware, external wiring uses a USB to RS-485 converter with a CH340 chip, and the software program employs the Serial suite in Python for action execution.

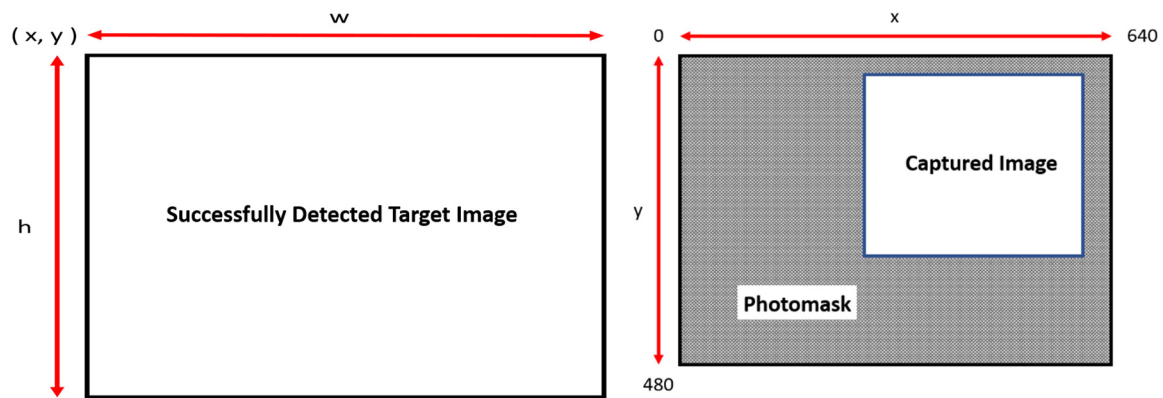


Figure 7. The image values are displayed as plots (Left) and schematic diagrams of the captured image (Right).

The Serial suite has an 8-bit limit to the transmission value, restricted to 0–255. Any over-performing or negative integer values will force the program to interrupt execution. The coefficients 0.8 and 0.75 were empirically derived based on the physical dimensions of the plug tray and the resolution of the captured image (640×480 pixels) (Figure 7). These scaling factors ensure that the pixel coordinates are proportionally mapped to the real-world grid of the tray. The offsets 225 and 10 correspond to the origin shift required to align the image capture box with the actual plug tray layout, compensating for the cropped region used during image acquisition, from pixel range (210, 10) to (530, 270).

However, the coordinates for capturing images are (210, 10) to (530, 270), and the rest of the exterior is black. Without additional processing, these measured coordinates will not be transmitted, causing the Raspberry Pi terminal program to be interrupted. Therefore, target coordinate values need to be converted using the following equations before transmission to the PLC:

$$x' = \left[\left(x + \frac{1}{2}w \right) - 225 \right] \times 0.8 \quad (1)$$

$$y' = \left[\left(y + \frac{1}{2}h \right) - 10 \right] \times 0.75 \quad (2)$$

where x and y represent the top-left pixel coordinates of the detected image block. w and h denote the width and height of the image block. x' and y' are the converted coordinates used by the PLC to control stepper motor positioning.

Before transmission, the floating point part is rounded, and the central value of the calibration box is calculated as the positioning coordinates. Values 225 and 10 are used to remove unnecessary parts based on the starting coordinates of the image capture box. The difference of 15 with 210 is due to the cell disk's left and right range, and the center point of the first cell grid usually being around the coordinate value 225. The upper and lower ranges are detected as much as possible without special modification. The multipliers 0.8 and 0.75 limit values to 0–255, balancing accuracy and size. If out-of-range values occur, values less than 0 are corrected to 0, and values greater than 255 are corrected to 255. Preliminary experiments show values are between -1 and 256, with minimal impact from corrections.

To prevent coordinate accumulation, an empty set is added before the detection program block, allowing direct transmission and updating with a new empty set before each detection. Pre-processed images, real-time images, and real-time light information are displayed on a window (Figure 8). Each time the PLC performs a new action, a screenshot of the window is synchronized and automatically saved to the specified folder.

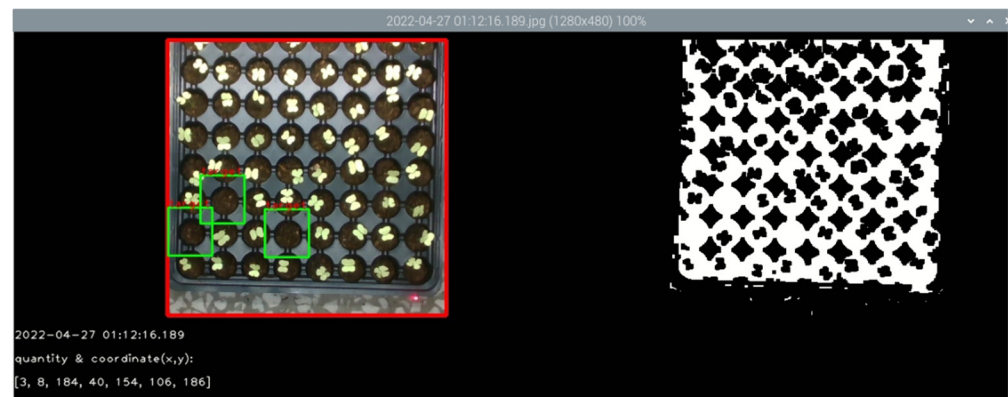


Figure 8. Example of an automatic screenshot.

2.2.6. Receiving Pixel Coordinates from Raspberry Pi on PLC

Coordinate reception utilizes the built-in Delta instruction API 80 RS (serial data transmission). Unlike other Modbus instructions, such as RTU (Remote Terminal Unit) and ASCII (American Standard Code for Information Interchange), this instruction does not require a CRC (Cyclic Redundancy Check) for the tail code, nor does it need to specify station numbers or read/write codes in the header. This streamlines communication transmission, makes it more efficient, and significantly enhances data transfer capabilities. It also allows for specifying data storage destinations, facilitating data management and usage. During the experimental phase, direct monitoring verifies if the transmitted coordinates match the values on the Raspberry Pi side.

2.2.7. Conversion of Received Pixel Coordinates to the Real-World Coordinates on PLC

After transmitting the signal from the Raspberry Pi, the data received by the PLC requires conversion. The received coordinates must be mapped to the actual target position of the compensating positioning slide. The coordinate conversion system maps the image pixel coordinate system to the world coordinate system. Instead of using standard length units, it is based on the step count of the stepper motor controlled by the PLC (Figure 9), facilitating intuitive control and operation within the program.

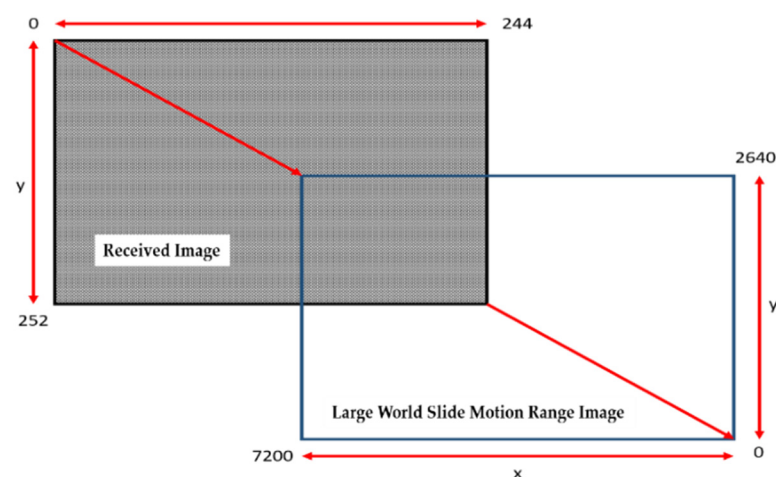


Figure 9. Schematic diagram of coordinate transformation.

The origin in the world coordinate system starts from the lower right corner (Figure 10), as the mechanism's reference origin is there. Each time the program starts or the positioning action is completed, the red laser point indicating the position is reset to the lower right corner origin. Programming control uses the PLC API 203 SCLP (Parameterized

Proportional Direct Operation Instruction) (Delta Electronics, Inc., Taipei, Taiwan), which performs proportional calculations on input values to achieve the conversion requirement.

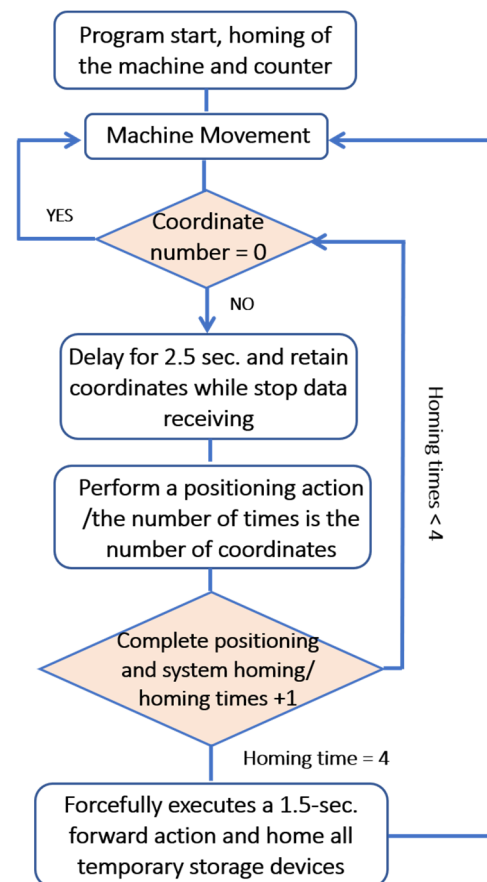


Figure 10. Flowchart of continuous mode with a non-fixed advance distance system.

2.3. Maneuver Control Strategies of the DDRP System for Deficiency Detection and Positioning

2.3.1. Design of the Deficiency Detection and Positioning System on PLC

For PLC control, besides real-time coordinate conversion, the focus is on overall mechanism control strategies and carrier movement control. Pixel coordinates of targets received from the Raspberry Pi via Modbus protocol are converted to real-world coordinates on the PLC module (Figure 5). Upon receiving the data, the DDRP machine starts replanting positioning and uses a red laser target designator to point out the empty cells in the plug tray. After replanting a seedling in one empty cell, the PLC proceeds to the next positioning task until it reaches the limit of positioning times (Figure 5), then moves to the next plug tray.

Two stepper motors provide high torque for the front wheels, driving the entire machine's movement. Additionally, two other stepper motors are used in the linear x/y -axis actuator for processing replantation positions, with one (3.7 V) for X -axis movement and another (24 V) for Y -axis movement (Figure 2). The laser target designator with a red laser beam points out the deficiency planting positions in the 128-cell plug trays. Manual replanting was applied whenever the DDRP machine detected empty cells. The PLC mechanism control in this study works by two different maneuver control strategies: (1) Continuous Detection (CD) mode with a non-fixed advance distance (Figure 10), and (2) Batch Detection (BD) mode with a fixed advance distance (Figure 11).

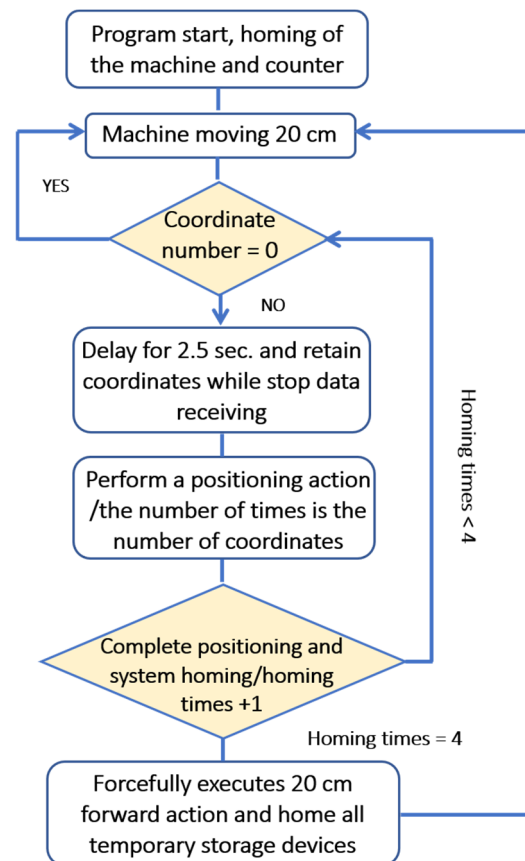


Figure 11. Flowchart of batch mode with a fixed advance distance system.

2.3.2. Continuous Detection Mode (CD Mode) of the DDRP System with Non-Fixed Advance Distance

In the CD mode without fixed forward distance control, the platform halts immediately upon detecting a deficient cell and initiates coordinate transformation and positioning operations. During this process, a signal is sent to the Raspberry Pi to trigger image capture, while the data transmission buffer is temporarily disabled to prevent overwriting of coordinate data. A left-shift operation is then performed on the buffer, allowing the new coordinate data to be stored in the corresponding transformation source. The number of executions is determined by the first digit of the transmitted data (Figure 8).

Following each detection of an empty cell, the system initiates a homing action and re-activates the data reception buffer to receive fresh input. Before executing the next forward movement, the system retains the accumulated homing count to prevent image recognition errors. If the homing count reaches four, the system forcibly performs a 1.5 s forward movement, resets all relevant variables, and reinitializes data reception. Upon detecting a target again, the platform halts and introduces a 2.5 s delay to allow image parameters—such as white balance—to stabilize before resuming data acquisition (Figure 10).

At system startup, the DDRP machine does not recognize the current position of the motors. Without initiating the homing process, the stepper motors may be misaligned. To ensure accurate positioning, the system must incorporate sensors capable of detecting a reference position either during startup or periodically throughout operation.

2.3.3. Batch Detection Mode (BD Mode) of the DDRP System with Fixed Advance Distance

In the BD mode with fixed-distance movement, aside from the predetermined forward step of 20 cm per cycle, the operational procedure remains identical to that of the non-fixed mode. Execution requires completing the fixed-distance advancement before initiating the

computation of coordinate data. Additionally, the forward movement is triggered only after four homing actions have been completed (Figure 11).

2.4. Experimental Setup

The main equipment used in this study is a custom-designed deficient detection and replanting positioning (DDRP) system on a pilot-scale automated DDRP machine (Figures 1 and 2). This machine is equipped with a moving function to simulate track travel mode with a fixed direction during actual operation. The steering wheels, controlled by stepper motors, maintain a fixed direction to reduce direction errors caused by the driving wheels' speed discrepancies.

The system module includes an image recognition and positioning system with a laser designator that marks and points to the actual positioning position with a red laser beam. This allows the program to be corrected according to the laser pointing and confirms the accuracy and location of the missing seedling cells in the plug trays.

The plug trays have been used in the Taiwanese seedling industry for over 30 years and come in various specifications, from 72 to 406 cells per tray. The cell shapes also vary, including basic square and round, special star, and triangular shapes. In this study, a 128-round cell (16×8 cells) plug tray was selected, measuring 60 cm $\times$ 30 cm, with artificial seedlings used instead of live seedlings due to indoor experiments without adequate solar radiation (Figures 12 and 13).

This approach was necessary due to laboratory constraints, particularly the lack of adequate solar radiation indoors. A fully controlled indoor environment provided uniform lighting conditions, minimizing external variables that could affect image processing accuracy. This ensured that the machine vision system's core performance was evaluated based on algorithmic precision rather than environmental fluctuations.

Additionally, since seedling replenishment typically occurs between 6 and 10 days after germination, maintaining a uniform appearance was crucial for recognition accuracy. Real seedlings at different growth stages exhibit variations in shape and texture, potentially compromising the system's ability to assess fundamental detection capabilities. Using standardized artificial seedlings allowed for consistent characteristics across trials, ensuring reliable evaluation of image processing techniques and deficiency detection algorithms. While artificial seedlings helped maintain controlled experimental conditions, we acknowledge that real-world applications require further validation with live seedlings under diverse lighting and environmental settings.

Seedling replenishment typically occurs 6–10 days after germination, with slight variations between summer and winter. Although a depth camera is used for image recognition, an additional network camera is mounted on top of the experimental stage to record the experimental process. This network camera captures the landing point of the laser marker and details of the entire experimental process for reanalysis (Figure 2). The experimental process is recorded using a network camera and a screen recording program. Data such as the current experiment time, PLC program data monitoring, and remote program monitoring from the Raspberry Pi are recorded for secondary analysis and confirmation. The total number of additional replanting actions, the duration of the experiment, and the number of successful seedling captures are also automatically screenshotted on the Raspberry Pi for saving coordinates and pre-processed images.

Based on practical experience in commercial seedling nurseries, a manual seedling process typically results in about 5 to 7 deficient seedlings per plug tray, with a maximum of 10 deficient seedlings. Therefore, in the systematic test experiment, 10 deficiency seedlings were pre-set for each plug tray (i.e., 10 empty cells randomly on each plug tray). The position of the 10 deficient seedlings was selected using a computer random function from

the seedling position of cell numbers 1 to 128 (Figure 12). The artificial seedlings were removed before each experiment, the missing seedlings' cell numbers were recorded, and then new deficient seedlings were randomly selected again.

While this study was conducted in a controlled indoor environment, real-world field validation is a crucial next step in assessing the system's robustness. The indoor setup ensured consistent lighting and environmental conditions, allowing a reliable evaluation of image-processing techniques and maneuver control strategies without external interference. To maintain uniformity in seedling shape, size, and color, artificial seedlings were used as experimental targets. This approach minimized growth-related variability that could affect detection accuracy, ensuring consistent evaluation parameters across trials. However, the controlled laboratory setting does not fully account for challenges such as variations in light intensity, occlusion effects, and complex backgrounds. Despite these limitations, the system incorporates photomask processing and HSV color-space conversion, significantly improving image segmentation performance under non-uniform lighting conditions.

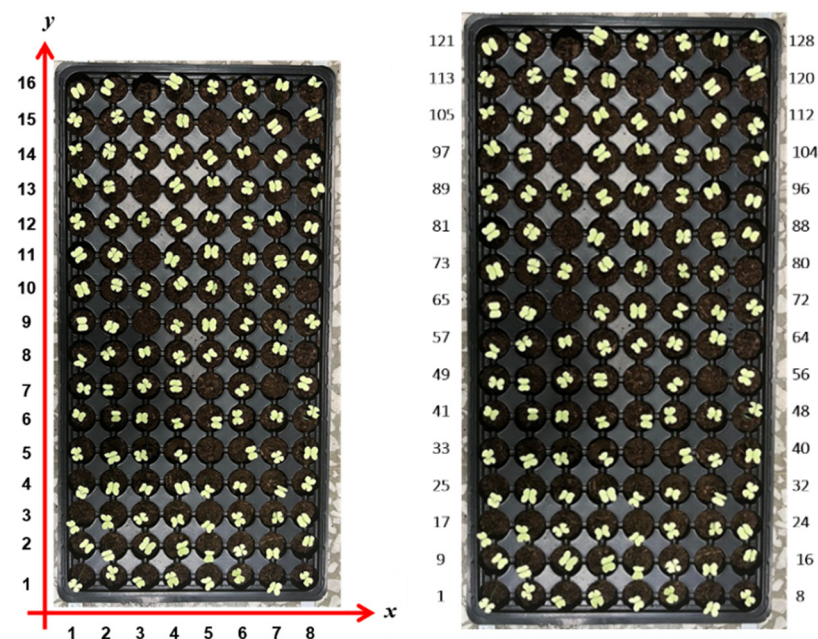


Figure 12. The tray size (Left) and cell numbering (Right) of the plug tray with seedlings.

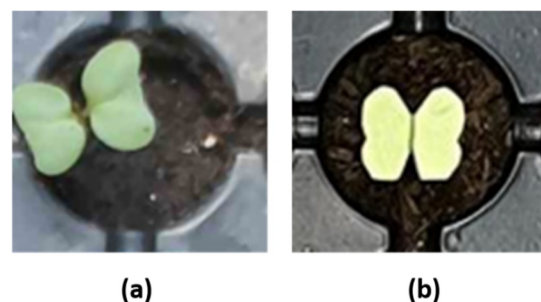


Figure 13. Comparative visualization of real 10-day-old seedlings from the nursery (a) and artificial seedlings (b). Both images were captured using a depth camera under controlled laboratory conditions.

The Effects of Different Image Processing Methods

The two methods—grayscale and Otsu thresholding processing—were compared in the control process design. The variable in the experiment was the image processing method, and the PLC adopted the batch mode of the DDRP system with fixed advance distance (Figure 10). Relevant experimental data were recorded, and all artificial seedling

replanting was carried out manually immediately after deficient detection and positioning by a laser designator.

In this phase of the experiment, the image pre-processing methods, Otsu thresholding, and HSV color space were used. Photos of the results were retained for subsequent analysis. The machine operated in batch detection mode (BD mode), detecting the target after moving a fixed distance. If the results of machine learning and image recognition are discussed separately, they are presented using the confusion matrix commonly used in machine learning (Table 1) [15]. Two of the most commonly used metrics for classification are precision and recall. These metrics are calculated using the following equations:

$$\text{Precision (\%)} = \frac{TP}{TP + FP} \times 100\% \quad (3)$$

$$\text{Recall (\%)} = \frac{TP}{TP + FN} \times 100\% \quad (4)$$

Table 1. Confusion matrix for binary classification of deficient seedling detection.

Machine Identify		Deficient Seedling (Positive)	Complete Seedling (Negative)
Practical Conditions	Deficient Seedling (Positive)	TP	FN
	Complete Seedling (Negative)	FP	TN

TN: True Negative; TP: True Positive; FP: False Positive; FN: False Negative.

For this study, each result is summed up and manually identified and recorded once per experiment. The total number of trials using Otsu thresholding processing alone is 30, and the average of these trials is used as the benchmark for comparing the total results. The number of trials processed with mixed HSV color space conversion was 90 trials. Therefore, 90 trials were individually calculated, then 30 trials were randomly selected for comparison under the same benchmark. In practice, seedling replenishment is carried out 6–10 days after germination, with slight variations in timing between summer and winter. To enable reuse and ensure uniformity in seedling age and morphological characteristics, this study employs green-colored paper and green iron wire to fabricate simulated cabbage seedlings aged 10 days as experimental materials (Figure 13).

The experiment was carried out using the same image pre-processing method, but the HSV color conversion space program was introduced for pre-processing. Variables in the experiments included the PLC's movement and detection control programs. Experiments applied both (i) continuous detection mode without a fixed distance (Figure 10) and (ii) batch detection mode with a fixed forward distance (Figure 11), and relevant data were recorded. All artificial seedling replanting was manually performed immediately after deficient detection and positioning by a laser designator.

2.5. Statistical Analysis

The statistical analysis for this trial was conducted with a one-way analysis of variance (ANOVA) and Student's *t*-test using Origin 2020b (OriginLab, Northampton, MA, USA) software. If the results from the variance analysis are significant, Tukey's honest significant difference (HSD) test would be used to compare the differences among the various factor grades. The tables were generated based on means and standard deviations and showed a significant difference when $p < 0.05$.

3. Results

3.1. The Effects of Different Image Processing Methods and the Number of Trials

Analytical results revealed that, in the fixed detection distance mode, using Process #A of the image processing strategy achieved a precision rate of $70.44\% \pm 15.69\%$ and a recall rate of $77.14\% \pm 17.36\%$ over 30 trials. On average, over 30 trials, when Process #B of the image processing strategy was applied, a precision rate of $82.14\% \pm 11.77\%$ and a recall rate of $69.35\% \pm 13.88\%$. Furthermore, when Process #B was applied and performed 90 trials, the average precision rate reached $83.78\% \pm 11.71\%$, while the recall rate was $64.12\% \pm 12.51\%$. Notably, despite a 7.79% decrease in recall rate, the precision rate improved by 11.7% when Process #B was applied and based on the results of 30 trials. The statistical analysis revealed no significant differences in precision rates between the 30-trial and 90-trial evaluations of Process #B (HGO), nor their recall rates ($p > 0.05$, Table 2). However, a significant difference was observed in the precision rates of the 30-trial evaluations between Process #A and Process #B ($p < 0.05$), while the recall rates remained statistically similar ($p > 0.05$, Table 2).

Table 2. The effects of different image processing methods and the number of trials.

Image Processing Strategies	Process #A (GO)	Process #B (HGO)	
Number of Trials	30	30	90
Precision rate (%)	70.44 ± 15.69^b	82.14 ± 11.77^a	83.78 ± 11.71^a
Recall rate (%)	77.14 ± 17.36^a	69.35 ± 13.88^{ab}	64.12 ± 12.51^b

Different lowercase letters indicate the significant difference in Tukey's test at the 5% level.

Process #A of the image processing strategy was implemented using batch detection (BD) mode in this experiment (Figure 14). The results indicated an average of 7.53 ± 1.28 seedlings with deficiencies successfully identified per tray, referred to as Seedling Identification per Deficiency (SID). The average time required to complete the detection process, Cycle Detection Time (CDT), was 3.64 ± 0.94 min (equivalent to 218.3 ± 56.5 s). Given that each plug tray was pre-set with 10 randomly selected deficient seedlings (Table 3), the SID accuracy was approximately 75.3% (i.e., 7.53 deficient seedlings detected/10 deficient seedlings $\times$ 100%). For future experiments, a program will be integrated to allow the PLC to automatically calculate the number of experimental cycles based on the executed positioning groups. The total number of positioning action groups performed throughout the experiment will also be recorded and analyzed for comparison.

Table 3. Summary of detection results for deficient seedlings in each plug tray under various image processing and maneuver control strategies.

Maneuver Control Strategy	Image Processing Strategy	
	Process #A (GO)	Process #B (HGO)
CD Mode	NA	CDT: 271.0 ± 74.4^a Accuracy of SID: $93.0\% \pm 11.9\%^a$ PIT: 23.40 ± 7.08 times ^{a}
BD Mode	CDT: 218.3 ± 56.6^b Accuracy of SID: $75.3 \pm 12.8\%^b$ PIT: NA	CDT: 244.2 ± 34.8^{ab} Accuracy of SID: $96.0\% \pm 8.0\%^a$ PIT: 16.53 ± 3.13 times ^{b} CDT: 226.5 ± 57.0^b Accuracy of SID: $98.7\% \pm 4.3\%^a$ PIT: 18.77 ± 5.18 times ^{b}

CDT (Cycle Detection Time): Average time required to complete the detection process for each tray. SID (Seedling Identification per Deficiency): Average number of seedlings successfully identified with deficiencies per tray. PIT (Position Identification Times): Average number of times the average number of times in which position identification is performed for each tray. NA: Not available. **Note:** Different lowercase letters indicate statistically significant differences according to Tukey's test at the 5% significance level.

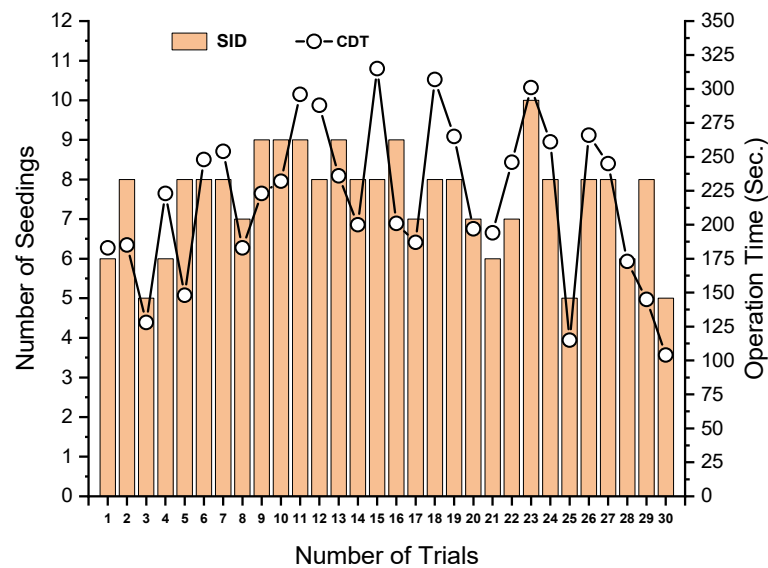


Figure 14. Results of the deficient seedling detection by Process #A of image processing with BD mode of the maneuver control strategy.

When Process #B of the image processing strategy was applied using batch detection (BD) mode, the experiment yielded a CDT value of 4.06 ± 0.58 min (equivalent to 244.2 ± 34.8 s). The SID value reached 9.6 ± 0.8 seedlings, resulting in an SID accuracy of approximately $96.0\% \pm 8.0\%$ (i.e., 9.6 deficient seedlings detected/10 deficient seedlings $\times 100\%$), based on the preset of 10 randomly selected deficient seedlings per plug tray (Table 3). The Position Identification Times (PIT)—representing the average number of times in which position identification is performed for each tray—was calculated by the PLC to be 16.53 ± 3.13 times (Figure 15).

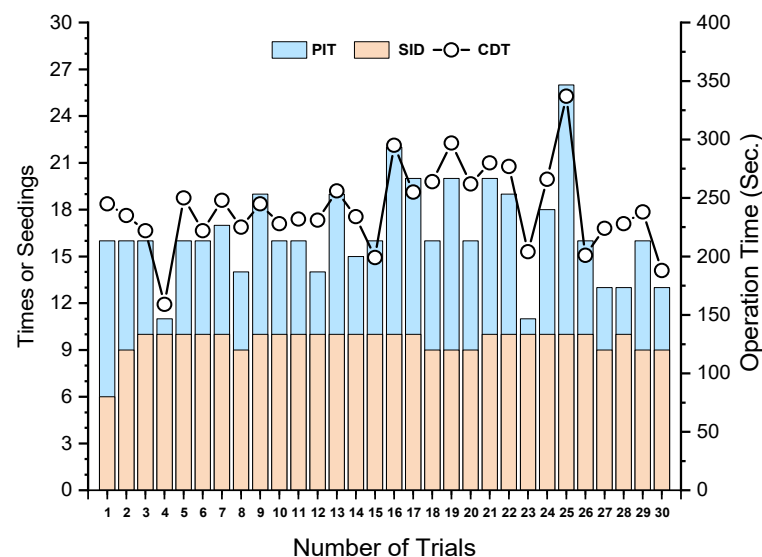


Figure 15. Results of the deficient seedlings detection by Process #B of image processing and BD mode of the maneuver control strategy.

A comparison of both image processing methods (Figures 14 and 15) reveals a significant difference in performance. However, the PLC program remained unchanged, aside from the addition of a direct count display in the monitoring window. The choice of image processing method notably influenced the detection range. When only Otsu thresholding was used, plug trays were intentionally placed in a non-uniform manner to simulate real-

world conditions. This led to excessive noise along the left and right edges, negatively impacting image recognition. Most of the unidentified empty cells were concentrated on these edges, necessitating a reduction in the identification range on both sides.

To address this issue, HSV color space conversion was integrated with Otsu thresholding. This hybrid approach effectively mitigated the noise problem and significantly enhanced the accuracy of deficient seedling identification.

3.2. The Effects of Different Maneuvering Detection Strategies

In this experiment, Process #B of the image processing strategy was applied with different maneuvering detection strategies (Figure 3). The maneuvering strategies include (1) a continuous detection mode (CD mode) without a fixed moving distance (Figure 16) and (2) a batch detection mode (BD mode) with a fixed moving distance (Figure 17).

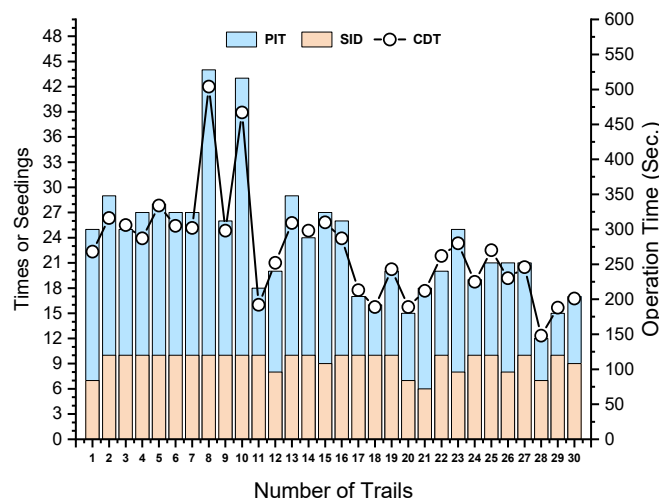


Figure 16. Results of the deficient seedling detection by Process #B of image processing and CD mode of the maneuver control strategy.

The performance of the DDRP machine operating in CD mode without fixed movement was solely dependent on the PLC program. Experimental results showed a CDT value of 4.52 ± 1.24 min (equivalent to 271.0 ± 74.4 s), with a SID value of 9.3 ± 1.19 seedlings successfully identified per tray. The Position Identification Times (PIT), calculated by the PLC, averaged 23.4 ± 7.1 times per tray (Figure 17). Based on the preset of 10 randomly placed deficient seedlings per plug tray (i.e., 10 empty cells per tray; Table 3), the SID accuracy was approximately 93.0% (i.e., 9.3 deficient seedlings detected/10 deficient seedlings $\times 100\%$).

In contrast, when the DDRP machine operated in BD mode with fixed-distance movement, the CDT was reduced to 3.77 ± 0.95 min (or 226.5 ± 57.0 s). The SID values increased to 9.87 ± 0.43 seedlings, and the PIT values averaged 18.7 ± 5.18 times per tray, as recorded by the PLC (Figure 17). Accordingly, the SID accuracy improved to approximately 98.7% (i.e., 9.87 deficient seedlings detected/10 deficient seedlings $\times 100\%$), based on the same pre-set of 10 randomly deficient seedlings per plug tray (Table 3).

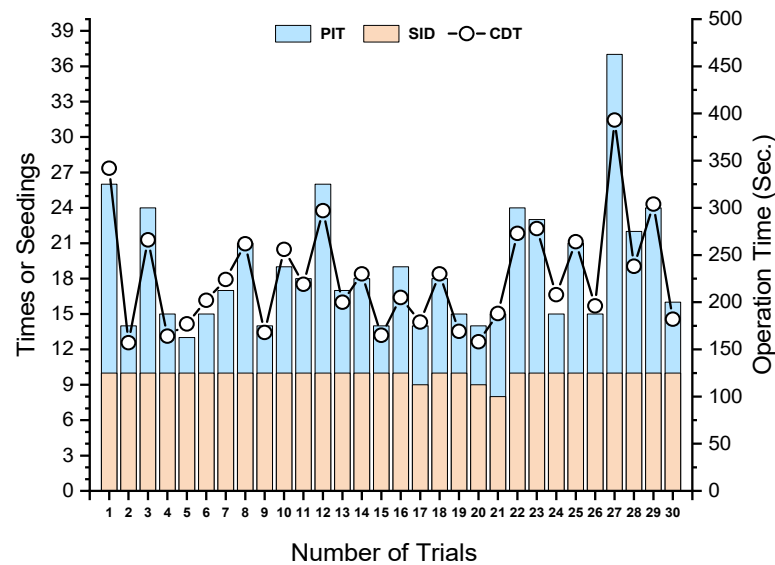


Figure 17. Results of the deficient seedling detection by Process #B of image processing and BD mode of the maneuver control strategy.

Based on the results from Figures 15–17, although the same image processing approach was used, different maneuvering strategies—namely, BD mode with fixed-distance movement and CD mode without fixed movement—produced varying outcomes in detecting deficient seedlings. The BD mode consistently achieved higher SID values, ranging from 9.6 ± 0.8 to 9.87 ± 0.43 seedlings, corresponding to an SID accuracy of 96% to 98.7%. In contrast, the CD mode yielded a SID value of 9.30 ± 1.18 seedlings, with an accuracy of approximately 93% (Table 3).

This performance gap is likely attributed to data transmission delays and the motor's inability to perform real-time positioning in CD mode. If continuous detection is preferred, enhancements in both software and hardware design will be necessary to improve system responsiveness and precision.

Statistical analysis reveals that under Process #B, there is no significant difference in SID detection accuracy between CD and BD modes ($p > 0.05$). However, significant differences were observed in CDT and PIT values ($p < 0.05$, Table 3). Additionally, BD mode under Process #A showed no significant difference in CDT ($p > 0.05$) but did exhibit a significant improvement in SID detection accuracy ($p < 0.05$, Table 3).

In repeated trials of BD mode within the Process #B strategy (Figures 15 and 17), CDT, PIT, and SID detection remained statistically consistent ($p > 0.05$, Table 3). Therefore, the most effective configuration for detecting deficient seedlings in plug trays is the BD mode paired with the image processing strategy Process #B.

4. Discussion

This study presents a low-cost, efficient machine vision system for detecting deficient seedlings in small-scale commercial nursery farms. By leveraging OpenCV's Haar cascade algorithm, a Raspberry Pi 4B, and an Intel RealSense Depth Camera D435i, we eliminate the need for expensive deep-learning models and high-performance GPUs, making automation more accessible for small-scale farms.

Compared to other machine vision approaches, our system balances cost-effectiveness with practical implementation. Previous studies highlight the trade-offs between detection accuracy, computational complexity, and hardware investment. For instance, Yan et al. [9] employed YOLOv5x to detect missing tomato seedlings, achieving 92.84% accuracy, but requiring GPU acceleration, making it cost-prohibitive for budget-constrained farms. Sim-

ilarly, Islam et al. [16] applied Mask R-CNN for seedling segmentation, attaining high precision but necessitating powerful AI hardware. Fu et al. [7] explored threshold segmentation, erosion, and expansion methods, like Otsu thresholding, which required additional pre-processing steps to enhance detection accuracy. Tseng et al. [5] designed a depth-camera-based system for rice seedling monitoring, demonstrating efficiency but introducing increased hardware complexity.

By integrating HSV conversion with Otsu thresholding in Batch Detection (BD) mode, our system achieved a 98.7% detection accuracy, surpassing the 93% accuracy of Continuous Detection (CD) mode, while reducing operation time per tray by 45 s. This performance enhancement is attributed to several core factors: (1) Lighting and Contrast Optimization: The combined HSV color-space conversion and Otsu thresholding improved segmentation under varying light conditions, enhancing recognition success across diverse plug tray environments. (2) Depth Camera Utilization: Unlike traditional RGB-based detection methods, depth imaging enables precise distance measurement, reducing errors associated with occlusion and background interference. (3) Batch Detection Efficiency: BD mode allows efficient sequential identification, minimizing redundant processing compared to CD mode, where continuous scanning introduces computational overhead and delays.

Compared to the following studies, our approach balances cost-effectiveness with automation feasibility, ensuring real-time performance without the need for expensive deep-learning infrastructure (Table 4). The hardware strategy adopted in this study was intentionally designed to democratize access to automated seedling detection, particularly for small- and medium-scale nursery farms that often lack the resources to invest in high-end computing infrastructure. By forgoing the use of a dedicated graphics card, this system dramatically lowers the barrier to entry without compromising essential performance. Our setup, which includes an Intel RealSense Depth Camera D435 (~USD 200), a Raspberry Pi 4B (~USD 75), stepper motors and actuators (~USD 40 per unit), and a PLC control system (~USD 120), totals approximately USD 435. This is a fraction of the cost of conventional deep-learning-based systems, which often exceed USD 1500 due to their reliance on industrial-grade GPUs and high-performance computing units.

Despite this lean configuration, the DDRP-Machine delivers competitive results. It achieves precision rates between 82.14% and 83.78% and maintains high accuracy (96.0–98.7%) in detecting deficient seedlings. These figures are particularly impressive given the absence of GPU acceleration and the system's real-time operational capability. As shown in Table 4, while state-of-the-art systems like YOLOv5x and Mask R-CNN offer marginally higher precision, they do so at a steep financial and technical cost. In contrast, our system strikes a rare and valuable balance: it is cost-effective, scalable, and robust enough to support real-time automation in practical nursery environments. This study clearly shows that high-performance seedling detection is possible without the burden of costly, high-end hardware. By leveraging efficient image processing and smart hardware integration, the DDRP-Machine offers a sustainable and accessible pathway toward agricultural automation, making advanced technology truly attainable for the broader farming community.

The benchmarking table compares the hardware setup of existing low-cost agricultural systems, showing that this study's DDRP-Machine system achieves a strong balance of cost-efficiency and automation without relying on GPU acceleration (Table 5). The DDRP-Machine is the only system focused on vision-based seedling detection, offering a unique solution for nursery automation. Other lab-scale systems focus on environmental sensing or irrigation, while our pilot-scale system targets vision-based seedling detection—a niche with fewer low-cost solutions. The absence of a GPU in this study is a major advantage for scalability in resource-constrained environments.

Table 4. The benchmarking table compares cost, hardware requirements, graphics card requirements, precision rates, and accuracy against existing vision-based seedling detection systems.

System	Estimated Cost (USD)	Hardware Requirements	Graphics Card Required	Precision Rate (%)	Accuracy (%)	Notes
This Study (DDRP-Machine)	~\$435	Raspberry Pi 4B, Intel RealSense D435	No	82.14–83.78	96.0–98.7	Low-cost, GPU-free, real-time detection
YOLOv5x [9]	>\$1500	High-end GPU, Industrial Camera, Workstation	Yes	92.84	92.84	Deep learning (YOLOv5x), high precision
Threshold-Based Detection [7]	~\$800	Depth Camera, PC with GPU	Yes	85.00	85.00	Threshold-based, moderate cost
Mask R-CNN [16]	>\$2000	GPU, Mask-R-CNN Framework, High-performance PC	Yes	90.00	90.00	Complex setup, high accuracy
Depth-Camera-Based Monitoring [5]	~\$1200	Depth Camera, Real-time Processing Unit	Yes	88.00	88.00	Effective for rice seedlings

Table 5. The benchmarking table compares the hardware setup of existing low-cost agricultural systems.

System	Primary Hardware Components	Estimated Cost (USD)	GPU Required	Functionality	Notes
DDRP-Machine (This Study)	Raspberry Pi 4B, Intel RealSense D435	~\$435	No	Seedling deficiency detection (pilot-scale)	No GPU; real-time automation; plug tray scanning
Smart Irrigation System (Arduino-based) LCABIS [17]	Arduino UNO, GSM module, Soil Moisture Sensor, Relay, Solenoid Valve	~\$60–100	No	Automated irrigation (lab-scale)	Simple, low-power, GSM-enabled
IoT Smart Farming (NodeMCU ESP8266) [18]	NodeMCU, DHT11, Soil Moisture Sensor, LDR, Water Pump	~\$50–80	No	Environmental monitoring, irrigation (lab-scale)	Cloud-connected via Adafruit IO
Smart Agriculture (Raspberry Pi-based) [19]	Raspberry Pi, Soil pH/NPK Sensors, Moisture Sensor, Solenoid Valve	~\$100–150	No	Soil analysis, bug detection, and irrigation (lab-scale)	Modular and scalable
YOLOv5x Vision System [9]	High-end GPU, Industrial Camera, Workstation	>\$1500	Yes	Seedling classification (pilot-scale)	High precision, but high cost and complexity

Comparison of the object detection methods for real-time nursery use among the Haar Cascade classifiers, Support Vector Machines, and Template Matching is organized in Table 6. Haar Cascade classifiers, originally developed for face detection, have been widely adopted in agricultural contexts due to their low computational cost and compatibility with lightweight hardware such as Raspberry Pi. Studies by Gowsikraja et al. [20] demonstrate detection accuracies ranging from 96 to 98% when combined with preprocessing techniques like HSV filtering and Otsu thresholding. This method is particularly suitable for real-time nursery use, offering high responsiveness and minimal training complexity. Support Vector Machines offer robust classification performance, especially when paired with handcrafted feature extraction techniques such as HOG or LBP. Wasala and Kryjak [21] highlight the method's high robustness to lighting variations and noise, making it ideal for environments with inconsistent illumination. However, the increased training complexity and hardware demands (e.g., CPU-intensive processing) limit its scalability in low-resource settings. Template Matching is a straightforward technique that compares input images to predefined templates. While it requires minimal hardware and no training, its sensitivity to scale, rotation, and lighting conditions significantly reduces its reliability in dynamic nursery environments. Mercier et al. [22] note that despite its simplicity, the method

lacks adaptability and is generally unsuitable for real-time applications involving variable seedling morphology.

Table 6. Comparison of object detection methods for real-time nursery use.

Algorithm	Computational Cost	Hardware Requirements	Detection Accuracy	Robustness to Lighting/Noise	Training Complexity	Suitability for Real-Time Nursery Use
Haar Cascade (OpenCV) [20]	Low	Runs on Raspberry Pi; no GPU	Moderate to High (96–98%)	Moderate (improved with HSV + Otsu)	Low	High-real-time capable, low-cost
SVM-Based Classification [21]	Moderate	Requires CPU; may need feature extraction tools	Moderate to High	High	High	Medium-effective but less scalable
Template Matching [22]	Low	Minimal hardware; no training	Low	Low-sensitive to scale and lighting	Low	Low-limited adaptability

5. Conclusions

This study successfully developed the DDRP-Machine, a low-cost, GPU-free machine vision system for detecting deficient seedlings in plug trays, tailored to the operational and financial constraints of small- and medium-scale nurseries. By integrating affordable components such as the Intel RealSense Depth Camera D435, Raspberry Pi 4B, stepper motors, and a PLC, the system achieved high detection accuracy (96.0–98.7%) and precision (82.14–83.78%) under controlled laboratory conditions, while maintaining a total hardware cost of approximately USD 435. Compared to deep-learning models like YOLOv5x and Mask R-CNN, which require significantly higher investment and computational resources, the DDRP-Machine offers real-time performance without GPU acceleration—making it a practical and scalable solution for resource-limited environments. The Batch Detection (BD) mode further enhances its suitability for nursery operations by reducing processing time. However, transitioning from laboratory to field conditions presents challenges such as biological variability, inconsistent lighting, and mixed crop types. To address these issues, future research should focus on integrating adaptive lighting, real-time correction mechanisms, and self-learning algorithms to improve robustness and field reliability. We recommend that agricultural technology developers and nursery operators explore the DDRP-Machine as a foundation for scalable automation. Further validation across diverse crop species and nursery environments is essential to confirm its adaptability and to promote broader adoption in precision agriculture.

Author Contributions: Investigation, Software, Conceptualization, Methodology, Validation, Formal analysis, Data curation, Writing—original draft, P.-J.S.; Supervision, T.-M.C.; Supervision, Funding acquisition, Writing—original draft, Writing—review and editing, J.-J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Council (NSTC), Taiwan, under contract No. NSTC 114-2321-B-002-017.

Data Availability Statement: The original contributions presented in this study are included in the article.

Acknowledgments: The authors also thank Wei-Chen Chen for the statistical analysis of the data.

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

References

- Guo, T.-J. Organic rice seedling cultivation operation. In *Hualien District Agricultural News 112 (In Chinese)*; Hualien District Agricultural Research and Extension Station, Council of Agriculture: Hualien, Taiwan, 2020. Available online: https://www.hdares.gov.tw/upload/hdares/files/web_structure/13154/03.pdf (accessed on 22 October 2025).
- An, J.H.; Liu, M.C.; Chang, J.S.; Chang, Y.L.; Chen, S.Y.; Chen, S.C.; Liu, C.H.; Tseng, S.Y. Investigation and analysis of the current situation of Taiwan's vegetable seeding industry in 2019. *Newsl. Seedl. Technol.* **2020**, *110*, 9–13. (In Chinese)
- Wen, Y.; Zhang, L.; Huang, X.; Yuan, T.; Zhang, J.; Tan, Y.; Feng, Z. Design of an experiment with seedling selection system for automatic transplanter for vegetable plug seedlings. *Agronomy* **2021**, *11*, 2031. [\[CrossRef\]](#)
- Syed, T.N.; Liu, J.; Xin, Z.; Zhao, S.; Yan, Y.; Mohamed, S.H.A.; Lakhari, I.A. Seedling-lump integrated non-destructive monitoring for automatic transplanting with Intel RealSense depth camera. *Artif. Intell. Agric.* **2019**, *3*, 18–32. [\[CrossRef\]](#)
- Tseng, H.-H.; Yang, M.-D.; Saminathan, R.; Wu, D.-H. Rice Seedling Detection in UAV Images Using Transfer Learning and Machine Learning. *Remote Sens.* **2022**, *14*, 2837. [\[CrossRef\]](#)
- Samiei, S.; Rasti, P.; Vu, J.L.; Buitink, J.; David Rousseau, D. Deep learning-based detection of seedling development. *Plant Methods* **2020**, *16*, 103. [\[CrossRef\]](#) [\[PubMed\]](#)
- Fu, W.; Gao, J.; Zhao, C.; Jiang, K.; Zheng, W.; Tian, Y. Detection method and experimental research of leafy vegetable seedlings transplanting based on a machine vision. *Agronomy* **2022**, *12*, 2899. [\[CrossRef\]](#)
- Xia, Y.; Zhu, Z.; Liu, X. SSM-based detection of rice seedling deficiency. *Sci. Rep.* **2025**, *15*, 22605. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yan, Z.; Zhao, Y.; Luo, W.; Ding, X.; Li, K.; He, Z.; Shi, Y.; Cui, Y. Machine vision-based tomato plug tray missed seedling detection and empty cell replanting. *Comput. Electron. Agric.* **2023**, *208*, 107800. [\[CrossRef\]](#)
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [\[CrossRef\]](#)
- Sezgin, M.; Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging* **2004**, *13*, 146–165. [\[CrossRef\]](#)
- Ali, S.S.; Al'Amri, J.H.; Abbas, T. Face Detection Using Haar Cascade Algorithm. In Proceedings of the 5th College of Science International Conference on Recent Trends in Information Technology (CSCTIT2022), Baghdad, Iraq, 15–16 November 2022.
- Al-Zaydi, Z.Q.H.; Ndzi, D.; Sanders, D.A. Cascade method for image processing-based people detection and counting. In Proceedings of the 2016 International Conference on Image Processing, Production and Computer Science (ICIPCS'2016), London, UK, 26–27 March 2016; pp. 30–36.
- Kulkarni, A.; Chong, D.; Batarseh, F.A. Foundations of data imbalance and solutions for a data democracy. In *Data Democracy at the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*; Academic Press: Cambridge, MA, USA, 2020; pp. 83–106. [\[CrossRef\]](#)
- Islam, S.; Reza, M.N.; Chowdhury, M.; Ahmed, S.; Lee, K.H.; Ali, M.; Cho, Y.J.; Noh, D.H.; Chung, S.O. Detection and segmentation of lettuce seedlings from seedling-growing tray imagery using an improved mask R-CNN method. *Smart Agric. Technol.* **2024**, *8*, 100455. [\[CrossRef\]](#)
- Sy, J.B.; Panganiban, E.B.; Abdulkarim Seid Endris, A.S. A low-cost Arduino-based smart irrigation system (LCABSIS). *Int. J. Emerg. Trends Eng. Res.* **2020**, *8*, 5645–5650. [\[CrossRef\]](#)
- Choudhary, A. *IoT-Based Smart Agriculture Monitoring System*; Circuit Digest: Rajasthan, India, 2021. Available online: <https://circuitdigest.com/microcontroller-projects/iot-based-smart-agriculture-monitoring-system> (accessed on 22 October 2025).
- Nowas, N. *IoT-Based Smart Agriculture Monitoring System*; Sixfab: Berlin, Germany, 2022. Available online: <https://sixfab.com/blog/smart-agriculture-projects-using-raspberry-pi/> (accessed on 22 October 2025).
- Gowsikraja, P.; Thevakumaresh, T.; Raveena, M.; Santhiya, J.; Vaishali, A.R.R. Object detection using the Haar Cascade machine learning algorithm. *Int. J. Creat. Res. Thoughts* **2022**, *10*, C742–C745.
- Wasala, M.; Tomasz Kryjak, T. Real-time HOG+SVM-based object detection using SoC FPGA for a UHD video stream. In Proceedings of the 2022 11th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 7–10 June 2022. [\[CrossRef\]](#)
- Mercier, J.-P.; Garon, M.; Gigue' re, P.; Lalonde, J.-F. Deep Template-based Object Instance Detection. *arXiv* **2021**. [\[CrossRef\]](#)

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