

Article

Intelligent Control System to Irrigate Orchids Based on Visual Recognition and 3D Positioning

Ting-Wei Chang, Wei-Cheng Wang and Rongshun Chen *

Power Mechanical Engineering, National Tsing Hua University, Hsinchu City 300, Taiwan; arsenelupin2512@gmail.com (T.-W.C.); s106033853@m106.nthu.edu.tw (W.-C.W.)

* Correspondence: rchen@pme.nthu.edu.tw; Tel.: +886-3574-2596

Abstract: This work develops a novel automatic irrigation system to implement the customized and accurate watering for an individual seedling. The system integrates the modules of visual recognition of the stem-leaf junction, identification of the stem-root junction as the watering point, and control of the spraying nozzle. The model of YOLOv3 is employed to screen the stem-leave junction of an orchid seedling, whose depth map then acquired by the method of Semi-Global Block Matching (SGBM) extracts the three-dimensional (3D) coordinates of the junction center. Next, the concept of leaf vector is introduced to identify the stem-root junction of the orchid seedling as the accurate watering point, which the spraying nozzle is controlled to reach for supplement of the specific amount of water. A number of experiments were conducted to verify the proposed irrigation system for orchid seedlings at different locations with various heights. The experimental results show that the rates of successful watering are 82% and 83.3% for the uni-pot and multi-pot orchid seedlings, respectively.

Keywords: automatic irrigation system; orchid seedling; object detection; stereo vision; depth map



Citation: Chang, T.-W.; Wang, W.-C.; Chen, R. Intelligent Control System to Irrigate Orchids Based on Visual Recognition and 3D Positioning. *Appl. Sci.* **2021**, *11*, 4531. <https://doi.org/10.3390/app11104531>

Academic Editor: Joachim Müller

Received: 31 March 2021

Accepted: 11 May 2021

Published: 16 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

Phalaenopsis are high-price plants, which need to be cautiously taken care of before they are sold. The orchid seedlings are traditionally irrigated by individually watering according to their growth condition. However, intensive manpower must be required for in-person watering, which would limit the massive production capacity of orchid seedlings, especially under the shortage of labors in some countries. For this intrinsic topic, the so-called *sprinkler irrigation* is one of the most popular and economical irrigation systems for the greenhouse, where the seedlings are watered automatically by means of *showerheads* installed over the assembly line. As being advantageous on watering a large area, with the addition of the machine vision, various schemes of global irrigation systems were proposed to locate and water the seedlings in the greenhouse [1,2]. Nevertheless, a high volume of water is poured onto the top of the seedlings, which then suffer overmuch moist leaves, uneven watering, or yet a deluge of water at the root. Consequently, Phalaenopsis plants would be diseased.

To solve this problem, this work aims at an effective vision-based watering approach to prevent plants on the assembly from being often mis-irrigated. In contemporary researches, the emergence of deep-learning based computer vision has contributed to detect and inspect the abnormal incidents in rather wide fields. For instance, a rapid recognition method was presented to examine the defects of electronic components [3]. Moreover, traffic conditions can be monitored by the computer vision techniques based on YOLOv3 (You Only Look Once version three) and spatial pyramid pooling (SPP) [4,5]. A low-cost swine surveillance system was accomplished by an automated vision-based object detector, for husbandmen to manage a large-scale swine farm in a cost-effective manner [6]. Additionally, even the machine vision with deep learning becomes more and more popular for the object detection in several agricultural applications, e.g., the coffee beans and the orchid seedlings [7–11].

Over the past decade, a plenty of deep-learning based techniques have been launched for the object detection and recognition. For example, the region-based convolutional neural network (R-CNN) [12] was proposed by Girshick et al. to succeed in applying the deep learning on the object recognition, where a convolutional neural network (CNN) and a selective search for region proposals were hybridized for object detection and classification. Fast R-CNN [13] advanced the computational efficiency by utilizing the region of interest pooling (RoIpooling) to create the feature map for only single computation on the same region proposal, which was repeatedly computed in the classical R-CNN.

Even though Fast R-CNN is superior to R-CNN in computational efficiency, both of them spend much time to acquire the region proposals by taking the same strategy—the selective search (SS). For this reason, the region proposal network (RPN) was firstly implemented in Faster R-CNN [14], to obtain the region proposals in an image frame by using neural networks to search positive and negative anchors, which then were classified by a *softmax* function and were positioned more accurately by an optimal *bbbox* regression. Despite satisfactory accuracies of the R-CNN based methods, high computational cost would be required to result in a low detection speed when using general computers, e.g., desktops and laptops.

To conquer the above-mentioned trouble, a new deep learning framework coined YOLO [15] was invented by Redmon et al. Unlike Faster R-CNN, YOLO considers the object detection as a regression problem and detects all interested objects of an entire image within the same period by the CNN, so it can perform better than R-CNN on the aspect of the computational efficiency. Soon afterwards, Redmon and Farhadi proposed YOLOv2 [16], which specialized in better speed and accuracy for the object detection and recognition. The algorithm of YOLOv2 obtained the aspect ratio of the image border by introducing the idea of *anchor box* in Faster R-CNN and applying the *k*-means clustering, replaced the dropout strategy with the batch normalization, and turned the CNN into the darknet-19. Later in 2018, to further raise the performance of YOLOv2, Redmon and Farhadi developed YOLOv3 [17], being involved with the following improvements. Firstly, the darknet-53 with deeper layers is substituted for the darknet-19. Secondly, the classifier is achieved by the *logistic* regression, instead of the *softmax* function. Moreover, the Feature Pyramid Network (FPN) is also introduced to realize the multi-scale inspections. Figure 1 proves that YOLOv3 spends only 22 ms to detect objects, and performs more accurate recognition than a large portion of other detection methods.

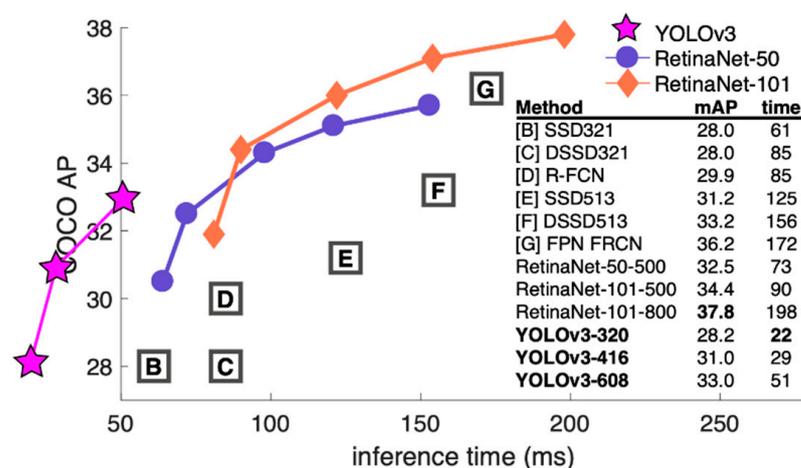


Figure 1. YOLOv3 competes with other detection methods for comparable performance, under the platform either an M40 or Titan X, with essentially the same GPU [17].

Accordingly, based on the object detection techniques of YOLOv3, this work develops a novel automatic irrigation system to implement the customized and accurate watering for an individual seedling, via the following ways: 1. Orchid seedlings are visually identified and selected by YOLOv3. 2. Orchid seedlings are accurately positioned in

three-dimensional (3D) space. 3. The stem-root junction is deemed the watering target to avoid damp leaves. 4. Intelligent control is introduced to adjust the watering amount.

The remaining parts of this work are listed as below. The system design and methodology are detailed in Section 2. The feasibility of the proposed system is verified through a number of experiments, as described in Section 3. Finally, Section 4 concludes this research and provides possible extensions.

2. System Design and Methodology

The proposed irrigation system for the orchid seedlings consists of several modules, such as the visual recognition, the construction of the 3D coordinate system, the positioning of the watering point, and the control for the system process, as shown in Figures 2 and 3. The framework of YOLOv3 is used for the visual recognition to identify and frame the seedlings. The 3D coordinate system is established through the depth maps corresponding to the output images of YOLOv3 for localization of the seedlings in the 3D space. The desired watering point is the stem-root junction of the seedling, which can be positioned and watered accurately by the spraying nozzle. The system process control takes into account the sequential control of automatic watering process.

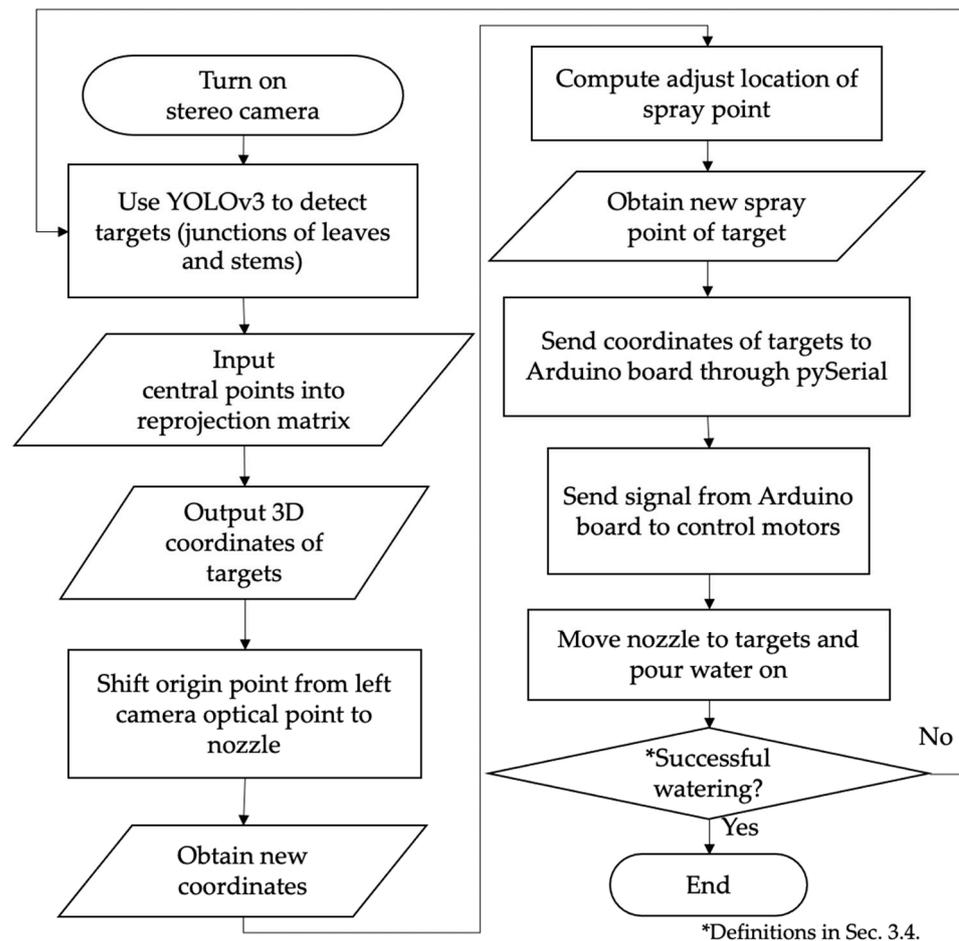


Figure 2. Flow chart from object detection (YOLOv3) to automatic watering process.

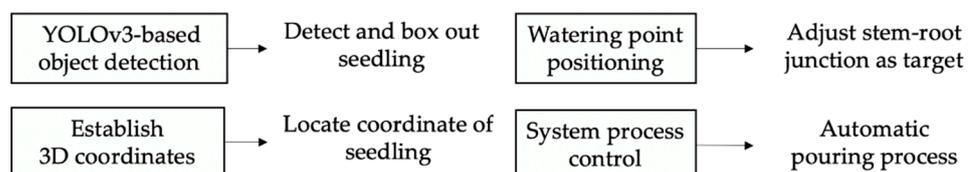


Figure 3. Functional blocks of the proposed visual recognition system.

2.1. Object Detection System

In order to accurately identify the location of each seedling to be watered and to distinguish the characteristics of each seedling from overlook view, this study frames the junction between the stem and leaf of the seedling, and collects hundreds of seedling over-view images to train the learning model of YOLOv3.

Firstly, this work manually frames the image datasets for junctions between stems and leaves of all seedlings via the software *LabelImg*. Secondly, the image datasets are divided into a high ratio of training, accompanied with a low ratio of validation and testing sets. Finally, the tactic of cross validation is selected to train the YOLOv3 model. The training and validation sets are used, respectively, to training the model and to prove the prediction accuracy of the trained model, which is followed eventually by the step that the use of the testing set is to provide an objective evaluation on a final model fitting the training set.

2.2. Construction of the 3D Coordinate System

The binocular visual system is established for the 3D coordinate system in this work. The distribution principle is to retrieve images of an object from different positions based on the binocular (dual lens). As illustrated in Figure 4, the true distance Z , between the object and the lens, can be calculated as follows:

$$Z = \frac{b * f}{X_L - X_R} = \frac{b * f}{d} \tag{1}$$

where $d = (X_L - X_R)$ is the disparity, X_L and X_R are the x -coordinates on the left and right images, respectively, b is the length of the base line (distance between the optical axes of both cameras), and f is the focal length of the camera.

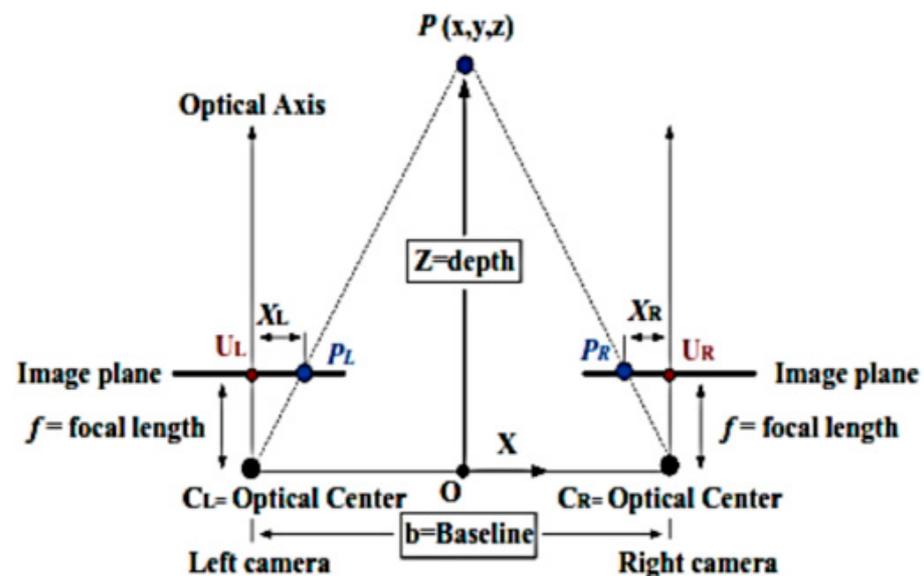


Figure 4. Binocular ranging principle [18].

The image coordinates (u, v) can be reprojected to the world coordinates (X, Y, Z) through the 4×4 reprojection matrix (Q) in order to obtain the actual 3D coordinates, which can be related according to Equation (2).

$$Q \begin{bmatrix} u \\ v \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{2}$$

where d is the disparity and W is the distance parameters of homogeneous coordinates.

The most widely used algorithm to match the feature point is Semi-Globe Block Matching (SGBM) [19–21], which obtains the Sum of Absolute Differences Window (SAD) to calculate the cost [22], as well as takes the left-camera image as the reference and the right-camera image as the target to perform pixel feature matching on the same epipolar line. Finally, the obtained parallax is also utilized by the SGBM algorithm to carry out the stereo vision. In this work, the left- and right-camera images are processed in grayscale, and then the feature matching is performed through the SGBM algorithm.

In the OpenCV function library [23], the function `cv2.StereoSGBM_create()` performs the execution of the SGBM algorithm. There are no fixed values for the parameters of SGBM to output the grayscale depth map, which further requires `cv2.reprojectImageTo3D()` to get the true 3D coordinates corresponding to the active pixel. After inputting the grayscale depth map and the reprojection matrix Q , the function `cv2.reprojectImageTo3D()` can output a depth map, each pixel of which tells 3D coordinates in the workspace with respect to the origin, i.e., the optical center of the left lens in Figure 4.

In order to make it convenient to observe the outputted depth map, the original noisy gray-scale depth map is converted into the pseudo-color depth map, which reduces the noises using the weighted least square (WLS) filter. It is a kind of edge-preserving filter that can smooth the whole image at the same time. OpenCV provides the `cv2.applyColormap()` function to convert gray-scale graphs to 12 color maps. Here, `COLORMAP_JET` mode is selected for the conversion process. On this mode, Figure 5 displays a pseudo-color map, where pixels with deeper red and deeper blue colors hold higher and lower gray values, respectively, in the original grayscale graph. Moreover, at the same pixel, the deeper red the color is, the closer the physical point is from the optical center of the left lens in Figure 4.

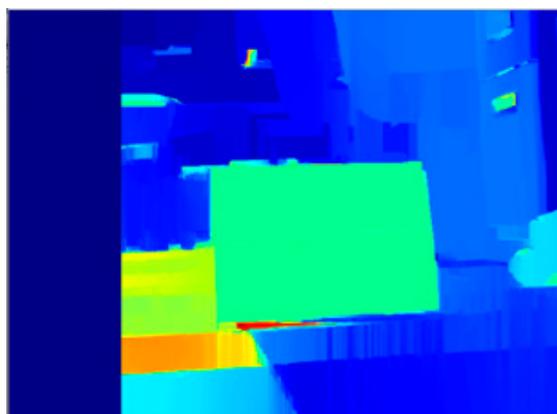


Figure 5. Effect after pseudo-color process, the deeper red (blue) is the closer (farther).

2.3. Positioning of the Watering Point

The established identification and depth map of YOLOv3 will be merged. According to actual requirements, when seedlings appear within the field of view of binocular lens, the following processes are executed: 1. Apply YOLOv3 to identify whether there is a target in the image (the junction between stems and leaves) through the left lens; 2. if the target object is detected, the coordinates of pixels at the upper-left and lower-right corners, on the prediction box in Figure 6, are, respectively, (x_{min}, y_{min}) and (x_{max}, y_{max}) , which are returned to calculate the central position of the prediction box; 3. input the center position to the map to obtain the corresponding real 3D coordinates.

After the 2D coordinates of the central point of the prediction box are obtained, its actual 3D coordinates can be obtained by feeding the 2D coordinates and the depth map to `cv2.ReprojectImageTo3D()`.

To refrain from diseases caused by directly watering on the leaves, the direction vector of the leaves (called *leaf vector* elsewhere), acquired by image processing, is defined as the fitted straight-line vector of the two main contours of leaves, as shown in Figure 7. After

the leaf vector is calculated, the desired watering point is locked to the stem-root junction of the seedling, as depicted in Figure 7.

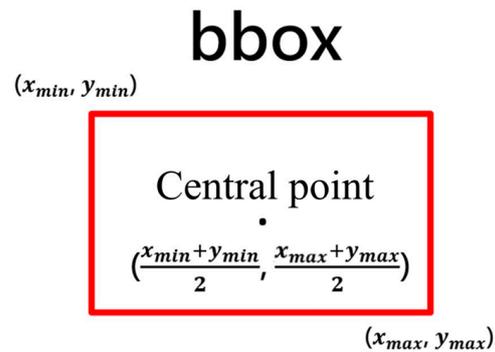


Figure 6. Illustration of center coordinates of the prediction box.

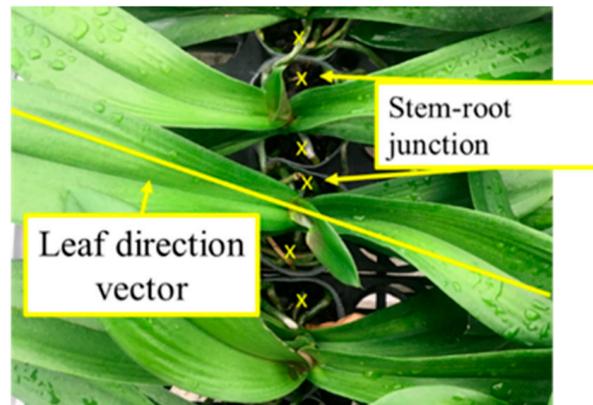


Figure 7. Direction vector of the leaf and target watering point to be modified (stem-root junction).

After selecting the junction position between the stem and the leaf through the YOLOv3 frame, the image of the center point of the prediction box is cropped, as provided in Figure 8. As different image sizes will induce redundant adjustments during the following erosion and expansion processes, the shorter length (or width) of the cropped image is set to be constant by the OpenCV function *cv2.resize()* to simplify the image processing.

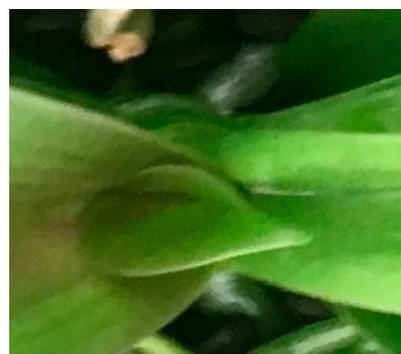


Figure 8. Cropped image of YOLOv3 prediction frame.

Then, to reduce noises, the function *cv2.cvtColor()* is used for grayscale processing of the cropped image. Moreover, the function of Gaussian blur, *cv2.GaussianBlur()* is utilized to blur the noises, such as spots, and finally the function *cv2.Canny()* is applied for edge detection. The effects of grayscale processing, Gaussian blur and edge detection are demonstrated in Figure 9a–c, respectively.

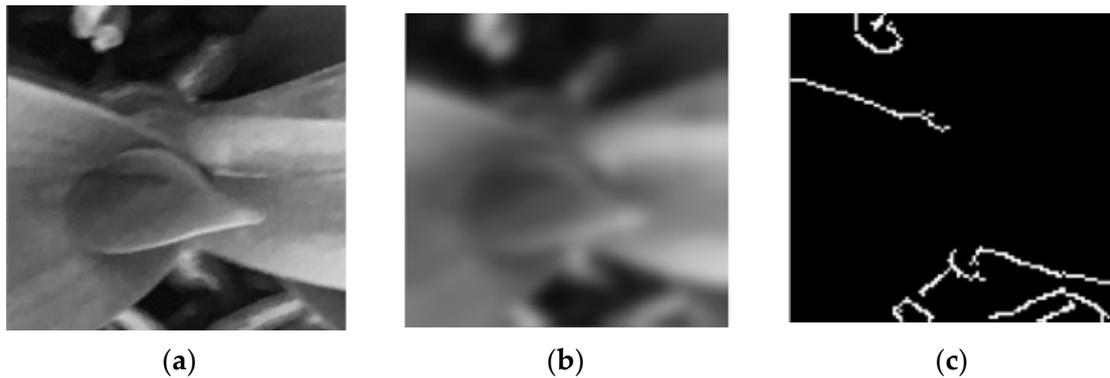


Figure 9. Edge detection effect: (a) grayscale process; (b) Gaussian blur; and (c) edge detection.

In order to merge the effective line segments at the edge of the main body, the contour is expanded using the function *cv2.dilate()*, as shown in Figure 10a. As the corner of the line segment may influence the calculation process of the leave vector, i.e., the folding of the line segment may reduce the single directionality of the contour area, the function *cv2.erode()* is invoked to properly erode the expanded contour, as illustrated in Figure 10b.



Figure 10. Effects of dilation and erode processes.

The function *cv2.findContours()* is utilized to calculate the areas of all blocks, the largest ones of which, selected through the function *sorted()*, represents the contour-of-interest of the image. As represented in Figure 11a, the area marked in red is the largest part, for which the function *cv2.fitLine()* is used to get the unit vector of the fitting line, clearly overlapped on the original image, as depicted in Figure 11b.



Figure 11. Fitted line effect: (a) Marked largest area; (b) fitted line.

The desired watering point on the stem-root junction is calculated after obtaining the actual 3D coordinates of the center point and the leaf vector. Through the customized

function, the original fitted line vector is inputted to compute the vertical unit vector by the inner product. Since the vertical unit vector can be represented in two directions, the system watering is set to water from the target with a smaller X-coordinate to the one with a larger X-coordinate, in order to reduce the total moving distance of the motor. Then, the distance between the new watering and central points for the seedling plant is set and multiplied by the selected unit vector to obtain the corrected distance in the X- and Y-directions. According to the size of the seedling plant, the new targeted watering point will be 1.5 to 2.5 cm far from the center of the seedling plant, which is on the straight line of the leaf growth, as illustrated in Figure 12.

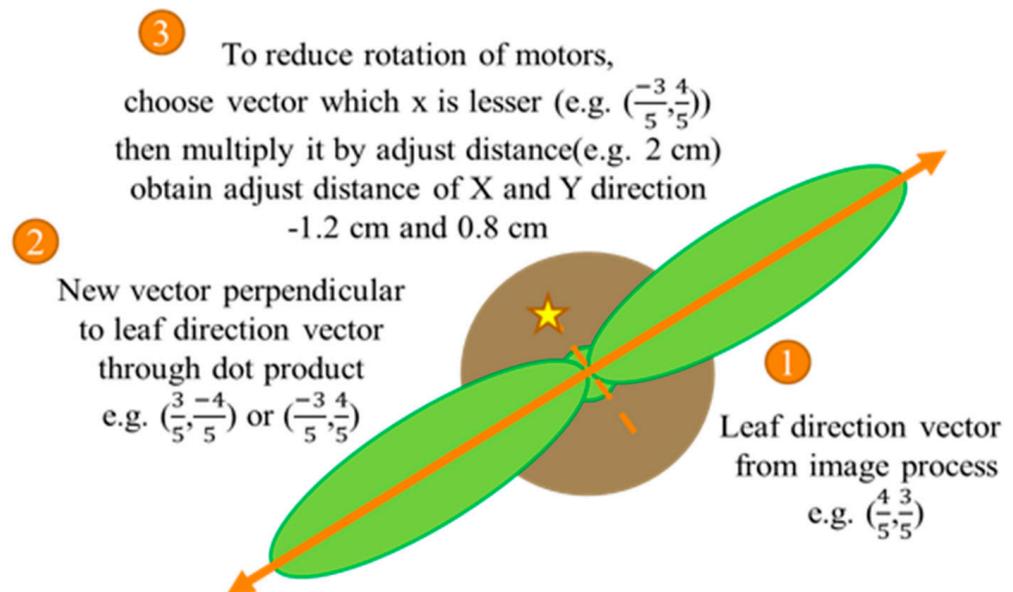


Figure 12. Elaboration on correction process of targeted watering point (stem-root junction).

In addition to the corrected distance in the direction of X- and Y- directions, since the height of the stem-root junction area will be slightly lower than the center of the seedling plant, an additional offset for the Z-direction height will be taken into account. After including the tri-axial correction values to relocate the seedling, the original 3D coordinates can be transferred to the real 3D coordinates for accurately targeting the watering point. If the leaf vector cannot be found after selecting the prediction box of YOLOv3, the distances in the X- and Y-directions shall not be corrected. However, for the Z-direction, the collision between the sprinkler and the center of the seedling shall be prevented by increasingly adjusting the height for watering via the customized functions. To make clear, the whole correction process for 3D coordinates of the seedling is described in Figure 13.

2.4. System Process Control

The system process control is divided into four parts: the motor control, the command control, the triggering process and the automatic watering process.

The three-axis sliding table is driven by three stepper motors with the synchronous wheel timing belts and screws. Firstly, for the motor control, the computer uses Python and Arduino to execute the operation instructions of the stepping motor through the Arduino control board. Secondly, the command control for the stepping motor to accomplish the specified task is implemented by mutual communications between Arduino and the computer via the package *pySerial* of Python. The function *mot.write()* is used to pass a set of strings combining letters and numbers (e.g., j0), which stand for the code of the task type and the relevant parameters, respectively. Afterwards, the functions, *Serial.read()* and *Serial.Parseint()*, are used in the Arduino to receive the commands to control the motor.

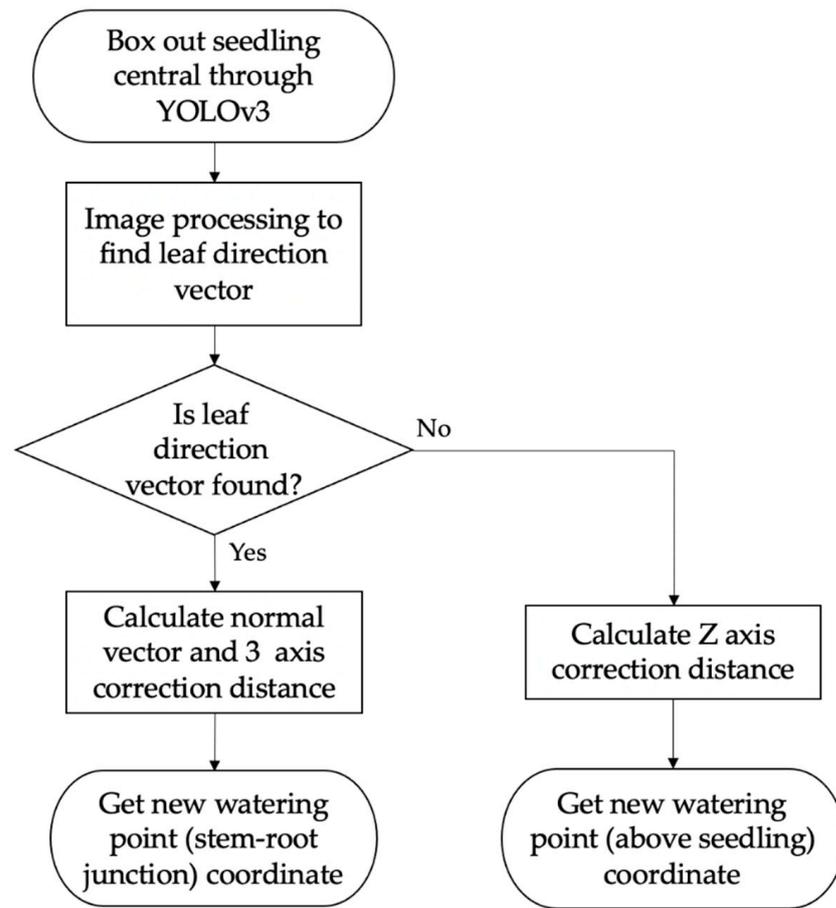


Figure 13. Correction process of watering point.

Thirdly, the triggering of the watering process is started to launch the automatic watering process, as addressed in Figure 14. If the seedlings enter the camera screen and the YOLOv3 boxes the junctions between stems and leaves to several blocks, the specific one of which is selected in the picture to trigger the watering process.

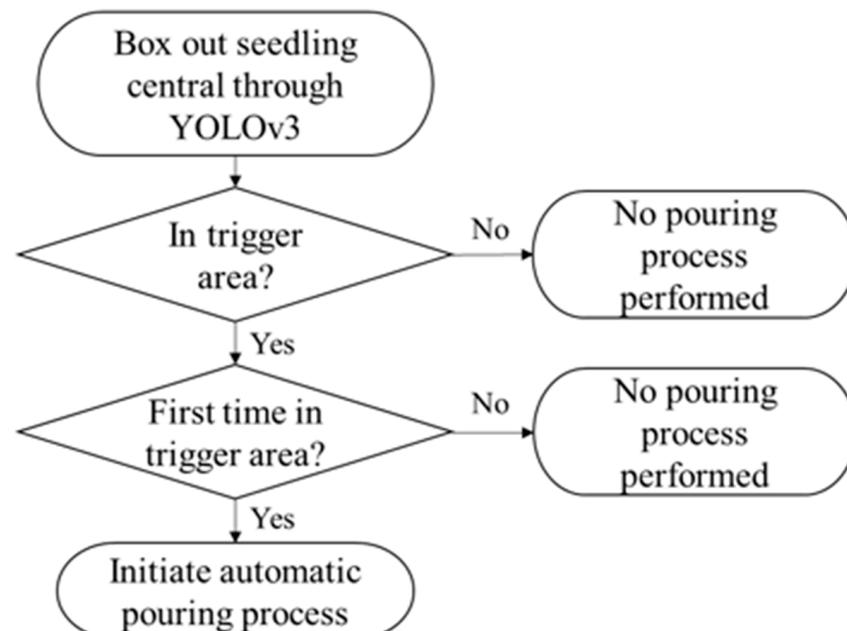


Figure 14. Flow chart of triggering for the watering process.

The area that triggers the process is represented as the function *rectangle()* with a green line in the real-time image of the left image. The triggering area is located above the center of the left lens, with a length of 240 pixels and a width of 120 pixels. When the central point of the prediction box enters into the triggering area, the watering process will be triggered.

Whether the prediction frame firstly enters the triggering area is determined by the fact that the number of prediction frames is greater than zero and the central points of all prediction frames are in the triggering area. At this time, the trigger signal is set to 1. If the number of frames is greater than zero, but the central point of the prediction frame does not enter the triggering area, the trigger signal is set to 0. When the number of prediction frames equals zero, the trigger signal is also set to 0. When the current trigger signal is greater than the previous one, the automatic watering process will be triggered. This approach ensures that the prediction frame staying in the triggering area will not cause repeated watering problems.

Finally, the watering process is completed by the customized functions, which at first, defines the moving boundary of each axis motor, and confirms whether each group of target coordinates exceeds the moving boundary. If any group of target coordinates exceeds the moving boundary, it will be removed. When the sprinkler reaches the plane position of the targeted watering point, the Z-axis motor is lowered to the specified height. Then, the valve is opened with a duration of time to satisfy the preset watering volume for completing the pouring step. Once the irrigation is accomplished, the Z-axis motor is raised to the specified height, e.g., 2 to 5 cm, based on the dimensions of the seedlings. The overall flowchart is detailed in Figure 15.

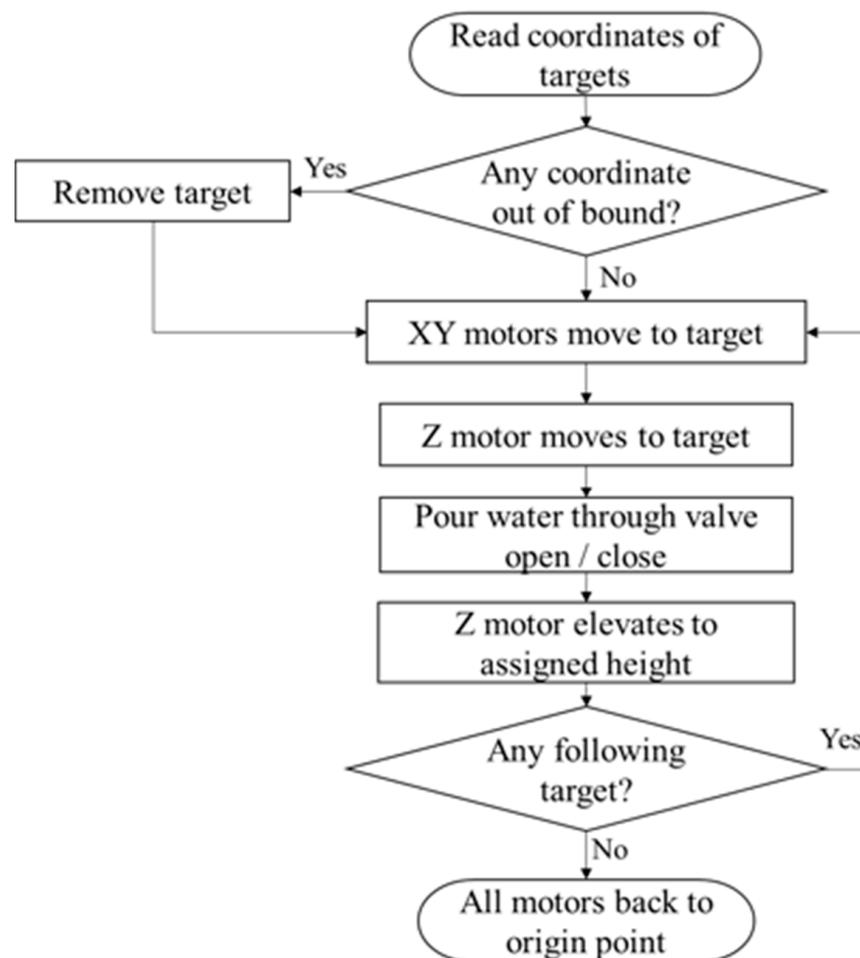


Figure 15. Flow chart of automatic watering process.

2.5. System Setup

The system platform is demonstrated in Figure 16, where an Arduino control board, a stepper motor drive board, a linear-slider driving system, a solenoid valve, a water storage container and a stereo camera are all installed on the three-axis motion platform. This platform is an aluminum extruded structure of 61.6 cm (width) \times 61.6 cm (length) \times 40.2 cm (height); the accessory carrier is an aluminum extruded structure of 21.0 cm (width) \times 40.2 cm (length) \times 62.5 cm (height). Furthermore, the watering nozzle is at a height of 22.9 cm from the ground.

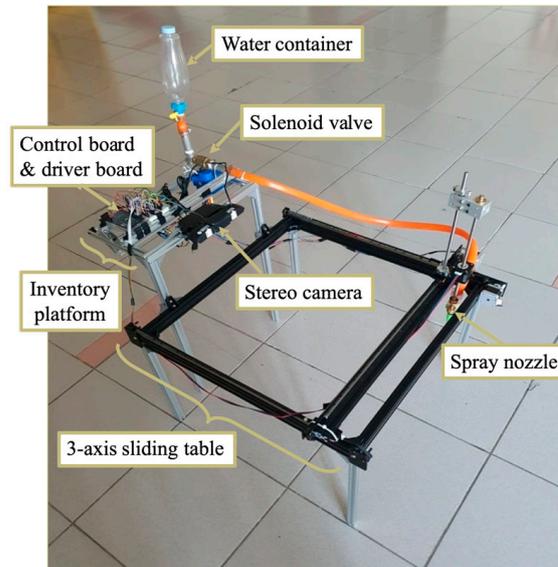


Figure 16. The platform of the automatic watering system.

For the image processing, the system employs the binocular cameras, being horizontally downward and parallel to the ground, fixed at a height of 63 cm from the ground. For the object detection and the obtainment of depth maps with YOLOv3, the left lens of the stereo camera is used as the main image acquisition device, so the center of the left lens is placed at the horizontally symmetric axis of the three-axis motion platform. Figure 17 defines positive directions indicated by yellow arrows labeled as X(+), Y(+) and Z(+), respectively towards positive directions of X-, Y-, and Z-axes of camera coordinates. As drawn in Figure 18, the origin of the three-axis motion platform is denoted as (0, 0, 0), and yellow arrows labeled with X(+), Y(+) and Z(+), representing the positive directions of X-, Y-, and Z-axes, respectively. Furthermore, the allowable travel lengths of the platform in the X-, Y-, and Z-directions are 40 cm, 40 cm and 16 cm, respectively.

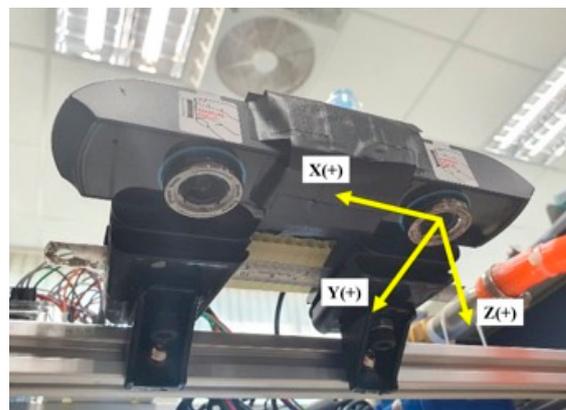


Figure 17. The coordinate system of the binocular camera.

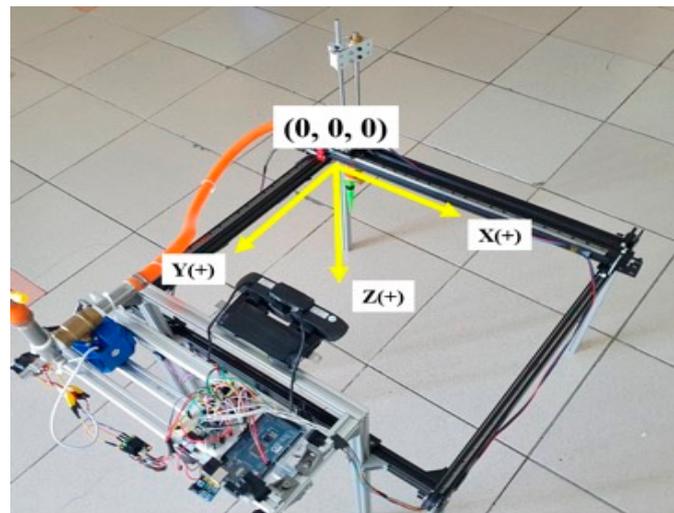


Figure 18. The coordinate system of the three-axis motion platform.

3. Results and Discussion

3.1. Model Training Results

There are totally 950 images in the dataset of the frame selection in this work, including 600, 150, and 200 ones in the training, validation, and test sets, respectively. The training and validation sets were put together into the model training. After the training process, the accuracy rates (also known as the average precision, AP) of the validation set and the test set are 92.63% and 86.23%, respectively, which implies that this work is of high practicability. The object detection results of the partial training set are shown in Figure 19.



Figure 19. Object detection for the testing set, junctions between stems and leaves are framed.

3.2. 3D Localization

There are two conditions for the experiments of 3D localization. Under Condition 1, the central point of the target is considered the measurement point and departs from the optical center of the left lens by 40 cm, as displayed in Figures 20 and 21.

Under Condition 2, a sample potted plant with the diameter of 8 cm, the length of 31 cm and the width of 7 cm, as labeled in Figure 22, is utilized for experiments in several 3D positions. They are given by the four coordinates (10, 30, 13.5), (10, 35, 13.5), (25, 30, 13.5) and (25, 35, 13.5) in the unit of centimeters, which are called Points 1 to 4, respectively, for the triggering areas in the field of view of the left lens, as demonstrated in Figure 23.

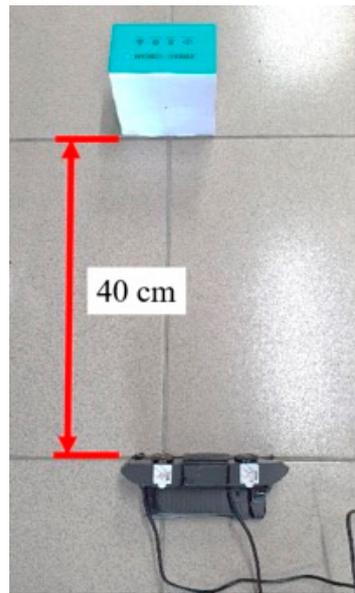


Figure 20. View of Condition 1, 40 cm far from the target to the optical center of the left lens of camera.

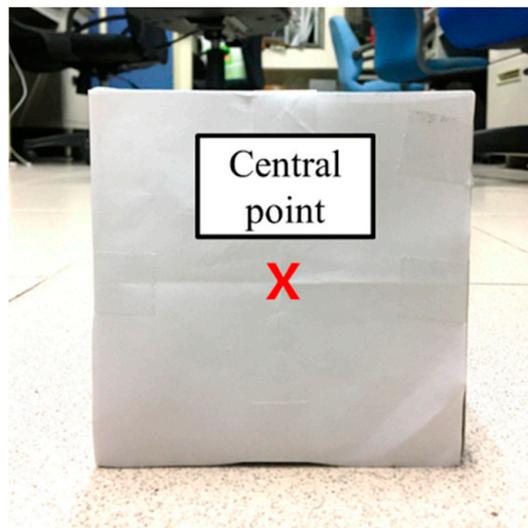


Figure 21. Target of Condition 1, target is located at the center (cross mark).

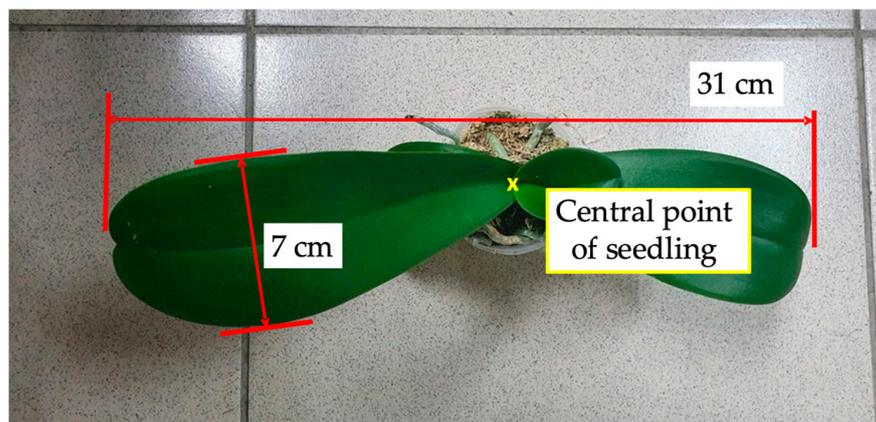


Figure 22. Target of Condition 2.

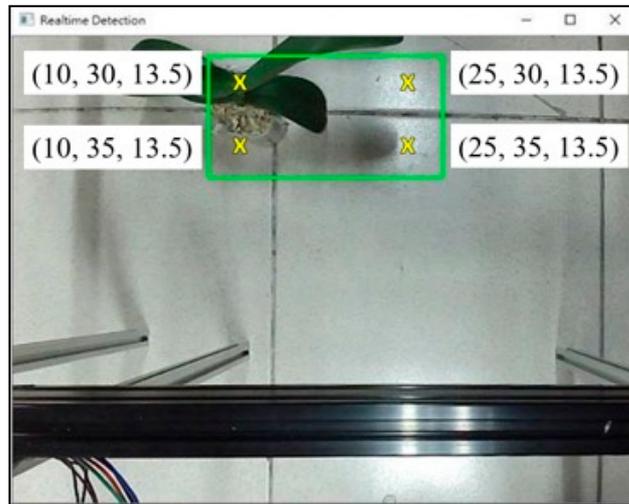


Figure 23. The schematic view of Condition 2, there are 4 measurement points in triggering area.

In the first experiment, Condition 1 provided the measuring method, and the results of measurement errors were recorded in Figure 24, corresponding to the Z-axis distance ranging from 20 to 80 cm. When the Z-axis distance ranges from 20 to 35 cm, the error is obviously large. Until the Z-axis reaches farther than 80 cm, the error starts to rise again, but negatively. Figure 25 clearly indicates that the best working distance of this binocular camera is from 35 to 75 cm, within which the absolute values of average, maximum, and minimum errors are 0.22 cm, 0.42 cm, and 0.02 cm, respectively.

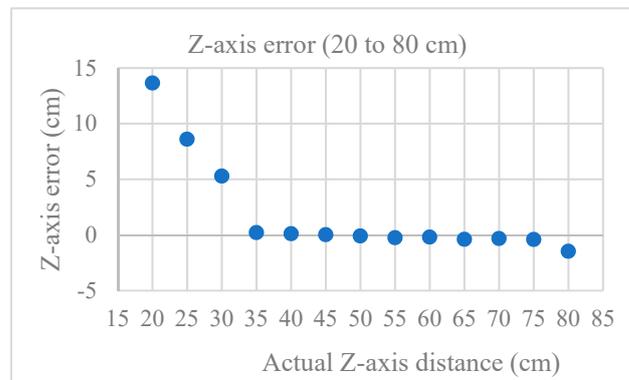


Figure 24. Z-axis error vs. Z-axis distance diagram (20 to 80 cm).

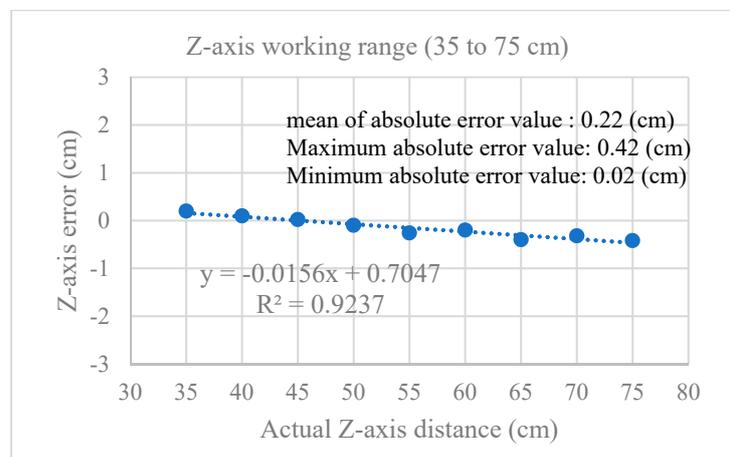


Figure 25. Z-axis error in working range vs. Z-axis distance relationship diagram (35 to 75 cm).

Additionally, Condition 2 was used for the second experiment. The measurement results were addressed in Table 1. The maximum absolute errors of the X-, Y-, and Z-axes of the four measurement points are 0.18 cm, 0.2 cm, and 0.32 cm, respectively. On the other hand, the minimum absolute errors of the X-, Y-, and Z-axes of the four measurement points are 0.1 cm, 0.12 cm, and 0.18 cm, respectively. It is shown that the resulted four measurements in Figure 23 deviate from the accurate positions with mm-scale errors, which is relatively little in comparison with the nozzle size. Therefore, it can be guaranteed that the nozzle still waters on the correct position, despite the presence of the measurement errors presented in Table 1.

Table 1. Measurement results of seedling center.

Measurements	Point 1	Point 2	Point 3	Point 4
X-axis average measurement (cm)	10.16	10.18	25.1	25.16
Y-axis average measurement (cm)	30.2	35.2	30.16	35.12
Z-axis average measurement (cm)	13.22	13.32	13.2	13.18
X-/Y-/Z-Axes absolute error (cm)	0.16/0.2/0.22	0.18/0.2/0.32	0.1/0.16/0.2	0.16/0.12/0.18

3.3. Water Flow Control Experiment of Solenoid Valve

This research also recorded experimental results in Figures 26 and 27, to relate the watering amount (g) with the duty time for valve opening (s). It was measured by the way that the duty time for valve opening started from 3 s and increased sequentially with the one-second interval. The experimental results are linear in the range between 4 and 11 s; hence, the relationship between the watering amount and the duty time for valve opening, formalized in Equation (3), is adopted to control the amount of watering.

$$y = 0.2313x + 1.57. \quad (3)$$

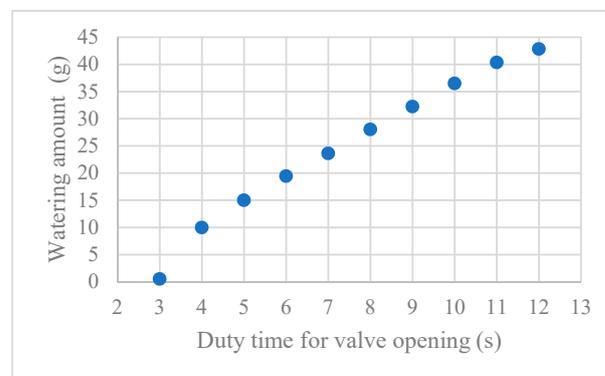


Figure 26. Relationship of duty time for valve opening (3 to 12 s) and watering amount.

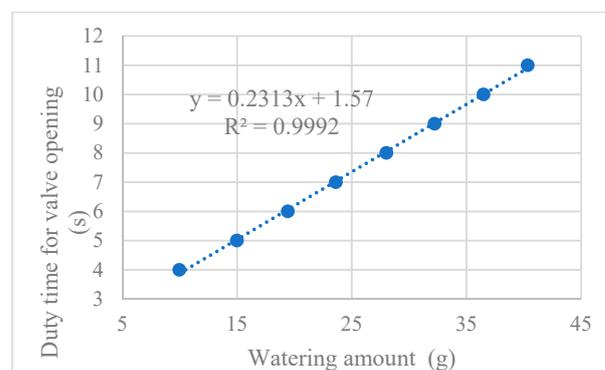


Figure 27. Relationship of duty time for valve opening (4 to 11 s) and watering amount.

3.4. Experiments of Automatic Irrigation System

Two respective experiments for the single and multiple seedlings were considered in the experimental configuration of the automatic irrigation system. In the configuration for the single-seedling experiments, as shown in Figure 28, the triggering area in the identification screen was split into five sections, named as Upper left, Lower left, Central, Upper right, and Lower right, as denoted in Figure 29.

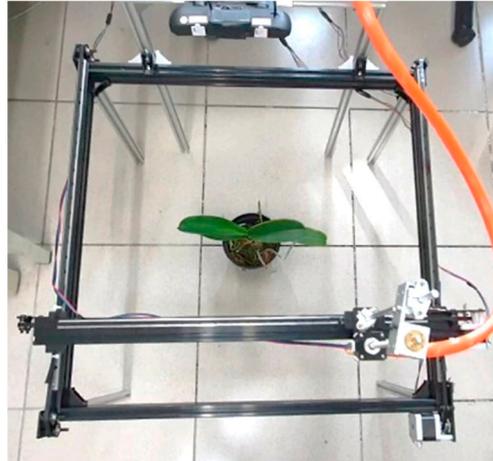


Figure 28. The configuration for the single-seedling experiments.

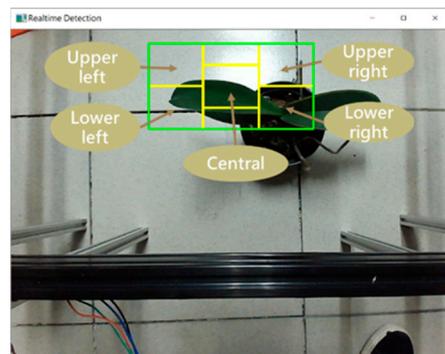


Figure 29. Five sections in the triggering area of the camera.

In order to numerically evaluate the experimental results, three types of indexes are defined. The first type is the number of correct watering to the stem-root junction, as depicted in Figure 30. The second type censuses the achievements of watering to the junction between stems and leaves, as highlighted in Figure 31. The third type stands for counting the incidents of improper watering to the leaf and colliding of nozzle with the leaf, as described in Figure 32. Two kinds of the watering effectiveness are formulated by Equations (4) and (5).

$$\text{Success watering rate (\%)} = \frac{(\# \text{ 1st type} + \# \text{ 2nd type})}{(\# \text{ 1st type} + \# \text{ 2nd type} + \# \text{ 3rd type})} \quad (4)$$

$$\text{Perfect watering rate (\%)} = \frac{\# \text{ 1st type}}{(\# \text{ 1st type} + \# \text{ 2nd type})} \quad (5)$$

where # 1st type, # 2nd type, and # 3rd type are the resultant counts of watering with the first, second, and third types, respectively.

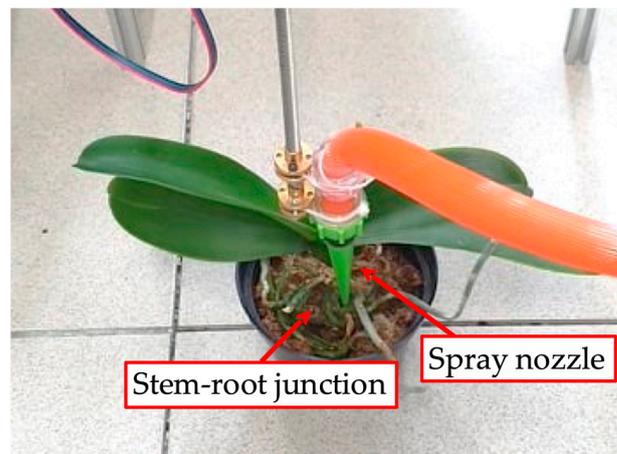


Figure 30. Demonstration of watering on stem-root junction.



Figure 31. Demonstration of watering on the junction between the stem and leaves.



Figure 32. Demonstration of watering on the wrong position (collision of sprinkler with leaves).

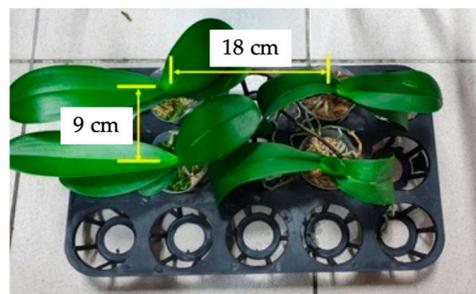
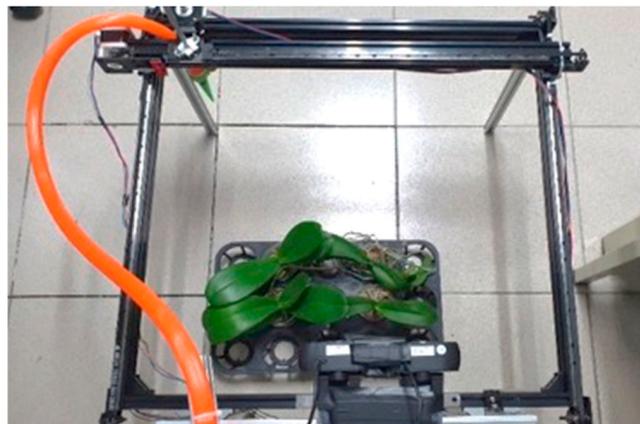
In this experiment, the distance from the optical center of the left lens of the camera to the ground is 64.1 cm. The potted plants were placed in five sections, as shown in Figure 29, and watering will be repeated with ten times. Finally, Table 2 addresses that the total successful and perfect watering rates are 82% and 70.7%, respectively. Whether the watering position is proper could be impacted by the obtainment of an incorrect leaf vector, due to surplus dilation and erosion on the framed orchid image. However, among the records in Table 2, it can be found that there is a low occurrence ratio for the improper watering, which shows that the system can be feasibly applied to practical customized watering onto orchids.

Table 2. Experimental results of automatic irrigation experiments with the same plant but different position of seedling.

Indexes for Single-Seedling	Lower Left	Lower Right	Central	Upper Right	Upper Left	Total
# of watering to stem-root junction	6	7	5	5	6	29
# of watering to stem-leaf junction	2	3	3	2	2	12
# of improper watering position	2	0	2	3	2	9
Successful watering (%)	80	100	80	70	80	82
Perfect watering (%)	75	70	62.5	71.4	75	70.7

For the multiple-seedling experiments, four pots of seedlings were placed in a rectangular manner in the pot holder, as displayed in Figure 33, where the seedlings were spaced by 9 cm and 18 cm, respectively, for the vertical and horizontal directions. Figure 34 demonstrates the experimental configuration for placing the four pots of seedlings in the triggering area. In addition to the three types of indexes to leverage the watering results of the single-pot experiments, the fourth-type index is also provided to accumulate the failures of watering due to the surplus Z-axis distance error, given by Equation (6).

$$\text{Watering operation rate (\%)} = \frac{(\# \text{ 1st type} + \# \text{ 2nd type} + \# \text{ 3rd type})}{(\# \text{ 1st type} + \# \text{ 2nd type} + \# \text{ 3rd type} + \# \text{ 4th type})} \quad (6)$$

**Figure 33.** The pot holder for the multiple-seedling experiments.**Figure 34.** The configuration of the multiple-seedling experiments.

During the multiple-seedling experiments, the distance between the ground and the optical center of the left lens of the camera is 64.1 cm. The experiment was repeated by 15 times to collect 60 resultant sets. Finally, the statistic results of watering provide that the rates of operated, successful, and perfect watering are 70%, 83.3%, and 62.9%, respectively. The experimental statistics are demonstrated in Table 3.

Table 3. Multiple-seedling experimental results.

Indexes for Multiple-Seedling	Results
# of watering to stem-root junction	22
# of watering to stem-leaf junction	13
# of improper watering position	7
# of non-operated watering	18
Operated watering (%)	70
Successful watering (%)	83.3
Perfect watering (%)	62.9

From Table 3, it can be found that there exists a ratio of 30% for no occurrence of watering, which could be the following two reasons. Firstly, the measurement errors of the depth map may be caused by insufficient environmental lighting onwards the feature points. Moreover, the watering process might be cancelled due to wrongly computed moving distance to the watering point for the nozzle, which thus moved too far to stay within the working boundary.

4. Conclusions

This work successfully develops an automated orchid irrigation system, which proves an accurate watering process on the stem-root junction of seedling, to reduce the possibility of diseases of Phalaenopsis plants. This research employs YOLOv3 for the object detection, to detect and to frame the centers of orchid seedlings. Then, in accordance with the derived depth map from the SGBM algorithm, the prediction frame of YOLOv3 is used to compute the actual 3D coordinates of the stem-root junction to be watered. Finally, the three-axis motion platform is controlled to move the spraying nozzle to accurately water the orchid seedlings. A list of vital conclusions are made as follows.

1. Under the total number of 950 training photos, the accuracy of the validation and test sets for the learning model of YOLOv3 are 92.63% and 86.23%, respectively, and hence the feasibility of object detection in this work is guaranteed.
2. The experiments show that the error components of the coordinates (X, Y, Z) for measuring the seedling center are (0.18, 0.2, 0.32) cm maximally, and (0.1, 0.12, 0.18) cm minimally.
3. According to the experiments, the slice from 4 to 11 s of the duty time for opening the spraying valve is linearly related to the watering amount for irrigation, which proves to reliably control the accurate watering amount to irrigate the orchid seedlings.
4. In the validation experiments for the automated irrigation, the successful watering rates of the single-pot and multiple-pot orchid seedlings are 82% and 83.3%, respectively. Moreover, the perfect watering rates of the single-pot and multiple-pot orchid seedlings are 70.7% and 62.9%, respectively.

In the future, the prediction accuracy of the model for the object detection can be raised by enlarging the training sets. As well, the mis-locating of leaf vectors could be solved by alternative detection algorithms, in order to further improve the successful rate of watering. In practice, the system can be also employed to similar applications that require automatic object detection, target positioning, motor control, etc.

Author Contributions: Conceptualization, T.-W.C. and R.C.; methodology, T.-W.C.; software, T.-W.C.; validation, T.-W.C.; formal analysis, T.-W.C. and W.-C.W.; investigation, T.-W.C.; writing—original draft preparation, T.-W.C. and W.-C.W.; writing—review and editing, W.-C.W. and R.C.; visualization, T.-W.C. and W.-C.W.; supervision, R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Ministry of Sciences and Technology, Taiwan, under the grant numbers MOST 108-2221-E-007-097 and MOST 109-2221-E-007-041.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data generated or analyzed in this study can be available from the corresponding author upon reasonable request. However, some data are disclosed under the protection of our partnership company.

Acknowledgments: The authors would like to thank Sunpride Company for providing the experimental sites and various supports to make this research practicable.

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

References

1. Choque, M.C.J.; Linares, O.P.E.; Alcorta, S.N.F.; Alva, A.J.L.; Prado, G.S.R. IoT Automated Orchard in a Domestic Environment. In Proceedings of the IEEE 1st Sustainable Cities Latin America Conference, Arequipa, Peru, 26–29 August 2019; pp. 1–6.
2. Chang, Y.-L. Development of an Automatic Spraying System for Phalaenopsis Potted Plants in Greenhouse. Master's Thesis, National Ilan University, Ilan, Taiwan, 2019.
3. Huang, R.; Gu, J.; Sun, X.; Hou, Y.; Uddin, S. A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network. *Electronics* **2019**, *8*, 825. [[CrossRef](#)]
4. Cao, C.-Y.; Zheng, J.-C.; Huang, Y.-Q.; Liu, J.; Yang, C.-F. Investigation of a Promoted You Only Look Once Algorithm and Its Application in Traffic Flow Monitoring. *Appl. Sci.* **2019**, *9*, 3619. [[CrossRef](#)]
5. Tai, S.-K.; Dewi, C.; Chen, R.-C.; Liu, Y.-T.; Jiang, X.; Yu, H. Deep Learning for Traffic Sign Recognition Based on Spatial Pyramid Pooling with Scale Analysis. *Appl. Sci.* **2020**, *10*, 6997. [[CrossRef](#)]
6. Seo, J.; Ahn, H.; Kim, D.; Lee, S.; Chung, Y.; Park, D. EmbeddedPigDet—Fast and Accurate Pig Detection for Embedded Board Implementations. *Appl. Sci.* **2020**, *10*, 2878. [[CrossRef](#)]
7. Chou, Y.-C.; Kuo, C.-J.; Chen, T.-T.; Horng, G.-J.; Pai, M.-Y.; Wu, M.-E.; Lin, Y.-C.; Hung, M.-H.; Su, W.-T.; Chen, Y.-C.; et al. Deep-Learning-Based Defective Bean Inspection with GAN-Structured Automated Labeled Data Augmentation in Coffee Industry. *Appl. Sci.* **2019**, *9*, 4166. [[CrossRef](#)]
8. Lin, C.-L. Visual Recognition System of Automation Orchid Seedling Lesion Based on Deep Learning. Master's Thesis, National Tsing Hua University, Hsinchu, Taiwan, 2019.
9. Vit, A.; Shani, G.; Bar-Hillel, A. Length Phenotyping with Interest Point Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 2609–2618.
10. Yang, J.-Y. Automatic Recognition System of Quality Classification for Orchid Based on Deep Learning. Master's Thesis, National Tsing Hua University, Hsinchu, Taiwan, 2020.
11. Chang, J.S.; Chen, R. Automated Counting System for Orchid Seedlings Based on Deep Learning. In Proceedings of the 17th International Conference on Automation Technology, Hualien, Taiwan, 12–14 November 2020.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
13. Girshick, R. Fast R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
15. 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, 2016; pp. 779–788.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Adi, K.; Widodo, C.E. Distance Measurement with a Stereo Camera. *Int. J. Innov. Res. Adv. Eng.* **2017**, *4*, 24–27.
19. Hirschmuller, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 807–814.
20. Hirschmuller, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 328–341. [[CrossRef](#)] [[PubMed](#)]
21. Hermann, S.; Morales, S.; Klette, R. Half-resolution semi-global stereo matching. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 201–206.
22. Hamzah, R.A.; Rahim, R.A.; Noh, Z.M. Sum of Absolute Differences algorithm in stereo correspondence problem for stereo matching in computer vision application. In Proceedings of the International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 1, pp. 652–657.
23. OpenCV. Available online: <https://opencv.org/releases/> (accessed on 10 July 2020).