

Article

Development and Evaluation of a Watermelon-Harvesting Robot Prototype: Vision System and End-Effector

Jiacheng Rong ¹, Jun Fu ¹, Zhiqin Zhang ¹, Jinliang Yin ¹, Yuzhi Tan ¹, Ting Yuan ^{1,*} and Pengbo Wang ^{2,*}¹ College of Engineering, China Agricultural University, Beijing 100083, China² Jiangsu Provincial Key Laboratory of Advanced Robotics, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215123, China

* Correspondence: yuanting122@cau.edu.cn (T.Y.); pbwang@suda.edu.cn (P.W.)

Abstract: Over the past decade, there have been increasing attempts to integrate robotic harvesting technology into agricultural scenarios to reduce growing labour costs and increase crop yields. In this paper, we demonstrate a prototype harvesting robot for picking watermelons in greenhouses. For robotic harvesting, we design a dedicated end-effector for grasping fruits and shearing pedicels, which mainly consists of a flexible gripper and a cutting device. The improved YOLOv5s-CBAM is employed to locate the watermelon fruits with 89.8% accuracy on the test dataset, while the K-means method is used to further refine the segmentation of the watermelon point cloud in the region of interest. Then, the ellipsoid is fitted with the segmented fruit point cloud to obtain the lowest point of the ellipsoid as the grasping point. A series of tests conducted in a laboratory simulation scenario proved that the overall harvesting success rate was 93.3% with a positioning error of 8.7 mm when the watermelon was unobstructed. The overall harvesting success rate was 85.0% with a positioning error of 14.6 mm when the watermelon was partially obscured by leaves.

Keywords: agricultural robot; watermelon; end-effector; fruit detection; deep learning



Citation: Rong, J.; Fu, J.; Zhang, Z.; Yin, J.; Tan, Y.; Yuan, T.; Wang, P. Development and Evaluation of a Watermelon-Harvesting Robot Prototype: Vision System and End-Effector. *Agronomy* **2022**, *12*, 2836. <https://doi.org/10.3390/agronomy12112836>

Academic Editor: Juntao Xiong

Received: 17 October 2022

Accepted: 10 November 2022

Published: 13 November 2022

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



Copyright: © 2022 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

Watermelon is an important cash crop, occupying an important position in the world's fruit production and consumption. World FAO statistics [1] show that the global area under watermelon cultivation in 2020 was 305 million hectares, and the production was over 100 million tons. China is the leading watermelon producer and consumer, with a production of 60.24 million tons in 2020. With the spread of the smart greenhouse watermelon-growing model in China, total watermelon production and unit yield will continue to rise. However, watermelon harvesting still relies on manual labour and often faces labour shortages during the harvesting period [2]. As the population ages, the number of young workers in agriculture will continue to decrease, and it will become more difficult to hire workers for watermelon production. Therefore, it is necessary to promote a transformation of watermelon harvesting from manual to robotic autonomous operation to cope with the risk of a mismatch between growing watermelon production and decreasing agricultural labour. Compared to industrial automation scenarios, agricultural operation scenarios are more complex. The robot not only has to face environmental problems such as background interference and occlusion [3,4], but the fruit-picking action is also distinctive, so the development of the vision system and the picking tool for a harvesting robot is a key factor in determining the potential of the robot [5].

To achieve the goal of automated crop harvesting, harvesting robotics has been actively developed over the past 40 years since the concept of harvesting robots was first introduced by Schertz and Brown [6]. Many scholars have conducted a large amount of exploration and research around the tasks of crop identification [7,8], end-effector design [9], obstacle localization [10], and motion planning [11] in the automated harvesting process, involving mainly mushrooms, apples, strawberries, and tomatoes.

Mushrooms are the most likely crop to be harvested automatically, as the growth environment will be relatively standardized due to the promotion of the factory farming model. As early as 2001, Reed et al. [12,13] developed an experimental platform for an *Agaricus bisporus* picking robot, including a vision system, a robotic arm, an end-effector, and a transfer mechanism. When the camera is positioned over the target mushroom area, the vision system acquires the image and locates the mushroom. The end-effector of the picking robot performs the grasping operation and places the grasped mushrooms onto the conveyor. Finally, the trimming device cuts the mushroom stalk and places it in the storage box. In the actual test, the picking robot attempted to grasp 2975 times and successfully picked 2427 mushrooms, with a harvesting success rate of 81.6% and an average of 9 mushrooms per minute. To improve harvesting efficiency, Yang et al. [14,15] developed a mushroom-harvesting robot consisting of a mobile platform, a harvesting unit, and a control module. During the harvesting operation, four harvesting units worked simultaneously to effectively improve the harvesting efficiency, and the average harvesting time for a single mushroom was only 8.85 s, with a harvesting success rate of 86.8%.

As a widely grown fruit, apples are also one of the main research targets for autonomous crop harvesting. Baeten et al. [16] developed an apple-harvesting robot in which the entire robot platform is towed by an agricultural tractor and a six-degrees-of-freedom industrial robot arm is mounted on a hydraulic platform to pick the target apples in view. The grasping function is mainly realized by a flexible suction cup driven by pneumatic pressure; a camera is also installed inside the suction cup, which simplifies the robot coordinate system conversion process and improves the positioning accuracy regarding the apple. However, for obscured apples, the vision system is not only unable to accurately identify the target location, but also the end-effector is easily affected by obstructions such as branches and trunks, resulting in harvesting failure. To circumvent obstacles that the robotic arm may touch during its motion, Kang et al. [17] developed robot vision perception and a three-dimensional reconstruction algorithm for autonomous apple harvesting. First, a deep-learning model that includes both detection and segmentation is presented for the recognition of apples in RGB images. Then, the obstacles in the working environment are modelled using the octrees method [18]. Finally, the fruits are modelled to locate the fruit centroids, and the grasping pose of each fruit is calculated based on the Hough transform. The harvesting success rate for overlapping or partially overlapping apples was increased from 62% to 81% using this vision algorithm for grasping. In addition, Kang et al. [19] used a deep learning-based PointNet network to estimate the apple grasping pose instead of the traditional algorithm, which can further improve the accuracy and stability of the grasping pose estimation.

Numerous studies have also been carried out for other crops. Adrian et al. [20] developed an asparagus-harvesting robot that is capable of travelling along a dam at a speed of 0.2 m/s, using a vision system based on 3D point cloud clustering to detect the location and size of asparagus stems, and an end-effector designed to clamp the entire asparagus and cut the roots, with an overall harvesting success rate of 90%. Williams et al. [21] designed a kiwifruit-harvesting robot equipped with four robotic arms and proposed a dynamic scheduling system capable of driving multiple robotic arms to work in concert, resulting in a more efficient picking operation with an overall harvesting success rate of 51% and an average harvesting cycle time of 5.5 s. In summary, there have been many studies focusing on the autonomous harvesting of mushrooms, strawberries, asparagus, kiwis, and many other crops [22–24], but until now few scholars have conducted studies on the automated harvesting of watermelons [25,26]. Especially for end-effector design, most research is now focused on the development of end-effectors for small and light crops [27–30], with less currently being conducted for the watermelon. This study aims to develop a vision system for watermelon grasping point positioning and a dedicated end-effector based on small watermelons. After reviewing the progress of harvesting robot research, we introduce the operation environment and the fruit parameters. We also show our watermelon-harvesting robot prototype and develop the watermelon identification algorithm and the grasping

point positioning method in Section 2. In Section 3, we perform algorithm validation and testing of the prototype in a laboratory simulation scenario. Finally, we conclude this paper in Section 4.

2. Materials and Methods

2.1. Greenhouse Environment and Fruit Parameters

In this paper, the subject of study is the watermelon grown in the greenhouse in an elevated soilless culture. A series of parameters were collected in advance for the greenhouse scenario and the watermelon fruit body for better design and development of the watermelon-harvesting robot. As shown in Figure 1, the target watermelons are planted on both sides of the corridor with a distance of 1.5 m between two rows of plants. Horticultural workers control the height of watermelon growth, ranging from 1.5 m to 2.5 m from the ground. For picking watermelons that grow too high, workers pick them aboard lifting platforms that travel on the standard greenhouse orbits.

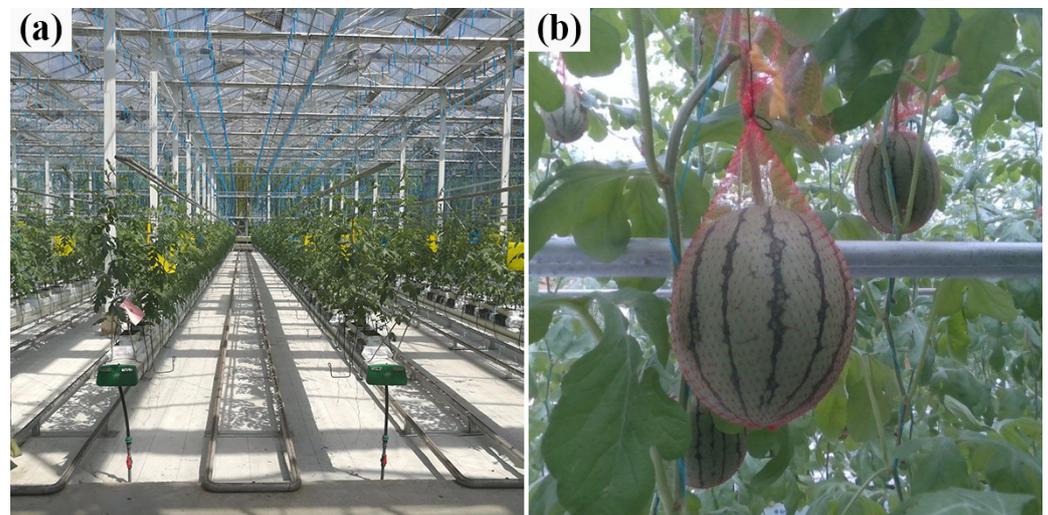


Figure 1. A modern greenhouse for growing watermelons: (a) the production greenhouse scenario; (b) elevated soilless watermelons.

Before designing the robot's end-effector, each physical parameter of the grasped object (the watermelon) needed to be fully investigated. We randomly sampled L600 watermelons grown on an agricultural site in Daxing District, Beijing, and 8424 watermelons grown on an agricultural site in Kunshan City, Jiangsu Province. First, the pedicel diameter, pedicel length, fruit width, and fruit length of each watermelon were measured using a vernier calliper. Then, we weighed each watermelon with an electronic scale.

We selected 30 watermelons of the L600 and 8424 varieties for parameter collection, as shown in Table 1. The average parameters of the L600 watermelon obtained from the measurements are as follows: a pedicel diameter of 5.5 mm, a pedicel length of 98 mm, a fruit width of 150 mm, a fruit length of 186 mm, and a fruit weight of 1.89 kg. Also, the measured average parameters of 8424 watermelons are close to those of L600 watermelons. Calculation of the watermelon fruit aspect ratio shows that the 8424 watermelon is closer to a sphere, while the L600 watermelon is an ellipsoid. Thus, we need to design a hand gripper that can hold different shapes of watermelon. Furthermore, the fruit pedicels of both watermelons are close in length, which provides a dimensional basis for the design of our cutting device.

2.2. System Design and Operation

The autonomous harvesting robot was developed to harvest watermelons in the greenhouse. As shown in Figure 2, the watermelon-harvesting robot we developed comprises an

industrial computer for overall control, a robotic mobile platform capable of moving on the orbits, an RGB-D camera providing sensing and positioning functions, a robotic arm used to perform picking tasks, and an end-effector able to hold the watermelon fruit and shear the pedicel. The industrial computer needs not only to control the rest of the harvesting robot but also to be able to process and analyze images in real time, so it is equipped with a high-performance graphics processing unit (Nvidia GTX-1060). To automate its movement in the greenhouse, the selected mobile platform can move on smooth ground and the specified orbit, which is enabled with radio frequency identification (RFID) and navigation line identification for autonomous navigation and automatic orbit transformation. The RGB-D camera is an Intel RealSense D435i, which is capable of obtaining colour images and aligned-depth images. The robotic arm (AUBO i5) has six degrees of freedom (DOF) and can be used for motion planning or position and pose control via the API interface or ROS. In addition, we developed a special end-effector for watermelon picking, which communicates with the host computer through an Arduino microcontroller.

Table 1. Physical parameters of L600 and 8424 watermelons.

Fruit Variety	Pedicel Diameter (mm)	Pedicel Length (mm)	Fruit Width (mm)	Fruit Length (mm)	Fruit Weight (kg)
L600	5.5	98	150	186	1.89
8424	5.6	99	153	167	1.97

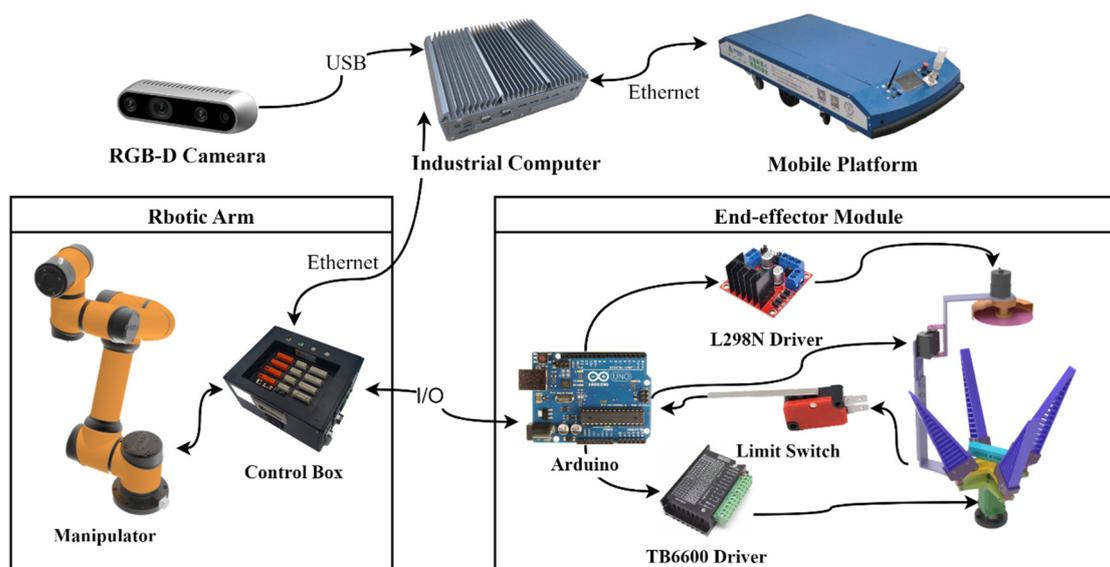


Figure 2. Schematic of the proposed watermelon-harvesting robot.

Based on the constructed harvesting robot platform, we designed the system operation flow of the watermelon-harvesting robot. The entire operation process of the harvesting robot is divided into three modules, corresponding to three working phases, as shown in Figure 3. In the first phase (the inspection phase), the watermelon-harvesting robot travels in the middle of the corridor to identify watermelon targets in the images captured by the camera. After detecting a pickable watermelon within the working area, the mobile platform stops moving forward. In this case, the watermelon-harvesting robot enters the second working phase.

In the positioning phase, the watermelon-harvesting robot needs to calculate the grasping point of the end-effector for harvesting the pickable watermelon fruit. Firstly, because of the signal delay and inertial motion between the issuance of the stop and the actual stop, the camera needs to reacquire the colour image and point-cloud information of the current position. After that, the watermelon target in the image is re-identified using

the object detector. Then, the watermelon grasping point localization method is used to further determine the grasping point of the end-effector. In the last phase (the execution phase), the robot arm performs picking operations on each pickable watermelon. Because the Arduino communicates with the robot arm through the IO port, the host computer controls the IO port of the robot arm to indirectly drive the end-effector to execute the grasping and shearing action. Finally, after the robotic arm performs the current picking task, the watermelon-harvesting robot re-enters the inspection phase to continue its work.

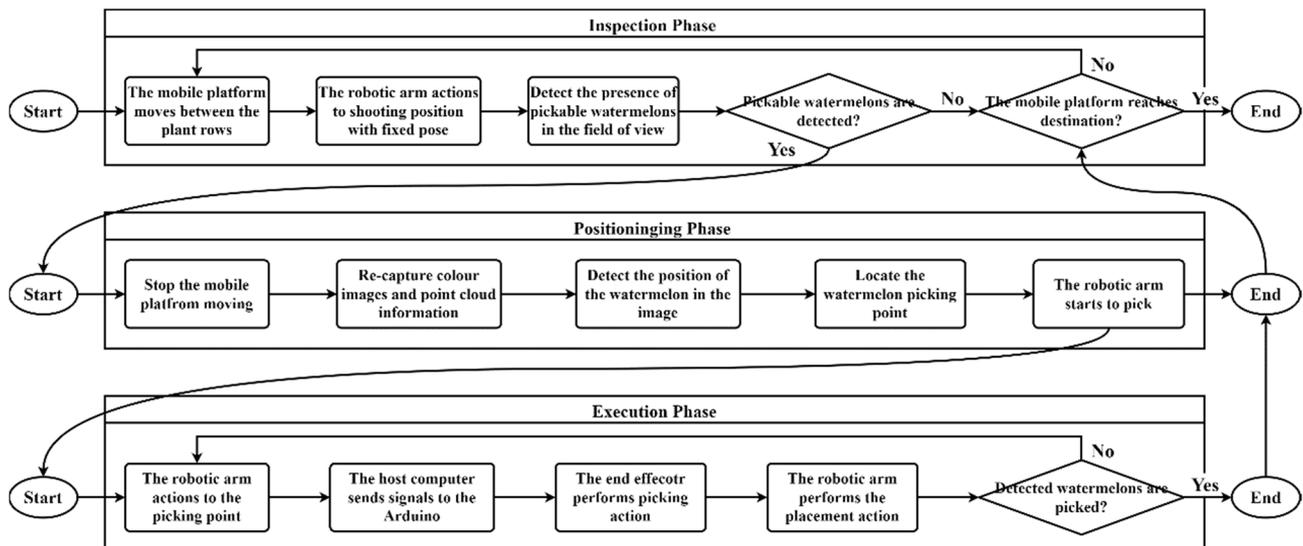


Figure 3. Workflow diagram of the watermelon-harvesting robot.

2.3. Design of the Watermelon Picking End-Effector

In the whole harvesting robot system, the vision system and the end-effector are complementary and directly affect the harvesting success of the robot. Because the shapes, dimensions, and growing environments of different varieties of crop fruits vary tremendously in agricultural scenarios, each end-effector designed for a specific fruit is not universal [28,31]. Therefore, as the only component that comes into physical contact with the fruit target, the design of the end-effector is very important.

Based on the investigated fruit parameters of L600 and 8424 watermelons, we designed the end-effector for watermelon harvesting using the SolidWorks software (Dassault Systems SolidWorks Corporation, Concord, MA, USA). After several version iterations, the design parameters of each component were optimized, and the final structure was simulated and fabricated, as shown in Figure 4. The custom end-effector is mainly composed of four flexible fingers, a Stepper Motor (42HB62F2SG-03, BOHONG, Shijiazhuang, China), a connecting slider, a connecting bracket, a flange, an RGB-D camera (RealSense D435i, Intel, Santa Clara, CA, USA), a cutting blade, a Servo (RDS3115, Dsservo, Dongguan, China), and a DC motor (XD-WS37GB3525, XIN DA, Shenzhen, China) (see Figure 4a).

The dimensions of the flexible fingers were designed based on the length, width, and weight of the watermelon fruit. To make the designed fingers fit the surface of the watermelon, the final optimized finger has a dimension of 190 mm in height, 20 mm in thickness, and 55 mm in maximum width, while the flexible fingers are made of thermoplastic polyurethanes (TPU) material to ensure the stability of clamping and a low fruit damage rate, as shown in Figure 4b. The installation position of the cutting blade is determined based on a combination of the fruit length, the pedicel length, and the opening angle of the four flexible fingers. After several actual tests, we determined that the vertical distance of the cutting blade from the top of the watermelon is 50–70 mm when the cutting blade is 290 mm away from the connecting slider.

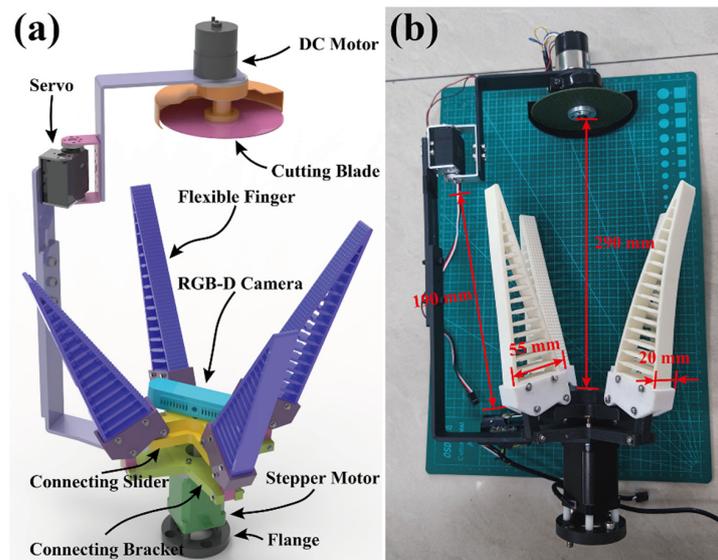


Figure 4. Structure diagram of the developed end-effector for autonomous harvesting of watermelon: (a) the three-dimensional structure schematic; (b) the end-effector prototype.

As shown in Figure 5, the end-effector performs the actual operational action on the watermelon. Firstly, the rotary motion of the stepper motor is converted into linear motion by using the screw–nut sub, which enables the connecting slider to move up and down. The flexible finger is linked to the connecting slider by a hinge, thus providing the opening and closing action of the four fingers (see Figure 5a,b). After the flexible fingers clamp the watermelon, the end-effector drags the fruit under the drive of the robotic arm to prevent the cutting blade from shearing the main stem. Then, the cutting blade starts to rotate at a high speed, driven by a DC motor. Finally, the cutting device is driven by a servo motor and swings to the shearing point to cut the pedicel, as can be seen in Figure 5d.

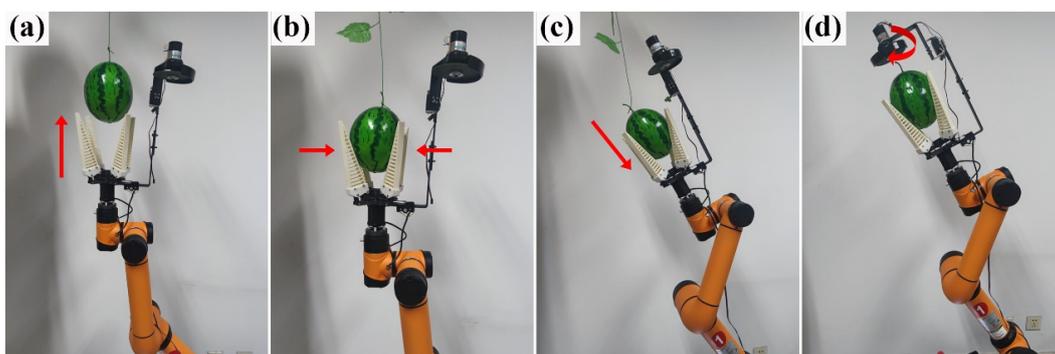


Figure 5. Operating process of the end-effector: (a) the end-effector reaches the pre-grasp position; (b) flexible fingers hold the watermelon tightly; (c) the end-effector drags the watermelon; (d) the cutting device trims the watermelon pedicel.

2.4. Watermelon Detection and Localization

2.4.1. Construction of the Object Detector

In the inspection phase, the watermelon-harvesting robot needs to move in the corridor between two rows of crops while the RGB-D camera acquires images and uploads them to the host computer for processing, with the whole image processing speed being higher than 25 FPS. However, in agricultural scenarios, the hardware conditions of the harvesting robot are limited by power consumption, robot volume, and heat dissipation, and the performance of the industrial computer does not meet the requirements for fast image processing. Large image processing delays can lead to longer stopping distances or missed

pickable fruits for the mobile platform. The abovementioned problem can be solved by reducing the robot's movement speed, but this greatly affects the overall efficiency of the robot. Therefore, it is necessary for the vision system to be able to recognize a single image in less than 40 ms.

To perform image processing using traditional machine learning methods, the researchers need to first analyze the image information and then manually select the appropriate features [32–34]. Compared with traditional manual feature extraction methods, the image processing method based on deep learning has better recognition capability and adaptability and has been widely used in agricultural scenarios in recent years [35–38]. As one of the latest masterpieces of single-stage object-detection algorithms, YOLOv5 (<https://github.com/ultralytics/yolov5>, accessed on 12 October 2021) has a range of models suitable for scenarios such as industry and agriculture, including N/S/M/L/X. Its architecture controls the network depths and number of intermediate layer channels by setting different depth factors and width factors. Considering its excellent detection performance and code readability, we chose YOLOv5 as the benchmark for code porting and improvement.

As shown in Figure 6, YOLOv5 consists of three primary components: the backbone network, the neck network, and the detection head. Its basic structure includes the Focus module, the CBS module, the SPP module, and the C3 module; the Focus module is used to subsample the input image without information loss to achieve faster computation, and the SPP module serves to address the problem of inconsistent input image size through multiple receptive field fusion. The backbone network is composed mainly of the CBS module and the C3 module stacked in series to extract feature information from images. The neck network is designed using a strategy of fusing a feature pyramid network (FPN) and pixel aggregation network (PAN) to increase feature reuse and better exploit the extracted features. Finally, the detection head performs a prediction with these extracted image features and outputs the location and category information of the object.

YOLOv5s has a simple structure and fast inference, which can meet the requirements of real-time detection. However, the detection accuracy of YOLOv5s decreases when the watermelon is obscured by leaves. To solve the problem of insufficient accuracy due to the complex background of the field environment, this paper proposes an improved YOLOv5s by adding the CBAM attention mechanism module [39] at key positions of the backbone network, as shown in Figure 7. In the backbone network, the output results of the last three C3 modules are fed into the neck network, and we replace the original C3 module with the CBAM-C3 module. As shown in Figure 7, after performing multiple convolutions, the CBAM attention mechanism module is added to weigh the feature maps in the channel and spatial dimensions. In addition, the computational cost associated with CBAM can be ignored, given that it is a lightweight module.

2.4.2. Watermelon Grasping Point Localization Method

After YOLOv5 outputs the 2D coordinates of the watermelon in the image coordinate system, the watermelon-harvesting robot is still unable to directly localize the grasping point. Combined with the design of the end-effector, the harvesting robot needs to adopt the lowest point of the watermelon as the grasping point. If the grasping point is confirmed directly in the 2D image, the point-cloud information of this grasping point is largely unreliable because the grasping point is at the location of the depth's abrupt change. Therefore, we designed a simple method to fit the watermelon point of the fitted ellipsoid as the picking point, as shown in Figure 8. Firstly, the point-cloud information is cropped to obtain the corresponding point cloud inside the rectangular box (see Figure 8a,b). Then, outlier points in the z-axis direction (as shown in Figure 8c) are filtered out using a guide filter. In addition, the watermelon point cloud in the foreground is then clustered using the K-means algorithm to reduce the effect of leaf occlusion (see Figure 8d). Finally, the ellipsoid is fitted to the watermelon point cloud to obtain the lowest point of the ellipsoid, as shown in Figure 8e.

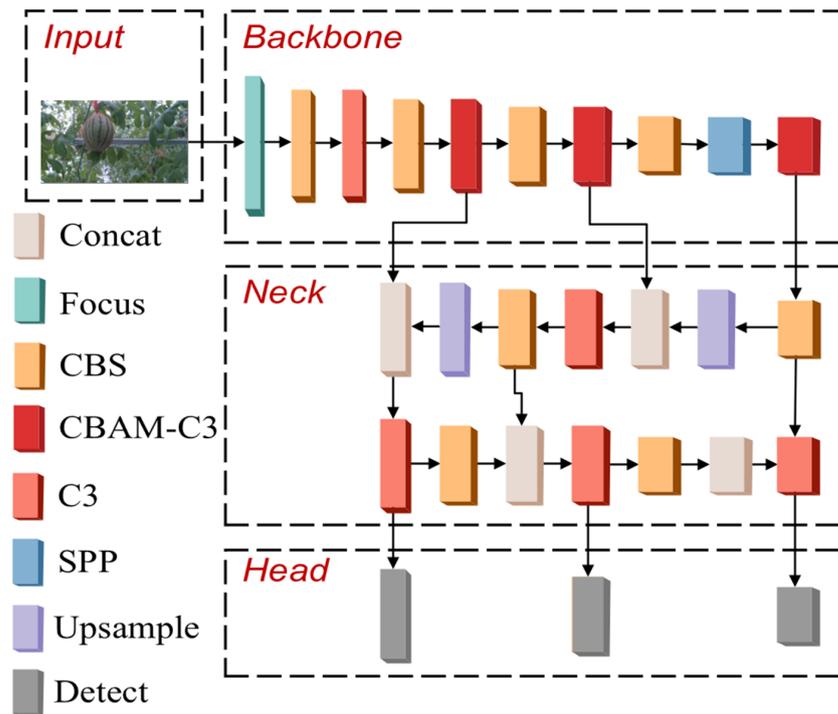


Figure 6. The structure of the improved YOLOv5 detector.

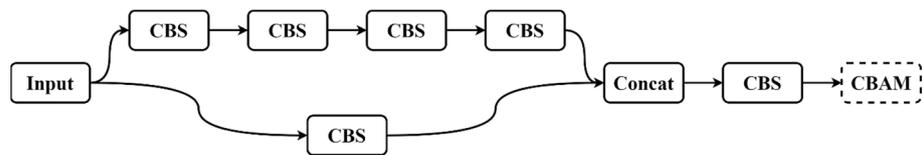


Figure 7. The improvement of the C3 module by adding the CBAM attention mechanism.

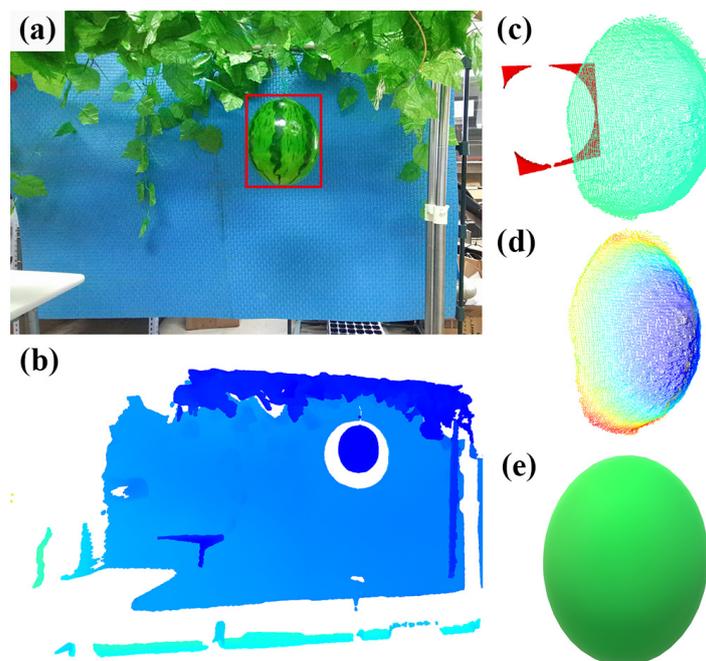


Figure 8. Grasping point positioning method: (a) the colour image; (b) the point-cloud information; (c) the point cloud in the rectangular box; (d) the watermelon point cloud obtained after performing clustering; (e) the fitted ellipsoid.

3. Experiments and Results

3.1. Hand-Eye Calibration

After obtaining the spatial position and pose of the target in the camera coordinate system, the position-based vision servo system needs to establish the spatial relationship between the camera and the robotic arm. Then, the pose of the target relative to the base coordinate system of the robotic arm can be deduced after completing hand-eye matrix calibration. As shown in Figure 9, the watermelon-harvesting robot developed in this paper uses the Tsai–Lenz hand-eye calibration method for hand-eye matrix solving [40]. Because the RGB-D camera is mounted on the end joint of the robotic arm, the relative positions of the robotic arm base and the calibration board are sure to be constant, and then the robotic arm is moved to shoot the calibration board. As shown in Figure 10, the robotic arm is controlled to shoot the calibration board several times at different viewpoints. We develop a hand-eye calibration script (<https://github.com/JCRONG96/-Hand-Eye-with-opencv>, accessed on 29 September 2022) based on the OpenCV library in this paper. First, the inner corner points of the chessboard are directly localized using the `findChessboardCorners` function. Then, the PnP method is used to solve the external parameter matrix of the camera, and the internal parameters of the camera (focal length, optical centre, scale factor, and aberration factor) are obtained by the `calibrateCamera` function. Finally, the transformation matrix from the end to the base coordinate system and the external parameters of the calibrated camera are passed to the `calibrateHandEye` function as input parameters to obtain the hand-eye matrix.

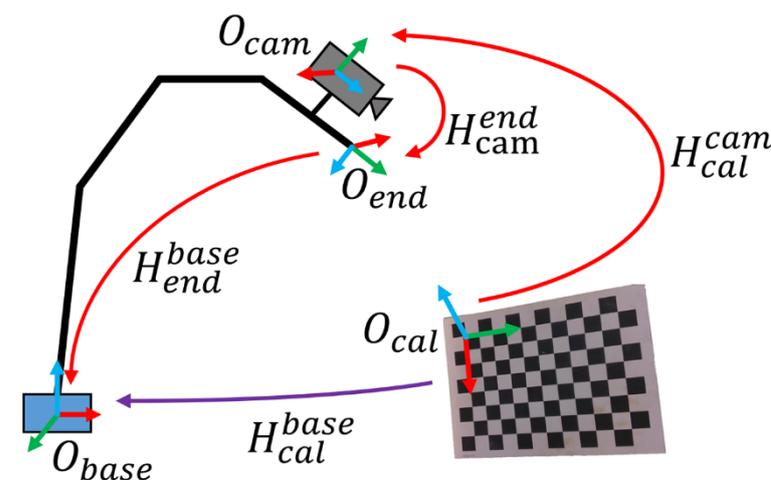


Figure 9. Schematic diagram of hand-eye calibration solution.

Because the accuracy of the hand-eye calibration is one of the keys to determining the precise positioning of the robot vision servo system, it is required to measure the positioning error of the calibrated vision servo system. In this paper, we tested the arrival spatial position of the end tool and the actual spatial position of the target point (the position reached by manually dragging the robotic arm) and calculated the Euclidean distance between the two positions as the positioning error, where the positioning accuracy of the robotic arm itself is ± 0.02 mm. In the test experiment, we first printed a chessboard on paper. The chessboard was then photographed by a camera mounted on a robotic arm, and 40 of the inner corner points were found, after which the end tool of the robotic arm touched these inner corner points in turn. The position touched by the robotic arm is the position converted by the hand-eye matrix, which deviates from the real corner point position. Therefore, we also need to manually move the end tool to the corner point position to record the current spatial position. We performed five hand-eye calibrations and conducted accuracy tests, and the accuracy test results are shown in Figure 11. To perform hand-eye calibration, images and corresponding robotic arm positions were acquired from 20 different viewpoints for hand-eye calibration each time. For precision measurement,

we controlled the distance between the camera and the target point to be about 500 mm. The average positioning errors from the five hand-eye calibrations (Serial A, B, C, D and E in Figure 11) were 3.88 mm, 3.84 mm, 3.94 mm, 4.12 mm, and 3.67 mm. The error of the vision servo system of the developed watermelon-harvesting robot can be controlled within 4.41 mm when the vertical distance from the target is less than 500 mm.



Figure 10. Calibration board images taken from different perspectives.

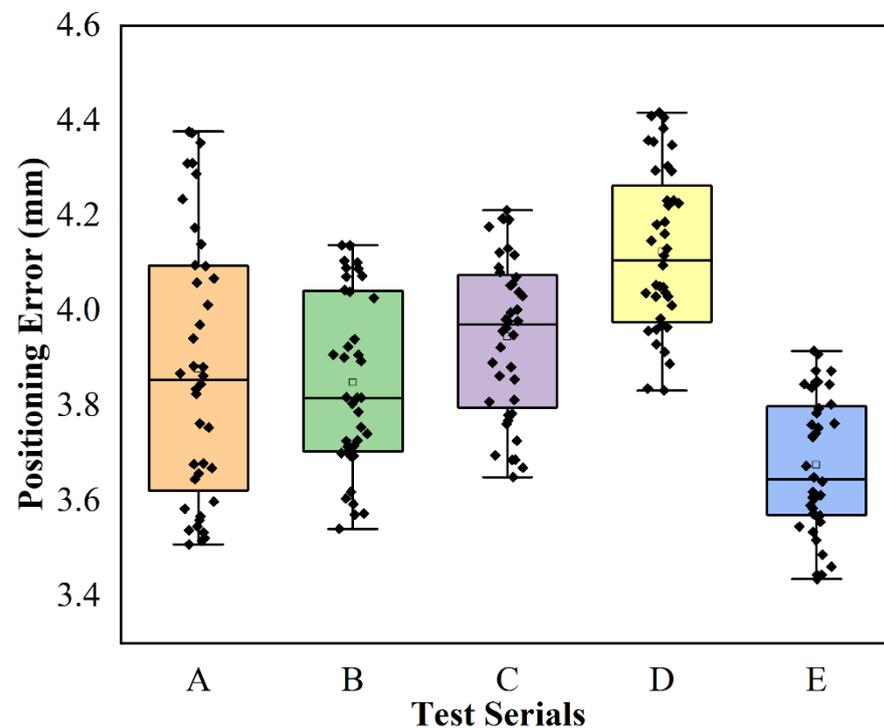


Figure 11. Accuracy test results of the hand-eye calibration.

3.2. Evaluation of the Detection and Localization Method

3.2.1. Image Acquisition and Dataset Creation

The watermelon dataset was collected from May 2020 to July 2020 on an agricultural site in Daxing District, Beijing, and an agricultural site in Kunshan City, Jiangsu Province. The camera used for image acquisition was the RealSense D435i from Intel, and the camera resolution was set to 1280×720 . Watermelon images were acquired under different

weather conditions to make the vision algorithm have a strong generalization capability in unstructured environments, and a total of 1026 images were acquired. As this watermelon-harvesting robot has no auxiliary lighting added, no image acquisition at night was carried out. Finally, we manually processed all the images using the annotation software ‘Labelme’, marking all the foreground watermelons with a rectangular box (see Figure 12).



Figure 12. RealSense camera image example: (a) the RGB image; (b) the RGB image overlaid with one class ground truth.

3.2.2. Model Training and Testing

As the watermelon-harvesting robot moves through a corridor, the vision system needs to detect pickable watermelons in the image vision field in real time, and the accuracy and speed of the object-detection algorithms are critical in the practical application of the harvesting robot. Therefore, several mainstream object-detection algorithms (YOLOv5s, our improved YOLOv5s-CBAM, YOLOv4 [41], and SSD [42]) were compared using the watermelon dataset to find an object detector with the best balance of speed and accuracy.

The device used in this experiment is a laptop computer with an Intel Core i7-10870H CPU and an NVIDIA GeForce RTX 2070 GPU. All object-detection algorithms were trained for 1500 epochs. The labelled 1026 images were divided into the training set, the validation set, and the test set at a ratio of 7:2:1. The initial learning rate and minimum learning rate of the optimizer were set to 0.01 and 0.002, respectively.

In this experiment, four principal metrics (speed, precision, recall, and mAP) were utilized to evaluate these object-detection algorithms. The speed indicates the time consumed to infer an image. The precision represents the ratio of the correct detection results to the total detection results, and recall is defined as the proportion of the correct detection results in all ground truth results, as shown in Equations (1) and (2). Furthermore, the mean average precision (*mAP*) was introduced to better reflect the overall detection performance of the models. The *AP* in Equation (3) is defined as the area under the precision–recall curve, and the *mAP* in Equation (4) is the average value of each category (*c*) of *AP*.

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$recall = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \int_0^1 p \cdot r \cdot dr \quad (3)$$

$$mAP = \frac{1}{c} \sum_{i=1}^c AP_i \quad (4)$$

After the models were trained, the performance of each model was evaluated on the watermelon test dataset, and the speed, precision, recall, and mAP were calculated. As shown in Table 2, compared to YOLOv4 and SSD, YOLOv5s achieved a substantial advance in speed, inferring an image in less than 9 ms. In terms of recall, YOLOv4 performed the best, surpassing the other three methods. The accuracy of the improved YOLOv5s-CBAM

was 89.8%, which was 2.4%, 0.6%, and 3.0% higher than YOLOv5s, YOLOv4, and SSD, respectively. As shown in Figure 13, all the object-detection algorithms could accurately detect all foreground watermelons. However, YOLOv5s, YOLOv4, and SSD misidentified the background watermelons to different degrees, which reduced the detection accuracy of the model. Finally, YOLOv5s–CBAM also achieved the highest score on mAP, a metric that takes into account both accuracy and recall. It is experimentally demonstrated that the CBAM attention mechanism can effectively improve the detection accuracy of the model on the watermelon dataset with a small increase in model computation. Therefore, we use the improved YOLOv5s–CBAM for watermelon identification and localization.

Table 2. Performance of detectors on the watermelon test dataset.

Model	Size (Pixels)	Speed (ms)	Recall	Precision	mAP
YOLOv5s	640	8.6	94.6	87.4	94.1
YOLOv5s–CBAM	640	8.9	94.9	89.8	96.3
YOLOv4	416	40.9	95.1	89.2	95.6
SSD	512	27.5	93.0	86.8	92.2

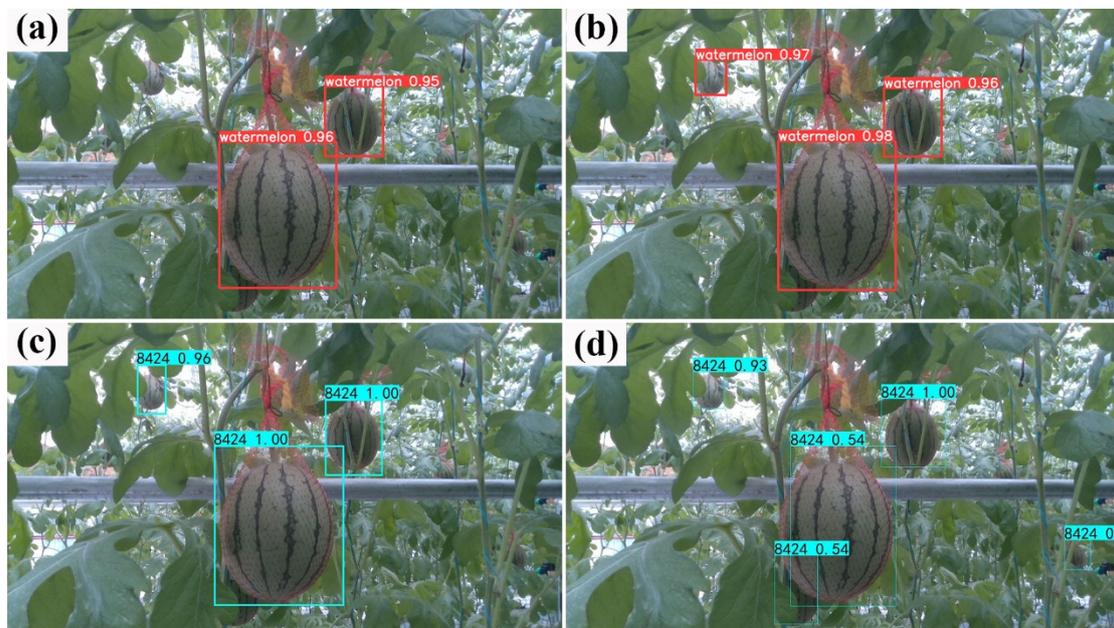


Figure 13. Inference results of different detectors: (a) YOLOv5s–CBAM; (b) YOLOv5s; (c) YOLOv4; (d) SSD.

3.3. Evaluation of the Watermelon-Harvesting Robot Prototype

The automatic harvesting experiments were conducted using a prototype watermelon-harvesting robot, and the machine construction and the motion control of the prototype are shown in Figures 3 and 4. We set up a highly reversible simulation environment in the laboratory and performed the end-effector clamping performance verification and the whole-machine performance evaluation of the watermelon-harvesting robot, as shown in Figure 14.

3.3.1. End-Effector Clamping Performance Verification

We randomly purchased 60 mature L600 and 8424 varieties of small watermelons (30 of each variety) from a market for individual testing of the clamping performance of the end-effector. In this testing process, we did not consider picking failure caused by the positioning error of the vision system, so the end-effector needed to be manually moved to the bottom of the watermelon. Then, the watermelon-harvesting robot entered

the execution phase of the workflow to execute the grasping and separating action on the watermelon, as shown in Figure 4. The main indicators to assess the performance of the end-effector are the successful clamping rate, the maximum shearing time, and the average shearing time.



Figure 14. Validation of the watermelon-harvesting robot in a simulation environment.

The results of the individual tests on the end-effector are shown in Table 3. For the two different varieties of watermelon, the end-effector successfully clamped 28 and 29 watermelons (the successful clamping rate reached 95%), respectively, indicating that the flexible finger design was able to effectively clamp watermelons. The other three watermelons failed to be clamped because the watermelons were too wide, exceeding the average width of the fruit by 30 mm. After the end-effector successfully clamped a watermelon, the cutting device started to shear the pedicel of the fruit, and the average shearing times for cutting the stalks of the two watermelon varieties were 5.0 s and 5.2 s, and the maximum shearing times were 5.5 s and 5.9 s, respectively. The experiment proved that a shearing time setting of 6 s can ensure the pedicel of the watermelon fruit is effectively cut off.

Table 3. End-effector experiment results.

Varieties	Number of Experiments	Successful Clamping	Maximum Shearing Time (s)	Average Shearing Time (s)
L600	30	28	5.5	5.0
8424	30	29	5.9	5.2

3.3.2. Evaluation of Automatic Harvesting

To validate the overall performance of the watermelon-harvesting robot, we developed a watermelon-harvesting robot system that integrates mobile platform control, vision recognition algorithms, and robotic arm motion planning, thus automating the inspection phase, the positioning phase, and the execution phase (Video S1). The whole-machine test experiment was conducted in the laboratory of the College of Engineering of China Agricultural University. Two experiments were conducted to better reproduce a real greenhouse scenario, one of which was to set up local shading on the surfaces of the watermelons. In these experiments, we manually recorded the average positioning error, the number of fruits successfully clamped, and the number of fruits successfully sheared by the watermelon-harvesting robot. In each operation process, the robot recorded the three-dimensional coordinates of each grasping point. After first arriving at the 3D coordinates of the grasping point, the robotic arm stopped moving, and we manually fine-tuned the 3D position of the end-effector. And we recorded the two position coordinates before and after adjustment. The Euclidean distance between two positions is the positioning error of the watermelon-harvesting robot. Finally, the robot arm returned to the initial position and performed the complete operation process using the recorded original grasping point.

The results of the experiments for automatic harvesting in the simulation scenario are shown in Table 4. In the first set of tests, we did not apply any shading to the surfaces of the watermelons. Considering errors of hand–eye calibration, the average value of the fruit positioning error of the watermelon-harvesting robot was 8.7 mm. In addition, 56 fruits were successfully clamped and sheared in a total of 60 tests. In the second set of tests, we arranged leaves on the surfaces of the watermelons and set the watermelons to be partially obscured. Compared with the clustered fruit point-cloud information in the unobstructed case, the clustered point clouds in the obscured case still had some of the leaf point clouds, which were misclassified as fruit point clouds. As a result, the average localization error of the vision system for an obscured watermelon increased to 14.6 mm, which also led to a decrease in the clamping success rate of the harvesting robot from 93.3% to 85.0%. The success rate of the cutting device was 100% after the watermelon fruit was successfully grasped by the flexible fingers.

Table 4. The results of the experiments for automatic harvesting.

Number of Experiments	Partial Obstruction	Average Positioning Error	Successful Clamping	Successful Shearing
60	×	8.7	56	56
60	✓	14.6	51	51

3.4. Discussion

The development and experimental study of a prototype watermelon-harvesting robot provide valuable insights into the automated harvesting of watermelons in greenhouses, especially in the development of the vision system and the design of the end-effector.

In the problem of watermelon object-detection and localization, the complex background is an important factor affecting the accuracy of the object-detection algorithm. The lack of spatial information perception makes it difficult for the target detection algorithm to correctly distinguish between the front and back rows of watermelons using only colour images, and the utilization of an attention mechanism can alleviate these conditions, but it cannot solve the root cause. Therefore, the object-detection algorithm can be improved by multimodal fusion to obtain both colour information and location information by fusing colour and depth images as the model input [43].

For the problem of locating the grasping points, the K-means clustering method takes a large amount of time and tends to misclassify the leaf point cloud covering the fruit surface into the fruit point cloud. Incorrect point cloud classification introduces noise to the fruit point cloud, which increases the error of the ellipsoidal fit. We are now trying to

solve this problem; after the object detector detects the watermelon target, we segment the cropped ROI image pixel by pixel using a semantic segmentation algorithm, and it is fast.

Finally, the end-effector we designed works on small watermelons (see Table 1), which is difficult to clamp larger watermelons. Also, the cutting knife tends to cut the vines when the front of the watermelon is obscured by the vines. Therefore, the design of the end-effector should be more dexterous, and some safety considerations must be made. For example, the end-effector of the strawberry harvesting robot designed by Xiong et al. [44] has a cutting device located inside the end-effector. The knife inside the finger rotates quickly to cut the pedicel only when the strawberry fruit falls inside the finger. In addition, our cutting device drives the rotating blade to cut the pedicel by unprotected oscillation, which could potentially damage other vines. Furthermore, we need more field experiments to discover the flaws of our designed watermelon-harvesting robot and to optimize it. All of the above issues will be the focus of future work and are listed as potential research directions for us.

4. Conclusions

This paper presents a fully integrated robot for the automatic harvesting of watermelon fruits, consisting of hardware components, robot control, and a vision system. The vision system of the harvesting robot includes watermelon object recognition and grasping point localization. First, we proposed an improved YOLOv5 object-detection algorithm for recognizing watermelon targets in greenhouse scenarios with 89.8% accuracy on a test dataset. Then, we adopted the K-means algorithm to cluster the point clouds in the region of interest to obtain the watermelon point cloud for fitting the ellipsoid. In addition, we designed an end-effector for harvesting watermelons, which includes two parts: the clamping mechanism and the cutting mechanism. To evaluate the performance of the end-effector and the vision system, we built a prototype watermelon-harvesting robot and conducted a series of tests in a laboratory simulation environment. The overall harvesting success rate was 93.3% with a positioning error of 8.7 mm in the unobstructed condition of watermelon. Under the condition of the watermelon being partially obscured by the leaves, the overall harvesting success rate was 85.0%, and the positioning error was 14.6 mm. The purpose of this study was to explore the potential of robotic watermelon harvesting, design a dedicated end-effector, and propose a grasping point positioning method that provides a possibility and a reference base for future research on watermelon-harvesting robots. The design of the end-effector and the development of the overall vision system provide a research basis and reference for the future advancement of watermelon-harvesting robots.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/agronomy12112836/s1>, Video S1: watermelon-harvesting robot tested in a laboratory simulation scenario.

Author Contributions: Conceptualization, T.Y.; funding acquisition, T.Y.; methodology, T.Y., J.R., J.F. and P.W.; image annotation, J.R. and Z.Z.; software, J.R. and J.F.; validation, J.R., J.F. and J.Y.; writing—original draft, J.R., Y.T. and J.F.; writing—review and editing, J.R., T.Y. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2017YFD0701303.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. FAO. 2022. Available online: <https://www.fao.org> (accessed on 2 September 2022).
2. Bac, C.W.; van Henten, E.J.; Hemming, J.; Edan, Y. Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. *J. Field Robot.* **2014**, *31*, 888–911. [[CrossRef](#)]
3. Tsolakis, N.; Bechtsis, D.; Bochtis, D. AgROS: A Robot Operating System Based Emulation Tool for Agricultural Robotics. *Agronomy* **2019**, *9*, 403. [[CrossRef](#)]
4. Zhang, B.; Xie, Y.; Zhou, J.; Wang, K.; Zhang, Z. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. *Comput. Electron. Agric.* **2020**, *177*, 105694. [[CrossRef](#)]
5. Kootstra, G.; Wang, X.; Blok, P.M.; Hemming, J.; van Henten, E. Selective Harvesting Robotics: Current Research, Trends, and Future Directions. *Curr. Robot. Rep.* **2021**, *2*, 95–104. [[CrossRef](#)]
6. Schertz, C.E.; Brown, G.K. Basic Considerations in Mechanizing Citrus Harvest. *Trans. ASAE* **1968**, *11*, 343–346. [[CrossRef](#)]
7. Vasconez, J.P.; Delpiano, J.; Vougioukas, S.; Auat Cheein, F. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Comput. Electron. Agric.* **2020**, *173*, 105348. [[CrossRef](#)]
8. Gené-Mola, J.; Gregorio, E.; Guevara, J.; Auat, F.; Sanz-Cortiella, R.; Escolà, A.; Llorens, J.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V.; et al. Fruit detection in an apple orchard using a mobile terrestrial laser scanner. *Biosyst. Eng.* **2019**, *187*, 171–184. [[CrossRef](#)]
9. Kondo, N.; Yata, K.; Iida, M.; Shiigi, T.; Monta, M.; Kurita, M.; Omori, H. Development of an End-Effector for a Tomato Cluster Harvesting Robot. *Eng. Agric. Environ. Food* **2010**, *3*, 20–24. [[CrossRef](#)]
10. Lin, G.; Zhu, L.; Li, J.; Zou, X.; Tang, Y. Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning. *Comput. Electron. Agric.* **2021**, *188*, 106350. [[CrossRef](#)]
11. He, Z.; Ma, L.; Wang, Y.; Wei, Y.; Ding, X.; Li, K.; Cui, Y. Double-Arm Cooperation and Implementing for Harvesting Kiwifruit. *Agriculture* **2022**, *12*, 1763. [[CrossRef](#)]
12. Reed, J.N.; Tillett, R.D. Initial experiments in robotic mushroom harvesting. *Mechatronics* **1994**, *4*, 265–279. [[CrossRef](#)]
13. Reed, J.N.; Miles, S.J.; Butler, J.; Baldwin, M.; Noble, R. AE—Automation and Emerging Technologies: Automatic Mushroom Harvester Development. *J. Agric. Eng. Res.* **2001**, *78*, 15–23. [[CrossRef](#)]
14. Qian, Y.; Jiacheng, R.; Pengbo, W.; Zhan, Y.; Changxing, G. Real-time detection and localization using SSD method for oyster mushroom picking robot. In Proceedings of the 2020 IEEE International Conference on Real-time Computing and Robotics (RCAR), Asahikawa, Japan, 28–29 September 2020; pp. 158–163.
15. Rong, J.; Wang, P.; Yang, Q.; Huang, F. A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse. *Agronomy* **2021**, *11*, 1210. [[CrossRef](#)]
16. Baeten, J.; Donné, K.; Boedrij, S.; Beckers, W.; Claesen, E. Autonomous Fruit Picking Machine: A Robotic Apple Harvester. In *Field and Service Robotics: Results of the 6th International Conference*; Laugier, C., Siegwart, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 531–539.
17. Kang, H.; Zhou, H.; Chen, C. Visual Perception and Modeling for Autonomous Apple Harvesting. *IEEE Access* **2020**, *8*, 62151–62163. [[CrossRef](#)]
18. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [[CrossRef](#)]
19. Kang, H.; Zhou, H.; Wang, X.; Chen, C. Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting. *Sensors* **2020**, *20*, 5670. [[CrossRef](#)]
20. Leu, A.; Razavi, M.; Langstädtler, L.; Ristić-Durrant, D.; Raffel, H.; Schenck, C.; Gräser, A.; Kuhfuss, B. Robotic Green Asparagus Selective Harvesting. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2401–2410. [[CrossRef](#)]
21. Williams, H.A.M.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [[CrossRef](#)]
22. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [[CrossRef](#)]
23. Arad, B.; Balendonck, J.; Barth, R.; Ben-Shahar, O.; Edan, Y.; Hellström, T.; Hemming, J.; Kurtser, P.; Ringdahl, O.; Tielen, T.; et al. Development of a sweet pepper harvesting robot. *J. Field Robot.* **2020**, *37*, 1027–1039. [[CrossRef](#)]
24. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [[CrossRef](#)]
25. Sakai, S.; Osuka, K.; Fukushima, H.; Iida, M. Watermelon harvesting experiment of a heavy material handling agricultural robot with LQ control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; Volume 761, pp. 769–774.
26. Hu, Z.; Zhang, X.; Zhang, W.; Wang, L. Precise control of clamping force for watermelon picking end-effector. *Trans. Chin. Soc. Agric. Eng.* **2014**, *30*, 43–49.
27. Ji, W.; Zhang, J.; Xu, B.; Tang, C.; Zhao, D. Grasping mode analysis and adaptive impedance control for apple harvesting robotic grippers. *Comput. Electron. Agric.* **2021**, *186*, 106210. [[CrossRef](#)]
28. Gao, J.; Zhang, F.; Zhang, J.; Yuan, T.; Yin, J.; Guo, H.; Yang, C. Development and evaluation of a pneumatic finger-like end-effector for cherry tomato harvesting robot in greenhouse. *Comput. Electron. Agric.* **2022**, *197*, 106879. [[CrossRef](#)]

29. Jun, J.; Kim, J.; Seol, J.; Kim, J.; Son, H.I. Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector. *IEEE Access* **2021**, *9*, 17631–17640. [[CrossRef](#)]
30. Wang, Y.; Yang, Y.; Yang, C.; Zhao, H.; Chen, G.; Zhang, Z.; Fu, S.; Zhang, M.; Xu, H. End-effector with a bite mode for harvesting citrus fruit in random stalk orientation environment. *Comput. Electron. Agric.* **2019**, *157*, 454–470. [[CrossRef](#)]
31. Bac, C.W.; Hemming, J.; van Tuijl, B.A.J.; Barth, R.; Wais, E.; van Henten, E.J. Performance Evaluation of a Harvesting Robot for Sweet Pepper. *J. Field Robot.* **2017**, *34*, 1123–1139. [[CrossRef](#)]
32. Afonso, M.; Fonteijn, H.; Fiorentin, F.S.; Lensink, D.; Mooij, M.; Faber, N.; Polder, G.; Wehrens, R. Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning. *Front. Plant Sci.* **2020**, *11*, 571299. [[CrossRef](#)]
33. Zhang, C.; Zou, K.; Pan, Y. A Method of Apple Image Segmentation Based on Color-Texture Fusion Feature and Machine Learning. *Agronomy* **2020**, *10*, 972. [[CrossRef](#)]
34. Mokhtar, U.; Ali, M.A.S.; Hassenian, A.E.; Hefny, H. Tomato leaves diseases detection approach based on Support Vector Machines. In Proceedings of the 2015 11th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 29–30 December 2015; pp. 246–250.
35. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [[CrossRef](#)]
36. Darwin, B.; Dharmaraj, P.; Prince, S.; Popescu, D.E.; Hemanth, D.J. Recognition of Bloom/Yield in Crop Images Using Deep Learning Models for Smart Agriculture: A Review. *Agronomy* **2021**, *11*, 646. [[CrossRef](#)]
37. Saleem, M.H.; Potgieter, J.; Arif, K.M. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. *Precis. Agric.* **2021**, *22*, 2053–2091. [[CrossRef](#)]
38. Chakraborty, S.K.; Chandel, N.S.; Jat, D.; Tiwari, M.K.; Rajwade, Y.A.; Subeesh, A. Deep learning approaches and interventions for futuristic engineering in agriculture. *Neural Comput. Appl.* **2022**, *34*, 20539–20573. [[CrossRef](#)]
39. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
40. Tsai, R.Y.; Lenz, R.K. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [[CrossRef](#)]
41. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
42. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision–ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
43. Sun, Q.; Chai, X.; Zeng, Z.; Zhou, G.; Sun, T. Noise-tolerant RGB-D feature fusion network for outdoor fruit detection. *Comput. Electron. Agric.* **2022**, *198*, 107034. [[CrossRef](#)]
44. Xiong, Y.; From, P.J.; Isler, V. Design and Evaluation of a Novel Cable-Driven Gripper with Perception Capabilities for Strawberry Picking Robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7384–7391.