agronomy

Article

Cherry Tomato Detection for Harvesting Using Multimodal Perception and an Improved YOLOv7-Tiny Neural Network

Yingqi Cai [1,†], Bo Cui [2,3,†], Hong Deng [3,†], Zhi Zeng [2], Qicong Wang [1], Dajiang Lu [1], Yukang Cui [1] and Yibin Tian [1,*]

[1] College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
[2] School of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
[3] R&D Division, Litemaze Technology, Shenzhen 518060, China
* Correspondence: ybtian@szu.edu.cn
† These authors contributed equally to this work.

Abstract: Robotic fruit harvesting has great potential to revolutionize agriculture, but detecting cherry tomatoes in farming environments still faces challenges in accuracy and efficiency. To overcome the shortcomings of existing cherry tomato detection methods for harvesting, this study introduces a deep-learning-based cherry tomato detection scheme for robotic harvesting in greenhouses using multimodal RGB-D perception and an improved YOLOv7-tiny Cherry Tomato Detection (YOLOv7-tiny-CTD) network, which has been modified from the original YOLOv7-tiny by eliminating the "Objectness" output layer, introducing a new "Classness" method for the prediction box, and incorporating a new hybrid non-maximum suppression. Acquired RGB-D images undergo preprocessing such as color space transformation, point cloud normal vector angle computation, and multimodal regions of interest segmentation before being fed into the YOLOv7-tiny-CTD. The proposed method was tested using an AGV-based robot in a greenhouse cherry tomato farming facility. The results indicate that the multimodal perception and deep learning method improves detection precision and accuracy over existing methods while running in real time, and the robot achieved over 80% successful picking rates in two-trial mode in the greenhouse farm, showing promising potential for practical harvesting applications.

Keywords: cherry tomato; fruit harvesting; multimodal perception; RGB image; point cloud; deep neural network

Citation: Cai, Y.; Cui, B.; Deng, H.; Zeng, Z.; Wang, Q.; Lu, D.; Cui, Y.; Tian, Y. Cherry Tomato Detection for Harvesting Using Multimodal Perception and an Improved YOLOv7-Tiny Neural Network. *Agronomy* **2024**, *14*, 2320. https://doi.org/10.3390/agronomy14102320

Academic Editors: Baohua Zhang and Yongliang Qiao

Received: 15 August 2024
Revised: 29 September 2024
Accepted: 5 October 2024
Published: 9 October 2024

1. Introduction

Fruit harvesting robots have been studied over the decades, mostly in research settings [1–4]. With the advancement of robotics and deep learning, they are gaining increasing attention in recent years [5–7]. For harvesting robots, rapid and accurate identification and localization of fruits have become some of the key challenges. Previous studies have primarily focused on the recognition and localization of fruits using color images. Bulanon et al. used an automatic threshold selection algorithm based on the red color difference histogram, effectively improving the accuracy of segmentation and the success rate of recognition [8]. Payne et al. separated mangoes from the background by performing color segmentation in the RGB and YCbCr color spaces [9]. Senthinath et al. proposed a co-spectral space classification method that first uses the naive Bayes algorithm to obtain the optimal number of image clusters, followed by spectral clustering using K-means, expectation maximization, and self-organizing maps for tomato identification [10]. Luo et al. introduced an automatic detection method for ripe grapes based on visual sensors, integrating the AdaBoost framework with multiple color components, which can suppress the impact of environmental noise to some extent [11]. Teixidó et al. employed a classification method based on multiple color hyper-planes implicit in the color space, calculating the distance of pixel points to different hyper-planes to improve detection accuracy [12]. Kurtulmus et al. extracted features through circular Gabor texture analysis and achieved the recognition of immature peach fruits in natural images using various classifiers [13]. Zhou et al. utilized the K-means algorithm for cherry tomato picking [14]. Fruit picking robots often encounter issues such as drastic changes in lighting, complex backgrounds, and random occlusions of target objects when operating in outdoor unstructured environments. These factors lead to insufficient stability of target detection methods that only rely on features such as color, spectrum, and shape. Although the aforementioned algorithms have high detection efficiency in well-controlled environments, they still face challenges in unstructured field environments.

Due to the strong representational capability of deep learning for images, it can effectively compensate for the lack of robustness in algorithms that solely depend on color, shape, and texture features. Many studies have introduced deep learning into fruit harvesting robots [15]. Chen et al. introduced a dual-path network based on YOLOv3 to enhance the semantic feature extraction for cherry tomatoes [16]. Zheng et al. proposed YOLOvX-Dense-CT for cherry tomatoes based on the YOLOvX model. It uses the DenseNet as its core framework and integrates the convolutional block attention module to significantly improve the model's recognition [17]. Yan et al. constructed a new Si-YOLO network to identify cherry tomatoes. It is based on YOLOv5, integrating the SIMAM attention module and generative adversarial network to enhance the model's generalization performance [18]. Wang et al. proposed a lightweight model for detecting cherry tomatoes based on YOLOv5. It calculates the size and aspect ratio of the anchor box through K-means++ and the coordinate attention mechanism and the weighted intersection over union (IOU) loss function [19]. The generated weight file is only 4 MB in size.

The more recent YOLOv7 model consists of Input, Backbone, Neck, and Head and provides three basic models of YOLOv7-tiny, YOLOv7, and YOLOv7-W6 [20]. Compared with other YOLO series, the model construction of YOLOv7 is similar to that of YOLOv4 and YOLOv5 [21]. Its core structure mainly consists of convolutions and the Extended-ELAN (E-ELAN), MPConv, and SPPCSPC modules. The E-ELAN module can greatly improve the learning performance of the network by changing the structure of the calculation block without affecting the original gradient path and using techniques to expand, shuffle, and merge cardinality. The SPPCSPC module greatly reduces the distortion and repetitive feature problems that may occur in the image processing by introducing multiple parallel MaxPool operations, thereby effectively improving the performance. In the MPConv module, the MaxPool operation can greatly expand the sensing range of the current feature layer and combine it with the feature processed by convolutions, thus greatly improving the generalization ability of the network.

In order to ensure the recall rate, the existing target detection algorithms often output multiple candidate boxes for the same target. However, multiple redundant candidate boxes will affect the detection accuracy, thus it is necessary to use postprocessing to filter out the redundant ones and output the candidate boxes with the highest confidence scores. The original YOLOv7 uses non-maximum suppression (NMS) to eliminate redundant candidate boxes, but the operation efficiency of this method is low and only IOU is used as the evaluation metric, which is insufficient to fully describe the overlapping relationship. In addition, using a manually set threshold will make the results not robust. Although other studies have proposed various solutions, such as Weighted NMS [22], DIOU NMS [23], and CIOU NMS [24], it is still difficult to solve the problem effectively. Learning-based NMS has also been introduced [25], but its implementation is not simple. Table 1 shows the advantages and disadvantages of various NMS methods.

Table 1. Comparison of different Non-Maximum Suppression (NMS) methods.

NMS	Advantage	Disadvantage
Regular	Simple	Sequential processing, IOU selected based on experience
Weighted [22]	High precision	Sequential processing, low efficiency
DIOU [23]	High recall; can be combined with other methods	Low efficiency; abnormal conditions when centers coincide
CIOU [24]	Overcome DIOU anomalies	Low efficiency; increasing number of iterations
Learning [25]	No hand-crafted settings	Complex implementation

In addition, the aforementioned algorithms primarily rely on color images and underutilize depth information. During the operation of the harvesting robot, these models need high-performance GPU hardware, which leads to a significant increase in cost. Depth information can improve the detection of cherry tomatoes as distance can help reduce the ambiguities arising from partial occlusions by neighboring cherry tomatoes. To overcome these shortcomings of existing cherry tomato detection methods for harvesting, this study combines multimodal perception (color and depth, RGB-D for short) with an improved YOLOv7-tiny network to achieve real-time detection and positioning of cherry tomatoes in commercial greenhouse farms, named YOLOv7-tiny-CTD. It was evaluated with a harvesting robot consisting of an off-the-shelf collaborative robotic arm, a custom automated ground vehicle (AGV) platform, and a custom end effector with force sensors and an RGB-D sensor. Results indicate that the proposed YOLOv7-tiny-CTD model has significantly enhanced the recognition capability for cherry tomatoes. Moreover, harvesting experiments demonstrate that the robot achieved a more than 80% picking success rate and has potential for applications in commercial farming facilities. Partial results have been previously presented at a conference [26].

Cherry tomatoes can be harvested either individually or by bunches. The latter is much more efficient, if and only if we look at picking from the vines alone. For many farms, if harvested in bunches, individual fruits still need to be separated from them and examined and classified before being packaged, which needs much more time than picking bunches from the vines. Some farms envisioned the use of robots to classify and pick individual fruits and directly package them. Whether this is more efficient needs to be studied in the future. In the current study, we chose detecting and picking individual cherry tomatoes at the request of the farm that we worked with.

The main contributions of the paper are:

(1) Multimodal RGB-D images are utilized in combination with simple preprocessing methods to screen regions of interest (ROIs) for cherry tomato detection to improve efficiency.

(2) To better utilize depth information for cherry tomato detection, the normal vector angles of a point cloud are introduced and combined with the non-luminance color channels in the Lab color space of an RGB image as the input to neural networks.

(3) In addition to the multimodal image input, YOLOv7-tiny has been improved from three different aspects: eliminating the “Objectness” output layer; introducing a new “Classness” method for prediction box; and improving the NMS by using a hybrid method.

(4) The proposed approach has been evaluated with a cherry tomato harvesting robot in a commercial greenhouse farm, and it outperforms several state-of-the-art detection neural networks in precision, recall, and accuracy while running at 26 FPS on Nvidia Jetson TX1. And cherry tomato picking based on the detection results shows promising potential for practical applications.

2. Materials and Methods

2.1. Cherry Tomato Picking Robot

We first introduce the cherry tomato harvesting robot. The functional module diagram of the robotic system is shown in Figure 1. The main hardware modules include a customized AGV platform, a collaborative robotic arm (cobot), a customized end effector with force sensors and an RGB-D sensor, and computing, communication, and control modules. And the main software modules are obstacle avoidance, eye–hand calibration, cherry tomato detection, robotic arm and end effector control, communication and data transfer, and debugging tool and graphic user interface (GUI).

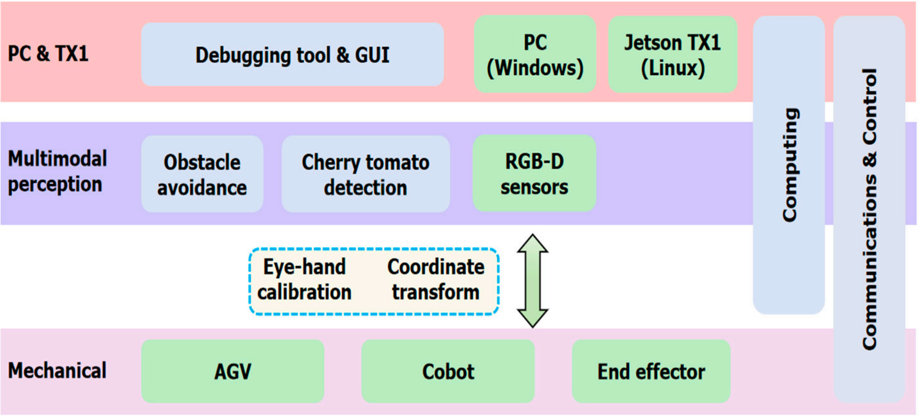

Figure 1. The function module diagram of the cherry tomato picking robotic system.

2.1.1. Main Hardware Modules

As the robot is intended for use in greenhouse cherry tomato farming facilities, the AGV only runs on embedded tracks on the ground (Figure 2a), and no extra navigation is needed. For obstacle avoidance, two Litemaze time-of-flight (TOF30) sensors (the highest resolutions of color and depth images are 5 MP and 0.3 MP (640 × 480); Litemaze Technology, Shenzhen, China) are installed at the front and back sides of the AGV, respectively. These TOF sensors have short working distance of 3 m, which is sufficient for the low-speed AGV whose speed is capped at 1 m/s.

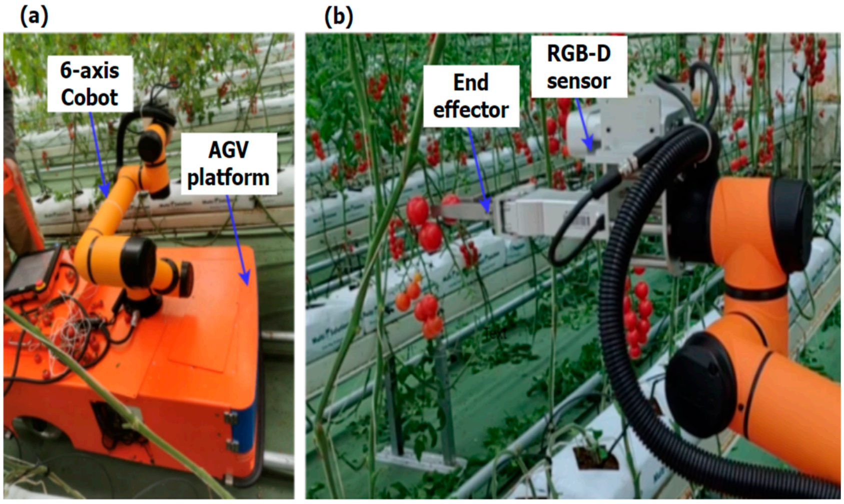

Figure 2. The cherry tomato harvesting robot in the greenhouse commercial farming facility. (**a**) The robotic system on the track in the greenhouse. (**b**) Single cherry tomato picking.

A cobot arm Aubo i5 (Aubo Robotics, Beijing, China) is mounted on top of the AGV to offer 6 degrees of manipulation freedom. It has arm reach of 88.65 cm and payload of 5 Kg. A customized end effector with two straight parallel fingers is attached to the end arm of the cobot. A force sensor is built into each finger of the end effector. The two fingers can move towards or away from each other driven by an electrical motor. A multimodal RGB-D sensor, either the Microsoft Kinect DK (Microsoft, Redmond, WA, USA) or the Litemaze TOF30, can be mounted on the cobot end arm. The Kinect DK is a developer kit, and its highest resolutions of color and depth images are 12 MP and 1 MP, respectively. The total weight of the end effector, RGB-D sensor, and the installation parts, illustrated in Figure 2b, is around 3 Kg, leaving sufficient payload capacity for cherry tomato picking. Two-arm solutions have been used to imitate humans for fruit picking [27], with more complex systems at a much higher cost. The embedded computing platform for perception and robotic control is the Nvidia Jetson TX1 (Nvidia, Santa Clara, CA, USA). A separate personal computer (PC) with Intel i7 CPU and 16 GB of RAM (Lenovo, Beijing, China) is used for development and hosting the user interface.

The cost of the build of materials (BOM) for the hardware is about USD 28,000 (the AGV platform: USD 14,000, the cobot arm: USD 8000, the end effector and sensors: USD 3000, the PC and TX1: USD 1800, other parts and cables: USD 1200).

2.1.2. Main Software Modules

The main software modules are obstacle avoidance, cherry tomato detection, robotic arm and end effector control, communication and data transfer, and debugging tool and graphic user interface (GUI). The algorithms involved in the obstacle avoidance, eye–hand calibration, and cherry tomato detection will be discussed in detail later. The debugging and GUI are shown in Figure 3. System parameters can be configured via the settings interface. The communications between the PC, TX1, and cobot are by TCP/IP protocol. And the TX1 communicates with the RGB-D sensor by USB.

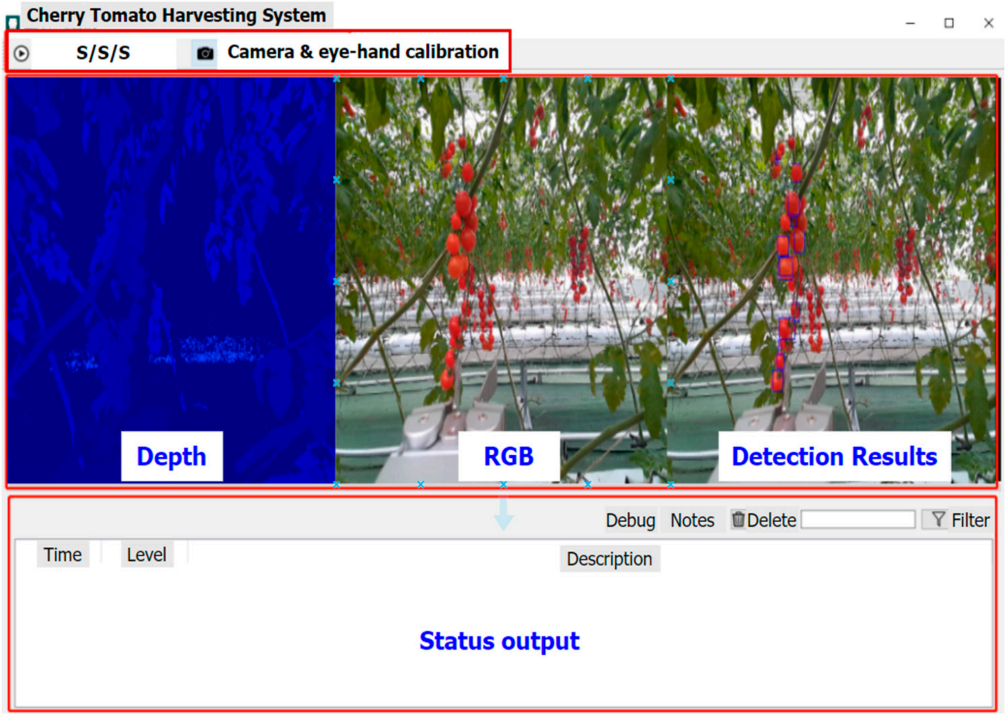

Figure 3. The graphic user interface of the robotic system. S/S/S represent Start/Stop/Settings.

2.2. *Multimodal Perception*

Multimodal perception here refers to the utilization of color, depth, and force sensors for the cherry tomato picking robot. The force sensor feedback is used to prevent damage to the cherry tomatoes.

2.2.1. Obstacle Avoidance

As the robot moves around in the greenhouse in autonomous mode, it needs to be able to avoid obstacles, such as farm workers, unexpected objects falling on the ground, or loose cherry tomato vines. As the AGV is on the track, obstacle avoidance is only offered for forward and backward movements. Both RGB and depth images are used for obstacle detection, using a strategy similar to the salient feature learning in a previous study [28]. The control strategy is simply distance based; the space in front in the moving direction is divided into three different zones: danger/red (<1 m), risk/yellow (1–2 m), and safe/green (>2 m). If an obstacle of a diameter greater than a given size (in this study, it was set to 2 cm) is detected in the red or yellow zone, the robot immediately stops or slows down, respectively. Otherwise, it continues moving at the current speed. Obstacles smaller than the given size are ignored as they are unlikely to pose a risk. In the study, obstacle avoidance testing was carried out by dropping objects on the ground in front of the AGV, and human operators were not allowed to be in front of it.

2.2.2. Cherry Tomato ROI Image Patches

Mature cherry tomatoes have distinct red colors and are in great contrast with the surrounding environment. At the same time, most cherry tomatoes are in strings. Based on these characteristics, we propose a preprocessing method to segment cherry tomato ROIs by superimposing the depth information onto the RGB image to obtain a new multimodal image.

Compared with RGB color space, the Lab color space has more compact color range and higher contrast. Acquired RGB images are converted into Lab space [29] with which thresholding is carried out to obtain a binary mask B_C for each color channel where 1s represent candidate ROIs.

$$B_C(i,j) = \begin{cases} 1 & C_L \leq C(i,j) \leq C_H \\ 0 & otherwise \end{cases} \tag{1}$$

where $C(i,j)$ is the color channel ($C \in [L,a,b]$) of pixel (i,j) under consideration, C_L and C_H the low and high thresholds of the color channel (for the dataset in this study, C_L and C_H were set to 30 and 70 for L (range [0, 100]), 18 and 127 for a (range [−128, 127]), and 10 and 50 for b (range [−128, 127]). Similarly, for the depth channel, thresholding is carried out to obtain a binary mask B_D where 1s are candidate ROIs.

$$B_D(u,v) = \begin{cases} 1 & d_L \leq D(u,v) \leq d_H \\ 0 & otherwise \end{cases} \tag{2}$$

where $D(u,v)$ is the depth of pixel (u,v), d_L and d_H the low and high depth thresholds (in this study, d_L and d_H were set to 30 to 100 cm). The initial cherry tomato ROIs are determined as

$$B_{ROI}(i,j) = [B_L(i,j) \cap B_a(i,j) \cap B_b(i,j)] \cap f[D(u,v)] > 0 \tag{3}$$

where $f[\cdot]$ is the rotation and translation operations that match the depth camera coordinates in (u,v) to the RGB camera coordinates in (i,j), which can be obtained from camera calibration. B_{ROI} is the intersection of all candidate ROIs in color and depth, where pixels of true values (1s) are ROIs. Figure 4 shows an example of the process of obtaining the initial ROIs, which can be further cleaned up by morphological operations, such as open and close [30,31].

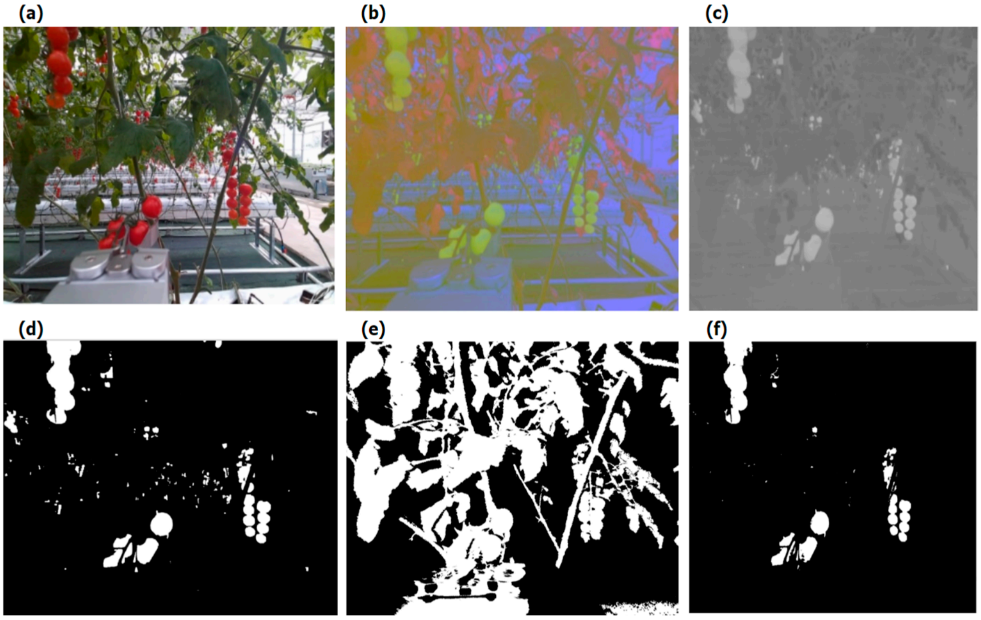

Figure 4. Initial cherry tomato ROI segmentation. (**a**) RGB image. (**b**) Image in Lab space. (**c**) Single channel in the Lab space. (**d**) Binary mask from all color channels. (**e**) Binary mask from the depth channel. (**f**) Initial cherry tomato ROIs.

As the resolution of the original images is high, if they are down-sampled to feed to deep neural networks (DNNs), the image will be compressed, resulting in the loss of information. We can crop the input images into patches of a fixed size and feed them into DNNs in batches, but the computational cost is high. To address this issue, we introduce a threshold parameter for each image patch as

$$S_p = \sum_{\mathrm{i}=0}^{\mathrm{m}-1} \sum_{\mathrm{j}=0}^{\mathrm{n}-1} B_{ROI}(i,j) \tag{4}$$

where m and n are the row and column sizes of the patch. When S_p is less than the set threshold (based on the image size of the cherry tomatoes in the dataset, this study set the threshold to 10), there are no cherry tomatoes in the patch, and it is discarded. The sequentially cropped image and the filtered cropped image are shown in Figure 5.

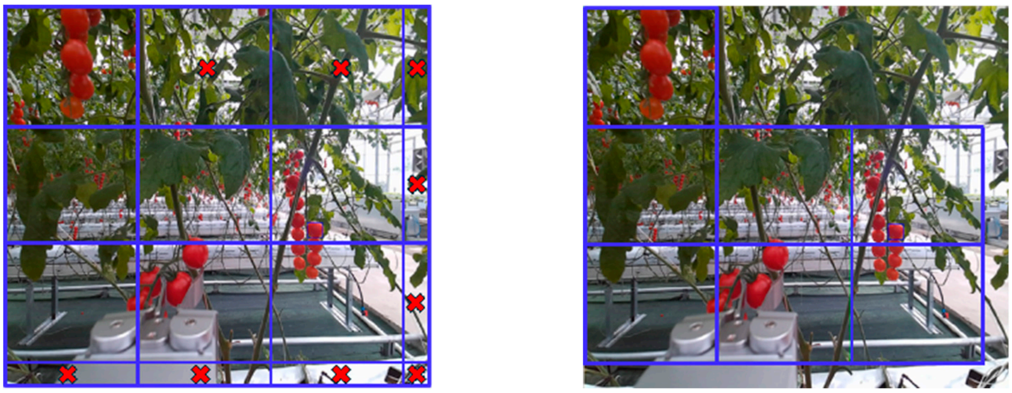

Figure 5. The sequential cropped patches before (**left**) and after (**right**) filtering. Some patches (with red cross signs in the middle) on the top right, right, and bottom are discarded.

With the simple cropping, a string of cherry tomatoes or even single fruits may be divided into different patches. This leads to inaccurate coordinates of the cherry tomatoes, which greatly reduces the accuracy and robustness of the detection results. Using sliding windows to generate many patches with overlaps can overcome this issue, but at the cost of a dramatic increase in computations. Therefore, we use a patch merging strategy; patches are merged if (1) the pixel distances between the patch borders and the image edges are less than a given threshold, or (2) the pixel distances between the adjacent patches are within a given threshold, which can be accomplished by the distance transform [32].

2.2.3. Normal Vector Angles of a Point Cloud

The image captured by the TOF camera only contains depth. In Euclidean space, the horizontal and vertical coordinates are needed to represent the position information, forming XYZ coordinate values. Therefore, it is necessary to calculate the coordinate value of a point in the depth map according to the internal parameters of the camera, and then convert it into point cloud. Because of the small size of cherry tomatoes, their depth differences are very small. To make better use of depth information, we calculate the normal vector angles of a point cloud, as illustrated in Figure 6, and convert them into an image.

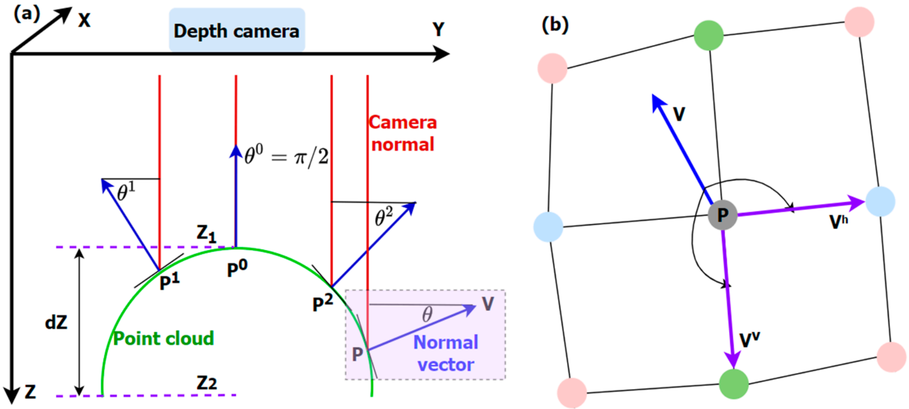

Figure 6. Illustration of the normal vector angles of a point cloud. (**a**) Normal vector angles. A cherry tomato is a small spherical object, the depth range (dZ $= Z_2 - Z_1$) is only a tiny fraction of its distance to the camera (Z_1). (**b**) A local point cloud neighborhood and its normal vector (depiction of the purple rectangular area in (**a**)). Each colored circle represents a point in the local neighborhood.

(1) The integral image of the Z data of the point cloud is constructed, which is the sum of all Z coordinate values in the rectangular area from $Z(0,\ 0)$ to $Z(P,\ Q)$.

$$I_Z = \sum_{i=0}^{P} \sum_{j=0}^{Q} Z(i,\ j) \tag{5}$$

where P and Q are the row and column sizes of the depth map.

(2) The horizontal and vertical vectors are computed as [33]

$$\begin{cases} V_x^h = 0.5[X(m+r,n) - X(m-r,n)] \\ V_y^h = 0.5[Y(m+r,n) - Y(m-r,n)] \\ V_z^h = 0.5[S(I_Z,m+1,n,r-1) - S(I_Z,m-1,n,r-1)] \end{cases}$$
$$\begin{cases} V_x^v = 0.5[X(m,n+r) - X(m,n-r)] \\ V_y^v = 0.5[Y(m,n+r) - Y(m,n-r)] \\ V_z^v = 0.5[S(I_Z,m,n+1,r-1) - S(I_Z,m,n-r,r-1)] \end{cases} \tag{6}$$

where X and Y are the values of the pixels at the corresponding positions of the XY channels in the Euclidean coordinate system, r is the radius of a smooth rectangular region, and S the average value in the specified area, which can be calculated as

$$S(I_Z,m,b,r) = [I_Z(m+r,n+r) - I_Z(m-r,n+r) - I_Z(m+r,n-r) \\ +I_Z(m-r,n-r))]/\left(4r^2\right) \tag{7}$$

(3) The local point cloud normal vector, as shown in Figure 6a, is

$$V = (V_x, V_y, V_z) = \left(V_x^h, V_y^h, V_z^h\right) \times \left(V_x^v, V_y^v, V_z^v\right) \tag{8}$$

where × represents the outer product of the two vectors. It is further transformed into an angle and converted to a pixel value.

$$\begin{gathered} \theta = atan\left[V_z / \sqrt{(V_x)^2 + (V_y)^2 + (V_z)^2}\right] \\ P_\theta = P_{max}\theta/(4\pi) \end{gathered} \tag{9}$$

where P_{max} is the maximum pixel value used to convert the normal vector angles to image pixel values. For 8-bit images, P_{max} is 255.

2.2.4. Depth Value in the Prediction Box

For the "Classness" prediction described below, using individual pixel depth values is volatile. To improve the stability, we utilize a depth value (Z_p) corresponding to a percentile of the cumulative distribution function of the depth values in the prediction box, as shown in Figure 7.

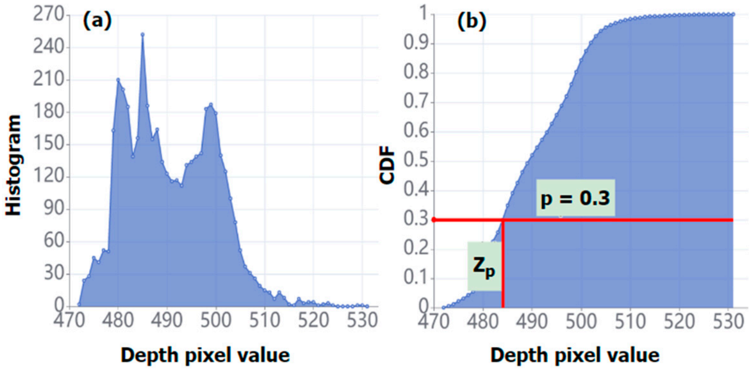

Figure 7. Illustration of depth percentile in prediction box. (**a**) Histogram of depth values. (**b**) Cumulative distribution function (CDF) of the depth values.

2.2.5. YOLOv7-Tiny-CTD: Improvement of YOLOv7-Tiny for Cherry Tomato Detection

We introduce an improved YOLOv7-tiny for cherry tomato detection as YOLOv7-tiny-CTD. YOLOv7-tiny is chosen as it is compact, with less than 1/6 of the network parameters of the other two YOLOv7 versions. The following improvements are made to YOLOv7-tiny:

(1) Replacing the color image input with a multimodal image input. Most of the object detection models, including YOLOv7, use color image input. In some methods using RGB-D input, the color image is first sent into the model to obtain the preliminary results and combined with the depth map for further processing, which struggles to make full use of the depth information. Hybrid RGB-D DNNs have been proposed to better utilize both color and depth information [34–36]. But they usually introduce significant complexity. For simplicity, we adjust the network structure of the input of the YOLOv7-tiny and add the depth map as a separate channel to the color channels in Lab space to feed into the network. As shown in Figure 8, an RGB image is replaced by a 4-channel image by adding the mask map described above.

(2) Eliminating the "Objectness" output layer of YOLOv7-tiny given that there is only one type of detection target for cherry tomatoes so for every prediction box the object inside must be a cherry tomato.

(3) Defining a new "Classness" method of the prediction box:

$$P_{class} = \frac{A_d}{2wh} + \frac{1}{2}\left(1 - \frac{Z_{top} - A_d}{Z_{max} - Z_{min}}\right) \tag{10}$$

where w and h are the width and height of the prediction box, respectively, Z_{max} and Z_{min} the maximum and minimum depth thresholds for cherry tomato detection, and A_d is the area with the depth value $d = \mathrm{Z_p}$ in the prediction box, which is explained in the previous section.

(4) Improving the NMS in YOLOv7-tiny. We propose a custom CWD-NMS combining CIOU NMS [24] and Weighted NMS [22] to solve the problem.

$$M = \frac{\sum_i W_i B_i}{\sum_i W_i},\ B_i \in \{B|CIOU(M,B) > T\},\ W_i = S_i * CIOU(M,B) \tag{11}$$

where M is the box with the current highest score, B the set of detected prediction boxes, S_i the prediction box score, and T the threshold.

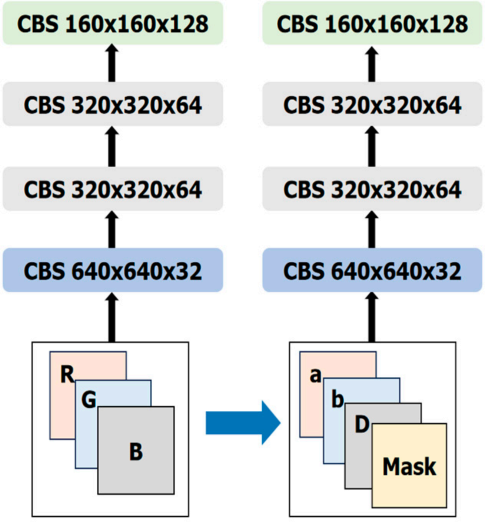

Figure 8. The adjustment to the network input for YOLOv7-tiny-CTD.

2.2.6. Evaluation Metrics

A few common metrics are used for evaluating the proposed YOLOv7-tiny-CTD cherry tomato detection method, namely, mean average precision (mAP), accuracy, and recall. For robot picking, a success rate is computed as the number of successful pickings after 2 attempts divided by the number of fruits with which the system initiated the attempts.

2.3. Eye–Hand Calibration

The cherry tomato detection RGB-D sensor is mounted on the cobot end arm in the eye-in-hand mode. It is calibrated by placing a plane calibration board with an array of small circles on the ground or the AGV platform while the cobot arm moves the sensor to different viewing angles and distances. Intrinsic and extrinsic parameters are obtained simultaneously with about 30 calibration images [37]. Hand–eye calibration is performed with the relatively simple 9-point method [38,39].

2.4. End Effector Trajectory Planning and Cherry Tomato Picking

Once cherry tomatoes are detected, their distances to the fingertips of the end effector can be calculated using the coordinates of their centers and the camera center, as well as the spatial relationship between the camera and end effector obtained from the hand–eye calibration. The simplest strategy is employed; using a straight-line trajectory to carry out cherry tomato picking. The two end effector fingers open up to about 1 cm wider than the diameter of the current target cherry tomato, which is the closest and fully exposed one in the current camera field of view. The two fingers grip the target fruit and make a turn of about 90 degrees while pulling it away from the vine by moving about 5–10 cm. If the fruit has not been detached after this process, the robot gives up and moves to the next cycle. This helps the fruit detach easily without tearing the stem or damaging the plant. In this process, the gripping force of the two fingers is capped at 10 N as measured by the force sensor to avoid bruising of the fruits. It should be noted that most cherry tomatoes occluded by others will be exposed once the latter are picked. And those occluded by leaves are ignored.

3. Experimental Results

The cherry tomato picking robot was built and tested in a commercial greenhouse farming facility. The tests and experiments ran from April to July to cover a long period such that different harvesting conditions were covered. The overall system evaluations were carried out to test autonomous running and obstacle avoidance. No collision with obstacles occurred during the test period.

More importantly, two types of specific experiments were performed to evaluate the proposed YOLOv7-tiny-CTD cherry tomato detection method and the cherry tomato picking success rate.

3.1. Dataset

The dataset was collected in the commercial greenhouse framing facility over the 4-month period. The RGB-D sensor on the cobot's arm collected both color and depth images while the AGV autonomously ran through the facility. The images were taken strictly according to the posture and distance of the cobot arm, and the collection times corresponded to the peak and end period of cherry tomato maturity, respectively. It contains 799 RGB-D image pairs and 6312 postannotated labels. The dataset was divided into training and testing at a ratio of 4 to 1 (639 and 160 images for training and testing, respectively).

The cherry tomatoes are mostly large and dense at the mature stage and relatively small and sparse at the end stage. In the latter case, the leaf color is yellowish, and the fruit is shaded. Due to the continuous mode during image acquisition, some cherry tomatoes appear blurry in the images.

3.2. Model Training and Testing

The hardware and software environments for training and testing the models are summarized in Table 2.

Table 2. Model training and evaluation conditions.

Environment	Parameters/Version
Operating system	Ubuntu18.04
CPU	Intel i7-10700F
Memory	16G
GPU	NVIDIA RTX3070
CUDA	11.2
CUDNN	8.1.1
Python	3.8
PaddlePaddle-GPU	2.2.1

3.3. YOLOv7-Tiny-CTD Compared with Existing Models

YOLOv7-tiny-CTD is compared to several existing deep learning models using the same RGB-D dataset described above, including YOLOv5-s [40], Faster R-CNN [41], SSDLite [42], and YOLOv7-tiny. As expected, for all existing models, using multimodal RGB-D images improves the cherry tomato detection performance in terms of all the metrics over usage of RGB images only (Table 3). In addition, YOLOv7-tiny outperforms all other existing models for both RGB and RGB-D inputs. More importantly, YOLOv7-tiny-CTD further improves over YOLOv7-tiny by 2.1%, 4.9%, and 4.2% in mAP, recall, and accuracy, respectively, while both of them ran at about 26 frames per second (FPS), achieving real-time detection. Figure 9 shows an example where YOLOv7-tiny-CTD detected all six cherry tomatoes while YOLOv7-tiny only detected four of them in the top left corner of the input image. The two fruits missed by YOLOv7-tiny (pointed to by orange arrows in Figure 9d) are slightly further away from the robot and partially occluded by the neighboring ones on the same string, showcasing the advantages of making good use of the depth information.

Table 3. Cherry tomato detection comparison between YOLOv7-tiny-CTD and 4 existing deep learning models using RGB-only and RGB-D images. YOLOv7-tiny-CTD has no RGB-only mode as it utilizes the normal vector angles of the point cloud.

Model	Input	mAP	Recall	Accuracy
YOLOv5-s	RGB	86.0%	87.6%	88.2%
	RGB-D	91.1%	89.7%	91.2%
Faster R-CNN	RGB	88.0%	90.1%	89.1%
	RGB-D	91.8%	91.3%	91.5%
SSDLite	RGB	87.3%	87.9%	88.7%
	RGB-D	89.6%	90.2%	91.1%
YOLOv7-tiny	RGB	90.4%	90.2%	89.9%
	RGB-D	*92.8%*	*91.6%*	*91.8%*
YOLOv7-tiny-CTD	**RGB-D**	**94.9%**	**96.1%**	**95.7%**

Note: **Bold** is the best value, and *italic underlined* is the second-best value in each column.

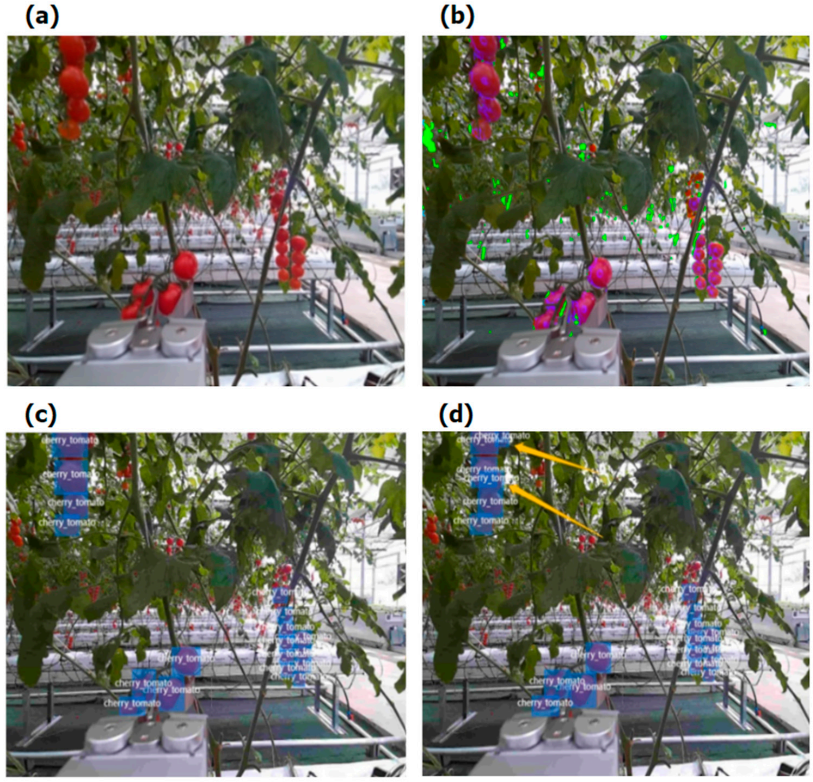

Figure 9. Cherry tomato detection by the YOLOv7-tiny and YOLOv7-tiny-CTD. (**a**) RGB image. (**b**) Point cloud normal vector angles added to (**a**), green areas indicate small, far away, or partially occluded cherry tomatoes, and purple areas large or close ones. (**c**) Detection output from the YOLOv7-tiny. (**d**) Detection output from the YOLOv7-tiny-CTD. The two orange arrows point to cherry tomatoes that are detected by the YOLOv7-tiny-CTD but missed by the YOLOv7-tiny.

3.4. Cherry Tomato Robot Picking Results

Cherry tomato robot picking experiments were carried out for five runs with the same number of image acquisitions of 80, and the number of picking actions performed by the cobot arm and the number of successful pickings were counted, respectively. Because the end effector may collide with other objects around the cherry tomatoes, the picking could fail even if the positioning is accurate. Therefore, the force sensor feedback was used to judge whether the picking was successful. In the case of the first failure, the second picking was carried out with the same positioning coordinates. The results are shown in Table 4.

Table 4. Cherry tomato picking success rates over five different runs, each with 80 image acquisitions and cherry tomato detections.

Trial	Num of Fruits with Picking Action	Num of Successes on 1st Attempt	Num of Successes on 2nd Attempt	Success Rate with Two Attempts
1	77	56	62	81%
2	78	57	63	81%
3	78	60	65	83%
4	72	61	63	87%
5	71	61	61	86%

4. Discussion

The proposed YOLOv7-tiny-CTD was tested in a commercial cherry tomato farming greenhouse with the harvesting robot. It obtains better cherry tomato detection under the transformed multimodal image inputs. Incorporating depth improves cherry tomato detection precision and accuracy for all deep learning models compared. And the YOLOv7-tiny-CTD outperforms all compared models without sacrificing efficiency. Combined with the eye-in-hand configuration and the force sensing feedback built into the custom end effector, robot pickings of cherry tomatoes were carried out in the greenhouse using very simple trajectory planning and grasping strategy. In 400 picking attempts at five different rack positions, the average success rate was 83.5%, indicating that the picking robot system has potential for application in a real commercial farming environment.

However, there are still many shortcomings in the harvesting robot used. For example, the robot picks one single fruit at a time, and the cobot arm picking movement is slow, thus the overall system efficiency is low. It is foreseeable that with a proper end effector design and more sophisticated picking strategy, cherry tomatoes may be harvested in bunches instead of individual ones as it may be desired in some cases. This needs a different detection strategy, but we think the multimodal perception scheme is likely to be of great help for cherry tomato bunch detection with some adjustment.

In addition, though the cherry tomato detection accuracy is high (about 95%), the picking success rate is about 10% lower even with a 2nd try allowed. The limitations are more on the hardware side. It should be noted that there are more sophisticated techniques available, such as humanoid end effectors [43–45], but at the moment the cost and complexities associated with such devices are prohibitive for practical fruit harvesting robot applications. The hardware cost of the robotic system in the current study is only about USD 28,000, comparable to the yearly labor cost of two workers in the commercial farm. The affordability is an important factor in practical adoptions of fruit picking robots.

In the future, we plan to improve the robotic system in several aspects: (1) design a new end effector to support both single picking and bunch picking to improve efficiency; (2) incorporate bunch detection into the neural network model; (3) increase the tactile sensor embedded in the fingertip of the end effector to achieve a more delicate visual–tactile-guided fruit picking; (4) develop a miniature and portable multispectral sensor to facilitate integration into the end effector to judge the maturity and status of cherry tomatoes before picking.

5. Conclusions

This study designed, integrated, and tested a cherry tomato detection scheme using a multimodal RGB-D sensor and an improved YOLOv7-tiny cherry tomato detection network named YOLOv7-tiny-CTD for a large greenhouse farming environment. In order to fully utilize the depth information in RGB-D images and improve cherry tomato detection performance, the RGB-D images are segmented using both color and depth to obtain cherry tomato ROIs, and a normal vector angle transformation of the point cloud is introduced. At the same time, the YOLOv7-tiny model is improved in multiple aspects. The color image input is replaced by a four-channel image containing the normal vector angle map. New Classness and hybrid NMS methods are also utilized.

The proposed method was tested using a harvesting robot in a commercial cherry tomato farming greenhouse. The experimental results show that the improved YOLOv7-tiny-CTD model obtains better cherry tomato detection under the multimodal image inputs. Incorporating depth improves cherry tomato detection precision and accuracy for all deep learning models compared. And the proposed YOLOv7-tiny-CTD outperforms all compared models without sacrificing efficiency, being able to run at 26 FPS on Nvidia Jetson TX1.

6. Patents

A patent application was filed with China National Intellectual Property Administration in 2023 (No. CN202310505684.X).

Author Contributions: Conceptualization, Z.Z. and Y.T.; methodology, B.C., H.D. and Y.T.; software, B.C. and H.D.; validation, Y.C. (Yingqi Cai), B.C. and Q.W.; formal analysis, B.C. and Y.C. (Yingqi Cai); investigation, D.L. and Y.C. (Yukang Cui); resources, H.D. and Y.T.; data curation, B.C. and H.D.; writing—original draft preparation, Y.C. (Yingqi Cai), B.C. and Q.W.; writing—review and editing, D.L., Z.Z., Y.C. (Yukang Cui) and Y.T.; visualization, Y.C. (Yingqi Cai) and Q.W.; supervision, Y.T.; project administration, H.D.; funding acquisition, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by Shenzhen University (No. 868/03010315 and 86902/00248).

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: B.C. and H.D. are with Litemaze Technology of Shenzhen, China, a 3D sensor manufacturer. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Bogue, R. Fruit picking robots: Has their time come? *Ind. Robot Int. J. Robot. Res. Appl.* **2020**, *47*, 141–145. [CrossRef]
2. Ceres, R.; Pons, J.L.; Jimenez, A.; Martin, J.; Calderon, L. Design and implementation of an aided fruit-harvesting robot (Agribot). *Ind. Robot Int. J.* **1998**, *25*, 337–346. [CrossRef]
3. Muscato, G.; Prestifilippo, M.; Abbate, N.; Rizzuto, I. A prototype of an orange picking robot: Past history, the new robot and experimental results. *Ind. Robot Int. J.* **2005**, *32*, 128–138. [CrossRef]
4. Scarfe, A.J.; Flemmer, R.C.; Bakker, H.; Flemmer, C.L. Development of an autonomous kiwifruit picking robot. In Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 10–12 February 2009; pp. 380–384.
5. Hua, X.; Li, H.; Zeng, J.; Han, C.; Chen, T.; Tang, L.; Luo, Y. A review of target recognition technology for fruit picking robots: From digital image processing to deep learning. *Appl. Sci.* **2023**, *13*, 4160. [CrossRef]
6. Pal, A.; Leite, A.C.; From, P.J. A novel end-to-end vision-based architecture for agricultural human–robot collaboration in fruit picking operations. *Robot. Auton. Syst.* **2024**, *172*, 104567. [CrossRef]
7. Chen, B.; Gong, L.; Yu, C.; Du, X.; Chen, J.; Xie, S.; Le, X.; Li, Y.; Liu, C. Workspace decomposition based path planning for fruit-picking robot in complex greenhouse environment. *Comput. Electron. Agric.* **2023**, *215*, 108353. [CrossRef]
8. Bulanon, D.M.; Kataoka, T.; Ota, Y.; Hiroma, T. AE—Automation and emerging technologies: A segmentation algorithm for the automatic recognition of Fuji apples at harvest. *Biosyst. Eng.* **2002**, *83*, 405–412. [CrossRef]
9. Payne, A.B.; Walsh, K.B.; Subedi, P.; Jarvis, D. Estimation of mango crop yield using image analysis–segmentation method. *Comput. Electron. Agric.* **2013**, *91*, 57–64. [CrossRef]

10. Senthilnath, J.; Dokania, A.; Kandukuri, M.; Ramesh, K.; Anand, G.; Omkar, S. Detection of tomatoes using spectral-spatial methods in remotely sensed RGB images captured by UAV. *Biosyst. Eng.* **2016**, *146*, 16–32. [CrossRef]
11. Luo, L.; Tang, Y.; Zou, X.; Wang, C.; Zhang, P.; Feng, W. Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components. *Sensors* **2016**, *16*, 2098. [CrossRef]
12. Teixidó, M.; Font, D.; Pallejà, T.; Tresanchez, M.; Nogués, M.; Palacín, J. Definition of linear color models in the RGB vector color space to detect red peaches in orchard images taken under natural illumination. *Sensors* **2012**, *12*, 7701–7718. [CrossRef] [PubMed]
13. Kurtulmus, F.; Lee, W.S.; Vardar, A. Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. *Precis. Agric.* **2014**, *15*, 57–79. [CrossRef]
14. Zhou, W.; Meng, F.; Li, K. A cherry tomato classification-picking Robot based on the K-means algorithm. *J. Phys. Conf. Ser.* **2020**, *1651*, 012126. [CrossRef]
15. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
16. Chen, J.; Wang, Z.; Wu, J.; Hu, Q.; Zhao, C.; Tan, C.; Teng, L.; Luo, T. An improved Yolov3 based on dual path network for cherry tomatoes detection. *J. Food Process Eng.* **2021**, *44*, e13803. [CrossRef]
17. Zheng, H.; Wang, G.; Li, X. YOLOX-Dense-CT: A detection algorithm for cherry tomatoes based on YOLOX and DenseNet. *J. Food Meas. Charact.* **2022**, *16*, 4788–4799. [CrossRef]
18. Yan, Y.; Zhang, J.; Bi, Z.; Wang, P. Identification and Location Method of Cherry Tomato Picking Point Based on Si-YOLO. In Proceedings of the IEEE 13th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Qinhuangdao, China, 11–14 July 2023; pp. 373–378.
19. Wang, C.; Wang, C.; Wang, L.; Wang, J.; Liao, J.; Li, Y.; Lan, Y. A lightweight cherry tomato maturity real-time detection algorithm based on improved YOLOV5n. *Agronomy* **2023**, *13*, 2106. [CrossRef]
20. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
21. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
22. Zhou, H.; Li, Z.; Ning, C.; Tang, J. Cad: Scale invariant framework for real-time object detection. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 760–768.
23. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]
24. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586.
25. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4507–4515.
26. Cui, B.; Zeng, Z.; Tian, Y. A Yolov7 cherry tomato identification method that integrates depth information. In Proceedings of the Third International Conference on Optics and Image Processing (ICOIP 2023), Hangzhou, China, 14–16 April 2023; Volume 12747, pp. 312–320.
27. Gursoy, E.; Navarro, B.; Cosgun, A.; Kulić, D.; Cherubini, A. Towards vision-based dual arm robotic fruit harvesting. In Proceedings of the IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 26–30 August 2023; pp. 1–6.
28. Wang, H.; Cui, B.; Wen, X.; Jiang, Y.; Gao, C.; Tian, Y. Pallet detection and estimation with RGB-D salient feature learning. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 17–19 November 2023; pp. 8914–8919.
29. Durmus, D. CIELAB color space boundaries under theoretical spectra and 99 test color samples. *Color Res. Appl.* **2020**, *45*, 796–802. [CrossRef]
30. Tian, Y. Dynamic focus window selection using a statistical color model. *Digit. Photogr. II* **2006**, *6069*, 98–106.
31. Serra, J.; Vincent, L. An overview of morphological filtering. *Circuits Syst. Signal Process.* **1992**, *11*, 47–108. [CrossRef]
32. Fabbri, R.; Costa, L.D.F.; Torelli, J.C.; Bruno, O.M. 2D Euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv. (CSUR)* **2008**, *40*, 1–44. [CrossRef]
33. Holzer, S.; Rusu, R.B.; Dixon, M.; Gedikli, S.; Navab, N. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 2684–2689.
34. Zia, S.; Yuksel, B.; Yuret, D.; Yemez, Y. RGB-D object recognition using deep convolutional neural networks. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 896–903.
35. Gené-Mola, J.; Vilaplana, V.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Gregorio, E. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. *Comput. Electron. Agric.* **2019**, *162*, 689–698. [CrossRef]
36. Eitel, A.; Springenberg, J.T.; Spinello, L.; Riedmiller, M.; Burgard, W. Multimodal deep learning for robust RGB-D object recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 681–687.

37. Guan, L.; Wang, F.; Li, B.; Tang, R.; Wei, R.; Deng, H.; Tian, Y. Adaptive automotive chassis welding joint inspection using a cobot and a multi-modal vision sensor. In Proceedings of the International Conference on Digital Economy and Artificial Intelligence, Shenzhen, China, 24–26 June 2024; pp. 841–849.
38. Jiang, J.; Luo, X.; Luo, Q.; Qiao, L.; Li, M. An overview of hand-eye calibration. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 77–97. [CrossRef]
39. Enebuse, I.; Foo, M.; Ibrahim, B.S.K.K.; Ahmed, H.; Supmak, F.; Eyobu, O.S. A comparative review of hand-eye calibration techniques for vision guided robots. *IEEE Access* **2021**, *9*, 113143–113155. [CrossRef]
40. Zhou, Q.; Zhang, W.; Li, R.; Wang, J.; Zhen, S.; Niu, F. Improved YOLOv5-S object detection method for optical remote sensing images based on contextual transformer. *J. Electron. Imaging* **2022**, *31*, 043049. [CrossRef]
41. Ren, S.; He, K.; Girshick, R.; Sun, J. *Faster R-CNN:* Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
42. Kim, S.; Na, S.; Kong, B.Y.; Choi, J.; Park, I.-C. Real-time SSDLite object detection on FPGA. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *29*, 1192–1205. [CrossRef]
43. Fukaya, N.; Toyama, S.; Asfour, T.; Dillmann, R. Design of the TUAT/Karlsruhe humanoid hand. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Takamatsu, Japan, 30 October–5 November 2000; Volume 3, pp. 1754–1759.
44. Parlikar, S.; Jagannath, V. Application of pneumatic soft actuators as end-effectors on a humanoid torso playing percussion instrument. In Proceedings of the 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 676–680.
45. Ramón, J.L.; Calvo, R.; Trujillo, A.; Pomares, J.; Felicetti, L. Trajectory optimization and control of a free-floating two-arm humanoid robot. *J. Guid. Control Dyn.* **2022**, *45*, 1661–1675. [CrossRef]

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