agronomy

MDPI

Article

Detection and Localization of Tea Bud Based on Improved YOLOv5s and 3D Point Cloud Processing

Lixue Zhu [1,*], Zhihao Zhang [1], Guichao Lin [1,2], Pinlan Chen [1], Xiaomin Li [1] and Shiang Zhang [3]

[1] School of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China; zhangzhihao1014@126.com (Z.Z.)
[2] Zhongkai Guangmei Research Institute, Meizhou 514700, China
[3] College of Innovation and Entrepreneurship, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
* Correspondence: zhulixue@zhku.edu.cn

Abstract: Currently, the detection and localization of tea buds within the unstructured tea plantation environment are greatly challenged due to their small size, significant morphological and growth height variations, and dense spatial distribution. To solve this problem, this study applies an enhanced version of the YOLOv5 algorithm for tea bud detection in a wide field of view. Also, small-size tea bud localization based on 3D point cloud technology is used to facilitate the detection of tea buds and the identification of picking points for a renowned tea-picking robot. To enhance the YOLOv5 network, the Efficient Channel Attention Network (ECANet) module and Bi-directional Feature Pyramid Network (BiFPN) are incorporated. After acquiring the 3D point cloud for the region of interest in the detection results, the 3D point cloud of the tea bud is extracted using the DBSCAN clustering algorithm to determine the 3D coordinates of the tea bud picking points. Principal component analysis is then utilized to fit the minimum outer cuboid to the 3D point cloud of tea buds, thereby solving for the 3D coordinates of the picking points. To evaluate the effectiveness of the proposed algorithm, an experiment is conducted using a collected tea image test set, resulting in a detection precision of 94.4% and a recall rate of 90.38%. Additionally, a field experiment is conducted in a tea experimental field to assess localization accuracy, with mean absolute errors of 3.159 mm, 6.918 mm, and 7.185 mm observed in the x, y, and z directions, respectively. The average time consumed for detection and localization is 0.129 s, which fulfills the requirements of well-known tea plucking robots in outdoor tea gardens for quick identification and exact placement of small-sized tea shoots with a wide field of view.

Keywords: the famous tea; tea buds; locating and detecting; YOLOv5; point cloud

check for updates

Citation: Zhu, L.; Zhang, Z.; Lin, G.; Chen, P.; Li, X.; Zhang, S. Detection and Localization of Tea Bud Based on Improved YOLOv5s and 3D Point Cloud Processing. *Agronomy* **2023**, *13*, 2412. https://doi.org/10.3390/agronomy13092412

Academic Editor: Silvia Arazuri

Received: 26 August 2023
Revised: 11 September 2023
Accepted: 17 September 2023
Published: 19 September 2023

Copyright: © 2023 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

Tea, being a naturally green beverage with a rich historical and cultural heritage, offers various micronutrients and possesses anti-aging properties, among other benefits [1]. In recent years, China has made significant efforts to cultivate and produce renowned teas, effectively boosting the income of tea farmers. According to statistical data, in 2021, Guangdong Province alone yielded 87,000 tons of tea annually, with famous tea accounting for 70% of the total output [2]. Currently, tea bud picking primarily relies on manual labor, which offers advantages such as strong selectivity, high accuracy, and minimal damage to tea leaves. However, this approach leads to increased labor intensity for tea farmers and significant labor costs due to the inefficiency associated with manual picking. Therefore, an essential challenge in intelligent tea picking lies in achieving precise and mechanized picking to prevent leakage, incorrect picking, and tea breakage through effective target detection of tea buds and accurate positioning of picking points.

In recent years, the application of vision-based automatic picking robots in the identification and picking of renowned tea has gained significant traction. In a recent study, Zhang

et al. [3] conducted image processing on collected tea samples to derive the R-component, G-component, and B-component. By applying a threshold value greater than 0 to the B-component of tea buds within the highlighted area, the differentiation between the old leaves and the tea buds was enhanced through the segmental linear transformation of the G-B component. Subsequently, the watershed function was employed for tea bud segmentation, resulting in an average segmentation precision of 95.79% and an overall segmentation precision of 94.26% across 100 samples. Xu et al. [4] proposed a two-stage fusion network detection and classification method. This method combines the fast detection ability of YOLOv3 with the high-precision classification capability of DenseNet201. Experimental results revealed a detection precision of 95.71% for tea buds captured from the side view, which was 10.60% higher than that for tea buds captured from the top view. Yang et al. [5] first trained and tested the R, G, and B components of young leaves and their backgrounds and then used gradient descent and Adam's algorithm to optimize the objective function. The results showed that the average accuracy of young leaf recognition was 92.62%, with an 18.86% misclassification rate. In a similar vein, Gui et al. [6] enhanced the YOLOv5 model by replacing the original convolution with the Ghostconv module. They also introduced the Bottleneck Attention Module (BAM) into the backbone network. The improved model demonstrated an average precision that exceeded the original YOLOv5 model by 9.66%. In addition, the enhanced model achieved a reduction of 52.402 G in floating-point operations and 22.71 M in parameters. Li et al. [7] employed Ghost Net as the backbone feature extraction network for YOLOv4, integrated the CBAM into PANet and introduced the SIoU loss function. This approach increased the precision of detecting a bud with a leaf/two leaves to 85.15%, an improvement of 1.08% over the original YOLOv4 network. Moreover, it reduced the average computational complexity by 89.11% and the number of parameters by 82.36%. Zhang et al. [8] proposed using MobileNetV3 as the backbone network of YOLOv4, replacing the original convolution with a depth-separable convolution and introducing a deformable convolutional layer and a coordinate attention module, and the experimental results show that under different lighting conditions, the detection accuracy, recall, and AP are 85.35%, 78.42%, and 82.12%, respectively. Zhang et al. [9] achieve the goal of reducing the model size by removing the focus layer and replacing the original feature extraction network of YOLOv5 with the ShuffleNetv2 algorithm, followed by channel pruning at the head of the neck layer, and the experimental results show that the detection speed can be up to 8.6 frames/second. These studies reveal that, based on color and morphological features, the initial separation of tea buds is accomplished through image processing methods. However, such methods are susceptible to factors including tea bud posture variations at different shooting locations, diverse light intensities, and color thresholds. The advent of deep learning has paved the way for the widespread utilization of semantic segmentation and target detection methods in tea bud recognition.

Currently, scholars have proposed various methods for tea bud-picking point positioning to achieve the automation of tea picking. Yang et al. [10] extracted the skeleton of tea bud images. Then, they solved the two-dimensional coordinates after determining the lowest point of the skeleton as the picking point location. Chen et al. [11] utilized Faster R-CNN to detect regions of interest in tea images, followed by the recognition of picking points applying FCN-16s. Experimental results showed that FCN achieved an average precision of 84.91%, with an IOU average precision of 70.72%. Li et al. [12] utilized a depth camera to acquire depth images. Then, depth images were fused with RGB images to obtain a 3D point cloud of the target area. By employing point cloud preprocessing, Euclidean clustering, and a target point cloud extraction algorithm, they determined the 3D harvesting positions of tea buds based on their growth characteristics, point cloud properties, and end-effector scheme. The tea bud localization experiment achieved an average positioning time of approximately 24 ms per bud. The team [13] achieved a localization success rate of 78.90% in a subsequent field experiment. Chen et al. [14] proposed a combination of skeleton extraction and minimum enclosing rectangle for localizing tea bud picking points. Experimental results showed an average depth localization error of

4.2 mm and a tea bud picking point precision rate of 83%. These studies indicate that machine vision and deep learning methods can achieve a two-dimensional localization of tea bud-picking points. However, due to variations and irregularities in the growth patterns of tea buds, depth information is required to realize the three-dimensional localization of tea bud picking points.

The current methods for shoot detection and small-size shoot localization within a large field of view exhibit common issues such as low precision and efficiency. These methods are unable to swiftly and accurately detect and localize tea buds, which is essential for meeting the efficiency and quality requirements of mechanized tea plucking. This study aims to address the detection and localization challenges associated with tea bud targets in unstructured tea garden environments. The proposed innovations are as follows: (1) To detect small-sized tea buds within a large field of view, a bidirectional feature pyramid network and a channel attention module are employed to construct a feature map with comprehensive semantic information. This facilitates the development of an enhanced YOLOv5 network model for target detection. (2) To address the localization problem of tea bud picking points, a 3D point cloud of tea leaf buds is extracted by generating a region of interest 3D point cloud based on detection results and point cloud clustering. Principal component analysis is utilized to fit the minimum outer cuboid of the 3D point cloud, with the bottom center of the cuboid serving as the 3D coordinates for the tea bud picking points.

This paper's succeeding sections are organized as follows: Section 2 outlines the fundamental theory and the novel methods used in this investigation. The third section examines the experimental test findings achieved in this investigation. Section 4 describes the study's findings. Finally, Section 5 brings the paper to a close.

2. Materials and Methods

2.1. Data Acquisition and Preprocessing

In this study, the Intel RealSenseD435 depth camera is employed to capture images of Yinghong No. 9 tea leaves under natural illumination within an angular range between 30° and 60°. The camera is positioned at a distance of 40–60 cm from the subject, operating at 30 FPS. The resulting images are saved in jpg format as RGB images with a resolution of 1280 × 720 pixels, while depth images were saved in png format. A total of approximately 50,000 tea leaf images were collected at the White Cloud Experimental Base of Guangdong Academy of Agricultural Sciences between 8:00 and 17:00 during the months of mid-March and mid-May in both 2021 and 2022.

To minimize the impact of harvesting on tea tree growth, preserve the tea trees' inherent growth potential, and facilitate the continuous sprouting of new tea buds, this study seeks to identify a single bud and a single leaf that retain the characteristics of healthy leaves, as demonstrated in Figure 1. During the process of data acquisition, the automatic continuous shooting method, which captures images at a rate of 30 FPS, results in consecutive images exhibiting similar content. Additionally, the captured images are susceptible to variations in natural lighting conditions, leading to image exposure issues. Accordingly, the collected tea leaf images are unsuitable for direct usage. Therefore, a subset of appropriate images is selected and subsequently annotated, yielding a total of 1226 images that meet the established harvesting criteria. To enhance the diversity of the experimental dataset, local transformations are applied to the image dataset, simulating the growth state of tea leaves in a natural environment. These transformations include horizontal flipping, brightness adjustments, and the introduction of Gaussian noise. Through this process, the number of samples is expanded [15], resulting in a tea image dataset including a total of 3678 images. Among these, 2944 images are allocated for the training set, while the remaining 734 images are designated for the testing set.

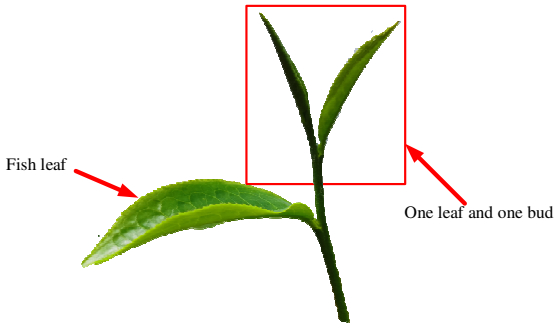

Figure 1. Picking targets in this study: A bud and a leaf.

2.2. Methodology

This study focuses on shoot detection issues across a wide field of view and the localization of picking points for small-sized tea buds. The workflow employed to solve this problem is illustrated in Figure 2. Firstly, the RGB images and depth images of the tea buds are acquired using a depth camera. Subsequently, the RGB images are fed into the target detection network for identification, and the resulting detection results are fused with the corresponding depth images to generate a 3D point cloud representing the region of interest. The 3D point cloud derived from the clustering process is then utilized to determine the picking points of the tea buds by applying cloud principal component analysis.

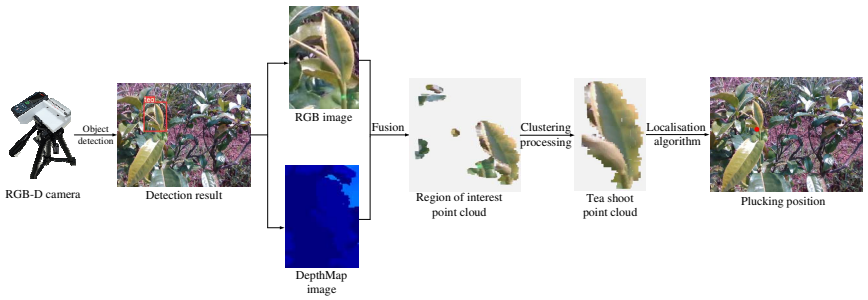

Figure 2. Detection and localization process of tea buds. The red dot in the last image is the picking point position generated by the algorithm and projected over the color image.

2.2.1. Tea Bud Detection

The Detection of tea buds in large, unstructured tea garden environments presents challenges, including dense growth, similarity in color between the target and the background, and occlusion resulting from overlapping foliage. These factors reduce the distinctiveness of tea bud features, leading to incomplete detection results by existing models. To ensure the operational efficiency of the tea-picking robot and fulfill the real-time detection requirements within the constraints of the edge computing server device, this study adopts the YOLOv5s network model. The chosen model exhibits a minimal model file and comprises four main components: the input end, the backbone network, the neck network, and the detection head [16]. The neck network incorporates PANet [17], which is susceptible to a limitation involving a sole top-down path and a solitary bottom-up path, where the features from individual input edges are not effectively fused and contain minimal information. Meanwhile, the feature extraction process in YOLOv5, accomplished through convolutional layers, is prone to information loss and redundancy across feature maps. Therefore, to further enhance the detection performance of small-sized tea buds within a wide field of view, the introduction of a lightweight channel attention module, namely ECANet [18], and a bidirectional feature pyramid structure (BiFPN) [19] is considered. These additions enable better extraction and utilization of multi-scale feature information, allowing the model to focus on tea bud characteristics and enhance the precision of tea bud detection. Therefore, the network becomes more suitable for embedded devices and mobile terminals.

The improved model is denoted as YOLOv5s-Tea, and its network structure is depicted in Figure 3.

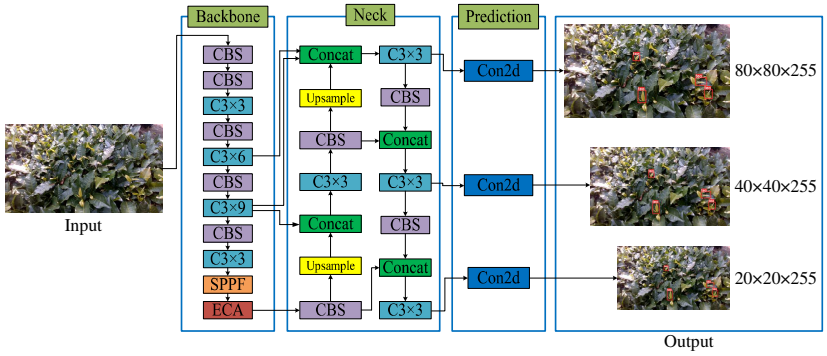

Figure 3. YOLOv5s-Tea Chart.

1. Bidirectional feature pyramid network structure

To enhance feature fusion performance, this study incorporates a BiFPN to replace the original PANet structure. Firstly, nodes with only one input edge or minimal contributions to feature fusion are eliminated, and then extra connections are established between input and output nodes within the same layer. Subsequently, each bidirectional (top-down and bottom-up) path is treated as a feature network layer and repeated multiple times within the same layer [20]. Finally, a weight is assigned to the fused features at each scale, and the weighted feature fusion is conducted using fast normalized fusion, as shown in Equation (1), yielding the final weighted bidirectional feature pyramid network.

$$o = \sum_i w_i \cdot I_i \tag{1}$$

Figure 4 illustrates a comparison between the PANet structure and the BiFPN structure, and level 6 is taken as an example in Equations (2) and (3).

$$P_6^{td} = conv\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot \mathrm{Resize}(P_7^{in})}{w_1 + w_2 + \varepsilon}\right) \tag{2}$$

$$p_6^{out} = conv\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot \mathrm{Resize}(P_5^{out})}{w_1' + w_2' + w_3' + \varepsilon}\right) \tag{3}$$

where P_6^{td} is the intermediate feature at level 6 on the top-down pathway, and $P_6^{out\ t}$ is the output feature at level 6 on the bottom-up pathway.

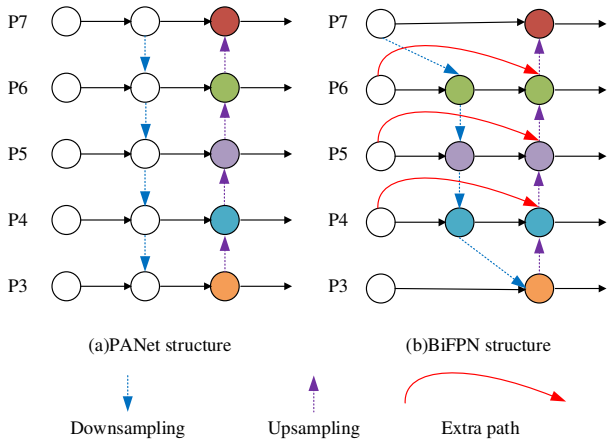

Figure 4. Comparing the structures of PANet structure and BiPFN structure. Circles of the same color represent the same hierarchy's input, intermediate, and output nodes.

2. Feature fusion network of channel attention mechanism

To enhance the focus of the network on target detection and improve the detection efficacy, an Efficient Channel Attention (ECA) model is incorporated into the backbone network [1]. The input feature map is first globally average-pooled, converting its dimensions from a matrix [h, w, c] to a vector [1, 1, c]. This vector is then utilized to calculate the size, k, of the adaptive convolution kernel, as shown in Equation (4):

$$k = \varphi(c) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd}, \tag{4}$$

where k represents the convolution kernel size; C represents the number of channels; odd represents that k can only take an odd number; γ and b are used to change the ratio between the number of channels, C, and the convolution kernel size, k.

Subsequently, a 1×1 convolution is performed to acquire channel weight vectors in the feature map. These vectors undergo processing through a Sigmoid activation function and are multiplied channel-by-channel with the input feature map, yielding a weighted feature map. The weighted feature map is then scaled and panned to generate the final feature map [21]. The specific structure of the ECANet module is depicted in Figure 5.

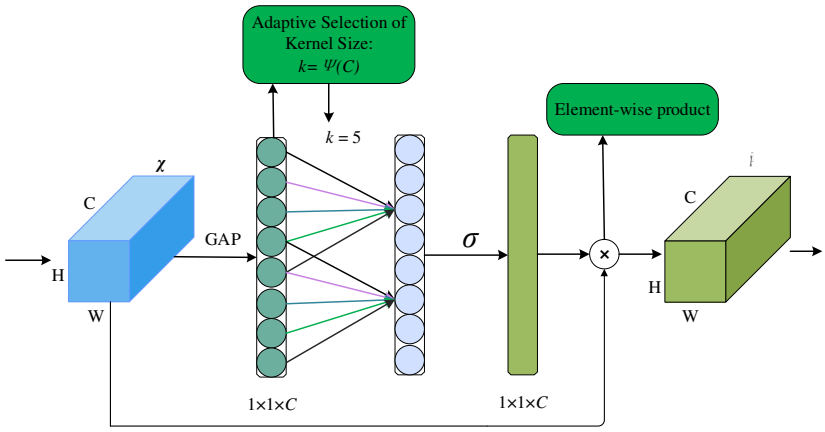

Figure 5. ECANet Module Specific Structure. The different colored arrows represent the process of interacting information across channels, while the light blue circles represent the feature map's channel weights.

The hardware configurations employed in the model training platform used for this study consist of an Intel(R) Xeon(R) Gold 5218 CPU operating at a frequency of 2.30 GHz, along with a Quadro RTX 5000 graphics card equipped with 16 G video memory. The Ubuntu18.04 operating system is installed, accompanied by CUDA10.2 and CUDNN8.6.0 configurations. Network training is conducted within the Anaconda virtual environment pre-installed in the system, utilizing PyTorch1.7.1, paddlepaddle-gpu2.4.1, openCV4.6, Open3D, and other related libraries. Python3.7 serves as the programming language of choice. The training phase involves employing the data-enhanced Tea Tree dataset, followed by the verification of the model's stability and reliability using the testing set. The training parameters are set as follows: an initial learning rate of 0.01, a Batch Size of 32, a Momentum of 0.937, a Decay of 0.0005, an IOU threshold of 0.05, and enhancement coefficients of 0.015, 0.7, and 0.4 for Hue (H), Saturation (S), and value (V) respectively, totaling 300 rounds (Epochs).

2.2.2. Localization Method of Picking Point

Building upon the aforementioned information, this study proposes a 3D localization method for identifying tea bud picking points. Initial results obtained from the YOLOv5s-Tea model are combined with depth images to generate a 3D point cloud for the designated

area. Subsequently, DBSCAN clustering is applied to the 3D point cloud in order to extract the relevant tea bud points. The minimum outer cuboid of the tea bud 3D point cloud is then fitted to accurately locate the tea bud picking points. Finally, the bottom center point of the cuboid is identified as the picking point location, enabling the realization of 3D positioning for the picking points.

(1) Point cloud generation. Firstly, the region of interest (ROI) pertaining to the tea buds is isolated from the input RGB and depth maps. Subsequently, the 3D coordinates of each pixel point in the camera coordinate system are computed based on the internal parameters of the depth camera, the pixel values from the RGB images, and the depth value information extracted from the depth images, as presented in Equation (5) below.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{5}$$

where $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ represents the coordinate of the pixel point; $\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ represents the 3D coordinate point in the camera coordinate system; z_c represents the depth value; f_x and f_y represent the focal length of the cameras; c_x and c_y represent the optical center of the camera.

(2) Tea bud point cloud clustering. The 3D point cloud data of the region of interest comprises the tea buds and the surrounding environment, tea trees, and additional background point clouds. In this study, the Density-Based Spatial Clustering of Applications with Noise [22] method is employed to cluster the point cloud data and differentiate between the data from the target 3D point cloud and other 3D point clouds. It is assumed that the 3D point clouds of tea buds exhibit concentration and completeness within the area of interest. Firstly, an empty set of cluster groups is initialized. Thereafter, the distances between points are calculated, as shown in Equation (6), and points with distances less than or equal to eps are identified based on the eps and distance matrix.

$$d = \sqrt[2]{(x_{core} - x_{border})^2 + (y_{core} - y_{border})^2 + (z_{core} - z_{border})^2} \tag{6}$$

where x_{core}, y_{core}, and z_{core} are the abscissa and the ordinate of the core point, respectively, and x_{border}, y_{border}, and z_{border} are the abscissa and ordinate of the border point, respectively.

Subsequently, a core point is randomly chosen from the space, and its neighborhood is assessed to determine if it meets the density criteria. If the density requirements are met, the neighborhood points are iteratively grouped into the same cluster until all points within the core point's neighborhood have been accessed. The process then continues by selecting the next unvisited core point and performing the same operation recursively. Boundary points within the cluster group, where the nearest core point is situated, are considered noise points if they do not belong to any cluster group [23]. This method enables the detection of arbitrary shape classes without requiring a predetermined number of cluster groups and effectively handles noisy point data, as illustrated in Figure 6. The effect of 3D point cloud clustering is shown in Figure 7.

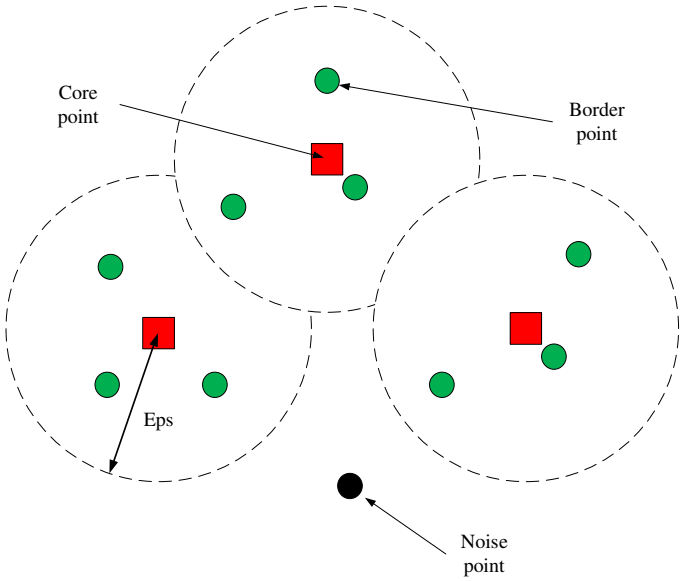

Figure 6. Schematic diagram of the DBSCAN algorithm. A cluster group is represented by each dotted circle.

Figure 7. Clustering Flowchart.

(3) Picking point determination Utilizing the 3D point cloud of tea buds after DBSCAN clustering in the previous step, Principal Component Analysis [24] is applied to analyze the tea bud 3D point cloud and obtain the minimum outer cuboid enclosing the point cloud. Firstly, the centroid coordinate of the tea bud 3D point cloud is calculated using Equation (7).

$$m = \frac{1}{N}\sum_{i=1}^{n} P_i \tag{7}$$

where m represents the centroid coordinate of the 3D point cloud, n represents the number of 3D point clouds, and P_i represents the 3D coordinate of the ith point.

Subsequently, the covariance matrix C is derived from the centroid coordinate, as presented in Equation (8).

$$C = \frac{1}{N}\sum_{i=1}^{n} (P_i - m)(P_i - m)^T \tag{8}$$

where C represents the covariance matrix C of the 3D point cloud.

To perform eigenvalue decomposition of the covariance matrix C, Equation (9) is employed to obtain the eigenvalues $\lambda = (\lambda 1, \lambda 2, \lambda 3)$ and the corresponding eigenvectors $v = (v1, v2, v3)$ [25].

$$Cv = \lambda v \tag{9}$$

where the eigenvector v represents the principal component.

Thereafter, the eigenvectors are sorted based on the corresponding eigenvalues, resulting in the three principal component directions: X, Y, and Z. Each point in the point cloud is projected onto these sorted eigenvectors. The projection lengths of the point cloud in the three directions are denoted as l_1, l_2, and l_3, and l_1 is taken as an example, as shown in Equation (10).

$$l_1 = \min_{1 \le i \le N} \{P_i \cdot X\} + \max_{1 \le i \le N} \{P_i \cdot X\} \tag{10}$$

where min and max are the minimum and maximum values in the x, y, and z directions, respectively.

By integrating the centroids and eigenvectors, the coordinates of the eight vertices of the smallest cuboid can be calculated. Finally, the coordinates of the four points with the smallest y-coordinate are averaged to obtain the coordinates of the center of the base.

2.2.3. Experimental Methodology

The detection and localization algorithm utilized in this study is carried out on a TW-T600 Xavier edge computing server equipped with an 8-core ARM v8.2 64-bit CPU, a 512-core Volta with Tensor cores, and 32 GB of RAM. The tea bud detection algorithm is derived from the OpenCV open-source computer vision library, while the picking point localization algorithm is implemented using the Open3D point cloud library, programmed in Python, and deployed on the edge computing server. (1) Detection experimental method for tea buds to assess the detection performance of the YOLOv5s-Tea model, six metrics are employed: Precision (P), Recall (R), Average Precision (AP), Param, Weight Size, and frames per second (fps). The Precision, Recall, AP, and Param are computed according to the Equations (11)–(14) presented below.

$$IoU = \frac{A \cap B}{A \cup B} \tag{11}$$

$$P = \frac{TP}{(TP + FP)} \times 100\% \tag{12}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{13}$$

$$AP = \int_0^1 P(R) dR \tag{14}$$

In the aforementioned formulas, A denotes the area of the detected bounding box, while B represents the area of the actual bounding box. In Equations (8) and (9), TP corresponds to the number of accurately detected tea buds, FP signifies the number of falsely detected tea buds, and FN represents the number of missed tea bud detections. Weight Size indicates the memory space required to store the model in Mb, and fps denotes the rate at which images are processed per second [26]. The intersection and union ratio IoU ≥ 0.5 signifies a true case, IoU < 0.5 indicates a false positive case, while IoU = 0 expresses a false negative case.

(1) Feature Information Analysis Experiment

A gradient-weighted class activation mapping (Grad-CAM) heat map was employed as an analytical tool to evaluate the performance of several structural YOLOv5s models for the task of tea shot detection. Grad-CAM [27] is a class-discriminative localization strategy for any CNN-based network that outputs visual interpretations. Firstly, given an image and a category of interest tea, forward propagate the model through the CNN component to produce task-specific category scores y. Then, set the specified category gradient tea to 1 and all other gradients to 0. Then, apply the supplied category scores y^c to the convolutional feature maps, combine the calculations to produce coarse gradient-CAM localizations (the blue heatmaps), and lastly, multiply the results with the heatmaps to obtain high-resolution Grad-CAM visualizations. Finally, the heat map is dot-multiplied with the backpropagation

data to provide a high-resolution customized Grad-CAM visualization. The calculation of Grad-CAM is shown in Equation (15):

$$L^c_{Grad-CAM} = \mathrm{ReLU}(\sum_k \alpha^c_k A^k) \tag{15}$$

where A is the feature layer to be shown, and the output of the final convolutional layer is commonly chosen; k is the kth channel in feature layer A; and c is the category c. α^c_k represents the weight of the pin category c on the kth channel of the feature layer A; A^k represents the weight matrix on the kth channel of the feature layer A; ReLU: makes the final output result greater than zero and suppresses the uninteresting weights.

The calculation about α^c_k is shown in Equation (16):

$$\alpha^c_k = \frac{1}{z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \tag{16}$$

where y^c is the score achieved via forward propagation for category c; A^k_{ij} is the data with coordinates (i, j) on the kth channel of feature layer A; and Z is the product of width i and height j. $\frac{\partial y^c}{\partial A^k_{ij}}$ represents the gradient information of category c acquired by backpropagation on feature layer A; the calculated gradients are pooled globally and averaged over the dimensions of width i and height j to get the important weight α^c_k.

The shade of the color indicates the degree of attention the model pays to the tea shoots in this visualisation, with darker red spots signifying greater attention from the model.

(2) Experimental method of tea bud picking point localization

The field experiment for tea bud picking point localization primarily involves a GLM50-23G laser range finder (Bosch, precision ± 1.5 mm), an Intel RealSense D435 depth camera, a tripod, and the edge computing server, as depicted in Figure 8. The depth camera captures depth information images with a resolution of 640 × 480 pixels. Complying with Intel's specified minimum depth detection distance of 175 mm for the RealSense D435, the distance between the depth camera and the tea buds is maintained at 200–300 mm, satisfying the localization requirements. In addition, the depth camera establishes USB communication with the edge computing server, and its installation angle relative to the horizontal plane ranges from 45°–60°, minimizing occlusion issues among the tea buds.

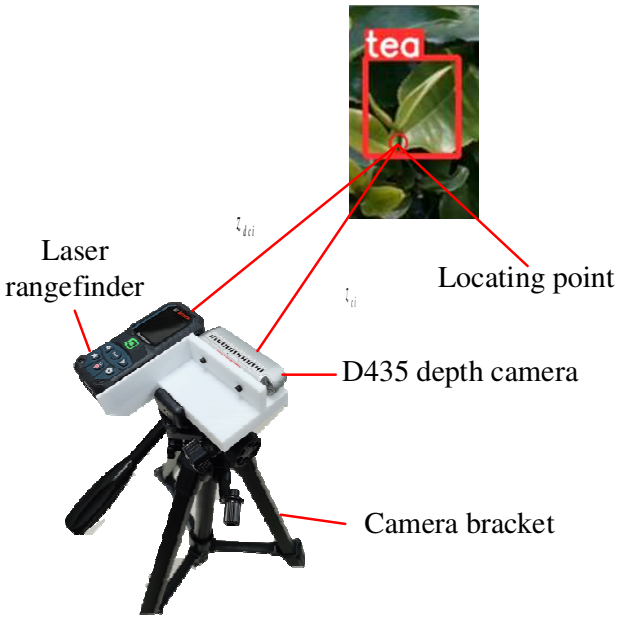

Figure 8. Graph of distance measurement.

To analyze localization errors of the picking point primarily originating from the X, Y, and Z directions, a test is conducted on the localization precision of tea buds' picking points, specifically focusing on the smallest external cuboid surrounding the buds. To determine the coordinates of the laser range finder relative to the depth camera within the coordinate system, a positional calibration is performed for both devices in a laboratory setting. A *9-square black and white checkerboard grid calibration plate is prepared, featuring small square grid sizes of 3030 mm. The normal direction of the calibration plate plane aligned with the Z-axis, positioned at a distance of 400 mm from the depth camera. The laser range finder and depth camera are mounted on the plane of a 3D-printed component. The following specific coordinate measurement procedure is executed: Firstly, the laser beam of the range finder is directed toward the intersection of the black and white squares on the calibration plate. Subsequently, the ranging code retrieves the coordinates (X′, Y′, Z′) of the point in the depth camera coordinate system. The X and Y coordinates represent the horizontal coordinates of the laser range finder (X, Y*), while the depth coordinate Z* is obtained by reading the distance from the calibration plate interface on the range finder. The horizontal coordinates of the laser range finder are obtained using the following method: The pixel coordinates of the points hit by the laser range finder, corresponding to the depth, are first mapped by aligning the depth images of the depth camera with the RGB images. Subsequently, the coordinates are calculated by combining them with the conversion matrix of the depth camera's internal and external parameters. The field experiment took place in June 2023 at the tea test base of the Digital Agriculture Demonstration Park, Baiyun Campus, Zhongkai Agricultural Engineering College, as depicted in Figure 9. The experimental subject is the Yinghong No. 9 tea variety. Firstly, a localization algorithm is applied to determine the locations of tea bud picking points on different tea trees within the tea garden, utilizing the Intel RealSense D435 depth camera. Then, the distance from the depth camera's plane to the tea stem of the tea buds is measured three times using a laser range finder. The average value of these measurements represents the imaging distance of the outer surface of the tea stems. Additionally, the average value of three measurements of the tea buds' tea stem diameter, obtained using a vernier caliper, is considered as the final diameter value, as shown in Figure 10.

(a) (b)

Figure 9. Trial location for tea. (**a**) Trial Tea Plantation, (**b**) Tea leaf shoots on the canopy's surface.

(a) (b)

Figure 10. Schematic diagram of the measurement. (**a**) Distance measurement, (**b**) Diameter measurement.

3. Results

3.1. Tea Bud Detection Experiment

To demonstrate the effectiveness of the model improvement, the datasets generated in this study are employed for training and obtaining the optimal model. Subsequently, these models are deployed to edge computing servers for test set testing. A comprehensive comparison of the YOLO versions (specifically YOLOv5s, YOLOv7, and YOLOv8s) with the Ours model for a comprehensive comparison. Table 1 shows that our model surpasses the others in terms of detection accuracy and speed. When compared to the original YOLOv5, ours improves in all elements of detection performance. With an AP of 94.58%, the accuracy is 94.4% and increased by 0.69%. The addition of the BiFPN and ECANet modules enhances the model's complexity and size by introducing new procedures and parameters. However, the model detected at 37.139 fps, which is roughly 22.5% faster.

Table 1. Test Set Results.

Network	P (%)	R (%)	AP (%)	Weight Size	Param (M)	fps [1]
YOLOv5s	93.71	89.78	94.36	13.6	7,012,822	30.329
YOLOv5s + bifpn	94.08	89.72	94.58	13.8	7,078,367	34.896
YOLOv7	92.7	89.7	90.5	298.4	36,481,772	14.12
YOLOv8s	93.7	89	94.4	89.6	11,125,971	24.94
Ours	94.40	89.66	94.71	13.8	7,078,370	37.139

[1] Calculate the inference time required to iterate 100 rounds.

Figure 11 compares our results to the original YOLOv5s ablation detection trials, where the quantity of tea shoots, distance, and light intensity are varied in each of the eight photos. The incorporation of BiFPN and ECANet into YOLOv5 increases the model's feature extraction and fusion capabilities, improves detection performance for obstructed or tiny objects, and decreases missed detections even more.

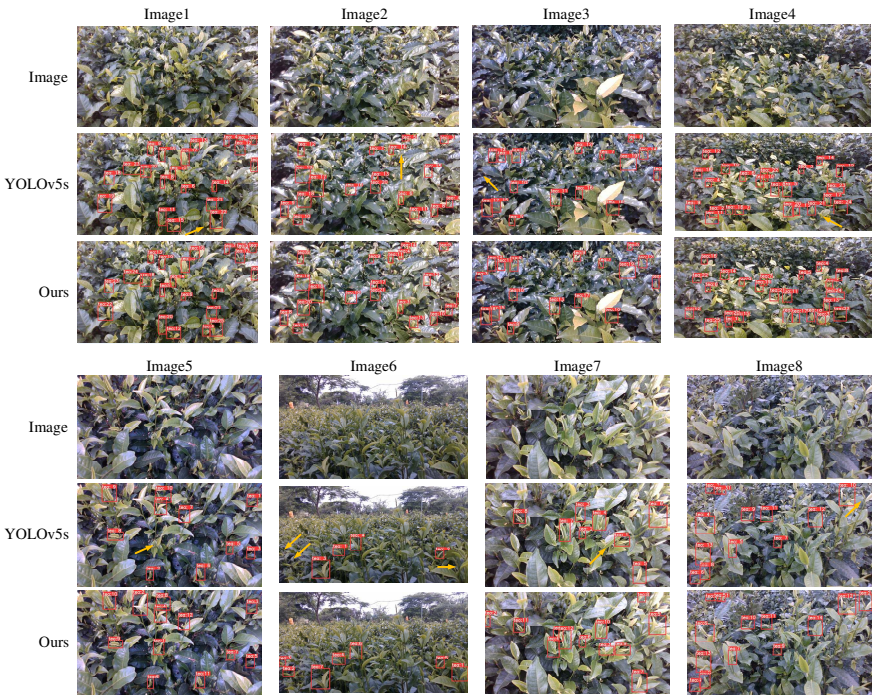

Figure 11. Comparison of detection results.

We used the Grad-CAM technique to visually compare the feature maps of YOLOv5 and Ours to show the correctness of the Ours model. The outcomes of YOLOv5 and Ours

on various degrees of feature production are shown in Figure 12. The figure shows that by efficiently filtering the background data and optimizing the computing resources for the detection of tea shoots, ours beats the YOLOv5 model.

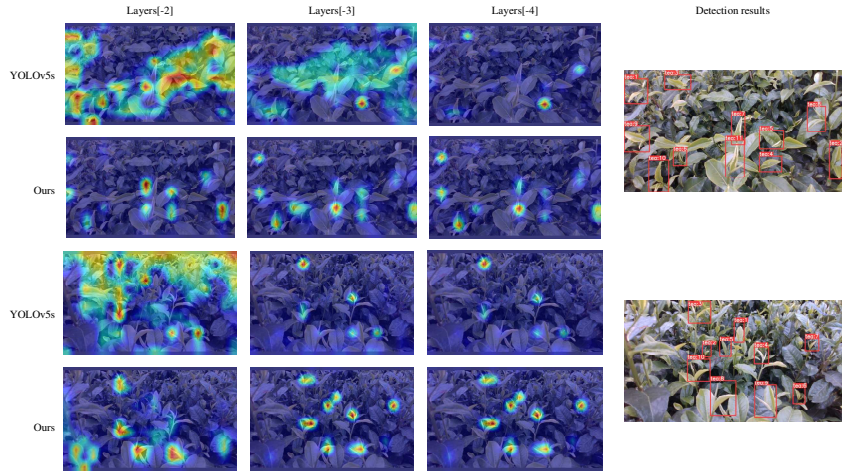

Figure 12. Grad-CAM comparison result.

3.2. Tea Bud Picking Point Localization Experiment

First, the distance between the depth camera imaging camera and the laser rangefinder was measured using a straightedge, providing a value of approximately 43 mm in the X-direction and about 1 mm in the Y-direction. The results of the experiment involving the calibration of the laser range finder and depth camera positions are presented in Table 2. These results illustrate that, under the coordinate system of the depth camera, the average distances recorded in the X-direction, Y-direction, and Z-direction by the laser range finder are 43.632 mm, 1.346 mm, and 241.967 mm, respectively. The corresponding standard deviations are 0.674 mm, 0.349 mm, and 13.497 mm. Considering an average measurement distance of 236 mm and a standard deviation of 13.520 mm for the laser range finder, the distance between the depth camera's imaging position and the laser range finder's beam launching position is calculated to be approximately 5.967 mm and the distances in the X and Y directions are within the uncertainty of the straightedge measurements. Table 3 showcases the measured values for the field localization of the tea bud picking points, including their errors. The mean values of the picking points' three-dimensional coordinates in the X, Y, and Z directions are 43.101 mm, 8.207 mm, and 288.291 mm, respectively, and the average absolute errors calculated with the calibration results in Table 2 are found to be 3.159 mm, 6.918 mm, and 7.185 mm, respectively. The corresponding standard deviations are 10.763 mm, 1.899 mm, and 2.759 mm. The result is shown in Figure 13. Moreover, the average time consumed for target detection during the field experiment is 0.042 s, the average time for the localization process is 0.087 s, and the overall average time for the complete tea bud detection and localization process is 0.129 s.

Table 2. Position calibration results of laser rangefinder and depth camera.

Result	X′/mm	Y′/mm	Z′/mm	Rangefinder Results Z*/mm
Average value	−43.632	1.346	241.967	236
Standard deviation	0.674	0.349	13.497	13.520

Table 3. Measured values of tea shoot picking point positioning and their errors.

Image No.	Positioning Results			Rangefinder Results	Positioning Error Results		
	X	Y	Z		$\|X\text{-}X'\|$/mm	$\|Y\text{-}Y'\|$/mm	$\|Z\text{-}Z^*\|$/mm
Average value	−43.101	8.207	288.291	295.475	3.159	6.918	7.185
Standard deviation	11.205	2.097	32.493	31.899	10.763	1.899	2.759

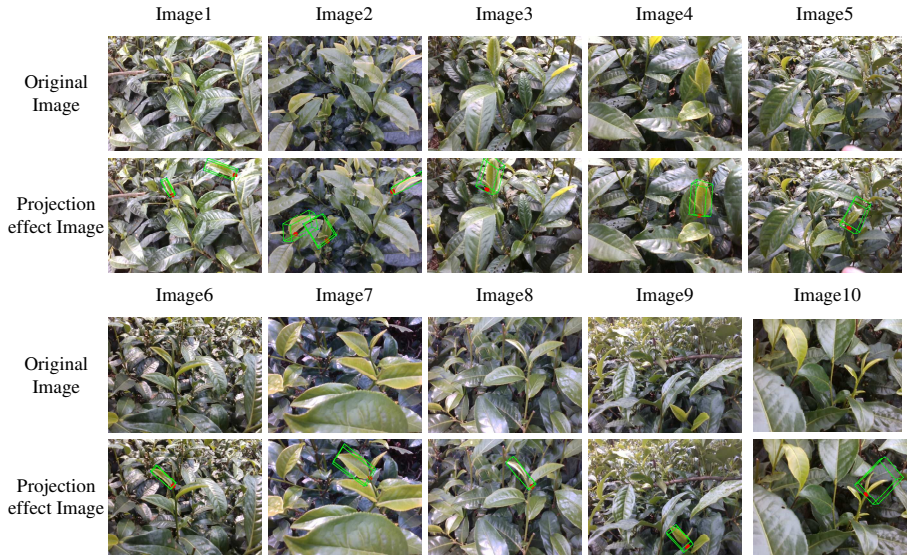

Figure 13. Picking position for tea shoots and the effect of the minimal exterior rectangle. The green box in the image represents the projection impact of the minimal exterior rectangle created by the 3D point cloud fitting, and the red dot represents the projection effect of the algorithm's choosing point.

4. Discussion

4.1. Experimental Analysis of Tea Bud Detection

This study evaluates the precision and real-time performance of the enhanced YOLOv5s-Tea network for tea bud detection. The findings indicate that the incorporation of ECANet and BiFPN yielded superior results. To begin, ECANet can assist the model in better focusing on the correlation between different channels in the image and extracting more discriminative features by weighting the features of different channels in the image, allowing it to better distinguish tea shoots from non-tea shoots and improve detection accuracy [28]. Second, tea shoots can appear at various scales, and their size in the photograph can vary substantially. The BiFPN module can perform feature fusion and scale adjustment efficiently by fusing features at different scales utilizing top-down and bottom-up feature propagation, allowing the model to perform target identification at multiple scales and increase its shot target detection capacity [29]. As a consequence, in a wide field of view, tea shoots seem identical to non-tea shoots, and the introduction of ECAnet and BiFPN modules in YOLOv5 allows the network to comprehend the target more fully and precisely, potentially improving the model's resilience and stability.

4.2. Locating Experimental Analysis of Tea Bud Picking Points

During the field experiment, the depth information is successfully obtained by most of the tea buds, and the 3D point cloud is accurately acquired and localized. However, a few picking points exhibit positioning errors. Analyzing only the point cloud detected within the target area, rather than the entire scene's point cloud, reduces the amount of data used for processing. Therefore, the speed of the tea-picking robot in localizing the tea bud-picking points improves significantly. The analysis of the experimental data presented in Table 3 reveals that the visual localization errors primarily originated from the following factors: (1) The presence of small tea buds and other distractions such as tea buds, tree

branches, and tea leaves in the complex background hinders the correct extraction of the tea bud's 3D point cloud after clustering. (2) The intense direct light on the tea buds, along with the depth camera, tends to overexpose the tea bud images and interfere with the infrared structured light emitted by the depth camera [30]. As a result, the fusion of depth and RGB images is compromised, leading to the loss of point clouds for some tea buds and introducing errors in the horizontal direction during the calculation of the minimum outer cuboid [31]. (3) The experimental method utilized in this study excludes a few laser range finder points from the target detection identification box. This results in a systematic error between the coordinates of the picking points solved by the picking point localization algorithm and the calibrated coordinates of the laser range finder.

5. Conclusions

In order to achieve the detection of small-sized tea shoots and picking point localization under a large field of view, this study developed an algorithmic system containing the detection of tea shoots and picking point localization, which includes the YOLOv5s target detection method, generation of the 3D point cloud in the region of interest, clustering of the 3D point cloud, fitting and generation of the minimum outer rectangle, and 3D localization of picking points. Field tests were used to assess the method's performance, and the following results were reached:

(1) The results show a tea bud detection precision of 94.4% and a recall rate of 90.38%. This indicates the effective applicability of the method in identifying tea buds within small targets and complex environments.

(2) The average absolute errors in the X-direction, Y-direction, and Z-direction are found to be 3.159 mm, 6.918 mm, and 7.185 mm, respectively. This allows tea shoots to be positioned in three dimensions.

(3) The overall average time for the complete tea bud detection and localization process is 0.129 s.

In conclusion, the method is robust to tea shoot detection and localization in an unstructured tea garden environment, and the algorithm has high detection and localization accuracy and reasonable time consumption for the Famous Tea Picking Robot, which can be effectively applied to the Famous Tea Picking Robot's real-time picking work in hilly and mountainous areas. Future research will primarily concentrate on localizing tea buds in scenarios involving overlapping and obstruction.

Author Contributions: Conceptualization, Z.Z. and P.C.; methodology, Z.Z.; software, Z.Z.; validation, Z.Z.; formal analysis, Z.Z.; investigation, Z.Z.; resources, Z.Z.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, L.Z.; visualization, X.L.; supervision, L.Z.; project administration, S.Z.; funding acquisition, Z.Z and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2022 Guangdong Science and Technology Innovation Strategy Special Funds (Grant pdjh2022b0249), and Science and Technology Program of Meizhou, China (Grant No. 2021A0304004).

Data Availability Statement: Data recorded in the current study are available in all tables and figures of the manuscript.

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

References

1. Zhang, D.-Y.; Zhang, W.; Cheng, T.; Zhou, X.-G.; Yan, Z.; Wu, Y.; Zhang, G.; Yang, X. Detection of wheat scab fungus spores utilizing the Yolov5-ECA-ASFF network structure. *Comput. Electron. Agric.* **2023**, *210*, 107953. [CrossRef]
2. Department of Agriculture and Rural Affairs of Guangdong Province. Analysis of Tea Production and Marketing Situation in Guangdong Province in 2021. Available online: http://dara.gd.gov.cn/gkmlpt/content/3/3802/mmpost_3802733.html3045 (accessed on 29 January 2022).

3. Zhang, L.; Zou, L.; Wu, C.; Jia, J.; Chen, J. Method of famous tea sprout identification and segmentation based on improved watershed algorithm. *Comput. Electron. Agric.* **2021**, *184*, 106108. [CrossRef]
4. Xu, W.; Zhao, L.; Li, J.; Shang, S.; Ding, X.; Wang, T. Detection and classification of tea buds based on deep learning. *Comput. Electron. Agric.* **2022**, *192*, 106547. [CrossRef]
5. Yang, J.; Chen, Y. Tender Leaf Identification for Early-Spring Green Tea Based on Semi-Supervised Learning and Image Processing. *Agronomy* **2022**, *12*, 1958. [CrossRef]
6. Gui, Z.; Chen, J.; Li, Y.; Chen, Z.; Wu, C.; Dong, C. A lightweight tea bud detection model based on Yolov5. *Comput. Electron. Agric.* **2023**, *205*, 107636. [CrossRef]
7. Li, J.; Li, J.; Zhao, X.; Su, X.; Wu, W. Lightweight detection networks for tea bud on complex agricultural environment via improved YOLO v4. *Comput. Electron. Agric.* **2023**, *211*, 107955. [CrossRef]
8. Zhang, Z.; Lu, Y.; Zhao, Y.; Pan, Q.; Jin, K.; Xu, G.; Hu, Y. TS-YOLO: An All-Day and Lightweight Tea Canopy Shoots Detection Model. *Agronomy* **2023**, *13*, 1411. [CrossRef]
9. Zhang, S.; Yang, H.; Yang, C.; Yuan, W.; Li, X.; Wang, X.; Zhang, Y.; Cai, X.; Sheng, Y.; Deng, X.; et al. Edge Device Detection of Tea Leaves with One Bud and Two Leaves Based on ShuffleNetv2-YOLOv5-Lite-E. *Agronomy* **2023**, *13*, 577. [CrossRef]
10. Yang, H.; Chen, L.; Chen, M.; Ma, Z.; Deng, F.; Li, M.; Li, X. Tender Tea Shoots Recognition and Positioning for Picking Robot Using Improved YOLO-V3 Model. *IEEE Access* **2019**, *7*, 180998–181011. [CrossRef]
11. Chen, Y.; Chen, S. Localizing plucking points of tea leaves using deep convolutional neural networks. *Comput. Electron. Agric.* **2008**, *171*, 105298. [CrossRef]
12. Li, Y.; He, L.; Jia, J.; Lv, J.; Chen, J.; Qiao, X.; Wu, C. In-field tea shoot detection and 3D localization using an RGB-D camera. *Comput. Electron. Agric.* **2023**, *185*, 106149. [CrossRef]
13. Li, Y.; Wu, S.; He, L.; Tong, J.; Zhao, R.; Jia, J.; Chen, J.; Wu, C. Development and field evaluation of a robotic harvesting system for plucking high-quality tea. *Comput. Electron. Agric.* **2023**, *206*, 107659. [CrossRef]
14. Chen, C.; Lu, J.; Zhou, M.; Yi, J.; Liao, M.; Gao, Z. A YOLOv3-based computer vision system for identification of tea buds and the picking point. *Comput. Electron. Agric.* **2022**, *198*, 107116. [CrossRef]
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*; Cornell University Library: Ithaca, NY, USA, 2014.
16. Ultralytics. YOLOv5: v6.1. Available online: https://github.com/ultralytics/yolov5 (accessed on 19 May 2022).
17. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
18. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Supplementary material for ECA-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 13–19.
19. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
20. Li, S.; Zhang, S.; Xue, J.; Sun, H. Lightweight target detection for the field flat jujube based on improved YOLOv5. *Comput. Electron. Agric.* **2023**, *202*, 107391. [CrossRef]
21. Xu, L.; Wang, Y.; Shi, X.; Tang, Z.; Chen, X.; Wang, Y.; Zou, Z.; Huang, P.; Liu, B.; Yang, N.; et al. Real-time and accurate detection of citrus in complex scenes based on HPL-YOLOv4. *Comput. Electron. Agric.* **2023**, *205*, 107590. [CrossRef]
22. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996.
23. Zhang, X.; Chen, Y.; Jia, J.; Kuang, K.; Lan, Y.; Wu, C. Multi-view density-based field-road classifi- cation for agricultural machinery: DBSCAN and object detection. *Comput. Electron. Agric.* **2023**, *200*, 107263.
24. Pearson, K. LIII. On lines and planes of closest fit to systemsof points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci. Comput. Electron. Agric.* **1901**, *2*, 559–572.
25. Hussain, M.; He, L.; Schupp, J.; Lyons, D.; Heinemann, P. Green fruit segmentation and orientation estimation for robotic green fruit thinning of apples. *Comput. Electron. Agric.* **2023**, *207*, 107734. [CrossRef]
26. Fang, M.; Lü, J.; Ruan, J.; Bian, L.; Wu, C.; Yao, Q. Tea Buds Detection Model Using Improved YOLOv4-tiny. *J. Tea Sci.* **2022**, *42*, 549–560.
27. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
28. Zhao, S.; Lei, X.; Liu, J.; Jin, Y.; Bai, Z.; Yi, Z.; Liu, J. Transient multi-indicator detection for seedling sorting in high-speed transplanting based on a lightweight model. *Comput. Electron. Agric.* **2023**, *211*, 107996. [CrossRef]
29. Liang, J.; Chen, X.; Liang, C.; Long, T.; Tang, X.; Shi, Z.; Zhou, M.; Zhao, J.; Lan, Y.; Long, Y. A detection approach for late-autumn shoots of litchi based on unmanned aerial vehicle (UAV) remote sensing. *Comput. Electron. Agric.* **2023**, *204*, 107535. [CrossRef]

30. Li, T.; Sun, M.; He, Q.; Zhang, G.; Shi, G.; Ding, X.; Lin, S. Tomato recognition and location algorithm based on improved YOLOv5. *Comput. Electron. Agric.* **2023**, *208*, 107759. [CrossRef]
31. Zhang, F.; Gao, J.; Zhou, H.; Zhang, J.; Zou, K.; Yuan, T. Three-dimensional pose detection method based on keypoints detection network for tomato bunch. *Comput. Electron. Agric.* **2022**, *195*, 106824. [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.