sustainability

MDPI

Article

Locating Tea Bud Keypoints by Keypoint Detection Method Based on Convolutional Neural Network

Yifan Cheng [1,2,†], Yang Li [2,†], Rentian Zhang [2], Zhiyong Gui [2], Chunwang Dong [3,*] and Rong Ma [1,*]

[1] College of Optical, Mechanical and Electrical Engineering, Zhejiang A&F University, Hangzhou 311300, China
[2] Tea Research Institute, Chinese Academy of Agricultural Sciences, Hangzhou 310008, China
[3] Tea Research Institute of Shandong Academy of Agricultural Sciences, Jinan 250100, China
* Correspondence: dongchunwang@163.com (C.D.); 20200001@zafu.edu.cn (R.M.)
† These authors contributed equally to this work.

Abstract: Tea is one of the most consumed beverages in the whole world. Premium tea is a kind of tea with high nutrition, quality, and economic value. This study solves the problem of detecting premium tea buds in automatic plucking by training a modified Mask R-CNN network for tea bud detection in images. A new anchor generation method by adding additional anchors and the CIoU loss function were used in this modified model. In this study, the keypoint detection branch was optimized to locate tea bud keypoints, which, containing a fully convolutional network (FCN), is also built to locate the keypoints of bud objects. The built convolutional neural network was trained through our dataset and obtained an 86.6% precision and 88.3% recall for the bud object detection. The keypoint localization had a precision of 85.9% and a recall of 83.3%. In addition, a dataset for the tea buds and picking points was constructed in study. The experiments show that the developed model can be robust for a range of tea-bud-harvesting scenarios and introduces the possibility and theoretical basis for fully automated tea bud harvesting.

Keywords: tea buds plucking; convolutional neural network; object detection; keypoint detection

check for updates

Citation: Cheng, Y.; Li, Y.; Zhang, R.; Gui, Z.; Dong, C.; Ma, R. Locating Tea Bud Keypoints by Keypoint Detection Method Based on Convolutional Neural Network. *Sustainability* **2023**, *15*, 6898. https://doi.org/10.3390/su15086898

Academic Editor: Teodor Rusu

Received: 13 March 2023
Revised: 5 April 2023
Accepted: 14 April 2023
Published: 19 April 2023

1. Introduction

Tea is one of the most popular and most consumed beverages in the world [1]. According to the Food and Agriculture Organization of the United Nations, the total global tea production in 2020 was 7.02 million tons. With the highest acreage and production volume, China is the largest producer and consumer of tea worldwide. The yearly increase in tea production poses a huge challenge to the labor force. At present, premium tea bud harvesting relies on manual plucking. The manual plucking process has many disadvantages, including a high work intensity and labor cost and strong subjective factors. More importantly, the short tea-harvesting period results in a great shortage of labor. The limitation and blockage of human movement during this COVID-19 pandemic has increased the shortage of professional pickers.

Today, there are already many studies that use computer vision methods to detect crops [2,3]. For example, Lin et al. proposed a detection algorithm based on color, depth, and shape information to detect the spherical or cylindrical fruits of plants in natural environments [4]. Liang et al. used the maximum interclass variance method for the target fruit bunch segmentation, which is a fast parallel algorithm for extracting the fruit pedicel skeleton, and the Harris corner point detection method for locating the keypoints of tomato picking [5]. They correctly located the keypoints of picking with 90% accuracy. Meanwhile, Liu et al. proposed a detection method based on the color and shape features of apples [6]. This method extracted color features from blocks by block mining, filtering candidate regions with non-fruit block proportions to improve the detection accuracy. However, the identified objects were significantly different from the background in terms of both color and shape, making it easier to extract features using these two factors. In the case of tea-harvesting, the color and the shape of the tea buds and leaves are similar.

In recent years, related research has led convolutional neural networks (CNNs) to show a remarkable performance in target detection for different crops [7,8]. Currently, two types of methods can detect objects using CNNs. The first type includes single-stage object detection methods, such as SSD [9] and YOLO [10], which can predict the bounding box directly from the input image without a region-suggestion step. Onishi et al. applied the SSD for apple detection and used a stereo camera to obtain three-dimensional (3D) positions [11]. The second type comprises two-stage target detection methods, such as R-CNN, Faster R-CNN, and Mask R-CNN. These methods generate region proposals from images and extract features from these regions for classification and positioning. Bargoti and Underwood used Faster R-CNN to detect apple and mango fruit trees in orchards [12]. Yu proposed a strawberry fruit detection method based on Mask R-CNN, which overcame the difficulties of using traditional machine-vision algorithms in unstructured environments [13]. A method based on an improved Faster R-CNN using color and depth images was also proposed for the robust detection of small fruits. The single-stage detection methods usually have faster speeds, while the two-stage detection methods have higher accuracies and can be used for difficult detection tasks.

Meanwhile, To solve the tea-harvesting problem, researchers have started to develop tea-harvesting machines [14]. However, existing automatic tea-harvesting machines use a large-scale "head-shaving" cutting method that destroys the buds and stems, causes the loss of flavor substances, and reduces the economic benefits of tea. At the same time, these traditional machine-learning detection methods based on a single feature or few features (e.g., color and shape) as the basis for detection cannot identify tea buds in tea bushes well. For these reasons, many experts and scholars have conducted a number of deep-learning researches on tea bud detection [15,16]. At the same time, these traditional machine-learning detection methods based on a single feature or few features (e.g., color and shape) as the basis for detection cannot identify tea buds in tea bushes well.

In this study, an image dataset of tea buds and key picking points was constructed. In addition, this study develops an algorithm that can accomplish tea bud detection in the field with different backgrounds and environments and locate the bud keypoints. The tea bud detection model used was based on Mask R-CNN. The Mask R-CNN model was improved with a new method for anchor generation, a better loss function for bud detection, and a new keypoint detection branch for locating the picking point. In an actual tea garden environment, this developed model can accurately identify tea buds and determine the keypoint of bud plucking. The remainder of this paper is structured as follows: Section 2 introduces the materials and methods used in this work; Section 3 compares the general experimental specific method implementation details and presents an analysis of the corresponding experimental results; and Section 4 discusses the experiment results and draws the study conclusions.

2. Materials and Methods

2.1. Data Preparation

The harvested objects were a single bud, one bud and one leaf, and one bud and two leaves of premium teas, including the Longjing 43 and Zhongcha 108 varieties. All images were collected by this research during the harvest season at the tea plantation of the Tea Research Institute, Chinese Academy of Agricultural Sciences, and taken from 15 March to 25 April 2022 between 8:30 and 17:30. The experimenter stood at a distance from a tea bush and used a smartphone to capture images of the buds to be harvested from that tea bush. The smartphone was held at 30–60° angles from the tea bush surface and approximately 0.5 m from the bud targets. This distance ensured that each image contained at least one clear, unobstructed set of bud targets for harvesting. The acquired images (Figure 1) include simple and complex shooting backgrounds (i.e., simple and complex backgrounds) and different target numbers (i.e., few and many bud targets) and shooting scenes (i.e., high-light and low-light scenes).

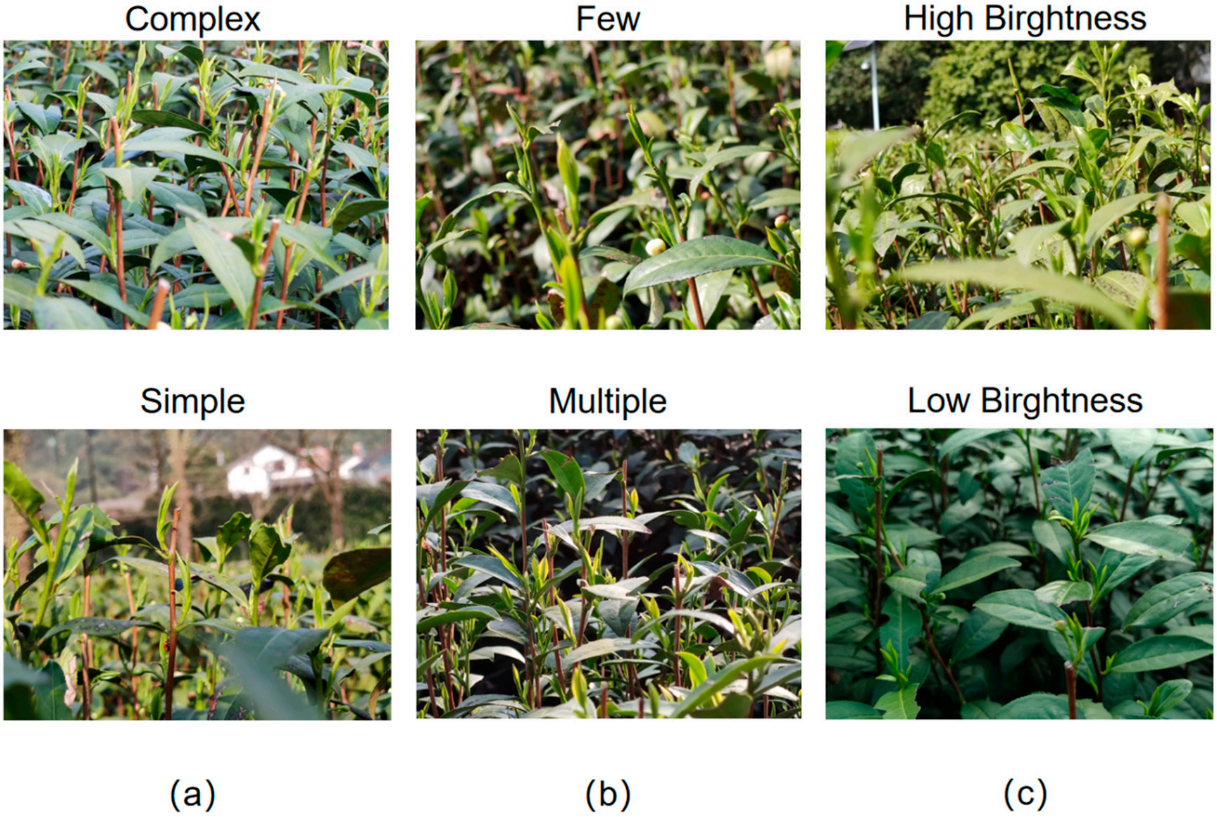

Figure 1. Tea bush images acquired under various conditions: (**a**) background complexity; (**b**) tea leaf complexity; and (**c**) illumination.

2.2. Image Labeling

According to the number of buds and leaves of tea buds, detectable buds can be regarded as three types: single-bud leaf picking, one-bud-one-leaf picking, and one-bud-two-leaf picking. According to the quality requirements, this study selected their optimal picking points as keypoints. The types of tea buds and the corresponding picking points are shown in Figure 2. In the Figure 2, (a) refers to the single bud and its picking point, (b) refers to the one-bud-and-one-leaf type and its picking point, and (c) refers to the one-bud-and-two-leaves type and its picking point. In this study, the picking priority of single bud is better than one bud and one leaf, and one bud and one leaf is better than one bud and two leaves. Therefore, the visible picking point of single bud is the optimal choice for annotation and identification in the research. Moreover, picking point of one bud and one leaf is better than picking point of one bud and two leaves.

Labelme software [17] was used to label the bud objects and their keypoint. In Figure 3, each bud object and keypoint to be picked in the picture was labeled with a rectangle and a point. The rectangle labeled as ‘bud’ is the bud object to be plucked. The point labeled as ‘k’ is the best picking point on the tea object. The rectangular box and the marked point are numbered separately to ensure that the picking points of the bud target can correspond to each bud target. According to the statistics of the above three types of bud and leaf, the number of the three bud leaf types was 2.4:4.3:3.3, respectively.

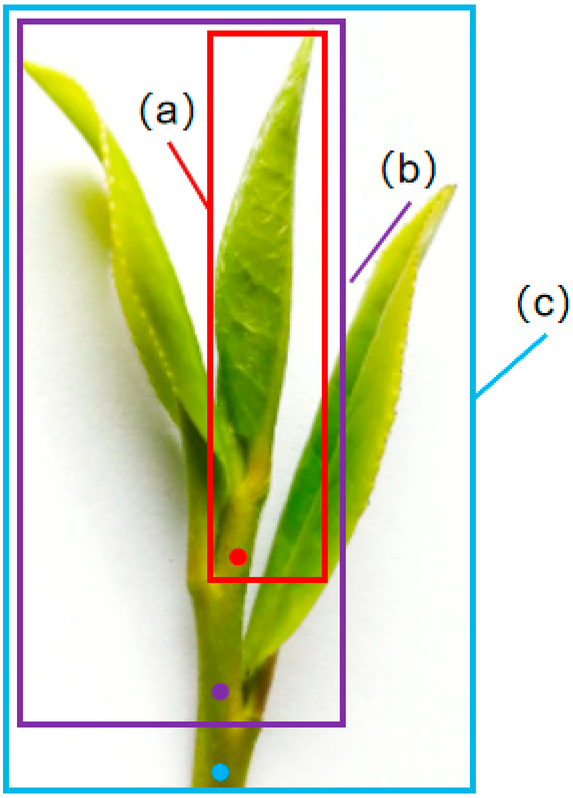

Figure 2. Bud type and plucking points: (**a**) single bud with picking point; (**b**) one bud and one leaf with picking point; and (**c**) one bud and two leaves with picking point.

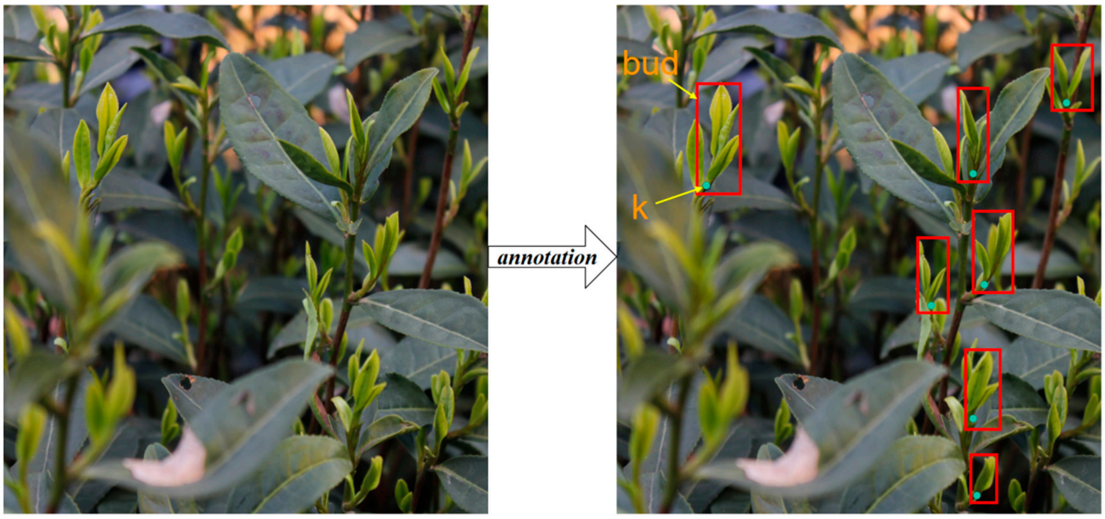

Figure 3. Visualized annotation examples from the tea bud and keypoint dataset. One box and one keypoint are annotated on each tea bud object. 'bud' is the tea sprout. 'k' is the most significant keypoint in one object.

The labeled Ground Truth consisted of the labels of the tea bud object and of their keypoints. The label of a bud object contains four elements. The first and second elements are the x- and y-axis co-ordinates of the lower left point of the frame. The third and fourth elements are the length and the height in the x- and y-axis directions, respectively. Similarly, the keypoint is labeled by three elements. The first and second elements are the x and y co-ordinates, respectively. The third element is the visibility flag (v). When v = 1, this means that the keypoint does not exist, while v = 2 indicates that the keypoint exists and is visible.

The collected pictures were sorted out, and some situations such as overexposure, dark light, blurry, jitter, serious target blocking of buds and leaves, and occlusion of bud and leaf picking points were screened out, which seriously affected the recognition results. At the same time, a total of 1260 images were collected and adjusted to 1600 × 1200 pixels. The dataset denoted a 5:1 ratio; 1050 images were used as the training set for training the parameters of the keypoint detection model, and 210 images were used as the test set for testing and evaluating this model. Data augmentation was used to randomly rotate, flip, stretch the image, and mixup. The term mixup means to fuse the two images. The dataset increased to 6300 after data augmentation.

2.3. Overall Approach

The stems, leaves, and buds of the tea bush were similar in color, depicting the tendency to block each other. They are difficult to accurately identify using the traditional threshold segmentation and color difference segmentation algorithms. To solve the problem of identifying the tea bud keypoints, the proposed model must identify the bud targets and complete the keypoint localization in the input image. Given that deep convolutional models have the unique advantages of feature extraction and recognition, the Mask R-CNN model [18] used here is an improvement on the Faster R-CNN [19] model. This model consisted of four modules: a feature extraction network composed of Resnet50 feature pyramid network (FPN); a region proposal network (RPN); an FC (full connected)-based object detection branch; and a FCN-based keypoint detection branch. Figure 4 illustrates the overall approach.

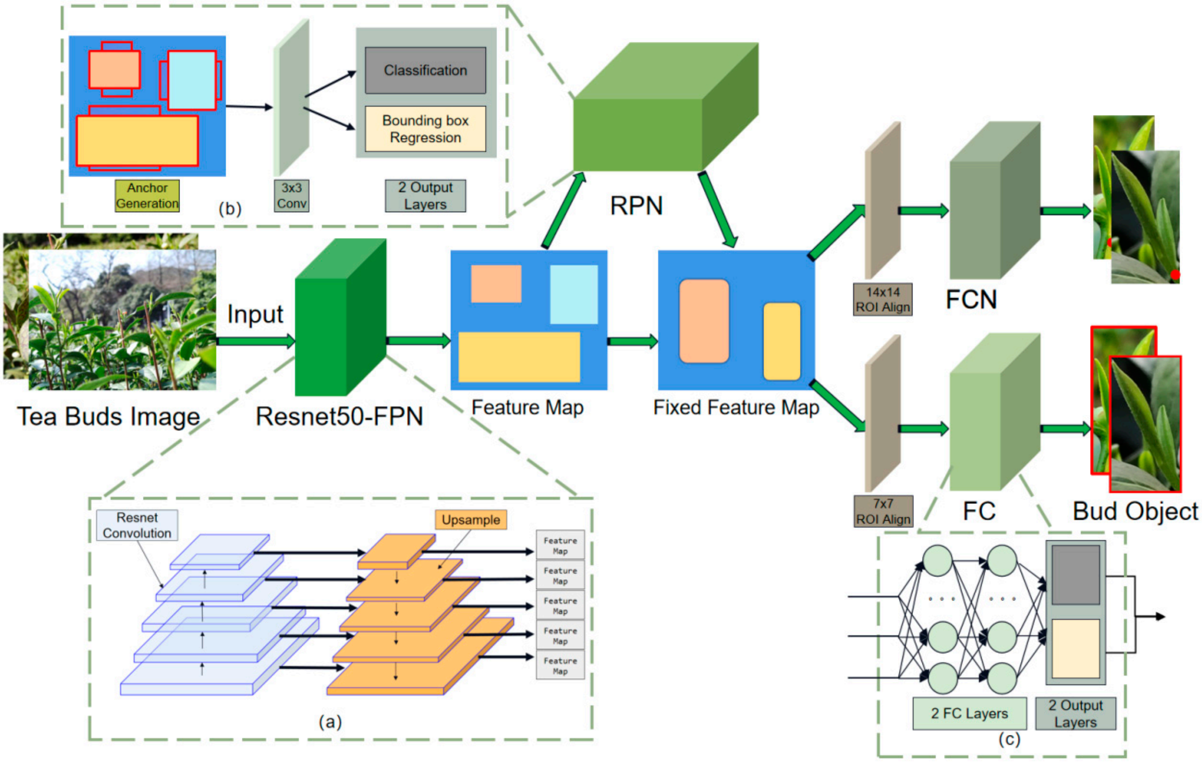

Figure 4. Architecture of the Mask R-CNN model: (**a**) feature extraction network composed of resnet50-FPN; (**b**) architecture of RPN, and (**c**) architecture of the FC model.

The keypoint detection branch attempts to locate the bud keypoints as the basis of inference for the later cutting points on a 3D space. This detection is very similar to human pose estimation methods in computer vision research. Human pose estimation methods can detect a specific set of keypoints, including knee and shoulder points, for each human object in an image [20–22]. Two types of keypoint detection both aim to detect the corresponding keypoints of multiple identical objects (e.g., humans or tea buds) in an image. This study accomplished the keypoint identification and localization for buds through transfer learning of human pose estimation methods [23]. The constructed model was also pre-trained by using the human pose estimation dataset in the Microsoft COCO dataset [24]. The pre-trained model laid a foundation for the bud target and keypoint detection.

2.3.1. Feature Extraction Network

The feature extraction network extracted the feature map from the input image (Figure 4a). The improved model network used a cascaded network consisting of Resnet50 and an FPN. The Resnet part contained five convolution stages. The features extracted from each convolution stage in the Resnet50 network were then fused by the FPN network. Finally, the top-down feature fusion was completed by five upsampling layers. Each FPN layer output the fused feature maps.

2.3.2. Improved Region Proposal Network

The RPN captures features from the feature extraction network by matching pre-defined anchors. These matched anchors become the region of interest (ROI). In Figure 4b, the model network consisted of an anchor generation module, a convolutional layer, and two output layers.

Anchor generation mainly provided several anchors for the subsequent matching. The bud objects were generally very dense with small aspect ratios; hence, the bud target interval was smaller than the step size of the generation function. The generated anchor was too sparse to match the appropriate bud objects. With reference to the anchored generation strategy proposed in the textbox++ model of M. Liao [25], an anchor generation algorithm was proposed in this research to better fit the tea bud detection task, specifically for specific bud object characteristics. Figure 5 shows that the algorithm used in this work generates additional box sets in the horizontal direction to better match the buds in the image.

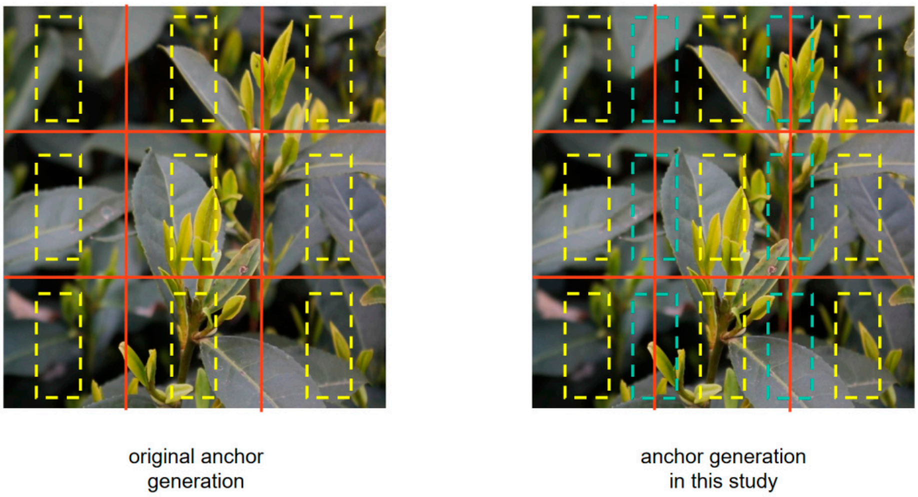

Figure 5. Yellow anchor: generated by the original anchor and only generates anchors at the center of each stride. Green anchor: additional anchor generated by the method used in this study.

The convolutional and output layers in the RPN comprised a convolutional layer with a 3 × 3-pixel kernel and two output layers with 1 × 1-pixel kernel, respectively. The first output layer determined the current bounding box class. The second layer calculated the determined bounding boxes of the proposal anchor.

2.3.3. Object Detection Branch

Mask R-CNN identified the ROI as a bud object using an object detection branch. The proposed region feature of different sizes was adjusted to a 7 × 7 size using the ROI align layers and sent to this branch. In the object detection branch (Figure 3c), the aligned features were calculated by two fully connected networks (FC). These FCs inferred the bounding boxes of the ROIs and determined the bounding box category.

2.3.4. Improved Keypoint Detection Branch

Mask-RCNN can be implemented with different functions by adding different branches. This part of the task was accomplished herein by adding a keypoint detection branch after the RPN network. The keypoint detection branch accepted the 14 × 14 ROI features aligned by the ROI align layers to locate the tea bud keypoints. All keypoints of buds were converted into the heatmap-offset format proposed by Google [26]. The first element indicates the probability. The second and third elements indicate the offset from the Ground Truth.

The keypoint detection branch proposed in this study mainly consisted of a fully convolutional network. Figure 6 depicts the FCN structure used in this work. The model used convolutional layer for completing the convolution operation and the bilinear interpolation method for generating the upsampling feature. Different upsampling multipliers can affect the test results. The most suitable configuration for the keypoint detection of the tea buds was found by comparing four upsampling multipliers (i.e., 4, 8, 16, and 32) in the FCN networks.

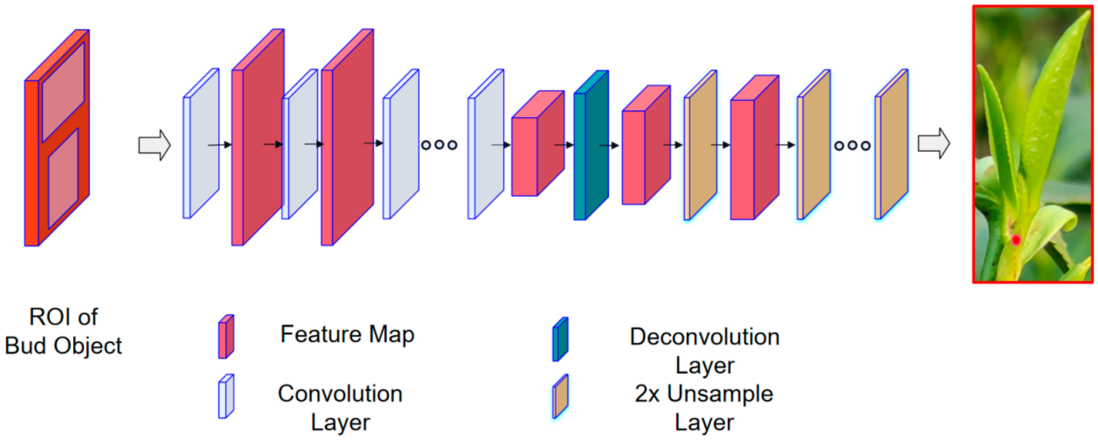

Figure 6. Architecture of the keypoint detection branch that controls the FCN structure by adjusting the numbers of convolutional layers and upsamples.

2.4. Evaluation Indicators

The same standard assessment metrics from the human estimation model were introduced to evaluate the performance of the present model and verify the feasibility of this study.

2.4.1. Average Precision and Average Recall of Object Detection

The COCO dataset previously defined Intersection over Union (IoU) for calculating the object detection precision. AP^{50} treats the prediction with an IoU greater than 50% as a correct prediction and calculates the proportion of correct predictions in the total number of predictions. Average precision (AP) refers to taking IoU threshold of every 0.05 from 0.50 to 0.95, calculating the average of 10 ranges. These will be referred to as $AP^{50}{}_{bbox}$ and AP_{bbox}

It calculated the proportion of correct prediction with IoU in the total number of Ground Truth. It will be referred to as AR_{bbox}

2.4.2. Average Precision and Average Recall of Keypoint Detection

The COCO dataset previously defined a threshold for calculating the keypoint detection precision. This threshold is called the object keypoint similarity *(OKS*). The *OKS* can normalize the distance between the predicted and target points by using detected bud areas of different sizes and is calculated as follows in Equation (1):

$$OKS = \frac{\sum_i \exp\left\{-d_{pi}^2 / 2S_p^2 \sigma_i^2\right\} \delta(V_{Pi} > 0)}{\sum_i \delta(V_{Pi} > 0)} \tag{1}$$

where P is the bud object ID in the Ground Truth; i is the keypoint ID; P^i is the keypoint i of the bud object P; d_{Pi} is the Euclidean distance between a detected keypoint and its corresponding Ground Truth; V_{Pi} is the visibility flag of this keypoint; S_p is the scale of the bud object; σ_i is the key control constant related to the keypoint type; $\delta(*)$ means "only calculate the keypoint marked in the Ground Truth". This σ_i value represents the standard deviation of the labeling process, and the larger the σ_i value, the more difficult the label. Thus, the values of the different kinds of points in the tea bud are not the same. For each keypoint, Equation (1) produces a similarity score between 0 and 1. These similarities were averaged over all the labeled keypoints. The unlabeled prediction keypoints did not affect the *OKS* score. *OKS* score participates in the assessment as a threshold value. The AP_{kpt} and AR_{kpt} metrics of the average precision and recall vary across thresholds of 0.5 to 0.95 with a 0.05 interval. The $\text{AP}^{50}{}_{\text{kpt}}$ value at a single *OKS* of 0.5 was also calculated, corresponding to the $\text{AP}^{50}{}_{\text{bbox}}$ metric. It calculated the proportion of correct prediction with *OKS* in the total number of Ground Truth. It will be referred to as AR_{kpt}.

The higher these values, the better the keypoint detection results.

2.4.3. Model Complexity

The model complexity is usually referred to as the number of parameters of the total computation process. This is a measure of how many parameters were computed by the model and how much storage space was spent to store the model parameters. The total number of model computations usually determines the model's speed during the detection process and is often measured using floating point operations (FLOPs) (i.e., 1 GFLOP = 10^9 FLOPs). This metric gives a good indication of how much computation is needed to run the current model.

2.5. Experimental Process Design

The bud image data were captured from the smartphone at the same settings. The length of the long side of the image was standardized to 640 to complete the size uniformity for the input image. The COCO human pose keypoint dataset was employed to pre-train the built model using the pre-training parameters to continue the following training:

PyTorch was used as the framework for the entire experiment. Nvidia RTX3070 was utilized for training. The cross-entropy function was used as the loss function in the FCN of the keypoint detection branch to calculate the loss between the predicted and Ground Truth keypoints. The initial learning rate was set to 0.0004. The batch size was 2. This experiment trained 30,000 iterations. The learning rate completed two decreases at 15,000 and 20,000 iterations. Learning rate and loss changes during model training are shown in Figure 7.

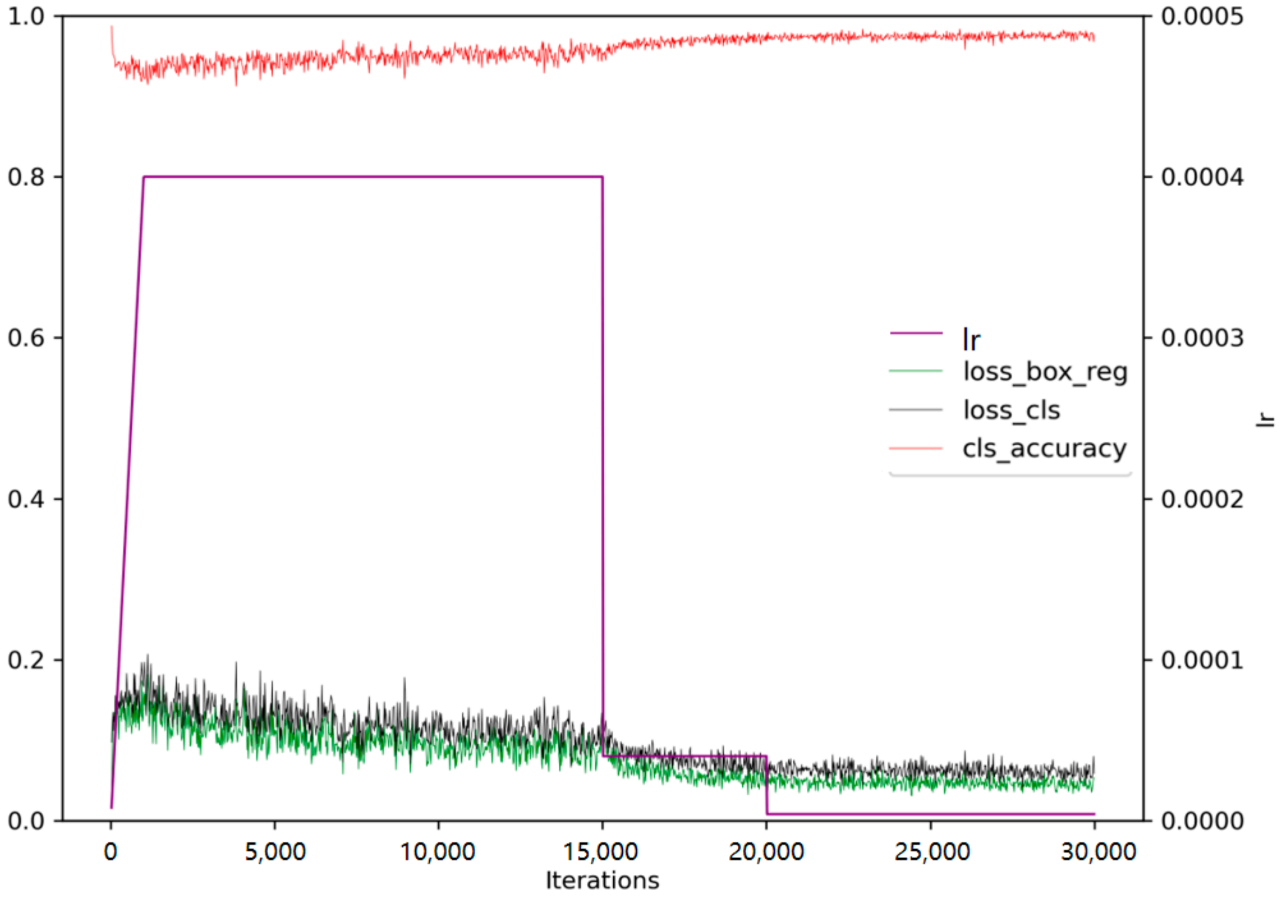

Figure 7. Loss curves and learning rate drop during training.

3. Results and Discussion

3.1. Target Area Identification Performance of Tea Buds

The trained bud target detection model aimed to accurately locate the bud object keypoints and was particularly important for obtaining an accurate bud region. Different backbone networks for feature extraction will obtain different feature results, affecting the bud object detection. Table 1 compares the effects of several popular backbone networks on the experimental results at this stage.

Table 1. Test results and model characteristics under different backbone networks.

Backbone	AP_{bbox}	AP^{50}_{bbox}	AR_{bbox}	Params	GFLOPs
ResNet34	55.3	71.8	66.9	157M	15.4
ResNet34-FPN	60.8	75.2	73.4	165M	17.5
ResNet50	70.2	79.4	78.5	217M	20.2
ResNet50-FPN	74.3	86.6	88.3	234M	22.5
ResNet101	73.4	80.2	76.4	298M	37.6
ResNet101-FPN	75.6	87.5	78.5	315M	48.3

As a backbone network, ResNet50-FPN showed a higher AP score for bud detection compared to ResNet50. The models with ResNet34-FPN and ResNet101-FPN as the backbone networks both showed improved bud detection results compared to those with ResNet34 and ResNet101, respectively. This result was attributed to the fact that FPN networks can combine the advantages of different feature scales to obtain both small information in small-scale features and larger semantic information in large-scale feature maps and can fuse to generate semantic-information-rich and spatially accurate feature maps with less computational effort. The comparison of the effects of the ResNet network depth on the detection effect showed that ResNet50-FPN significantly improved the detection accuracy, but correspondingly increased the amount of model computation compared to the ResNet34-FPN network. The ResNet101-FPN network with a convolutional depth of 101 layers required much more flops to be calculated compared to the ResNet50-FPN network, nearly twice as much as 50 layers. However, it did not have a correspondingly significant gain in the detection results.

After comparing the different backbone networks, ResNet50-FPN has the better AP performance; at the same time, the parameters and flops of this ResNet50-FPN are both not too large. Therefore, ResNet50-FPN was selected as the backbone network for the feature extraction.

3.2. Keypoint Positioning Performance of Buds

The keypoint detection branch consisted of an FCN with an upsampling module. The different upsampling multipliers of the FCN will gain features at different scales in the keypoint localization process, affecting the network effects on positioning the keypoints. The same test set was used to examine the ability of the different upsampling multipliers of the FCN to locate the keypoints. Table 2 shows a comparison of the performances of the four trained FCN models after the evaluation.

Table 2. Performances of the FCNs with different upsampling multipliers.

Model	$\mathbf{AP_{kpt}}$	$\mathbf{AP^{50}{}_{kpt}}$	$\mathbf{AR_{kpt}}$
FCN-4	54.3	68.3	62.5
FCN-8	62.8	75.9	78.3
FCN-16	55.9	69.6	74.5
FCN-32	41.3	64.5	68.7

As a small object, the keypoint feature is not always obvious enough to be recognized. Therefore, the keypoint localization may have a higher upsampling multiplier. Compared to FCN-4s, FCN-8s can produce features of a more suitable size scale after upsampling. In contrast, the upsampling multipliers of FCN-16s and FCN-32s were too large, resulting in a coarse feature map and much noise. Therefore, the identification of the extracted regions was not satisfactory. FCN-8s were used in the final model.

The heatmap channel of the keypoint detection branch can be converted into a visual image to intuitively compare the importance of the upsampling multipliers on feature extraction. Figure 8 illustrates the heatmap results of the keypoints. The heatmap values indicate the probability of a keypoint at that location. The higher the probability value in the heatmap, the warmer the color shown in the plot. In (a), the overall probability value of the response area is small and depicted as a yellow block covering the area around the point. In (b), the response region has a higher probability; hence, the heatmap color also changes from yellow to a warmer and more concentrated orange. In (c), some orange blocks can be seen on the heatmap, albeit being much less concentrated than in (b). Figure 8d clearly shows the regions with cooler colors and even less concentrated response regions, indicating that the overall probability is relatively small. Visualization heatmap images also proved that FCN-8s is a better choice.

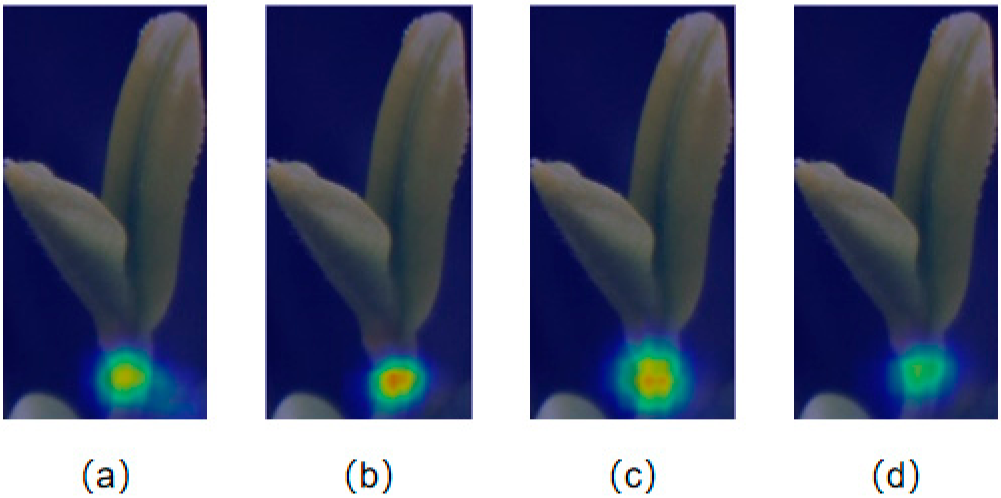

Figure 8. Visualization heatmap results of different FCNs. Left to right column: corresponding (**a**) FCN-4s, (**b**) FCN-8s, (**c**) FCN-16s, and (**d**) FCN-32s.

3.3. Relevant Experiments

During the overall model construction, some experiments showed that many modules of this model can have different effects on the final results. These effects were compared to investigate the effects of different module changes on the results.

3.3.1. Experiments of the Anchoring Generation Method

In the RPN of this model, the improved anchoring generation method adjusted the aspect ratio and generated additional anchor sets in the horizontal direction. Three anchor generation methods for the ablation experiments were compared to complete the experimental comparison. The three anchor generation methods are as follows: (a) original anchor generation; (b) the method adjusting the aspect ratio of the anchor during the anchor generation and generating the anchor with aspect ratios of 0.2, 0.3, 0.5, 0.8, and 1; and (c) the anchor generation in this study. Different anchor generation methods have different effects on the subsequent matching. The performances of the anchor generation methods for bud detection were examined herein. Table 3 presents the results of the three methods.

Table 3. Comparison of the detection performance results of different anchor generation methods.

Method	$\mathbf{AP_{bbox}}$	$\mathbf{AP^{50}{}_{bbox}}$	$\mathbf{AR_{bbox}}$
(a) Original anchor generation	65.3	77.4	80.1
(b) Adjusted aspect ratio	73.8	80.7	81.3
(c) Anchor generation in this study	79.6	85.4	84.5

Note that the bud objects are usually slim and elongated in shape and have a small aspect ratio. The adjusted aspect ratio showed a better performance than the original anchor generation. The anchor generation algorithm used in this study was adopted by adjusting the aspect ratio of the anchors and adding a horizontal anchor. This algorithm exhibited the best performance among the three. Therefore, there is reason to believe that this algorithm can better match the bud recognition and the keypoint localization.

3.3.2. Experiments of the Loss Function

In the original model, the smooth L1 function was used as the loss function in the RPN and the FC. The smooth L1 Loss function calculates the Euclidean distance between the co-ordinate values of the predicted Bounding Box and the Ground Truth box as a deviation. In order to determine the precise cause of the loss, this study attempts to use IoU as the loss value of the model and apply the GIoU loss function and the CIoU loss function [27] in the object detection model. The GIoU loss function used the loss of IoU as the regression loss of the detection process in this research. The CIoU loss function, which is optimized for GIoU, takes into account the distance between the prediction box and the Ground Truth box, the length–width ratio of the prediction box and the Ground Truth box, and the bud size scale. Table 4 presents the performance of the models with the three methods.

Table 4. Comparison of the detection performance of models with different loss functions.

Loss Function	**AP_{bbox}**	**AP^{50}_{bbox}**	**AR_{bbox}**
(a) Smooth L1 loss	71.3	78.4	82.1
(b) GIoU loss	75.6	79.8	83.4
(c) CIoU loss	79.6	85.4	84.5

The four points of the Bounding Box are independent of each other. The correlation of the four co-ordinates is not considered in the smooth L_1 loss, causing the model to be significantly worse for detecting small objects.

Compared to the GIoU loss function, the CIoU loss function can provide the movement direction for the bounding box, to make the prediction box regression robust. At the same time, this function adds multi-factor considerations by the length–width ratio, which can improve the performance of the model.

Compared to the original loss function, it can be seen from Table 4 that the AP^{50} and the AP for tea bud detection both increase by about 7% when using the CIoU loss function. Therefore, it is proven that modifying the CIoU loss function can optimize the object detection performance.

3.3.3. Study for Detecting the Number of Keypoints

In the keypoint detection branch, this model was designed to detect a single keypoint. The collection dataset was re-labeled by following the priority order described in Section 2.2 (i.e., (a), (b), and (c) for the three labeling methods) to complete the experimental comparison: (a) labeling only the single optimal keypoint, the keypoint constant σ was set as 1; (b) labeling two optimal visible keypoints on the bud objects, the keypoint constant was set as 0.7 and 0.9; and (c) labeling three cutting points of the single buds, the key control constant was set as 0.65, 0.7, and 0.8. The model to be trained three times on the three different training datasets obtained from the three annotation methods was adjusted. The results of the localization task using the three methods were then evaluated. Table 5 presents the keypoint detection results of the three methods.

Table 5. Comparison of the detection performance results for different numbers of critical point detections for calibrated frames.

Label Methods	**AP_{kpt}**	**AP^{50}_{kpt}**	**AR_{kpt}**	**AP^{50}_{bbox}**
(a)	61.2	75.9	84.1	86.2
(b)	58.2	71.9	76.4	78.3
(c)	42.5	72.6	74.5	75.9

Theoretically, an increased number of inference points in a candidate region will yield a good performance in the evaluation metrics because the detected points can interact with each other and consider more contextual semantics. However, as seen in the visualization results of the three methods in Figure 9, methods (b) and (c) illustrate the final output that may not be as good as earlier thought. The dense branches and leaves of the tea bush made it difficult to fully capture a complete object that contains three or two key points in one shot of a bud object. As a result, the keypoints detected by method (b) were intersected and the keypoints detected by (c) were in the wrong place.

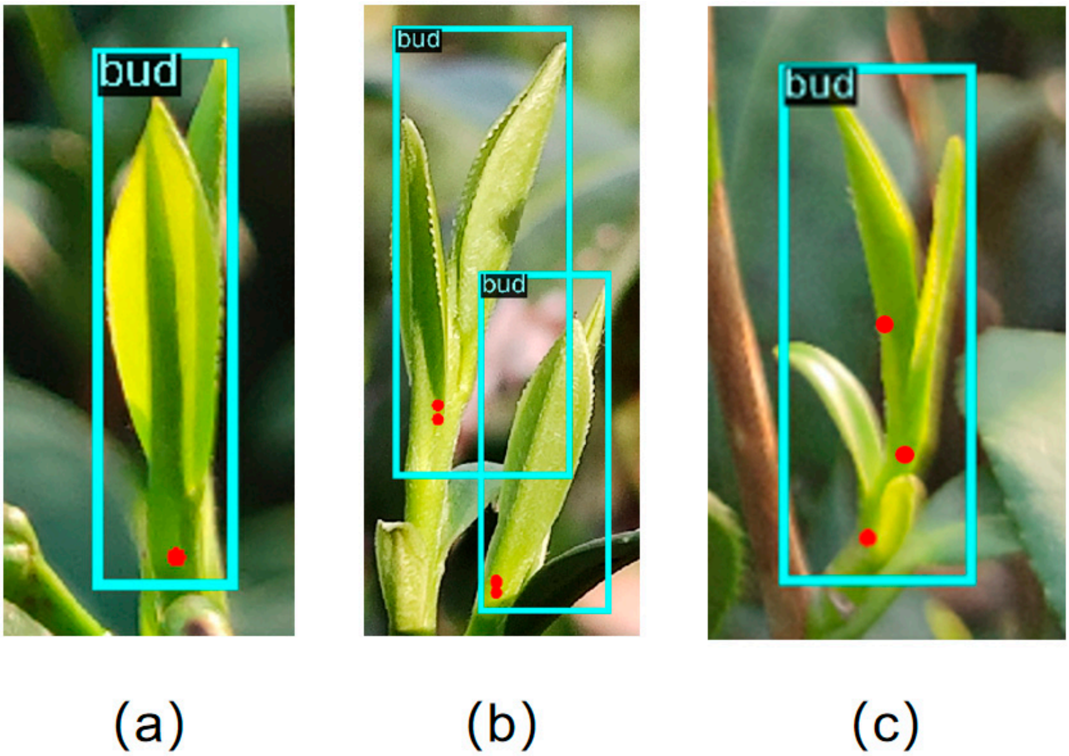

Figure 9. Visualization results of different detected keypoint numbers: (**a**)results of detecting single keypoint, (**b**)results of detecting two keypoints, (**c**) results of detecting three keypoints.

Therefore, this model finally only detects one tea-picking point as in method (a).

3.4. Overall Model Performance

A final model combined with a Mask R-CNN network and the FCN-8s for the keypoint localization was constructed. The final model's performance was evaluated using the test set. This model achieved 86.6% AP^{50} and 88.3% AR for the bud detection, and 85.9% AP^{50} and 83.3% AR for the keypoint localization.

3.4.1. Visualization Result Analysis

Figure 10 is a visual presentation of the detection results. The images illustrate the method applied to images obtained with different backgrounds (a), leaf complexity (b), and lighting conditions (c). The image background complexity varied depending on the angle and perspective. The final model showed an excellent performance in images with complex or simple backgrounds (a). A variation in the number of tea buds in the image can also be observed due to the variation in the distance between the camera and the tea bush. The model can locate bud objects and keypoints in images with different numbers of buds (b). The tea garden illumination was not controllable. The final model performed well on a large number of images, including those in low-illumination and high-illumination cases (c). This result indicates the robustness and adaptability of the final model.

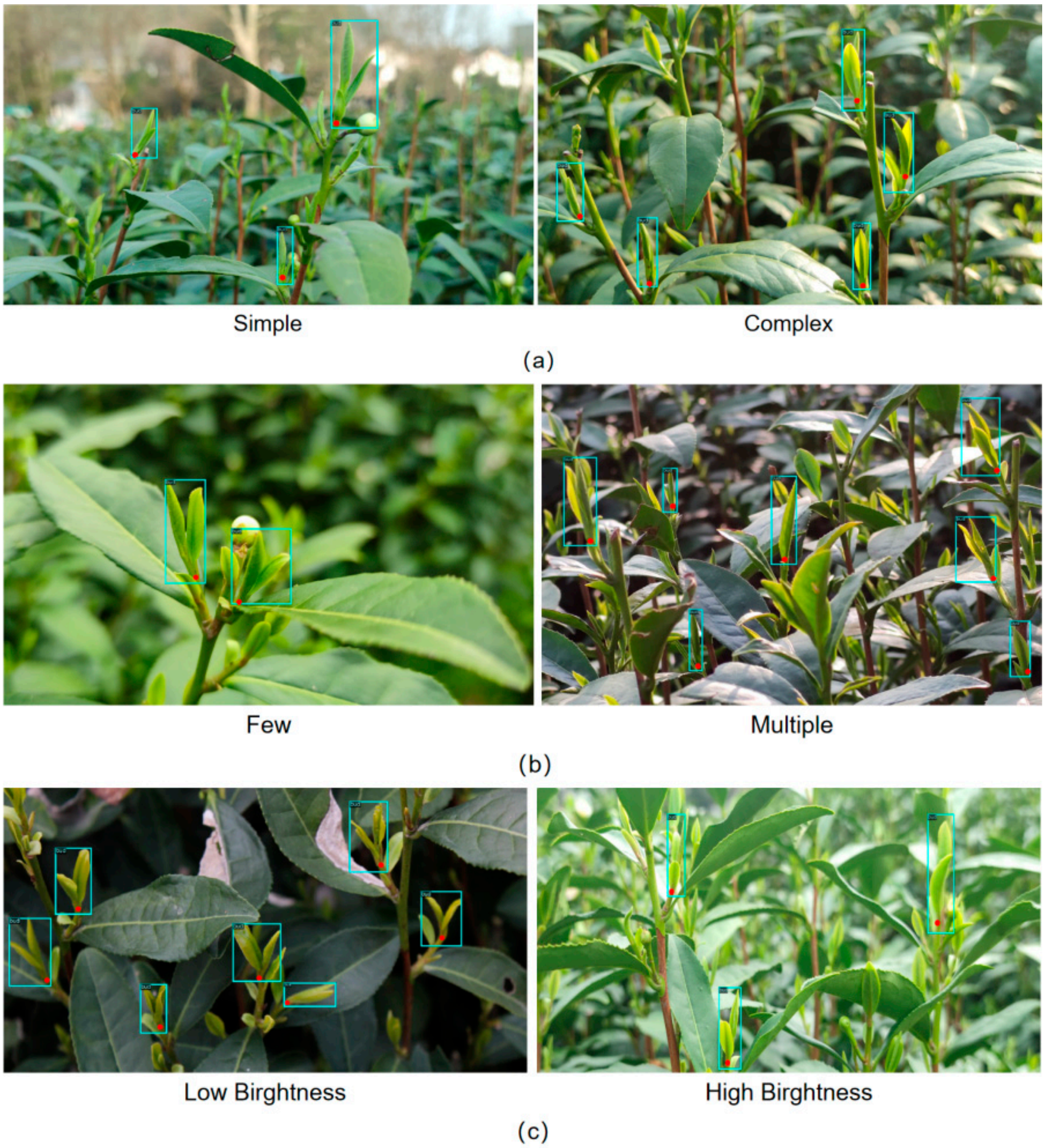

Figure 10. Detection of the tea bud regions and location of plucking points in images with various (**a**) backgrounds, (**b**) leaf complexity, and (**c**) brightness.

3.4.2. Comparison with State-Of-The-Art Methods

In this section, these state-of-the-art keypoint detection methods were compared with the proposed model on the keypoint dataset of this study, and its performance is evaluated. The results are shown in the Table 6.

Table 6. Comparison of the keypoint detection performance results for different models.

Model	$\mathbf{AP_{kpt}}$	$\mathbf{AP^{50}_{kpt}}$	$\mathbf{AR_{kpt}}$
Simple Baseline	59.2	78.5	69.5
HRnet	76.8	83.6	79.8
2-Stage Hourglass	67.2	78.0	75.6
Original Mask R-CNN	61.3	78.5	68.7
Our model	78.9	85.9	83.3

4. Conclusions

This study modified a stable model for the detection of tea buds and picking points in the field to solve the various problems faced by tea-picking machines. A Mask R-CNN-based object detection model was improved for the bud detection by a new anchor generation method and the CIoU loss function. It obtained an 86.6% precision and 88.3% recall after transfer learning. Adding a new keypoint detection branch led to the accurate detection of the keypoints in the bud region. The proposed final model yielded an 85.9% precision and 83.3% recall for the keypoint localization. The experimental results also showed that the model has the potential to detect tea buds and picking points in multiple scenarios and can be used to locate keypoints under different lighting scenarios and backgrounds based on its output. The modified model provides the basis for a machine that can automatically locate the cutting position of tea buds to complete a fully automated mechanical plucking operation.

Author Contributions: Conceptualization, Y.C., Y.L. and R.M.; methodology, Y.C.; software, Y.C.; validation, R.M.; formal analysis, Y.C.; investigation, Y.C., Y.L. and C.D.; resources, Y.C. and Z.G.; data curation, Y.C., R.Z. and Z.G.; writing—original draft, Y.C.; writing—review & editing, Y.L., C.D. and R.M.; supervision, Y.L., C.D. and R.M.; funding acquisition, Y.L. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by the Scientific Research Foundation of Zhejiang A and F University (2020FR063), Zhejiang Provincial Natural Science Foundation of China under Grant No. LTGN23C130004, the Central Public-Interest Scientific Institution Basal Research Fund (1610212021004, 1610212022004), and the Key R&D Program of Zhejiang (2022C02052, 2022C02042).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to another study related to this data is not yet publicly available.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Baba, R.; Kumazawa, K. Characterization of the potent odorants contributing to the characteristic aroma of Chinese green tea infusions by aroma extract dilution analysis. *J. Agric. Food Chem.* **2014**, *62*, 8308–8313. [CrossRef] [PubMed]
2. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [CrossRef]
3. Kuznetsova, A.; Maleva, T.; Soloviev, V. Using YOLOv3 algorithm with pre- and post-processing for apple detection in fruit-harvesting robot. *Agronomy* **2020**, *10*, 1016. [CrossRef]
4. Lin, G.; Tang, Y.; Zou, X.; Xiong, J.; Fang, Y. Color-, depth-, and shape-based 3D fruit detection. *Precis. Agric.* **2020**, *21*, 1–17. [CrossRef]
5. Liang, X.; Jin, C.; Ni, M.; Wang, Y. Acquisition and experiment on location information of picking point of tomato fruit clusters. *Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 163–169.
6. Liu, X.; Zhao, D.; Jia, W.; Ji, W.; Sun, Y. A Detection method for apple fruits based on color and shape features. *IEEE Access* **2019**, *7*, 67923–67933. [CrossRef]
7. Fu, L.; Tola, E.; Al-Mallahi, A.; Li, R.; Cui, Y. A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosyst. Eng.* **2019**, *183*, 184–195. [CrossRef]
8. Kang, H.; Chen, C. Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Comput. Electron. Agric.* **2020**, *171*, 105302. [CrossRef]
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
11. Onishi, Y.; Yoshida, T.; Kurita, H.; Fukao, T.; Arihara, H.; Iwai, A. An automated fruit harvesting robot by using deep learning. *Robomech J.* **2019**, *6*, 13. [CrossRef]

12. Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3626–3633.
13. Yu, Y.; Zhang, K.; Yang, L.; Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* **2019**, *163*, 104846. [CrossRef]
14. Du, Z.; Hu, Y.G.; Wang, S. Simulation and experiment of reciprocating cutter kinematics of portable tea picking machine. *Trans. Chin. Soc. Agric. Mach.* **2018**, *49*, 221–226.
15. Wang, T.; Zhang, K.; Zhang, W.; Wang, R.; Wan, S.; Rao, Y.; Jiang, Z.; Gu, L. Tea picking point detection and location based on Mask-RCNN. *Inf. Process. Agric.* **2021**, *in press*. [CrossRef]
16. Yan, L.; Wu, K.; Lin, J.; Xu, X.; Zhang, J.; Zhao, X.; Tayor, J.; Chen, D. Identification and picking point positioning of tender tea shoots based on MR3P-TS model. *Front. Plant Sci.* **2022**, *13*. [CrossRef]
17. Torralba, A.; Russell, B.C.; Yuen, J. LabelMe: Online image annotation and applications. *Proc. IEEE* **2010**, *98*, 1467–1484. [CrossRef]
18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
20. Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. Human pose estimation with iterative error feedback. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4733–4742.
21. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
22. Tompson, J.J.; Jain, A.; LeCun, Y.; Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1799–1807.
23. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 466–481.
24. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
25. Liao, M.; Shi, B.; Bai, X. TextBoxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. [CrossRef] [PubMed]
26. Papandreou, G.; Zhu, T.; Kanazawa, N.; Toshev, A.; Tompson, J.; Bregler, C.; Murphy, K. Towards accurate multi-person pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4903–4911.
27. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.

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.