

## Article

# A Dragon Fruit Picking Detection Method Based on YOLOv7 and PSP-Ellipse

Jialiing Zhou <sup>1,2</sup>, Yueyue Zhang <sup>1,2</sup> and Jinpeng Wang <sup>1,2,\*</sup> <sup>1</sup> School of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China<sup>2</sup> Co-Innovation Center of Efficient Processing and Utilization of Forest Resources, Nanjing Forestry University, Nanjing 210037, China

\* Correspondence: jpwang@njfu.edu.cn; Tel.: +86-025-85427765

**Abstract:** Dragon fruit is one of the most popular fruits in China and Southeast Asia. It, however, is mainly picked manually, imposing high labor intensity on farmers. The hard branches and complex postures of dragon fruit make it difficult to achieve automated picking. For picking dragon fruits with diverse postures, this paper proposes a new dragon fruit detection method, not only to identify and locate the dragon fruit, but also to detect the endpoints that are at the head and root of the dragon fruit, which can provide more visual information for the dragon fruit picking robot. First, YOLOv7 is used to locate and classify the dragon fruit. Then, we propose a PSP-Ellipse method to further detect the endpoints of the dragon fruit, including dragon fruit segmentation via PSPNet, endpoints positioning via an ellipse fitting algorithm and endpoints classification via ResNet. To test the proposed method, some experiments are conducted. In dragon fruit detection, the precision, recall and average precision of YOLOv7 are 0.844, 0.924 and 0.932, respectively. YOLOv7 also performs better compared with some other models. In dragon fruit segmentation, the segmentation performance of PSPNet on dragon fruit is better than some other commonly used semantic segmentation models, with the segmentation precision, recall and mean intersection over union being 0.959, 0.943 and 0.906, respectively. In endpoints detection, the distance error and angle error of endpoints positioning based on ellipse fitting are 39.8 pixels and 4.3°, and the classification accuracy of endpoints based on ResNet is 0.92. The proposed PSP-Ellipse method makes a great improvement compared with two kinds of keypoint regression method based on ResNet and UNet. Orchard picking experiments verified that the method proposed in this paper is effective. The detection method proposed in this paper not only promotes the progress of the automatic picking of dragon fruit, but it also provides a reference for other fruit detection.

**Keywords:** deep learning; picking robot; YOLOv7; PSP-Ellipse; dragon fruit

**Citation:** Zhou, J.; Zhang, Y.; Wang, J. A Dragon Fruit Picking Detection Method Based on YOLOv7 and PSP-Ellipse. *Sensors* **2023**, *23*, 3803. <https://doi.org/10.3390/s23083803>

Academic Editors: Yongwha Chung, Liqing Chen, Yuwei Wang, Peng Chen and Bolin Cai

Received: 25 February 2023

Revised: 3 April 2023

Accepted: 5 April 2023

Published: 7 April 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

The fruit picking robot is becoming one of the hottest topics in recent years [1,2]. China is the world's second largest dragon fruit producer. Nowadays, dragon fruit is usually harvested manually by cutting its branch that is connected with the root, which causes a lot of labor intensity. Research is being conducted on the dragon fruit picking robot to realize automatic picking, which could greatly reduce the labor force [3,4]. Visual intelligence is of key importance to the fruit picking robot [5]. However, most of the research on the visual systems for fruit picking only focuses on the recognition and position of fruits, which is not enough for dragon fruits [6]. The branches of dragon fruit are very hard, which means that the only way to pick dragon fruit is to cut the branches together with the fruit. The difficulty in the automatic picking of dragon fruit is that the complex postures of dragon fruit make it difficult to accurately find the root position for cutting. Therefore, in order to realize automatic picking, we need to develop ways to detect the position of dragon fruit and its root position.

The common methods used for fruit detection are traditional machine vision-based methods and deep learning-based methods. The former needs to manually extract features to recognize fruits, and it is generally less robust which makes it difficult to apply in complex natural environments [7–9]. In recent years, deep learning-based object detection algorithms have been widely considered to be promising tools for fruit detection because of their better generalization ability, which may be grouped into two categories: the one-stage algorithm and the two-stage algorithm [10–14]. Normally, the inference speed of the two-stage algorithm is slower than that of the one-stage algorithm, but the two-stage algorithm has a higher accuracy. For a picking robot, picking efficiency and picking accuracy are both important. YOLO and RCNN are the representatives of the one-stage algorithm and two-stage algorithm, and many researchers have improved the inference speed and the detection accuracy based on these models [15–17]. Xu et al. [18] proposed an improved CBF module and replaced the bottleneck CSP module with the Specter module, which improved the detection accuracy and detection speed of the YOLOv5s model. Similarly, CIoU Loss function was used in training YOLOv5s to decrease the missed detection rate and false detection rate [19]. Yang et al. [20] introduced BCAM into the network and added BCAM between the backbone neck of the YOLOv5s model, which obtained a high detection accuracy. As for dragon fruit detection, some researchers have made relevant contributions in detection. For example, the backbone of YOLOv4 was replaced with Mobilenet-v3 to detect dragon fruit, thereby reducing the model size and improving speed. However, this method makes the average precision of the model decrease slightly [21]. Zhang et al. [22] proposed an improved YOLOv5s to accurately detect dragon fruits under different light conditions. These methods above just locate and classify the fruits, which is difficult to apply to picking just some fruits, such as dragon fruit.

As for automatic picking of some special fruits, some researchers have made some contributions. Liang et al. [23] use YOLOv3 to detect litchi, and then use UNet to separate the stem one by one. Qi et al. [24] proposed a method for identifying litchi picking position. They used YOLOv5 to detect the litchi main stem and used PSPNet to segment the main stem and obtained the pixel coordinates of picking points in the segmented image. Sun et al. [25] proposed a novel method to detect the keypoint on the branch. They first obtained a candidate area according to the fruit-growing position and the fruit stem keypoint detection and used a multi-level feature fusion network to further detect the keypoint. The methods mentioned above not only detect the fruit, but also detect the stems or branches for accurately picking. However, the main problem of these methods is that the methods are not new enough and the accuracy is low.

To address the difficulties in the automatic picking of dragon fruit due to its postural diversity, the goal of this work is to propose a method to classify dragon fruit according to its growth posture and detect the head and root position of dragon fruit to identify the growth direction and realize automatic picking. We have developed this method using an edge computing machine and built a dragon fruit picking system. Using the picking system, we also have conducted some picking experiments.

The main contributions of this paper are as follows:

1. For the first time, dragon fruits are classified into two categories according to their postures. YOLOv7 is used to locate and classify dragon fruits, and some hot models are compared.
2. We propose a PSP-Ellipse method for endpoints detection and compare the proposed method with two kinds of keypoint regression method.
3. We have built a set of dragon fruit picking systems. To a certain extent, the problem of the picking difficulty caused by the complex growth posture of dragon fruit has been solved from the perspective of vision.

## 2. Materials and Methods

### 2.1. Dataset Construction

The dragon fruit pictures used in this article were taken from orchard Lile agricultural, Nanjing city. The camera used for collecting images is MV-CE200-10UC. They were taken under different weather conditions (cloudy and sunny), day times (morning, noon, evening and night), lighting conditions (strong light, weak light and artificial light) and shooting distances. All these pictures under complex natural environments make the network model more robust. We randomly cropped, randomly rotated and brightness modified the original pictures for data augmentation, and 2458 pictures used for the object detection model were obtained after manually eliminating invalid pictures. Labelme software was used to manually mark the rectangular box area and generate label files.

In the subsequent picking experiments, the camera was fixed eye-to-hand and mounted on the base of the robotic arm. According to the postures of the dragon fruit in the camera view, this paper classifies dragon fruit into two categories. First, when a dragon fruit grows on the left or right side of its branch, it is named as the dragon fruit in the side (DF\_S). This category of dragon fruit needs to detect its position, growth direction and endpoints position to provide a basis for the mechanical arm to pick. Second, when a dragon fruit grows on the front side of its branch, it is named as the dragon fruit in the front (DF\_F). This category of dragon fruit can be picked by detecting its position directly.

By using some of the label files and original pictures, we cropped all of the dragon fruits named DF\_S to train, validate and test the semantic segmentation model. Table 1 shows the datasets constructed.

**Table 1.** The dataset for object detection and semantic segmentation.

Dataset	Training	Validation	Test
Object detection	1966	196	296
Semantic segmentation	1279	143	146

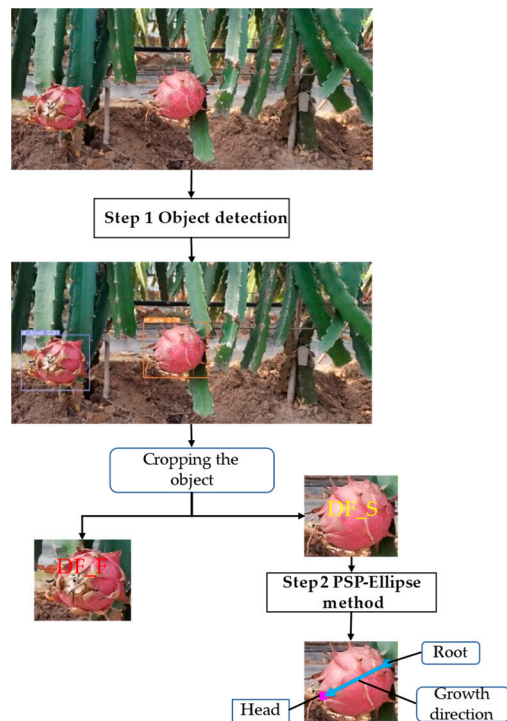
Figure 1 shows some pictures in the dataset taken at different times and includes the two categories of dragon fruit.



**Figure 1.** Data of two categories of dragon fruit taken at different times. DF\_S denotes dragon fruits in the side, and DF\_F denotes dragon fruits in the front.

## 2.2. Overview of the Method Proposed in this Paper

The flow chart of the method proposed in this paper is shown in Figure 2, which can be divided into the following two steps.



**Figure 2.** The flow chart of the proposed method.

- (1) Step 1: The dragon fruit image is processed by the trained YOLOv7 network to locate the fruit objects and classify them into DF\_S and DF\_F, and the DF\_S is cropped for further processing.
- (2) Step 2: The cropped image is processed by the PSP-Ellipse method proposed in this paper to detect the head and root of the dragon fruit.

## 2.3. Methodology

### 2.3.1. Related Theoretical Background

YOLO series models have been widely used in the agricultural field. YOLOv7 is currently the better performing object detector [26]. After pre-processing, the original image first enters the backbone network, which serves to extract features of several scales. Then, it enters the detection head to integrate the features extracted by the backbone network at several scales and perform feature fusion. Finally, the output of the YOLOv7 is obtained. In this paper, YOLOv7 is used for classifying the dragon fruit. The DF\_S detected will further be processed.

PSPNet is an efficient semantic segmentation network that classifies each pixel of the input image [27]. The core parts of this network are the feature extraction network and the Pyramid Pooling Module. The input images first enter the backbone to obtain the feature (F<sub>m</sub>). Some new features are obtained at different scales by pooling layers of different sizes; then, these features are upsampled separately to obtain the original scale features, and the newly obtained features are fused with the initial features. Finally, the segmented dragon fruit image is obtained by convolutional feature integration.

Ellipse fitting based on minimizing algebraic distance is a classic image processing algorithm to recognize the ellipse shape in the image. In the two-dimensional plane coordinate system, the ellipse can generally be expressed by Formula (1). The algebraic



distance from a point on the plane to the elliptic curve is defined as shown in Formula (2). For a given point set, Formula (3) is to be solved.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (1)$$

$$H_i = (x_i^2, x_i y_i, y_i^2, x_i, y_i, 1)$$

$$s = (A, B, C, D, E, F) \quad (2)$$

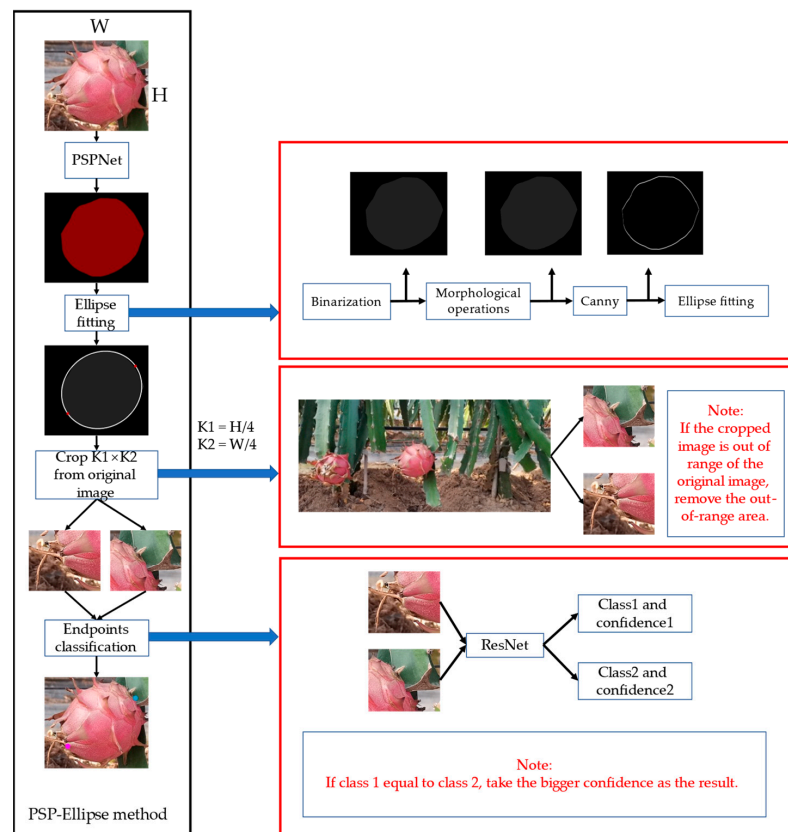
$$D_i = H_i \odot s$$

$$s^* = \text{Min} \sum_{i=1}^n D_i^2 \quad (3)$$

where  $A, B, C, D, E$  and  $F$  are coefficients of arbitrary elliptic equations, and  $x$  and  $y$  represent the coordinates of any point on the plane.

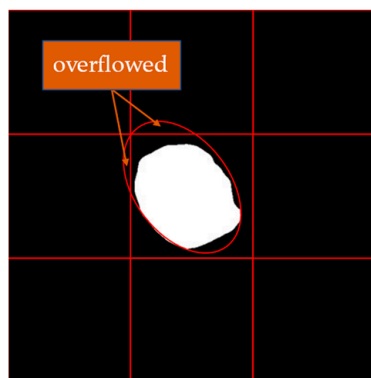
### 2.3.2. PSP-Ellipse Method for Endpoints Detection

The PSP-Ellipse method can be divided into three steps. First, the cropped images from YOLOv7 first enter the PSPNet to produce a segmented image. Second, the segmented image is processed by binarization, morphological operations, edges extraction and ellipse fitting. We use the fitted ellipse to calculate the position of the two endpoints of the long axis, and we regard the two endpoints as the endpoints of the dragon fruit. Third, taking the two endpoints obtained by ellipse fitting as the center, we crop two  $K1 \times K2$  rectangular areas from the original image and classify the rectangular image with a trained convolution neural network, which can directly output whether a point is the head or the root.  $K1$  and  $K2$  are calculated according to the size of the cropped dragon fruit image. Figure 3 shows the schematic of the PSP-Ellipse method.



**Figure 3.** The black box is the flowchart of the PSP-Ellipse method. The red box is a detailed description of each step. The red font is a note for this operation.

We have found two potential problems in the second step. First, the ellipse fitted by a small number of images will exceed the size of the segmented image and influence the subsequent calculations. Therefore, we will fill the surrounding of the segmented image with 0 value pixels as shown in Figure 4. Second, it was found that some noise points may influence ellipse fitting. Therefore, for all the ellipses fitted, we will calculate their IoU with the dragon fruit area; then, the ellipse with the largest IoU with the dragon fruit area is the fitting result.



**Figure 4.** The schematic of filling pixels.

In the third step, there are some details to handle. When the fruit appears near the edge of the original image, one of the rectangular areas may exceed the image range. For this, we will remove the parts beyond the image range. In addition, when ResNet processes two rectangular area images with the same output results, we believe that the one with a higher prediction confidence is correct.

#### 2.4. Experimental Environment and Training Strategies

The training environment for the YOLOv7 network and PSPNet is shown as follows: the operation system is Windows 10, the GPU is NVIDIA GeForce RTX 3060Ti, the CPU is AMD Ryzen 5 3600X and the deep learning framework is Pytorch.

For the training of the YOLOv7 network, we use mosaic and mixup for data enhancement, and the probabilities are 1.0 and 0.15. In label assignment, the intersection over union (IoU) threshold is 0.2, the aspect ratio threshold of the anchor is 4.0. Other hyperparameters used are as follows: the training epoch is 300, the training image size is  $640 \times 640$ , the initial learning rate is 0.01 and the parameters in convolution kernel are initialized randomly according to the Gaussian distribution. The training ultimately saves the weight file with the least loss on the validation set.

For the training of PSPNet, the hyperparameters used for training are shown as follows: the input image size is  $224 \times 224$ , the training epoch is 100, the batch size is 2, the weight decay coefficient is  $10^{-4}$ , the initial learning rate is  $10^{-2}$ , the learning rate decay type is cosine function, the minimum learning rate is  $10^{-4}$ , the optimizer is SGD and the momentum is 0.9. In this paper, the number of pixels in the background and the number of dragon fruit pixels may be unbalanced, so dice loss is used to reduce the effect of an unbalanced sample size. The training ultimately saves the weight file with the least loss on the validation set.

### 3. Results

All of the evaluation metrics used in this paper are described as follows.

Precision (P), recall (R), average precision (AP) and mean average precision (mAP) are used in the test experiments of YOLOv7. P, R, intersection over union (IoU) and mean intersection over union (MIoU) are used in the test experiments of PSPNet. Distance error (DE), angle error (AE) and classification accuracy (CA) are used in the test experiments of PSP-Ellipse. They are calculated as follows:

$$P = \frac{TP}{TP + FP} \quad (4)$$

$$R = \frac{TP}{TP + FN} \quad (5)$$

$$AP = \int_0^1 P(R)dR \quad (6)$$

$$mAP = \frac{\sum_{i=1}^n AP_i}{n} \quad (7)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (8)$$

$$MIoU = \frac{\sum_{i=0}^k IoU_i}{k + 1} \quad (9)$$

$$DE = \frac{\sqrt{(\hat{x}_h - x_h)^2 + (\hat{y}_h - y_h)^2} + \sqrt{(\hat{x}_r - x_r)^2 + (\hat{y}_r - y_r)^2}}{2} \quad (10)$$

$$AE = \left| \arctan\left(\frac{y_h - y_r}{x_h - x_r}\right) - \arctan\left(\frac{\hat{y}_h - \hat{y}_r}{\hat{x}_h - \hat{x}_r}\right) \right| \quad (11)$$

$$CA = \frac{T}{T + F} \quad (12)$$

where  $TP$  is the number of positive samples predicted to be positive class;  $FP$  is the number of negative samples predicted to be positive class;  $FN$  is the number of positive samples predicted to be negative class;  $AP$  is the area below the PR curve;  $mAP$  is the mean  $AP$  value for each category; and  $n$  and  $k$  represent the number of categories in object detection and semantic segmentation. In this paper,  $n = 2$  and  $k = 1$ , and  $x$  and  $y$  are horizontal and vertical coordinates of endpoints. The subscripts  $h$  and  $r$  represent the head and root, respectively. The superscript means it is ground truth; the other is the predicted results.  $T$  and  $F$  are the number of truly classified samples and falsely classified samples in endpoints classification.

### 3.1. Training and Test Results of YOLOv7

Using the dataset in Table 1, we train the YOLOv7 network. The loss of the training process is shown in Figure 5.

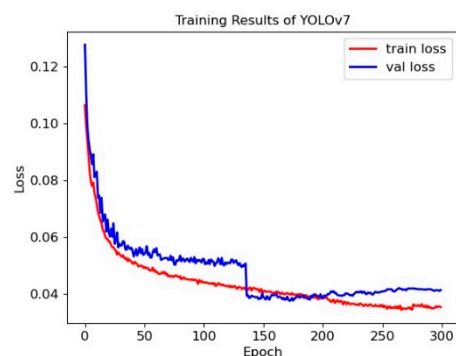


Figure 5. Training results of the YOLOv7 network.

As shown in Figure 5, the loss in the training set starts to drop faster and almost stops dropping later. The loss of the validation set starts to drop faster and drops sharply when

the training reaches 136 epochs, indicating that the model jumps out of the local optimal point and almost stops dropping at the later epoch.

To verify that the trained model has better robustness, the model is tested. The number of test images is 296, the number of labels for DF\_S and DF\_F is 245 and 212, respectively, the IoU threshold for non-maximum suppression is 0.5, the input image size is  $640 \times 640$ , and the batch size is 32. Table 2 shows the test results.

**Table 2.** Test results of YOLOv7.

Class	P	R	AP
DF_S	0.819	0.905	0.914
DF_F	0.870	0.943	0.95
Average	0.844	0.924	0.932

As shown in Table 2, the precision of the model is significantly lower than the recall, which means that the classification error of the model is large. A possible reason for this is that part of the fruit posture is between the two categories, which is difficult to strictly distinguish. We also found that the model performs better in detecting DF\_F due to more samples of DF\_F in training.

To further verify that the object detector used in this paper is superior, it is compared with other commonly used object detectors, and the comparison results are shown in Table 3.

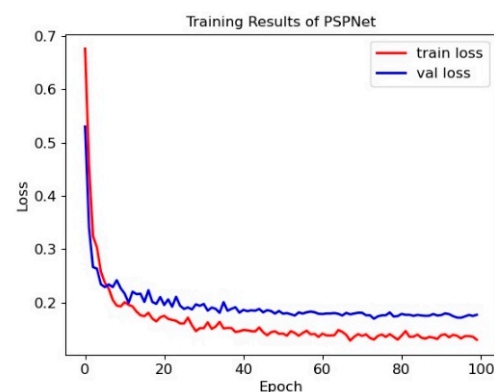
**Table 3.** Comparison results of YOLOv7 and other networks.

Networks	P	R	mAP
SSD	0.731	0.860	0.825
YOLOv4	0.662	0.763	0.631
YOLOv5	0.767	0.848	0.839
YOLOX	0.795	0.882	0.879
YOLOv6 [28]	0.779	0.868	0.847
YOLOv7	0.844	0.924	0.932

Comparative experimental results show that the accuracy of YOLOv7 is higher than some early networks. Compared with SSD, YOLOv4, YOLOv5, YOLOX and YOLOv6, the precision of YOLOv7 was improved by 0.113, 0.182, 0.077, 0.049 and 0.065, the recall was improved by 0.064, 0.161, 0.076, 0.042 and 0.056, and the average precision was improved by 0.107, 0.301, 0.093, 0.053 and 0.085.

### 3.2. Training and Test Results of PSPNet

Using the dataset in Table 1, we train the PSPNet. The loss of the training process is shown in Figure 6.



**Figure 6.** Training results of the PSPNet.



As shown in Figure 6, the loss in the training set and validation set start to drop faster and almost stop dropping later.

To verify that the trained model has better robustness, the model is tested. The number of test images is 146, the input image size is  $224 \times 224$ , and the batch size is 10. Table 4 shows the test results.

**Table 4.** Test results of PSPNet.

Class	P	R	IoU
Background	0.967	0.904	0.870
Dragon fruit	0.951	0.982	0.942
Average	0.959	0.943	0.906

As shown in Table 4, we pay more attention to the performance of the model in segmenting dragon fruit. All the precision, recall and IoU reach a very high value due to the dice loss in training.

We also compared the performance of PSPNet with other commonly used semantic segmentation models, and the results are shown in Table 5.

**Table 5.** Comparison results of PSPNet and other networks.

Networks	P	R	MIoU
FCN	0.929	0.904	0.885
DeepLabv3	0.921	0.893	0.843
LR-ASPP	0.914	0.871	0.802
UNet	0.938	0.917	0.892
PSPNet	0.959	0.943	0.906

Comparative experimental results show that the accuracy of PSPNet is higher than other commonly used semantic segmentation networks. Compared with FCN, DeepLabv3, LR-ASPP and UNet, the precision of PSPNet was improved by 0.03, 0.038, 0.045 and 0.021, the recall was improved by 0.039, 0.05, 0.072 and 0.026, and the mean intersection over union was improved by 0.021, 0.063, 0.104 and 0.014. Figure 7 shows the mixing results of the image segmented by PSPNet and the dragon fruit image.



**Figure 7.** The mixing results of the image segmented by PSPNet and the dragon fruit image.

### 3.3. Endpoints Detection Results of PSP-Ellipse Method

The head and root have different features. For example, there may be flowers near the head, and the shape of the fruit near the head may be different from the shape of the fruit near the root. There are always branches near the root. Convolutional neural networks can learn these features themselves and distinguish between the head and the root through these features. Based on this, we use ResNet to classify endpoints, which means that ResNet directly outputs classification results after processing the two rectangular area

images. Some other classification models are also compared with ResNet to verify its effectiveness. The classification models are trained by using some cropped rectangular images near the two endpoints. To verify the effectiveness of the PSP-Ellipse method proposed in this paper, the end-to-end test is carried out. We use 200 ground truth images which are cropped using YOLOv7 for the test, and these images are manually labeled the endpoints position.

Keypoint regression is also a method for detecting the endpoints of dragon fruit. We conducted two kinds of keypoint regression method. The first is direct regression. This method adds a fully connected layer to the end of a convolutional neural network, and finally outputs four values, which correspond to the head and root coordinates. The second is based on heatmaps. This method is to make a circular heatmap with the keypoint as the center. The closer to the center point, the larger the value, and the further away from the center point, the smaller the value. Therefore, we trained the ResNet and UNet for the two kinds of keypoint regression method as comparative experiments. The results are shown in Table 6.

**Table 6.** Comparison results of several endpoint detection methods.

Methods	DE (Pixels)	AE (°)	CA
PSP-Ellipse-ResNet	39.8	4.3	0.92
PSP-Ellipse-LeNet	39.8	4.3	0.87
PSP-Ellipse-VGGNet	39.8	4.3	0.90
PSP-Ellipse-MobileNet	39.8	4.3	0.89
Direct regression-ResNet	114.5	14.6	0.42
Heatmap-UNet	81.4	8.2	0.52

PSP-Ellipse-ResNet refers to the PSP-Ellipse method using ResNet for endpoints classification, PSP-Ellipse-LeNet refers to the PSP-Ellipse method using LeNet for endpoints classification, PSP-Ellipse-VGGNet refers to the PSP-Ellipse method using VGGNet for endpoints classification, PSP-Ellipse-MobileNet refers to the PSP-Ellipse method using MobileNet for endpoints classification, Direct regression-ResNet refers to the direct endpoints regression method using ResNet, and Heatmap-UNet refers to the endpoints regression based on the heatmap using UNet.

As shown in Table 6, using ResNet for endpoints classification has a higher accuracy. Compared with LeNet, VGGNet and MobileNet, the classification accuracy improves by 0.05, 0.02 and 0.03. In addition, the PSP-Ellipse method for endpoints detection obtains a great improvement compared with the keypoint regression method. As for an original image of size  $4608 \times 2064$ , using the PSP-Ellipse method proposed in this paper, the distance error and angle error are only 39.8 pixels and  $4.3^\circ$ . Compared with the two kinds of keypoint regression method, the distance error is decreased by 65.2% and 51.1%, and the angle error is decreased by 70.5% and 47.6%. At the same time, the classification accuracy is increased by 119.0% and 76.9%. The experimental results fully demonstrate that the PSP-Ellipse method proposed in this paper is very meaningful for detecting the endpoints of dragon fruit. Figure 8 shows some detection results of PSP-Ellipse. The blue point represents the position of the root, and the red point represents the position of the head.

### 3.4. Orchard Picking Experiment

In order to verify that the method proposed in this paper can be used to pick dragon fruit, we conducted an orchard picking experiment. The picking robot is shown in Figure 9. The edge computing machine is Jetson AGX Orin 32GB, the depth camera is ZED Mini and the mechanical arm is S6H4D\_Plus.

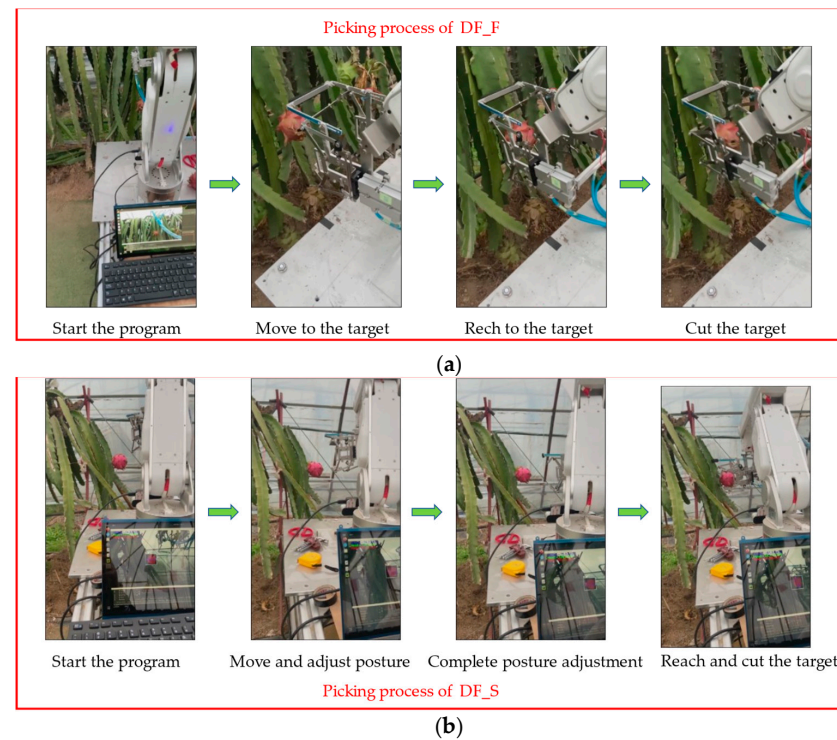
For the two classifications of dragon fruit, the picking strategy is slightly different. For DF\_F, directly send the three-coordinate command of moving to the center point to the mechanical arm. For DF\_S, we need to calculate its growth direction and posture according to the endpoints found by the method proposed in this paper; then, control the mechanical arm to approach the fruit in a specific posture to achieve picking. Figure 10 shows the picking process of the two-classification dragon fruit.



**Figure 8.** Dragon fruit detection results of the method proposed in this paper. df\_front denotes DF\_F and df\_side denotes DF\_S.



**Figure 9.** Dragon fruit picking experiment scene in orchard.



**Figure 10.** Picking process of the two categories of dragon fruit. (a) Picking process of DF\_F; (b) picking process of DF\_S.

#### 4. Conclusions

In order to solve the problem of picking difficulties caused by the complex growth of dragon fruit, this paper proposes a method for dragon fruit detection, including fruit detection and endpoints detection, to enable a picking robot's visual system to locate and determine the posture of dragon fruit. YOLOv7 is used for locating the dragon fruit and classifying dragon fruit into DF\_S and DF\_F. PSP-Ellipse is proposed for endpoints detection.

Several technological findings are made regarding the performances between competing algorithms along the two key steps of dragon fruit detection and endpoints detection. In dragon fruit detection, the precision, recall and average precision of YOLOv7 are 0.844, 0.924, 0.932, respectively; YOLOv7 therefore outperforms fellow object detection models in all metrics, including SSD, YOLOv4, YOLOv5, YOLOX and YOLOv6. In dragon fruit segmentation, the precision, recall and mean intersection over union of PSPNet are 0.959, 0.943, 0.906, respectively. PSPNet therefore also outperforms other commonly used models. In endpoints detection, the distance error, angle error and classification accuracy of PSP-Ellipse are 39.8 pixels, 4.3° and 0.92. Compared with the two kinds of keypoint regression method, the distance error is decreased by 65.2% and 51.1%, the angle error is decreased by 70.5% and 47.6%, and the classification accuracy increased by 119.0% and 76.9%. We built a dragon fruit picking system using the method proposed in this paper and conducted orchard picking experiments. We succeeded to pick some dragon fruit with different postures which further proves that the method proposed in this paper is effective.

**Author Contributions:** Conceptualization, J.W.; data curation, J.Z. and Y.Z.; formal analysis, Y.Z.; project administration, Y.Z.; resources, J.W.; software, J.Z.; supervision, J.W.; validation, J.Z. and Y.Z.; visualization, J.Z.; writing—original draft, J.Z.; and writing—review and editing, J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by Jiangsu Province Agricultural Science and Technology Independent Innovation Project (CX(22)3099), Key R&D Program of Jiangsu Modern Agricultural Machinery Equipment and Technology Promotion Project (NJ2021-18), the Key R&D plan of Jiangsu Province (BE2021016-2), emergency science and technology project of National Forestry and Grassland Administration (202202-3) and Self-made Experimental Teaching Instrument Project of Nanjing Forestry University (nlzzyq202106).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available on request from the corresponding author.

**Acknowledgments:** In addition, we would like to thank Lile Agriculture Science and Technology Ltd., Nanjing City, Jiangsu Province, for the dragon fruit data provided and Zhou Lei for proofreading the draft.

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

#### References

1. Wang, Z.; Xun, Y.; Wang, Y.; Yang, Q. Review of Smart Robots for Fruit and Vegetable Picking in Agriculture. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 33–54. [[CrossRef](#)]
2. Zhong, Z.; Xiong, J.; Zheng, Z.; Liu, B.; Liao, S.; Huo, Z.; Yang, Z. A Method for Litchi Picking Points Calculation in Natural Environment Based on Main Fruit Bearing Branch Detection. *Comput. Electron. Agric.* **2021**, *189*, 106398. [[CrossRef](#)]
3. Li, C. Badminton Motion Capture with Visual Image Detection of Picking Robotics. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988142096907. [[CrossRef](#)]
4. Zheng, C.; Chen, P.; Pang, J.; Yang, X.; Chen, C.; Tu, S.; Xue, Y. A Mango Picking Vision Algorithm on Instance Segmentation and Key Point Detection from RGB Images in an Open Orchard. *Biosyst. Eng.* **2021**, *206*, 32–54. [[CrossRef](#)]
5. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [[CrossRef](#)]
6. Wang; Walsh. Koirala Mango Fruit Load Estimation Using a Video Based MangoYOLO—Kalman Filter—Hungarian Algorithm Method. *Sensors* **2019**, *19*, 2742. [[CrossRef](#)]



7. Moreira, G.; Magalhães, S.A.; Pinho, T.; dos Santos, F.N.; Cunha, M. Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato. *Agronomy* **2022**, *12*, 356. [[CrossRef](#)]
8. Fu, L.; Yang, Z.; Wu, F.; Zou, X.; Lin, J.; Cao, Y.; Duan, J. YOLO-Banana: A Lightweight Neural Network for Rapid Detection of Banana Bunches and Stalks in the Natural Environment. *Agronomy* **2022**, *12*, 391. [[CrossRef](#)]
9. 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](#)]
10. Han, B.-G.; Lee, J.-G.; Lim, K.-T.; Choi, D.-H. Design of a Scalable and Fast YOLO for Edge-Computing Devices. *Sensors* **2020**, *20*, 6779. [[CrossRef](#)]
11. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using Channel Pruning-Based YOLOv4 Deep Learning Algorithm for the Real-Time and Accurate Detection of Apple Flowers in Natural Environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [[CrossRef](#)]
12. Zheng, Z.; Xiong, J.; Lin, H.; Han, Y.; Sun, B.; Xie, Z.; Yang, Z.; Wang, C. A Method of Green Citrus Detection in Natural Environments Using a Deep Convolutional Neural Network. *Front. Plant Sci.* **2021**, *12*, 705737. [[CrossRef](#)]
13. Dong, C.; Zhang, Z.; Yue, J.; Zhou, L. Automatic Recognition of Strawberry Diseases and Pests Using Convolutional Neural Network. *Smart Agric. Technol.* **2021**, *1*, 100009. [[CrossRef](#)]
14. Veeranampalayam Sivakumar, A.N.; Li, J.; Scott, S.; Psota, E.; Jhala, A.J.; Luck, J.D.; Shi, Y. Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery. *Remote Sens.* **2020**, *12*, 2136. [[CrossRef](#)]
15. Zhang, Y.; Sun, Y.; Wang, Z.; Jiang, Y. YOLOv7-RAR for Urban Vehicle Detection. *Sensors* **2023**, *23*, 1801. [[CrossRef](#)]
16. Jin, X.; Liu, T.; Chen, Y.; Yu, J. Deep Learning-Based Weed Detection in Turf: A Review. *Agronomy* **2022**, *12*, 3051. [[CrossRef](#)]
17. Sun, Y.; Li, M.; Dong, R.; Chen, W.; Jiang, D. Vision-Based Detection of Bolt Loosening Using YOLOv5. *Sensors* **2022**, *22*, 5184. [[CrossRef](#)]
18. Xu, Z.; Huang, X.; Huang, Y.; Sun, H.; Wan, F. A Real-Time Zanthoxylum Target Detection Method for an Intelligent Picking Robot under a Complex Background, Based on an Improved YOLOv5s Architecture. *Sensors* **2022**, *22*, 682. [[CrossRef](#)]
19. Xue, J.; Cheng, F.; Li, Y.; Song, Y.; Mao, T. Detection of Farmland Obstacles Based on an Improved YOLOv5s Algorithm by Using CloU and Anchor Box Scale Clustering. *Sensors* **2022**, *22*, 1790. [[CrossRef](#)]
20. Yang, R.; Hu, Y.; Yao, Y.; Gao, M.; Liu, R. Fruit Target Detection Based on BCo-YOLOv5 Model. *Mob. Inf. Syst.* **2022**, *2022*, 8457173. [[CrossRef](#)]
21. Wang, J.; Gao, K.; Jiang, H.; Zhou, H. Method for Detecting Dragon Fruit Based on Improved Lightweight Convolutional Neural Network. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 218–225. [[CrossRef](#)]
22. Zhang, B.; Wang, R.; Zhang, H.; Yin, C.; Xia, Y.; Fu, M.; Fu, W. Dragon Fruit Detection in Natural Orchard Environment by Integrating Lightweight Network and Attention Mechanism. *Front. Plant Sci.* **2022**, *13*, 1040923. [[CrossRef](#)] [[PubMed](#)]
23. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A Visual Detection Method for Nighttime Litchi Fruits and Fruiting Stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [[CrossRef](#)]
24. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [[CrossRef](#)]
25. Sun, Q.; Chai, X.; Zeng, Z.; Zhou, G.; Sun, T. Multi-Level Feature Fusion for Fruit Bearing Branch Keypoint Detection. *Comput. Electron. Agric.* **2021**, *191*, 106479. [[CrossRef](#)]
26. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
27. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. *arXiv* **2017**, arXiv:1612.01105.
28. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.

**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.