remote sensing

MDPI

Article

Real-Time Detection of Winter Jujubes Based on Improved YOLOX-Nano Network

Zhouzhou Zheng [1], Yaohua Hu [2], Yichen Qiao [1], Xing Hu [1] and Yuxiang Huang [1,*]

[1] College of Mechanical and Electronic Engineering, Northwest A&F University, Xianyang 712100, China
[2] College of Optical, Mechanical and Electrical Engineering, Zhejiang A&F University, Hangzhou 311300, China
* Correspondence: hyx@nwsuaf.edu.cn; Tel.: +86-029-87091111

Abstract: Achieving rapid and accurate localization of winter jujubes in trees is an indispensable step for the development of automated harvesting equipment. Unlike larger fruits such as apples, winter jujube is smaller with a higher density and serious occlusion, which obliges higher requirements for the identification and positioning. To address the issues, an accurate winter jujube localization method using improved YOLOX-Nano network was proposed. First, a winter jujube dataset containing a variety of complex scenes, such as backlit, occluded, and different fields of view, was established to train our model. Then, to improve its feature learning ability, an attention feature enhancement module was designed to strengthen useful features and weaken irrelevant features. Moreover, DIoU loss was used to optimize training and obtain a more robust model. A 3D positioning error experiment and a comparative experiment were conducted to validate the effectiveness of our method. The comparative experiment results showed that our method outperforms the state-of-the-art object detection networks and the lightweight networks. Specifically, the precision, recall, and AP of our method reached 93.08%, 87.83%, and 95.56%, respectively. The positioning error experiment results showed that the average positioning errors of the X, Y, Z coordinate axis were 5.8 mm, 5.4 mm, and 3.8 mm, respectively. The model size is only 4.47 MB and can meet the requirements of winter jujube picking for detection accuracy, positioning errors, and the deployment of embedded systems.

Keywords: winter jujubes; YOLOX-Nano; attention feature enhancement; 3D positioning; DIoU loss

Citation: Zheng, Z.; Hu, Y.; Qiao, Y.; Hu, X.; Huang, Y. Real-Time Detection of Winter Jujubes Based on Improved YOLOX-Nano Network. *Remote Sens.* **2022**, *14*, 4833. https://doi.org/10.3390/rs14194833

Academic Editors: Anup Basu, Chengcai Leng and Hemanth Venkateswara

Received: 19 August 2022
Accepted: 23 September 2022
Published: 28 September 2022

1. Introduction

Winter jujubes (Ziziphus mauritiana) are planted on a large scale in China due to its good taste and rich nutritional value. Owing to the complex environment and the randomness of fruit on unstructured jujube orchards, winter jujube harvesting relies heavily on manual picking, which is time consuming and high intensity. Therefore, it is urgent to develop an intelligent winter jujube harvesting device.

Intelligent fruit harvesting devices rely heavily on intelligent fruit recognition and positioning algorithms. Many works based on machine vision have been reported covering apples [1,2], kiwifruits [3], tomatoes [4], etc., which can be divided into two categories, including traditional image-processing methods and deep learning-based methods. For instance, Wang et al. [5] described an apple recognition method using the K-means clustering algorithm which achieved the extraction of occluded apple candidate regions in natural scenes. Further, an apple recognition system using a vector median filter and mathematical morphology operation was built. The method reached a recognition accuracy of 89% with an average recognition time of 0.352 s per image [1]. Tian et al. [6] proposed a graph-based segmentation algorithm combined with the depth information of test images to obtain the apples' location information, which reached the recognition rate of 96.61%. Traditional image-processing methods have difficulties in obtaining the optimal parameters to adapt to complex orchard environments. To solve this issue, deep convolution neural networks (DCNN) have made remarkable progress in parameter optimization and feature learning

and have been widely used in fruit classification and recognition [7]. Existing DCNN approaches can be roughly divided into three categories: image classification [8], object detection [9], and image segmentation [10], in which object detection-based methods are widely applied to locate fruits for intelligent picking.

Deep learning-based methods have been widely applied in many detection fields, including industrial defect detection [11,12], medical lesion detection [13,14] and quality inspection [15,16]. In recent years, object detection network-based deep learning has had tremendous developments and a series of representative networks including R-CNN [17], Fast R-CNN [18], YOLO [19,20], and SSD [21] have been reported, which can be divided into one-stage and two-stage networks. Two-stage object detection networks, such as R-CNN, Fast R-CNN, and Faster R-CNN, mainly include two steps in which region recommendation is utilized to generate regions that may contain targets, and then CNN is applied to classify these regions and provide a confidence. Many works about two-stage object detection networks are reported on fruit-picking fields. For instance, Fu et al. [7] achieved an average precision of 89.3% and recognition speed of 0.181 s per image in detecting apples using the Faster R-CNN-based architectures of VGG16. In addition, Faster R-CNN was utilized to estimate the locations for automated apple harvesting by identifying branches, apples, and trunks in the natural environment [22]. The aforementioned two-stage networks can accurately detect the locations of apples, but the recognition efficiency cannot meet the requirement of real-time localization due to its cumbersome region recommendation. To remedy the issue, one-stage networks abandoned regional recommendations and extract features of input image to predict the locations of objects. Among them, YOLO series and SSD are the most typical architectures, and a series of applications were also reported. For instance, Sozzi et al. [23] applied YOLOv3, YOLOv4, and YOLOv5 object-detection algorithms to achieve bunch detection in white grape varieties. An improved YOLOX-S with a new multi-scale feature integration structure was proposed to detect kiwifruit for automated harvesting [24]. A winter-jujube grading robot combined with the YOLOv3 algorithm was applied to sort winter jujube, which reported a mAP of 94.78% with a computational time of 0.042 s per image [25]. Moreover, Li et al. [26] introduced Efficient Channel Attention (ECA) and Coordinate Attention (CA) mechanisms on YOLOv5 to improve the accuracy of the model for jujube detection. At present, less research about the object detection of winter jujubes are reported, but apple target detection based on deep learning is relatively complete compared with other fruits' object detection. For example, Wu et al. [27] applied an improved YOLOv4 network to accomplish apple detection in complex scenes, and the study reached a detection accuracy of 95.52% with a computational time of 0.339 s per image of 416 × 416 pixels. Yan et al. [28] improved YOLOv5 with an SE module for apple detection in different environments and obtained a mAP of 86.75% with a computational time of 0.015 s per image. To further simplify the model and improve detection efficiency, a channel pruned YOLOv5s model was proposed to detect apple fruitlets accurately, which achieved satisfactory results under both backlight and direct sunlight conditions [29]. The one-stage networks, such as YOLOv4 and YOLOv5, perform well in apple recognition. However, few research studies based on deep learning have been studied in winter jujube detection. Different from apples, as shown in Figure 1, the challenge of winter jujubes detection includes its smaller fruit size, lower contrast between the fruit and the background, and higher density between fruits.

To improve the detection accuracy, the above research mainly applied various attention mechanisms to bring out useful features and weaken irrelevant features. In recent years, a range of original studies on attention mechanisms were conducted to strengthen feature learning ability [30]. For instance, Hu et al. [31] considered the importance of different channels and introduced channel weights to reflect the importance between channels in the Squeeze-and-Excitation Network (SENet). Furthermore, the Efficient Channel Attention network (ECANet) [32] established the relationship between channels more efficiently and without reducing the dimension. However, these methods can only focus on the importance of different channels and cannot build the relationship between channels and

spatial features. To address this issue, a Convolution Block Attention Module (CBAM) was proposed to take into account channel and spatial features for feature enhancement [33]. CBAM achieved feature enhancement by concatenating the spatial attention mechanism and the channel attention mechanism. However, this concatenation method reduces the computational efficiency. In recent years, some self-attention methods were proposed that integrated channel and spatial attention using a parallel manner, such as the Dual Attention Network (DANet) [34], which achieved satisfactory results on Cityscapes, PASCAL VOC, and COCO Stuff datasets. However, the method has higher computational complexity and larger parameter amounts.

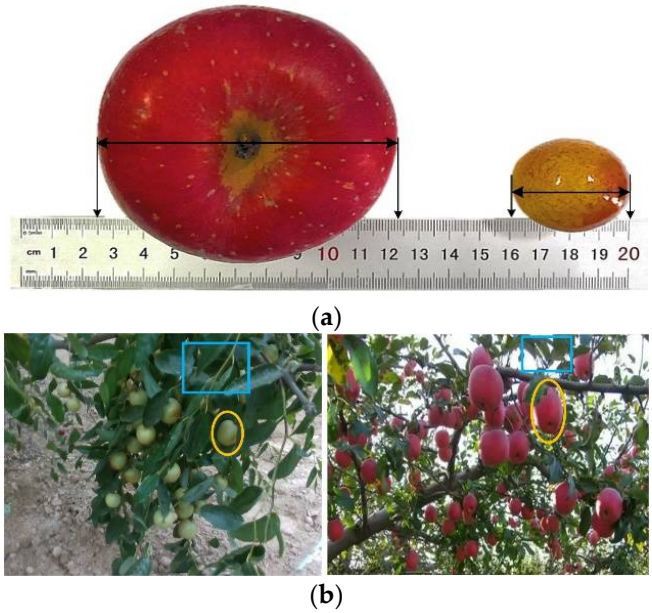

Figure 1. Challenges faced by winter jujube detection compared with apples. Yellow and blue boxes represent fruits and background, respectively. (**a**) Smaller fruit size. (**b**) Lower contrast and higher density.

It is equally important for smart harvesting to ensure the recognition speed. To accomplish this goal, two directions of research were carried out: one was to choose a lightweight model, and the other was to prune existing models to achieve a light weight. Zhang et al. [35] proposed a lightweight RTSD-Net deep neural network to embed in Jetson Nano for real-time strawberry detection, which achieved satisfactory results. A channel pruning-based YOLOv4 model was used to detect apple flowers in natural environments, and the experimental results showed that the model size was reduced by 231.51 MB and the new model size is only 12.46 MB [36]. Similarly, Fu et al. reduced the model weight from 244 to 137 MB by pruning for detection of banana bunches and stalks [37]. Reducing model size has become a prerequisite for embedded systems.

To overcome the challenges of winter jujube detection, an improved YOLOX-Nano network was proposed to improve detection accuracy and reduce the model size. The main contributions were as follows:

(1) An attention feature enhancement (AFE) module was proposed to establish connections between channels and spatial features for maximizing feature utilization.
(2) DIoU loss was used to replace IoU loss to optimize training and obtain a more robust model.
(3) A positioning error evaluation method was proposed to measure positioning error.
(4) Model size was only 4.47 MB and can meet the requirement of embedded systems deployment.
(5) An improved lightweight YOLOX-Nano network combined with an RGB-D camera was applied to provide 3D coordinates.

2. Materials and Methods

2.1. Image Acquisition

The images used in the experiment were obtained in a winter jujube orchard of Yangling District, Shaanxi Province in China from 1 September to 15 September 2021. An iPhone 11 with a resolution of 3024 × 3024 pixels was used to capture test images, as shown in Figure 2. To adapt for training the network and to save training memory, all the images were resized to 640 × 640 pixels for training and testing. The dataset contains 632 images and the corresponding 3659 winter jujubes were labeled in different environments including 39% front light, 32% backlight, and 29% occluded scene. The high-quality labels were marked using LabelImg for network training learning.

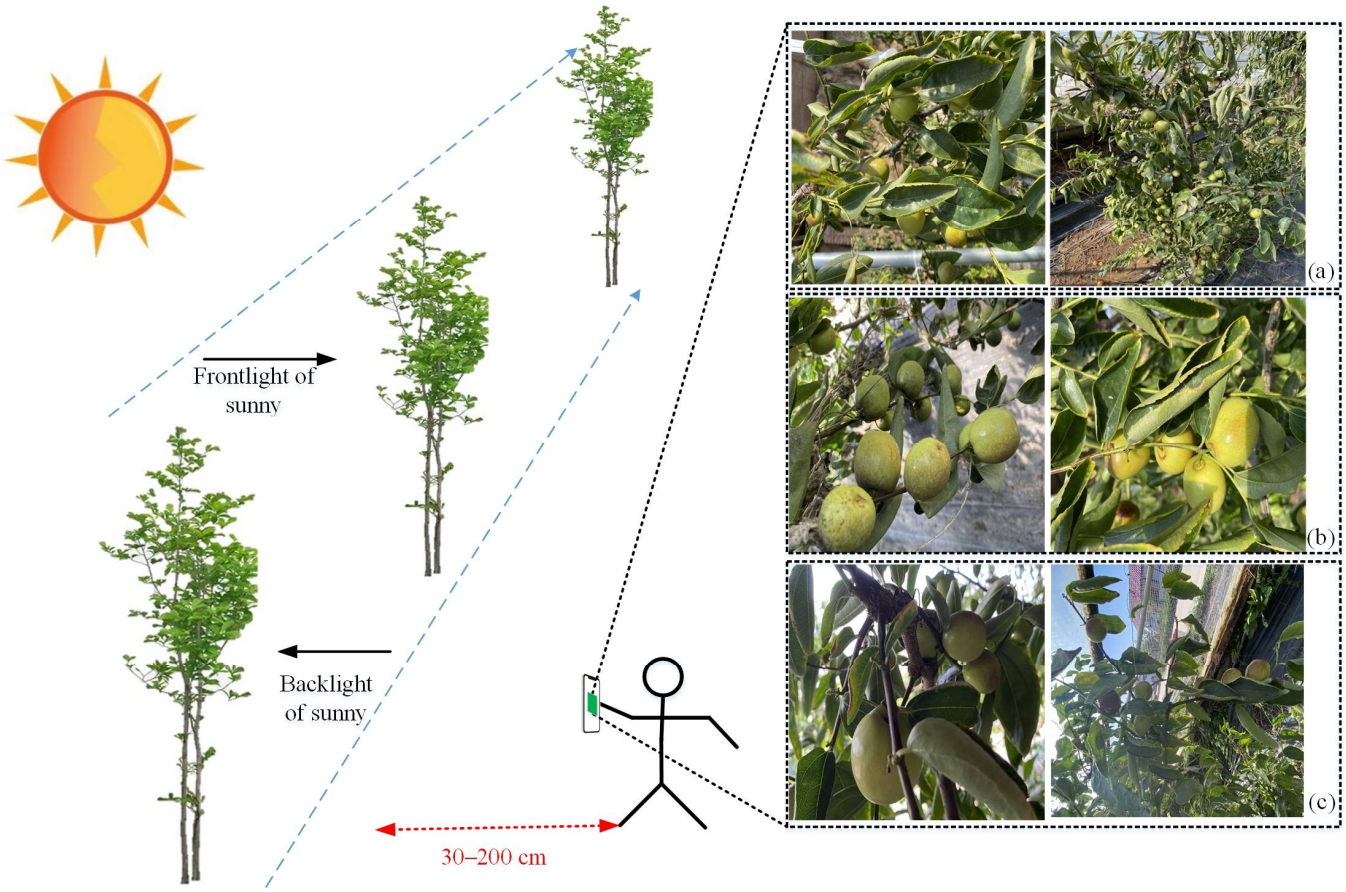

Figure 2. The acquisition process of the dataset. (**a**) Images under occlusion scene; (**b**) Images under front light; (**c**) Images under backlight scene.

2.2. Localization System

To test the proposed algorithm and achieve 3D positioning, as shown in Figure 3, a winter jujube localization system was built, including an RGB-D camera, a high-performance PC, a rangefinder, a tripod, a USB3.0 data cable, a calibration cardboard, and some winter jujubes. The RGB-D camera (Real sense D435i camera, Intel Corporation, Hillsboro, OR, USA) was used to obtain the localization information of winter jujubes. The tripod was utilized to carry the RGB-D camera and adjust the camera pose. The distance between the camera and winter jujubes was controlled by rangefinder. To evaluate the accuracy of positioning, a calibration cardboard with 30 × 30 mm cells was applied to control the relative position of winter jujubes. The USB3.0 data cable performed the transfer of information between the PC and RGB-D camera.

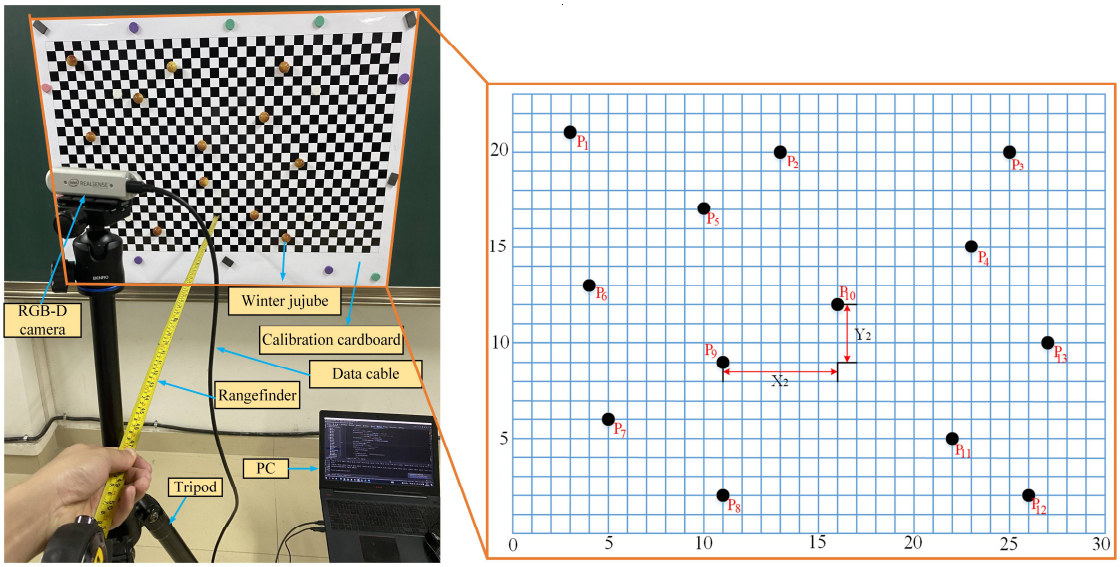

Figure 3. Winter jujube localization system.

2.3. Methodologies

With the development of high-performance GPUs [38], deep learning models have had great success due to their powerful parameters of self-optimization and feature self-learning capabilities. Object detection networks, as one of the most important branches in deep learning methods, are widely used in many different fields, which can be divided into two categories including anchor-based and anchor-free deep learning approaches. Anchor-based methods are domain-specific and less generalized that optimal anchor box need to be found using the K-means clustering algorithm for different datasets [39]. Thus, the state-of-the-art anchor-free deep learning architecture, namely, the YOLOX network, was chosen as the secondary development framework in our work. It combines the advantages of YOLOv3, v4, and v5 and proposes some new strategies, such as the anchor-free strategy, decoupled head, and SimOTA, which outperformed many existing one-stage and two-stage detectors on public datasets. The YOLOX network can be divided into 6 versions, including YOLOX-Nano, Tiny, S, M, L, and X according to the size of the model. To adapt to the deployment of embedded equipment, our work chose lightweight YOLOX-Nano for improvement.

2.3.1. Improved YOLOX-Nano

As shown in Figure 4, YOLOX mainly consists of three parts, including the backbone, neck, and YOLO head, respectively. The backbone feature extraction network of YOLOX is CSPDarkNet53, in which two novel modules called Focus and SPP are utilized to downsample feature maps and expand the receptive field, respectively. The Focus module can retain the feature map information to the maximum extent while performing downsampling operations. The SPP module implements feature extraction at different scales using 5×5, 9×9, and 13×13 pooling kernel sizes and obtains larger receptive fields. Multiple stacked CBS modules (Convolution, Batch Normalization, and SiLU layer) and CSP layers (CBS and Res_block modules) are used to transfer and extract features. In the neck stage, the structure of PA-FPN (FPN: feature pyramid network, PAN: path aggregation network) is used to efficiently fuse feature maps at different levels. Path aggregation greatly reduces the number of network layers by building bridges between different features. In the head stage, the YOLO head adapts the decoupled head strategy to perform classification and regression tasks separately, which is beneficial for accelerating the convergence of the network and obtaining a more robust model [39]. In our work, to enhance the learning ability of small objects, as shown in Figure 5, a novel attention feature enhancement (AFE) module was designed to strengthen feature extraction in the YOLOX-Nano network.

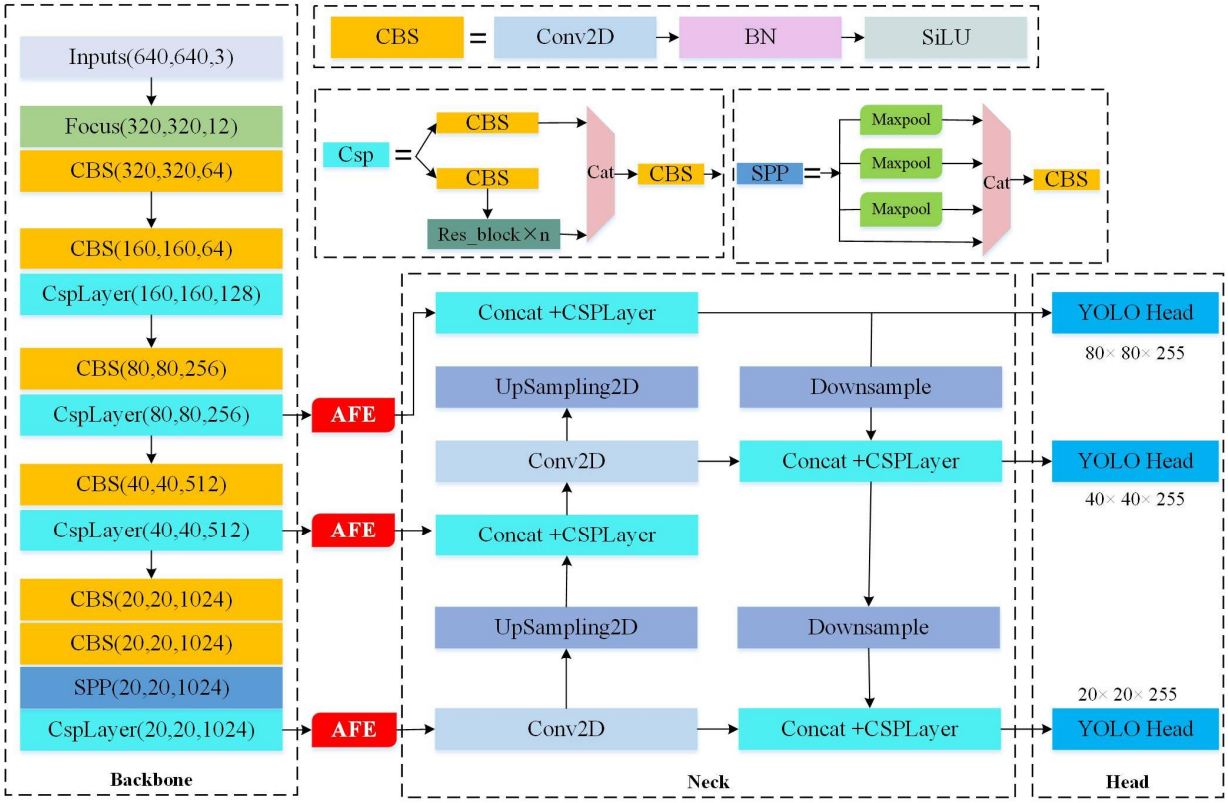

Figure 4. The structure of improved YOLOX-Nano.

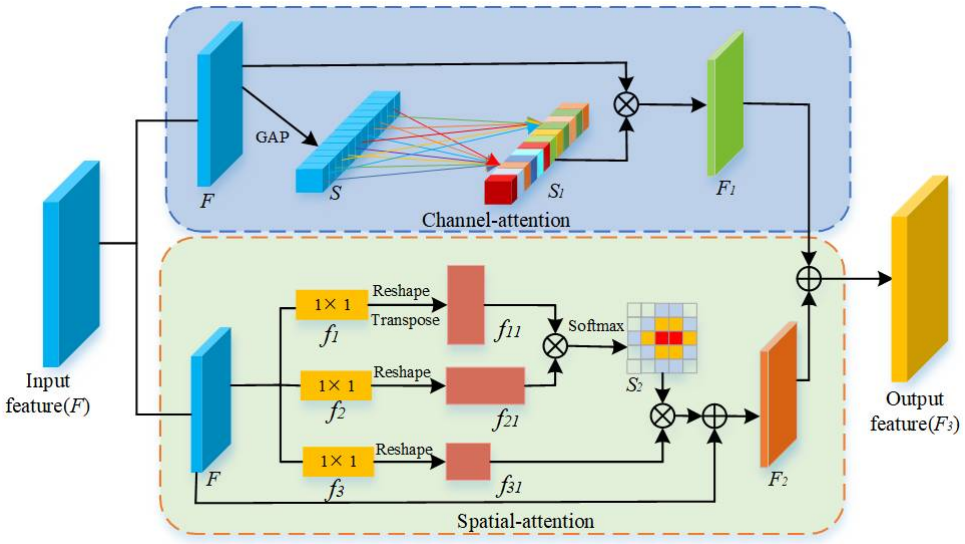

Figure 5. The architecture of attention feature enhancement module.

2.3.2. Attention Feature Enhancement Module

To remedy the problem of attention mechanisms described in the introduction, we rethought self-attention, aiming at reducing the number of parameters and performing channel and spatial attention in parallel. As shown in Figure 5, in our work, inspired by DANet [34] and ECANet [32], a novel AFE module was proposed to integrate channel and spatial information with smaller parameters, and strengthen insignificant feature map that can reflect object information. AFE speeds up feature transfer through parallel message transfer between the channel and spatial attention. Given the input feature $F \in R^{W \times H \times C}$, where W, H, and C are width, height, and channels, respectively. For the channel attention branch, first, global average pooling (*GAP*) is used to obtain channel

weights $S \in R^{1\times1\times C}$, then the dimension of channel weights is adjusted by *squeeze(sq)*, *transpose(tr) and unsqueeze(usq)* operations to fit 1D convolutions (*Conv1D*), which are utilized to build the relationship between channels. The new channel weights $S_1 \in R^{1\times1\times C}$ can be defined as

$$S_1 = Conv1D(GAP(F) \times sq \times tr) \times tr \times usq \tag{1}$$

Accordingly, the new feature maps after channel attention can be expressed as

$$F_1 = S_1 \otimes F \tag{2}$$

where $\otimes$ is multiplication operation.

In the spatial attention module, first, three *Conv1D* are used to obtain three feature maps $\{f_1, f_2, f_3\} \in R^{W\times H\times C}$, then, they are reshaped to $\{f_{11}, f_{21}, f_{31}\} \in R^{C\times N}(N = W \times H)$, to obtain spatial attention map $S_2 \in R^{N\times N}$. The operations are performed as:

$$S_2 = \sigma(tr(f_{11}) \otimes f_{21}) \tag{3}$$

where σ is softmax activation function.

Finally, the spatial attention mechanism assigns spatial weight to the original feature map for getting new spatial feature maps $F_2 \in R^{W\times H\times C}$. It can be defined as

$$F_2 = Re(\alpha(tr(S_2) \otimes f_{31})) \oplus F \tag{4}$$

where α and Re are scale parameters and reshape the operation, α is initialized as 0 and gradually learns to assign more weight [40], and $\oplus$ is an added operation.

After acquiring the channel and spatial feature maps, the resulting feature map F_3 can be obtain as follows:

$$F_3 = F_1 \oplus F_2 \tag{5}$$

2.3.3. Loss Function

The goal of network training is to reduce the loss function and make the prediction box close to the ground truth box to obtain a more robust model. The loss function of object detection always consists of three parts, which are bounding box location loss (L_{bou}), confidence loss (L_{Conf}), and classification loss (L_{Cls}), respectively. L_{Conf} is used to determine whether there is an object in the predicted box, which is in fact a binary classification problem. The closer the confidence is to 1, the greater the probability of the existence of the target. L_{Cls} is applied to reflect error in object classification, Cross-Entropy (CE) and Binary Cross-Entropy (BCE) loss [41], as these two most common classifications of loss functions are widely used in multi-class and binary classification tasks, respectively. In YOLOX, L_{Cls} and L_{Conf} both use the BCE loss function.

In object detection tasks, IoU loss (L_{IoU}) [42] is the most common bounding box location loss and is used in many object detection models, such as Faster R-CNN, YOLOv3, YOLOX, etc. However, when the prediction box and ground truth box do not intersect and $L_{\mathrm{IoU}} = 0$, the network cannot be trained. Furthermore, GIoU loss (L_{GIoU}) [43] introduced the C detection box (the smallest rectangular box that contains the ground truth and predicted box [18]) and considered the loss to be when the two boxes do not intersect based on L_{IoU}. However, when two boxes contained each other, the relative position of the boxes cannot be reflected by L_{GIoU}. As shown in Figure 6, to address the problem, DIoU loss (L_{DIoU}) [44] was proposed to consider the Euclidean distance between the prediction box and ground truth box. It can be defined as

$$L_{\mathrm{DIoU}} = 1 - L_{\mathrm{IoU}} + \frac{d^2}{C^2} \tag{6}$$

$$L_{\mathrm{IoU}} = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|} \tag{7}$$

where b and b^{gt} represent the predicted box and ground truth box, c is the diagonal length of the C detection box, and d is the distance between the center points of the predicted box and ground truth box.

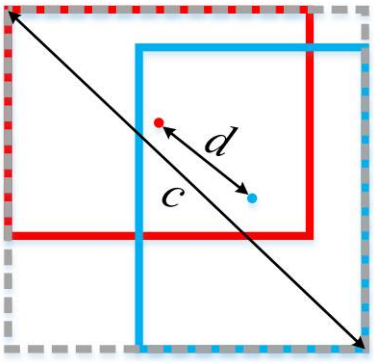

Figure 6. Details of DIoU loss. The red box, blue box, and gray box are the predicted box, ground truth box, and C detection box, respectively.

In our work, we chose L_{DIoU} to replace L_{IoU} to obtain a more robust detection model. Thus, our loss function can be defined as

$$\mathrm{Loss} = \lambda L_{\mathrm{DIoU}} + L_{\mathrm{Conf}} + L_{\mathrm{Cls}} \tag{8}$$

where λ is regression weight ($\lambda = 5$).

2.4. Network Training

The experiment environment was Pytorch 1.2.0, GeForce GTX 1080 (with 8 G memory) and CUDA10.0. All experiments were conducted using the same parameter settings. To accelerate network convergence, in the first 50 epochs, the batch size and learning rate were set to 16 and 0.001, respectively. After 50 epochs, we set the batch size and learning rate set to 8 and 0.0001 to obtain a robust detection model [45]. Our datasets were randomly divided into a training set (80%), testing set (10%), and validation set (10%). All comparative experiments were performed with the same hyper-parameters as our method.

2.5. Evaluation Indexes

In order to evaluate the performance of the models, the average precision (*AP*), recall, precision, and *PR* curve were used as model evaluation indicators, which can be defined as follows:

$$Precision = TP/(TP + FP) \times 100\% \tag{9}$$

$$Recall = TP/(TP + FN) \times 100\% \tag{10}$$

$$AP = \int_0^1 PRdr \tag{11}$$

where *TP* (True Positive) and *TN* (True Negative) are the number of actual positive and negative samples predicted as positive and negative samples, respectively. *FP* (False Positive) and *FN* (False Negative) are the number of negative samples predicted as positive and positive samples predicted as negative, respectively.

3. Results

In this section, to verify the effectiveness of our method, our method is compared with two classes of state-of-the-art models including object detection algorithms and their lightweight models. The relevant details are introduced in Sections 3.1 and 3.2. Additionally, we demonstrate the accuracy of our algorithm in real-world localization in Section 3.3.

3.1. Comparison of Different Object Detection Algorithms

To evaluate the performance of the proposed method, we compare our method with six state-of-the-art object detection networks including SSD [21], Faster R-CNN [46], CenterNet [47], Efficientdet [48], YOLOv3 [19], and YOLOv4 [49]. PR curves are applied to evaluate the performance of different models; the higher the curve, the better the model performance.

Consequently, as shown in Figure 7, our model outperforms other state-of-the-art object detection networks and has the best performance. Specifically, as shown in Table 1, our method reaches the highest AP (AP = 95.56%) and has the smallest model size (4.47 MB). It is worth noting that model size is critical for the deployment of embedded devices, and AP value is a comprehensive evaluation indicator which can reflect the comprehensive levels of precision and recall. The AP value of our method was 14.62%, 9.98%, 4.35%, 9.94%, 2.80%, and 1.37% higher than those of SSD, Faster R-CNN, CenterNet, Efficientdet, YOLOv3, and YOLOv4, respectively. Thus, in terms of accuracy and model size, our method has great advantages compared to other state-of-the-art object detection networks.

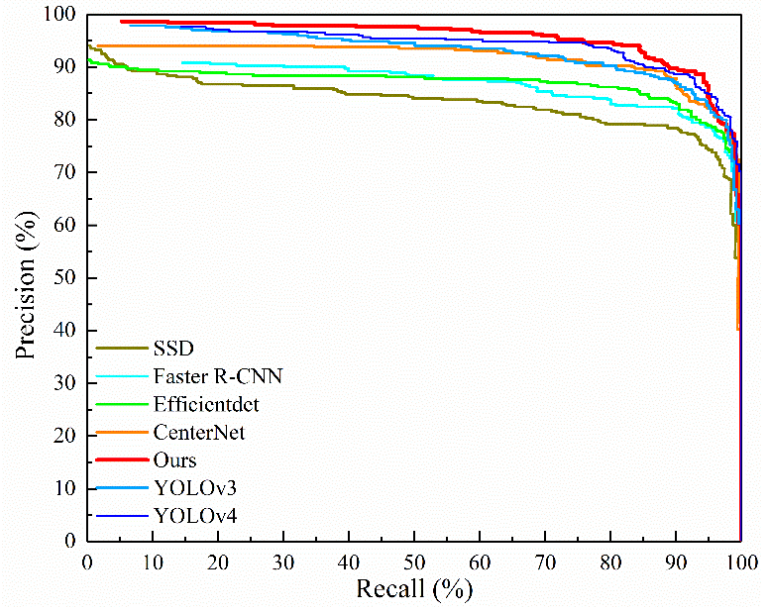

Figure 7. PR curves for the seven detection models.

To further qualitatively evaluate the detection results of our model, the detection results of different methods under two complex scenes including occluded and backlit are shown in Figures 8 and 9, respectively. It can be seen that some areas without jujubes (yellow box) were wrongly predicted as jujubes by Faster R-CNN. The main reason for these errors is that the region-proposed network of Faster R-CNN recommended too many prediction boxes in which duplicate target boxes were not suppressed by non-maximum suppression (NMS). In addition, it is clearly seen that the one-stage models including SSD, YOLOv3, YOLOv4, and YOLOv5-S all lost real objects (blue box) which contained occluded jujubes. The reason for this phenomenon is that these occluded jujubes had more complex features than non-occluded jujubes due to their lower feature scale and contrast. By contrast, our method and Efficientdet had satisfactory recognition results on the occluded scenes. However, as shown in Figure 9, Efficientdet had a false alarm under the backlit scene (yellow box) and our method still detected the winter jujubes correctly. Additionally, Faster R-CNN, Efficientdet, SSD, and YOLOv5-S also had false alarms on the backlit scene. It can be concluded that our method has strong generalization ability and good universality under backlit and occluded scenes.

Table 1. Detection results with different object detection networks.

Methods	Backbone	Precision (%)	Recall (%)	AP (%)	Model Size (MB)	Detection Time (s)
SSD	VGG16	94.83	74.32	83.37	99.76	0.041
Faster R-CNN	VGG16	66.46	86.76	86.89	522.91	0.054
CenterNet	ResNet50	97.93	76.76	91.58	124.61	0.031
Efficientdet	EfficientNet-D0	93.93	79.46	86.92	14.90	0.044
YOLOv3	DarkNet53	92.40	85.41	92.94	236.32	0.047
YOLOv4	CSPDarkNet53	91.36	88.65	94.27	245.53	0.060
Ours	CSPDarkNet53	93.08	87.83	95.56	4.47	0.022

Figure 8. Recognition results with different models for the occluded scene. (**a**) Faster R-CNN, (**b**) Efficientdet, (**c**) SSD, (**d**) YOLOv3, (**e**) YOLOv4, (**f**) YOLOv5-S, (**g**) Ours, (**h**) Ground Truth.

Figure 9. Recognition results with different models on a backlit scene. (**a**) Faster R-CNN, (**b**) Efficientdet, (**c**) SSD, (**d**) YOLOv3, (**e**) YOLOv4, (**f**) YOLOv5-S, (**g**) Ours, (**h**) Ground Truth.

Furthermore, as shown in Table 2, quantitative evaluation results using our proposed model were counted for the different scenes in our test set. It can be found that our method achieved the highest AP value under front light conditions compared to the backlit and occluded scenes. However, there were no significant differences of AP in terms of the different scenes, which proved that the model has strong adaptability to different scenes.

Table 2. The performance of our method under different scenes.

Scene	Precision (%)	Recall (%)	AP (%)
Front light	93.64	88.21	96.02
Backlight	93.01	87.92	95.63
Occluded scene	92.42	87.02	95.09

3.2. Comparison of Different Lightweight Models

In this section, to validate the performance of the proposed model, different lightweight models including YOLOv4-Tiny, YOLOv4-MobileNetv3, YOLOv5-S, YOLOX-S, and YOLOX-Tiny are compared with our method on the winter jujubes dataset. As shown in Figure 10, YOLOX-S, YOLOX-Tiny, and our method had the best, second-best, and third-best performances on our dataset. However, as shown in Table 3, the model sizes of YOLOX-S (34.21 MB) and YOLOX-Tiny (19.29 MB) are nearly 8 and 4 times that of our method (4.47 MB), respectively. Therefore, our method is more advantageous compared to YOLOX-S and YOLOX-Tiny. In addition, it can be also seen that the PR curves of our method are higher than those of YOLOX-Nano, which demonstrates the effectiveness of our modifications. Although the model size of our method is 0.99 MB larger than that of Nano, our method has higher detection accuracy than Nano, as shown in Table 4. As shown in Figure 11, it can be clearly seen that the lightweight version of YOLOX outperforms YOLOv4 and YOLOv5-S on model size and accuracy. Considering deployment of embedded models and detection accuracy, our method has better universality and robustness compared with other state-of-the-art object detectors. As shown in Table 4, ablation experiments have shown that as DIoU and AFE increase, the detection accuracy gradually improves while maintaining detection efficiency and the small model size, which further demonstrated the effectiveness of our modifications. In our work, our model size is only 4.47 MB, which provides the possibility for the deployment of our embedded systems (Jetson Nano) in winter jujube harvesting. Moreover, the recognition speed is 0.022 s (Fps = 45.45) for the size of 640 × 640 pixel images, outperforming the efficiency of manual picking.

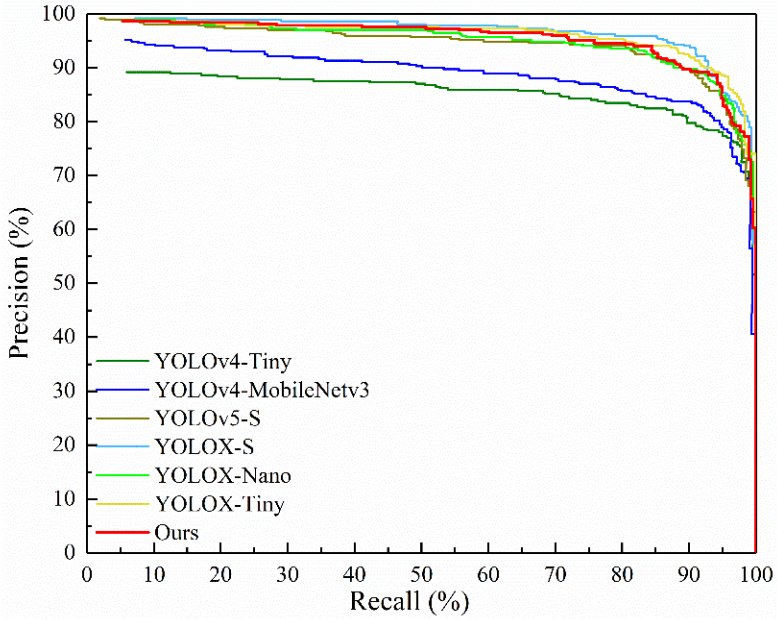

Figure 10. PR curves for the six lightweight detection models.

Table 3. Detection results with different lightweight networks.

Methods	Precision (%)	Recall (%)	AP (%)	Model Size (MB)	Detection Time (s)
YOLOv4-Tiny	93.55	78.38	85.52	23.10	0.010
YOLOv4-MobileNetv3	96.00	77.84	88.95	44.74	0.024
YOLOv5-S	92.49	86.49	94.41	27.76	0.016
YOLOX-S	91.42	92.16	96.66	34.21	0.018
YOLOX-Tiny	91.11	91.35	96.09	19.29	0.017
Ours	93.08	87.83	95.56	4.47	0.022

Table 4. The impact of each module on model performance.

YOLOX-Nano@640	IoU	DIoU	AFE	AP (%)	Model Size (MB)	Detection Time (s)
√	√			94.66	3.48	0.020
√	√		√	95.48	4.47	0.022
√		√	√	95.56	4.47	0.022

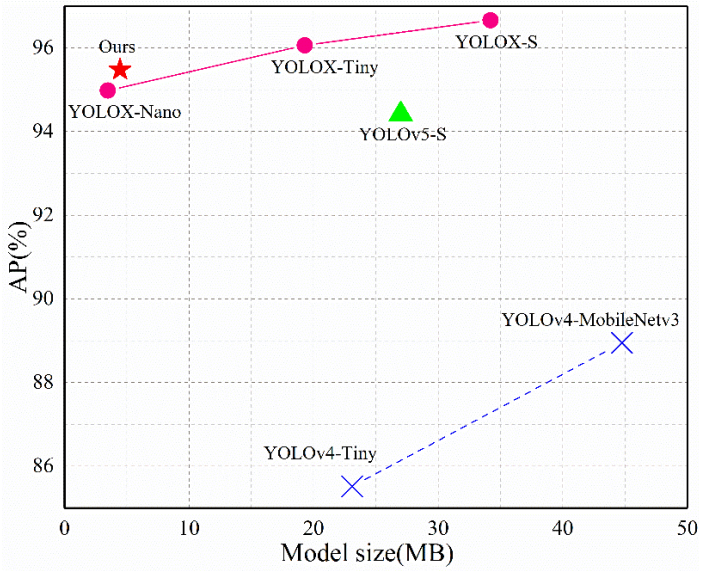

Figure 11. Size-accuracy curve for our method and other state-of-the-art object detectors.

3.3. Positioning Error Evaluation

To verify the 3D positioning accuracy, our method was used to obtain the X, Y, Z coordinates of winter jujubes. As shown in Figure 3, 13 winter jujubes were randomly arranged at the grid position of the calibration cardboard to reflect the relative positions of the jujubes (the grid size was 30 × 30 mm). We randomly selected 12 winter jujubes and divided them into six groups to calculate the relative positioning error. The predicted jujube coordinates (X_{P_i}, Y_{P_i}, Z_{P_i}) using our method are shown in Table 5. Since it is difficult to obtain real coordinates of the jujubes relative to the optical center of the camera, we proposed a positioning error evaluation method in which two winter jujubes at a fixed distance in the calibration cardboard were used to evaluate the localization error of the algorithm. We tried to control the RGB-D camera to be parallel to the calibration board as much as possible in Figure 3, so we think that the real Z-direction distance of the 13 points in the Z-direction is 0 mm (Z_2 = 0 mm). The corresponding real distance in the X, Y direction (X_2, Y_2) of the two targets can be obtained through the calibration board. The average positioning errors (APE) can be computed as

$$X_1 = X_{P_i} - X_{P_{i-1}} \qquad i = 2, 4, 6, 8, 10, 12 \tag{12}$$

$$X_2 = 30 \times N \tag{13}$$

$$\Delta X = |X_1 - X_2| \tag{14}$$

$$X_{APE} = \frac{\sum \Delta X}{6} \tag{15}$$

where N is the number of cells in the X direction of two targets. ΔX is the deviation of X direction. X_{APE} is the average positioning errors of X-direction. Y_{APE} and Z_{APE} are calculated the same as X_{APE}. As shown in Table 5, it is found that the X_{APE}, Y_{APE}, Z_{APE} are 5.8 mm, 5.4 mm, and 3.8 mm, respectively, which meet the accuracy requirements for locating winter jujubes.

Table 5. Error statistics between predicted and true positions.

Samples	X_{P_i}	Y_{P_i}	Z_{P_i}	X_1	X_2	ΔX	Y_1	Y_2	ΔY	Z_1	Z_2	ΔZ
P_1	−453.2	−275.3	1076.0	337.9	330.0	7.9	23.1	30.0	6.9	3.0	0.0	3.0
P_2	−115.3	−252.2	1073.0									
P_3	244.3	−254.1	1081.0	56.9	60.0	3.1	151.7	150.0	1.7	1.0	0.0	1.0
P_4	187.4	−102.4	1080.0									
P_5	−230.4	−155.1	1068.0	172.5	180.0	7.5	127.7	120.0	7.7	7.0	0.0	7.0
P_6	−402.9	−27.4	1061.0									
P_7	−375.3	182.9	1075.0	189.3	180.0	9.3	116.6	120.0	3.4	0.0	0.0	0.0
P_8	−186.0	299.5	1075.0									
P_9	−189.3	86.8	1080.0	173.7	180.0	6.3	94.6	90.0	4.6	8.0	0.0	8.0
P_{10}	−15.6	−7.8	1072.0									
P_{11}	169.0	224.3	1076.0	119.2	120.0	0.8	81.9	90.0	8.1	4.0	0.0	4.0
P_{12}	288.2	306.2	1082.0									
Average positioning errors						5.8			5.4			3.8

4. Discussion

4.1. Effect of the Illumination on the Winter Jujube Detection

During the operation of the jujube harvesting robot, it will face different light intensities in different operation periods. Some image enhancement methods, such as brightness transformation, color transformation, and contrast transformation [50,51], can be used to simulate different complex orchard environments. To explore the effect of different light intensities on our proposed model, in our work, the clip function in NumPy was used to adjust the brightness of the image. The images after brightness transformation is shown in Figure 12, the specific transformation process is as follows

$$I\_out = \delta I\_in + beta \tag{16}$$

where I_out and I_in are images after adjusting the brightness and raw image, respectively. δ is the brightness factor; when $\delta < 1$, the image will be darkened, otherwise, the image will be brightened. *beta* is 10.

Figure 12. Test images under different illuminations.

As shown in Figure 13, it can be found that the increase or decrease of light intensity affects the accuracy evaluation index (AP and Precision) of the model to a certain extent,

but it does not fluctuate greatly, reflecting the strong generalization ability of the model under different illuminations. The fact also should be accepted that different illuminations have a greater impact on recall due to having few samples under low-light and strong-light in our small sample dataset.

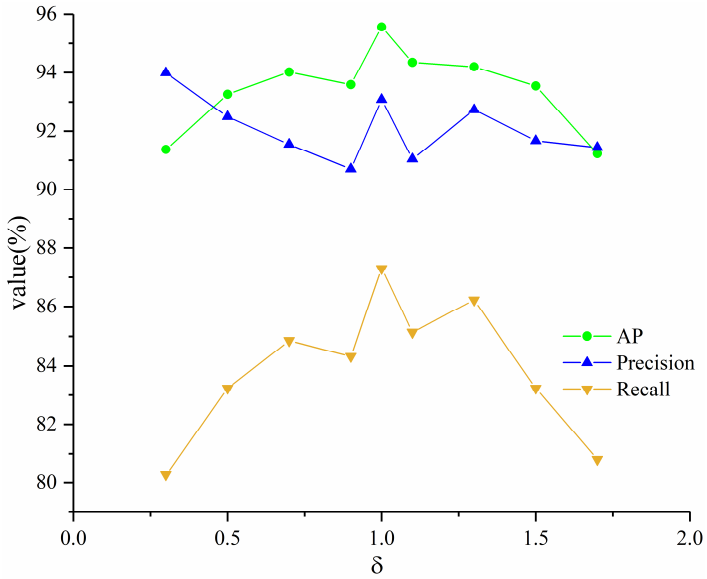

Figure 13. Performance analysis under different illuminations.

4.2. Failure Samples Analysis

Typical failure samples that cannot be detected under backlit and occlusion scenes (yellow box) are shown in Figure 14. The reason for these phenomena is that the coexistence of conditions such as occlusion of branches, the environment of backlight, and the occlusion between multiple fruits increased the difficulty of feature extraction and led to the loss of targets in object detection. Moreover, some winter jujubes that fell on the ground (white box) were also detected under a large field of view. This phenomenon largely resulted from the fact that a large field of view introduced irrelevant data, such as the fallen jujube, which had a great imbalance with the data we carried out in the target task and led to misidentification of irrelevant data.

4.3. Data Reliability Analysis

At present, small sample-based learning research is booming in deep learning fields. For instance, Hu et al. [52] achieved disease detection of tea leaf blight (TLB) images using Fast R-CNN based on a small sample dataset of 398 images, which obtained satisfactory detection results. Similarly, 546 wheat mite images were selected as a dataset to train Fast R-CNN for the recognition and counting of wheat mites, and the highest AP of 96.4% was achieved with VGG16 backbone [53]. Moreover, some image classification researches were conducted using small samples, e.g., Chen et al. [54] proposed a deep residual learning method for pest identification and classification (550 images). Zhang et al. [55] accomplished the recognition of cucumber leaf diseases with CNN and a small sample dataset (600 images).

To verify the reliability of our small sample dataset, as shown in Figure 15, existing data was augmented by lighting transformation with light factors δ = 1.5, 0.5 and mirroring operations along the X and Y axis. The corresponding expanded dataset reached 3160 images. The expanded dataset was trained using the same dataset division principles and training parameters as the unexpanded dataset. As shown in Figure 16, the detection result AP only improved 0.75% when the dataset was expanded by 4 times, which shows that the small sample data set can meet the requirements of our task.

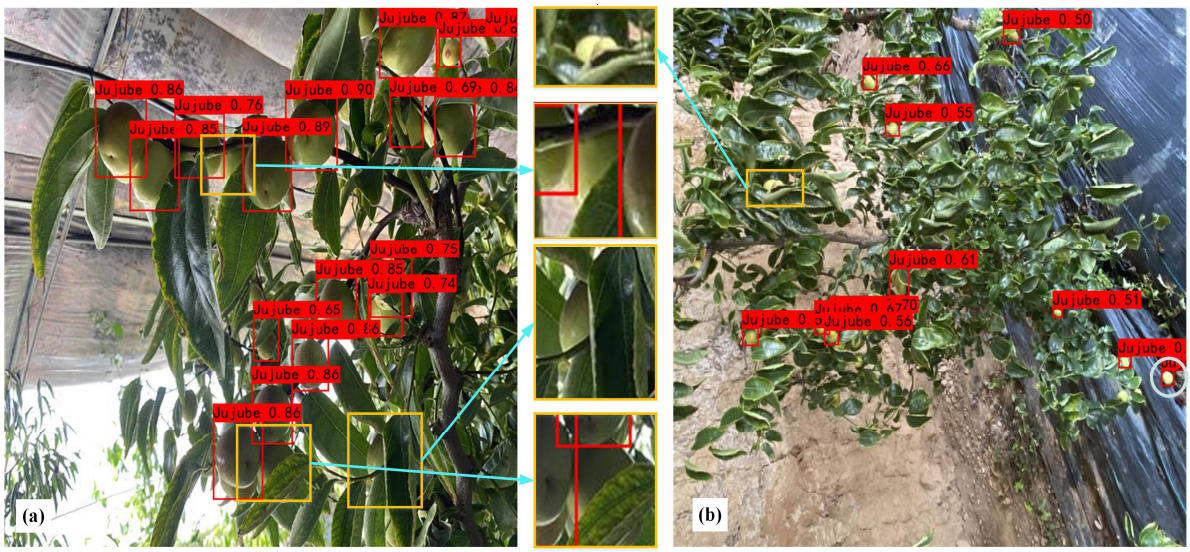

Figure 14. Typical failure samples. (**a**) Samples under backlight and occlusion scenes. (**b**) Samples under large field of view.

Figure 15. The result of data augmentation. (**a**) Raw images; (**b**) Lighting transformation with δ = 1.5; (**c**) Lighting transformation with δ = 0.5; (**d**) Mirror along the Y axis; (**e**) Mirror along the Y axis.

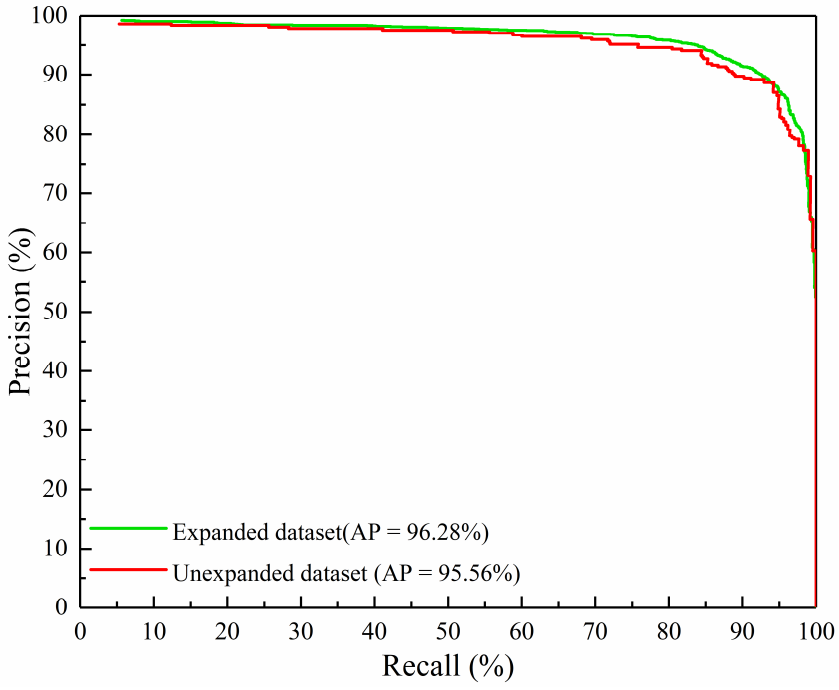

Figure 16. PR curves with the expanded and unexpanded dataset.

5. Conclusions

In this work, we proposed an object detection method based on YOLOX-Nano for detecting winter jujubes to address the problem of low recognition rate of winter jujubes in complex scenes, including backlit and occluded scenes. Firstly, an attention feature enhancement module was designed to strengthen the learning ability of features for identifying winter jujubes in complex scenes. Then, a DIoU loss was used to replace IoU loss to optimize the training process. Finally, ablation and comparative experimental results verified the effectiveness of our modifications and that our method can outperform other state-of-the-art object detectors on our datasets. In addition, combined with an RGB-D camera, our method can obtain the X, Y, Z real-time coordinates of winter jujubes, which can provide a reference for the precise harvesting of the robotic arm. In the study, a positioning error evaluation method was proposed to measure positioning error. The validation results showed that our method can meet the accuracy requirements for locating winter jujubes. Moreover, the reliability of small sample dataset, failure cases, and lighting effects were analyzed to more objectively evaluate the performance of the model.

In future work, we will carry our model to the embedded system to guide the harvest of winter jujubes. Additionally, we will also work toward the maturity identification of winter jujubes.

Author Contributions: Methodology, software, writing—original draft, visualization, Z.Z.; Writing—review & editing, supervision, funding acquisition, Y.H. (Yaohua Hu); investigation, validation, Y.Q.; formal analysis, validation, X.H.; writing—review and editing, supervision, Y.H. (Yuxiang Huang). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Talent start-up Project of Zhejiang A&F University Scientific Research Development Foundation (2021LFR066), the National Natural Science Foundation of China (C0043619, C0043628).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no conflict of interest to this research.

References

1. Ji, W.; Zhao, D.; Cheng, F.; Xu, B.; Zhang, Y.; Wang, J. Automatic recognition vision system guided for apple harvesting robot. *Comput. Electr. Eng.* **2012**, *38*, 1186–1195. [CrossRef]
2. Linker, R. Machine learning based analysis of night-time images for yield prediction in apple orchard. *Biosyst. Eng.* **2018**, *167*, 114–125. [CrossRef]
3. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J.; Li, X. A real-time detection algorithm for Kiwifruit defects based on YOLOv5. *Electronics* **2021**, *10*, 1711. [CrossRef]
4. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* **2020**, *20*, 2145. [CrossRef] [PubMed]
5. Wang, D.; Song, H.; Tie, Z.; Zhang, W.; He, D. Recognition and localization of occluded apples using K-means clustering algorithm and convex hull theory: A comparison. *Multimed. Tools Appl.* **2016**, *75*, 3177–3198. [CrossRef]
6. Tian, Y.; Duan, H.; Luo, R.; Zhang, Y.; Jia, W.; Lian, J.; Zheng, Y.; Ruan, C.; Li, C. Fast recognition and location of target fruit based on depth information. *IEEE Access* **2019**, *7*, 170553–170563. [CrossRef]
7. Fu, L.; Majeed, Y.; Zhang, X.; Karkee, M.; Zhang, Q. Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* **2020**, *197*, 245–256. [CrossRef]
8. Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; Xu, W. Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2285–2294.
9. Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 2896–2907. [CrossRef]
10. Li, Y.; Xu, W.; Chen, H.; Jiang, J.; Li, X. A novel framework based on mask R-CNN and Histogram thresholding for scalable segmentation of new and old rural buildings. *Remote Sens.* **2021**, *13*, 1070. [CrossRef]
11. Zheng, Z.; Yang, H.; Zhou, L.; Yu, B.; Zhang, Y. HLU 2-Net: A Residual U-Structure Embedded U-Net With Hybrid Loss for Tire Defect Inspection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11.
12. Zheng, Z.; Zhang, S.; Shen, J.; Shao, Y.; Zhang, Y. A two-stage CNN for automated tire defect inspection in radiographic image. *Meas. Sci. Technol.* **2021**, *32*, 115403. [CrossRef]

13. Zhao, M.; Jha, A.; Liu, Q.; Millis, B.A.; Mahadevan-Jansen, A.; Lu, L.; Landman, B.A.; Tyska, M.J.; Huo, Y. Faster mean-shift: GPU-accelerated clustering for cosine embedding-based cell segmentation and tracking. *Med. Image Anal.* **2021**, *71*, 102048. [CrossRef]
14. Zhao, M.; Liu, Q.; Jha, A.; Deng, R.; Yao, T.; Mahadevan-Jansen, A.; Tyska, M.J.; Millis, B.A.; Huo, Y. VoxelEmbed: 3D instance segmentation and tracking with voxel embedding based deep learning. In Proceedings of the International Workshop on Machine Learning in Medical Imaging, Strasbourg, France, 27 September 2021; pp. 437–446.
15. Fan, S.; Liang, X.; Huang, W.; Zhang, V.J.; Pang, Q.; He, X.; Li, L.; Zhang, C. Real-time defects detection for apple sorting using NIR cameras with pruning-based YOLOV4 network. *Comput. Electron. Agric.* **2022**, *193*, 106715. [CrossRef]
16. Zheng, Z.; Hu, Y.; Yang, H.; Qiao, Y.; He, Y.; Zhang, Y.; Huang, Y. AFFU-Net: Attention feature fusion U-Net with hybrid loss for winter jujube crack detection. *Comput. Electron. Agric.* **2022**, *198*, 107049. [CrossRef]
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
18. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
19. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
22. Zhang, J.; Karkee, M.; Zhang, Q.; Zhang, X.; Yaqoob, M.; Fu, L.; Wang, S. Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Comput. Electron. Agric.* **2020**, *173*, 105384. [CrossRef]
23. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. *Agronomy* **2022**, *12*, 319. [CrossRef]
24. Zhou, J.; Hu, W.; Zou, A.; Zhai, S.; Liu, T.; Yang, W.; Jiang, P. Lightweight Detection Algorithm of Kiwifruit Based on Improved YOLOX-S. *Agriculture* **2022**, *12*, 993. [CrossRef]
25. Lu, Z.; Zhao, M.; Luo, J.; Wang, G.; Wang, D. Design of a winter-jujube grading robot based on machine vision. *Comput. Electron. Agric.* **2021**, *186*, 106170. [CrossRef]
26. Li, S.; Zhang, S.; Xue, J.; Sun, H.; Ren, R. A Fast Neural Network Based on Attention Mechanisms for Detecting Field Flat Jujube. *Agriculture* **2022**, *12*, 717. [CrossRef]
27. Wu, L.; Ma, J.; Zhao, Y.; Liu, H. Apple detection in complex scene using the improved YOLOv4 model. *Agronomy* **2021**, *11*, 476. [CrossRef]
28. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [CrossRef]
29. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
30. Liu, R.; Tao, F.; Liu, X.; Na, J.; Leng, H.; Wu, J.; Zhou, T. RAANet: A Residual ASPP with Attention Framework for Semantic Segmentation of High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 3109. [CrossRef]
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
32. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Eca-net: Efficient channel attention for deep convolutional neural networks. *arXiv* **2020**, arXiv:1910.03151.
33. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
34. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
35. Zhang, Y.; Yu, J.; Chen, Y.; Yang, W.; Zhang, W.; He, Y. Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application. *Comput. Electron. Agric.* **2022**, *192*, 106586. [CrossRef]
36. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [CrossRef]
37. 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]
38. You, L.; Jiang, H.; Hu, J.; Chang, C.H.; Chen, L.; Cui, X.; Zhao, M. GPU-accelerated Faster Mean Shift with euclidean distance metrics. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 27 June–1 July 2022; pp. 211–216.
39. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
40. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 7354–7363.

41. De Boer, P.-T.; Kroese, D.P.; Mannor, S.; Rubinstein, R.Y. A tutorial on the cross-entropy method. *Ann. Oper. Res.* **2005**, *134*, 19–67. [CrossRef]
42. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
43. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
44. 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; pp. 12993–13000.
45. He, D.; Zou, Z.; Chen, Y.; Liu, B.; Yao, X.; Shan, S. Obstacle detection of rail transit based on deep learning. *Measurement* **2021**, *176*, 109241. [CrossRef]
46. 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*, 91–99. [CrossRef] [PubMed]
47. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6569–6578.
48. 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, 14–19 June 2020; pp. 10781–10790.
49. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
50. Zhou, J.; Zhang, D.; Zhang, W. Underwater image enhancement method via multi-feature prior fusion. *Appl. Intell.* **2022**, 1–23. [CrossRef]
51. Zhou, J.; Yang, T.; Chu, W.; Zhang, W. Underwater image restoration via backscatter pixel prior and color compensation. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104785. [CrossRef]
52. Hu, G.; Wang, H.; Zhang, Y.; Wan, M. Detection and severity analysis of tea leaf blight based on deep learning. *Comput. Electr. Eng.* **2021**, *90*, 107023. [CrossRef]
53. Chen, P.; Li, W.; Yao, S.; Ma, C.; Zhang, J.; Wang, B.; Zheng, C.; Xie, C.; Liang, D. Recognition and counting of wheat mites in wheat fields by a three-step deep learning method. *Neurocomputing* **2021**, *437*, 21–30. [CrossRef]
54. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* **2017**, *141*, 351–356. [CrossRef]
55. Zhang, J.; Rao, Y.; Man, C.; Jiang, Z.; Li, S. Identification of cucumber leaf diseases using deep learning and small sample size for agricultural Internet of Things. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 15501477211007407. [CrossRef]