symmetry

MDPI

Article

Pear Object Detection in Complex Orchard Environment Based on Improved YOLO11

Mingming Zhang, Shutong Ye, Shengyu Zhao, Wei Wang and Chao Xie *

College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing 210037, China
* Correspondence: chaoxie@njfu.edu.cn

Abstract: To address the issues of low detection accuracy and poor adaptability in complex orchard environments (such as varying lighting conditions, branch and leaf occlusion, fruit overlap, and small targets), this paper proposes an improved pear detection model based on YOLO11, called YOLO11-Pear. First, to improve the model's capability in detecting occluded pears, the C2PSS module is introduced to replace the original C2PSA module. Second, a small target detection layer is added to improve the model's ability to detect small pears. Finally, the upsampling process is replaced with DySample, which not only maintains a high efficiency but also improves the processing speed and expands the model's application range. To validate the effectiveness of the model, a dataset of images of Qiu Yue pears and Cui Guan pears was constructed. The experimental results showed that the improved YOLO11-Pear model achieved precision, recall, mAP50, and mAP50–95 values of 96.3%, 84.2%, 92.1%, and 80.2%, respectively, outperforming YOLO11n by 3.6%, 1%, 2.1%, and 3.2%. With only a 2.4% increase in the number of parameters compared to the original model, YOLO11-Pear enables fast and accurate pear detection in complex orchard environments.

Keywords: YOLO11; complex orchard environment; deep learning; fruit detection; machine vision

check for updates

Academic Editors: Sergei Odintsov, José Luis Vázquez-Noguera, Horacio Legal-Ayala and Julio César Mello-Román

Received: 10 January 2025
Revised: 24 January 2025
Accepted: 6 February 2025
Published: 8 February 2025

Citation: Zhang, M.; Ye, S.; Zhao, S.; Wang, W.; Xie, C. Pear Object Detection in Complex Orchard Environment Based on Improved YOLO11. *Symmetry* **2025**, *17*, 255. https://doi.org/10.3390/sym17020255

Copyright: © 2025 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

Pears are rich in nutrients and have a delicious taste. As an essential part of the fruit industry, China's pear cultivation has the largest planting area and total production in the world [1,2]. Over the past five years, China's pear harvest area and output have accounted for approximately 70% of the global total. However, the majority of pear harvesting is still carried out manually. With the continuous expansion of pear orchards, the drawbacks of manual harvesting have become increasingly evident: labor shortages, rising harvesting costs, and low efficiency are driving the industry towards automation, industrialization, and intelligent solutions [3]. As a result, the automation for pear picking has become an inevitable trend, and precise detection of pears during the harvesting process has become a critical task [4].

Fruit detection technology is a key technology in the intelligence industry. Early fruit detection methods were primarily based on machine learning, where image preprocessing and manual extraction of features such as shape, texture, and color were employed. Machine learning algorithms were then used to train and test models for fruit detection. These approaches required manual extraction of texture and color features from fruit images, which were influenced by subjectivity and environmental complexities, limiting the applicability of traditional fruit detection methods [5]. At the same time, the emergence and

rapid development of deep learning has helped address these issues. With continuous advancements in computer hardware and algorithm optimization, deep-learning-based fruit detection has become a hot topic in the field of intelligent recognition. In the fruit detection task, a deep learning model can directly process the original image data and complete the process from the image input to the target detection result output end-to-end, avoiding the error accumulation and efficiency problems caused by the multi-stage processing of machine learning. The performance of deep learning models on large-scale datasets will continue to improve with increases in the amount of data. They have good scalability and can meet the needs of different scenarios and tasks.

A comprehensive analysis revealed that current deep-learning-based fruit detection technologies have achieved certain successes in detection speed and accuracy. However, most experimental studies have not taken into account the impact of complex orchard environments, such as varying lighting, occlusion, overlap, and small object conditions, on a model's detection accuracy. Moreover, with regard to pear detection, most studies only focused on a single pear variety, lacking the ability to differentiate between different pear species [6]. To address these issues, this paper uses the Python language and proposes the YOLO11-Pear model for pear detection in complex orchard environments, based on the YOLO11 framework. The main contributions of this paper are as follows:

- A dataset comprising two pear varieties, Qiu Yue pears and Cui Guan pears, was constructed to enhance the sample diversity;
- C2PSS Module: The attention mechanism in the C2PSA module is replaced with the SEAM attention mechanism, giving the model improved pear fruit detection performance in the presence of leaf occlusions and fruit overlaps by learning the relationship between occluded and non-occluded objects;
- Small Object Detection Layer: To preserve more detailed features of pears, an additional feature map with a size of 160 $\times$ 160 is introduced for detecting small objects.
- DySample: In response to the dense distribution of pears, a dynamic upsampling module is introduced, enhancing the model's ability to detect densely clustered pear targets, while reducing the computational complexity.

2. Related Works

Early methods for fruit detection were mainly based on machine learning. For example, Kezhu Tan et al. [7] compared the KNN classifier with a newly proposed TMWE classifier. First, they constructed a fruit training set using patches, calculated the HOG feature vector, and then applied an SVM classifier for fast fruit detection and classification. The proposed TMWE classifier was able to quickly and accurately identify blueberries at different stages of ripeness, with a low computational cost. Jun Lu et al. [8] proposed an innovative method that uses only texture and intensity distribution to detect immature citrus fruits in images. This method employed an RUSBoost classifier with LBP texture features and circular Hough transform fitting for fruit detection, achieving an accuracy rate of 82.3%. Ferhat Kurtulmus et al. [9] developed a machine vision algorithm that utilized color threshold segmentation, an innovative "eigenface" method, and circular Gabor texture analysis to detect and count immature green citrus fruits. This method successfully detected 75.3% of green citrus fruits.

In recent years, deep learning has achieved remarkable performance in fruit detection tasks. For instance, Wenli Zhang et al. [10] proposed an improved model, OrangeYolo, based on the YOLOv3 algorithm. This method employs a receptive field matching strategy and a dual-attention multi-scale fusion mechanism to enhance fruit detection performance. The mAP50 of OrangeYolo was 95.7%, a 5.2% improvement over the original algorithm, allowing for the fast and accurate detection of green citrus targets of varying sizes in images.

To address challenges such as diverse fruit colors, overlapping fruits, and occlusions by branches and leaves in natural environments, Aobin Zhu et al. [11] proposed an improved YOLOv5s-CEDB model. This algorithm uses the lightweight GhostConv network, introduces a coordinate attention mechanism (CoordAtt) to enhance feature extraction, and employs a light-BiFPN structure to detect small and densely packed Camellia oleifera fruits. Additionally, they created their own dataset and conducted evaluation experiments. YOLOv5s-CEDB achieved an mAP50 of 91.4% and an F1 score of 89.6%, improving by 2.6% and 3.6%, respectively, compared to the original algorithm. This model performed better in detecting Camellia oleifera fruits under varying color, fruit overlap, and lighting conditions. Rongli Gai et al. [12] addressed the issue of inaccurate cherry maturity detection by proposing an improved YOLOv4 model. They first used DenseNet to reduce the model parameters and accelerate the inference speed, then replaced the original square bounding box with a circular bounding box, which is more suitable for detecting cherry fruits. The improved model achieved an F1 score of 94.7%, with mAP and F1 scores increasing by 15% and 1.2%, respectively, compared to the original YOLOv4, making it more suitable for detecting mature cherry fruits. Han Sun et al. [13] proposed an improved lightweight model, YOLO-P, for fast and accurate pear detection. This model uses inverted residual modules to efficiently extract features from distant pears, incorporates the CBAM attention mechanism for finer feature extraction, and replaces the activation function with the Hard-Swish activation function. The experimental results showed that YOLO-P reduced the size of the model by 39.4%, while improving the AP by 1.8% over YOLOv5, reaching 97.6%. This model enables fast and accurate pear detection in challenging natural environments.

3. Materials and Methods

3.1. Dataset Construction

3.1.1. Dataset Collection

To more effectively assess the performance of the proposed YOLO11-Pear model, we first established a pear fruit image dataset captured in a complex orchard environment. The dataset was collected at the pear cultivation base of the Jiangsu Academy of Agricultural Sciences in China, with pears as the target objects, divided into two varieties: Qiu Yue pears and Cui Guan pears. In July 2023, images were captured using a Nikon Z30 camera and an Apple camera, resulting in a total of 503 images—305 of Qiu Yue pear and 198 of Cui Guan pear. All images were stored in .JPG format. To simulate the most realistic complex orchard scenarios, the data collection process incorporated varying times of day, lighting conditions, and shooting angles, capturing images of pears with leaf occlusion, fruit overlap, and varying sizes. This approach enhanced the diversity of the dataset. Figure 1 shows some of the images of Qiu Yue pear and Cui Guan pear captured in the complex orchard environment.

3.1.2. Dataset Augmentation

The images of Qiu Yue pear and Cui Guan pear were annotated using Labelimg, with Qiu Yue pear labeled as “yellow pear” and Cui Guan pear labeled as “green pear”. To further enhance the model’s generalization ability and improve the robustness of the network, data augmentation was performed before training, to expand the dataset. Inspired by the concept of symmetry [14], geometric transformations were first applied to the images, including horizontal flip, vertical flip, and center cropping. Subsequently, random color feature transformations were applied to the processed images, including changes in brightness, contrast, saturation, and Gaussian noise [15]. These operations effectively simulated the lighting variations and growth posture of pears in a complex orchard environment. Through image augmentation, the dataset was expanded to 2012 images. Figure 2

shows the pear fruit images processed using the data augmentation techniques employed in this study.



Figure 1. Some images of Qiu Yue pear and Cui Guan pear captured in a complex orchard environment. Yellow pears are Qiu Yue pears and green pears are Cui Guan pears.

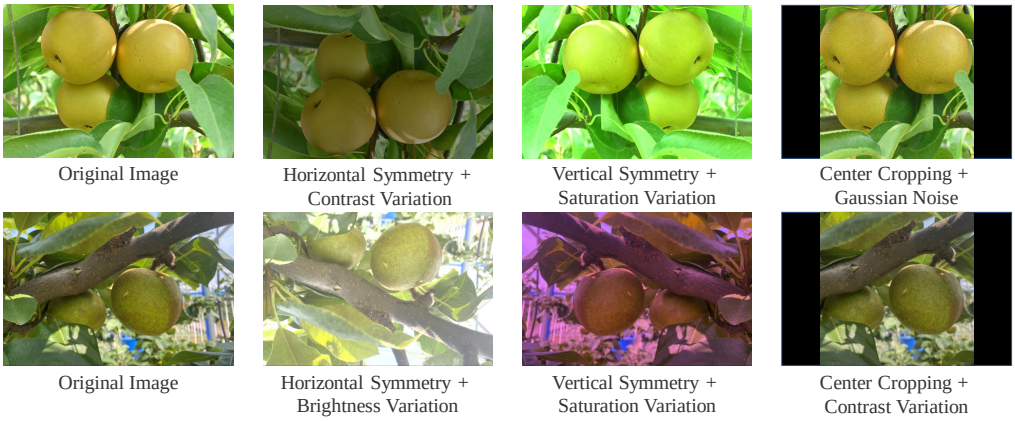


Figure 2. Dataset augmentation of pears.

3.2. YOLO11-Pear

After careful consideration from multiple perspectives, the YOLO11 model was selected as the base model for this study. The YOLO series has long been known for its excellent detection speed and end-to-end training process. As the latest model in the YOLO family, YOLO11 inherited these advantages, while further improving the detection accuracy and robustness through optimizations of the network architecture and training algorithms. Especially in complex background environments, YOLO11 effectively handles both object localization and classification tasks. For pear fruit detection in a complex orchard environment, YOLO11 has demonstrated significant adaptability, successfully identifying pears under varying lighting conditions and showing strong resistance to interference. Moreover, compared to other detection models, YOLO11 also has a computational efficiency advantage, allowing it to be deployed efficiently on resource-constrained devices, thus meeting real-time detection requirements. Orchard detection tasks typically require the model to

complete object recognition within a short time, and the efficiency of YOLO11 makes it particularly well-suited for real-time scenarios.

However, YOLO11 still has some limitations. For instance, pear fruits are often densely distributed on trees and prone to occlusion. In such cases, YOLO11 may experience detection failures or misidentifications, particularly when objects are very close or completely overlapping. Additionally, the performance of YOLO11 may not be as effective when detecting small objects as it is for larger ones. Due to the varying sizes of pears in the image—especially when partially obscured by leaves or other objects—YOLO11 may struggle to detect small-sized targets effectively, leading to missed detections or false positives. Based on these problems, the improved model, YOLO11-pear, proposed in this paper has been basically solved. The improved model architecture is shown in Figure 3.

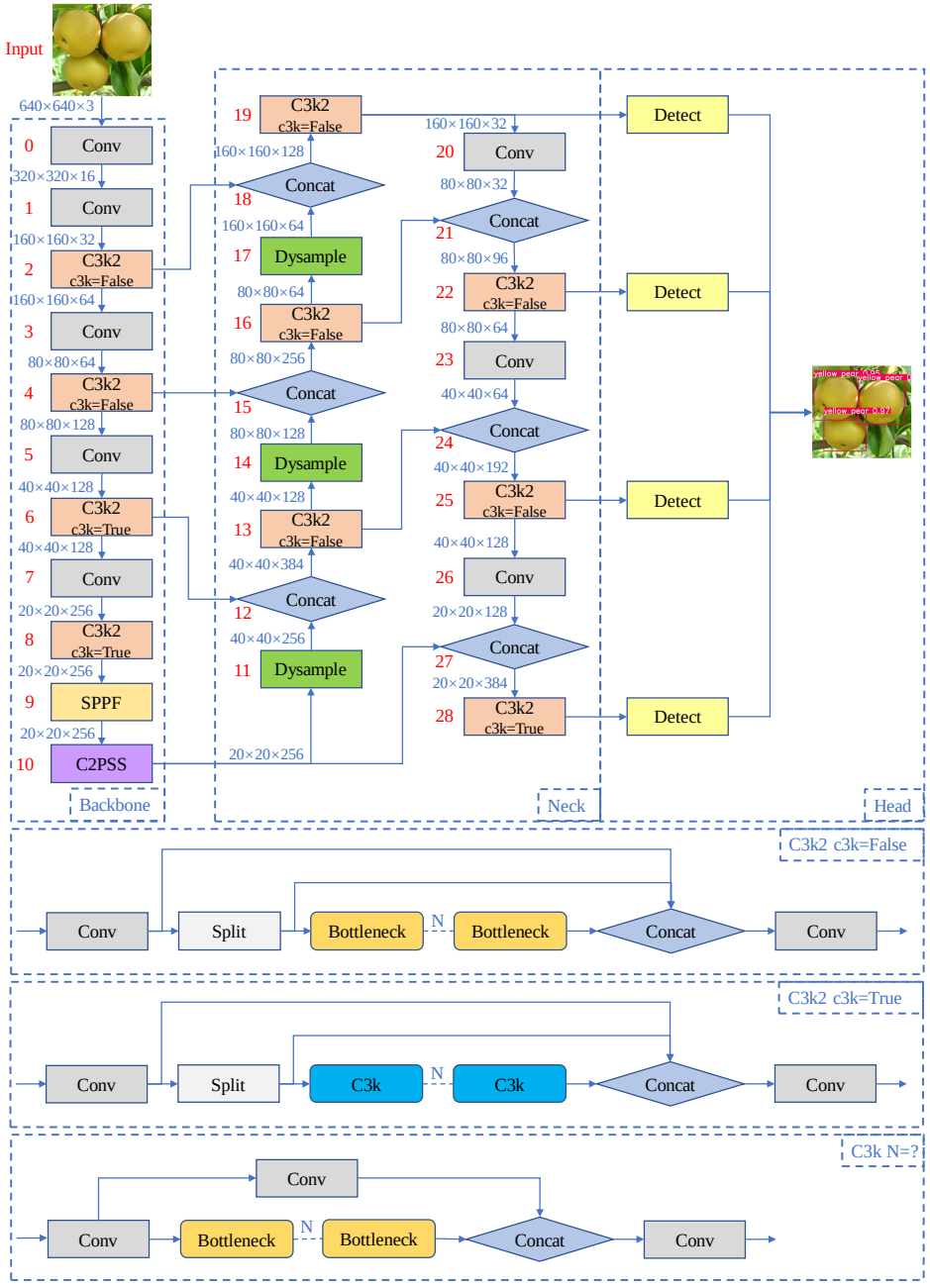


Figure 3. Model architecture of YOLO11-Pear. In the Backbone, replace C2PSA module with C2PSS module; In the neck, replace the original upsampler with DySample; In the head, add a small detection layer.The red number represents the number of model layers.

YOLO11-pear is divided into three parts: the backbone, neck, and head. YOLO11-pear continues to use the C3k2 module used by YOLO11 in the backbone and neck layers. This module is optimized based on the C3 module. By specifying the parameter N as 2, the two C3k modules are connected in series, which further improves the efficiency and ability of feature extraction. This design enables the C3k2 module to maintain the stability and improve the running speed of the model. The SPPF module is also used in the backbone, which uses global pooling to obtain more feature information. At the same time, the C2PSS module is proposed to replace the C2PSA module in the original model, which is a feature extraction and processing module with an attention mechanism that can effectively solve the problems of branch and leaf occlusion and fruit overlap in pear fruit detection. The upsampler is replaced with DySample in the neck layer to improve the ability of the model to detect dense pear targets. A 160 × 160 feature map is added to the head layer for small-target pear detection to improve the accuracy of the model when detecting different sizes of pear.

3.2.1. C2PSS

In orchard or agricultural detection, occlusion becomes a significant factor affecting model accuracy [16,17]. When pear fruits are partially obscured by leaves, branches, or other pears, YOLO11 may fail to accurately identify the position or type of the pear, leading to errors in detection. This paper proposes an improved C2PSA module, named C2PSS, which replaces the attention mechanism in the PSABlock module with a spatially enhanced attention module (SEAM) [18]. The experimental results demonstrated that the improved YOLO11 model effectively handled pear and branch occlusions in complex environments. The network architecture of the C2PSS module is shown in Figure 4. The network architecture of the SEAM module is shown in Figure 5.

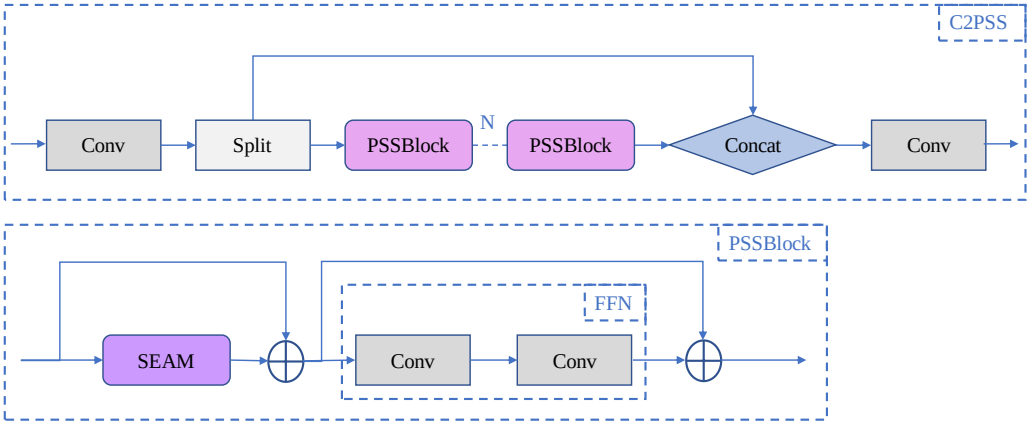


Figure 4. Network architecture of the C2PSS module.

The C2PSA module is an important addition to the backbone of the YOLO11 model, incorporating an attention mechanism to significantly enhance the model's feature extraction capabilities and accuracy. This module introduces a pyramid slice attention (PSA) mechanism, which improves the model's ability to extract and process features. Built upon the squeeze-and-excitation (SE) attention mechanism, the C2PSA module refines the feature processing and representation, thereby improving performance in object detection. However, in specific applications such as pear fruit detection, the model struggles with detecting pears and branches in occlusion scenarios. To address this, the attention mechanism is replaced with a spatially enhanced attention module (SEAM). The SEAM was first implemented as a DcovN module, which combines depthwise separable convolutions and residual connections to reduce the number of parameters, while learning

channel importance. To compensate for the loss of inter-channel relationships, the DcovN module uses 1 × 1 convolutions to fuse the outputs from different depthwise convolutions. Additionally, a two-layer fully connected network is employed to strengthen the relationships between channels, thus mitigating the information loss caused by occlusions.

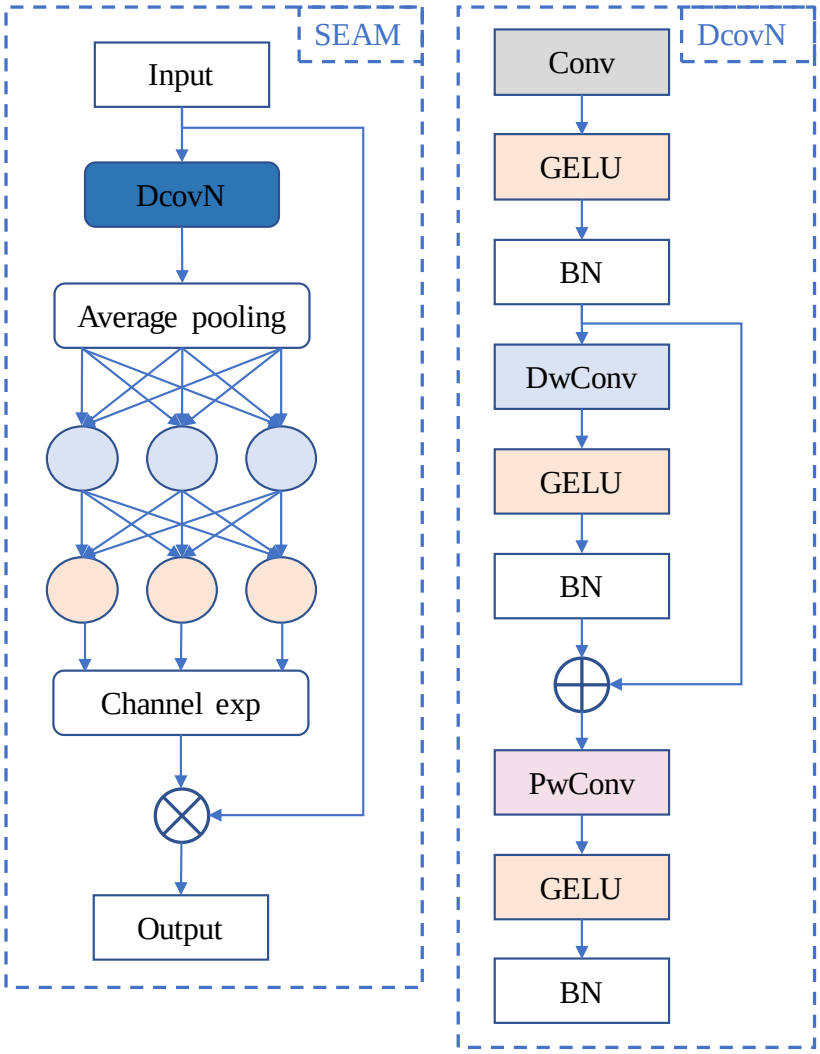


Figure 5. Network architecture of the SEAM module.

3.2.2. Small Object Detection Layer

In the YOLO11 model, the neck layer performs feature fusion after two upsampling operations, combining with the C3k2 module, and outputs feature maps with sizes of 80 × 80, 40 × 40, and 20 × 20. The deep network structure of YOLO11 is a double-edged sword. As the network depth increases, the model generally gains a stronger feature representation ability and improved performance. However, there are potential drawbacks: deeper networks reduce the spatial resolution of feature maps, causing each pixel to represent a larger region of the original image [19]. This change leads to a decrease in performance when detecting small objects. Due to the varying sizes of pear fruits in images, with the typical near-large and far-small pattern, the YOLO11 model is prone to missing small targets in pear fruit detection [20,21]. To address this issue, the feature map sizes in the neck layer were expanded to four scales: 160 × 160, 80 × 80, 40 × 40, and 20 × 20. The 160 × 160 feature map preserves more detailed information, which is beneficial for detecting small pear fruits [22]. As shown in Figure 3, the small object

detection layer fuses features through three upsampling operations with the first C3k2 module in the backbone, and outputs a feature map of size 160 × 160 at layer 19.

3.2.3. DySample

Small pear fruits typically have fewer features in images, and traditional upsampling methods may lead to a loss of edge details for small objects. Additionally, due to the dense growth pattern of pears, large areas of fruit may be occluded. To address these challenges, this paper introduces the DySample sampler, based on point sampling, with the aim of enhancing the model's ability to detect small and densely packed pears [23].

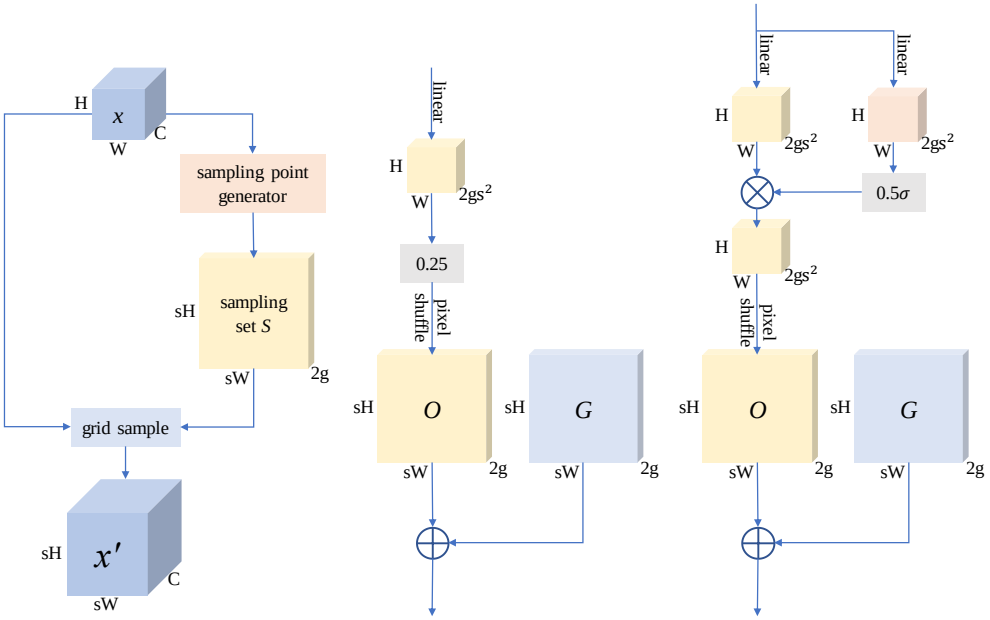
DySample avoids the use of dynamic convolution, significantly reducing the computational load of the model. Compared to traditional kernel-based upsampling methods, it employs a simpler and more efficient point sampling approach, which reduces the number of model parameters. The network structure of DySample is shown in Figure 6. Figure 6a illustrates sampling based on dynamic upsampling. The input feature map x is passed through a sampling point generator to produce a sampling set S, which is then resampled using the grid sample function to obtain an upsampled feature map x'. Figure 6b depicts the sampling point generator in DySample, which includes two methods for generating sampling points: the static range factor and the dynamic range factor. In the static method, the sampling set S is composed of the original sampling grid G and the generated offset O. The offset O is generated using a combination of a linear layer and pixel shuffle, with its range determined by both static and dynamic factors. In the dynamic method, besides using the linear layer and pixel shuffle, a dynamic range factor is introduced. This factor is first generated using a Sigmoid function, which then adjusts the offset O. The formulas used in these methods are as follows:

$$x' = grid_sample(x, S) \tag{1}$$

$$O_1 = 0.25[linear(x)] \tag{2}$$

$$O_2 = linear1(x) * 0.5\sigma[linear2(x)] \tag{3}$$

$$S = G + O \tag{4}$$



(a) Sampling based on dynamic upsampling (b) Sampling point generator in Dysample

Figure 6. Network architecture of DySample.

DySample enhances the sample diversity by generating neighboring samples, which helps address the data imbalance problem and reduces false positives and false negatives [24]. In pear fruit occlusion detection, DySample effectively increases the resolution of feature maps, making small objects clearer, even in occlusion scenarios. This improves the detection accuracy and reduces both false detections and missed detections. Additionally, a point-based sampling approach and the perspective of learning-based sampling for upsampling avoid dynamic convolution operations and extra sub-networks, thereby improving the model efficiency and performance, with minimal computational cost. This makes the module particularly suitable for detecting densely occluded pear fruits in environments with limited resources and computational power.

3.3. Evaluation Indicators

When evaluating the performance of the YOLO model, commonly used metrics include precision (P), recall (R), mAP50, and mAP50–95 [25]. In addition, this paper also used parameters to assess the model's complexity. These metrics provided a comprehensive evaluation of the model's pear fruit detection performance in complex orchard environments [26].

Precision (P) measures the proportion of correctly predicted pear fruit samples among all the samples predicted as pear fruits by the model. A higher precision indicates fewer false positive detections of pear fruits. The calculation formula is

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Here, true positive (TP) refers to the instances where the model correctly detects pear fruits as pear fruits, while false positive (FP) refers to the instances where the model incorrectly detects non-pear fruits as pear fruits.

Recall (R) measures the proportion of correctly detected pear fruit samples out of the actual pear fruit samples. A higher recall indicates fewer pear fruits are missed by the model. The calculation formula is

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Here, false negative (FN) refers to the instances where the model incorrectly detects pear fruits as non-pear fruits.

Average precision (AP) is commonly used to evaluate the overall performance of a model in detecting the single-class pear fruits by considering both precision and recall. Mean average precision (mAP) is the average of AP values across all categories of pear fruits. In pear fruit detection tasks, mAP50 (the mAP value when the intersection over union, IoU, between the predicted box and the ground truth box is greater than or equal to 0.5, typically used for initial model evaluation) and mAP50–95 (the mAP value when IoU ranges from 0.5 to 0.95, offering a more comprehensive assessment of the model's performance across different detection accuracy thresholds) are often the key metrics. The calculation formulas for AP and mAP are

$$AP = \int_0^1 P(r)dr \tag{7}$$

Here, $P(r)$ refers to the precision rate corresponding to the recall rate in the precision–recall rate curve (P–R Curve).

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i \tag{8}$$

Here, c is the number of categories and AP_i is the AP value of the category i.

3.4. Experimental Environment and Parameter Settings

In order to ensure that the training and evaluation process of the model was fair and reliable, an expanded image dataset was randomly divided into a training set and validation set at a ratio of 4:1, including 1610 images in the training set and 402 images in the validation set. The training set contained 3246 yellow pear tags and 2160 green pear tags, and the validation set contained 819 yellow pear tags and 510 green pear tags. The experimental environment used in this study is shown in Table 1. The hyperparameter settings are presented in Table 2.

Table 1. Experimental environment configuration.

Environment Configuration	Parameter
Operating system	Windows
CPU	13th Gen Intel(R) Core(TM) i5-13600KF @3.50 GHz
GPU	NVIDIA GeForce RTX 4070(12 GB)
Development environment	PyCharm 2023.2.4
Language	Python 3.9.18
Frame	Pytorch 2.1.0
Operating platform	CUDA 11.8

Table 2. Hyperparameter settings.

Hyperparameter	Parameter
Epochs	300
Batch size	32
Num_workers	8
Initial learning rate	0.01
Momentum	0.937
Weight decay	0.0005
Input image size	640

4. Results

4.1. Experimental Results of the Improved YOLO11 Model

Figure 7 shows the experimental results of the YOLO11-Pear model. Figure 8 shows the P–R Curve and the F1 Score Curve of YOLO11-Pear model.

As can be seen from the Figure 7, the various loss metrics of the model decreased continuously with the increase in the number of iterations, indicating that the model was gradually optimized during the training process and its fitting ability improved. The accuracy, recall, and mAP metrics increased as the number of iterations grew, suggesting that the model effectively learned more precise features from the training data, gradually reducing errors and enhancing its ability to detect pears. The precision, recall, mAP50, and mAP50–95 of the YOLO11-Pear model reached 96.3%, 84.2%, 92.1%, and 80.2%, respectively. Figure 8a shows the P–R Curve of YOLO11-Pear, which illustrates the relationship between precision and recall during the pear fruit detection training process. It can be observed that the curve mostly lies in the upper-right quadrant of the graph, with small fluctuations, indicating that the model had good robustness and generalization ability. In the detection task with two types of pear fruits, mAP50 reached 92.1%, demonstrating the excellent performance of the YOLO11-Pear model. In Figure 8b, the F1 score of YOLO11-Pear reached 90%. The F1 Score Curve provides a more comprehensive evaluation of the model's ability to detect different types of pears at different confidence thresholds. As the confidence increased, the F1 score rose rapidly, and when the confidence reached 0.8, the F1

score started to decline, indicating that within this range, the model excelled in balancing the precision and recall.

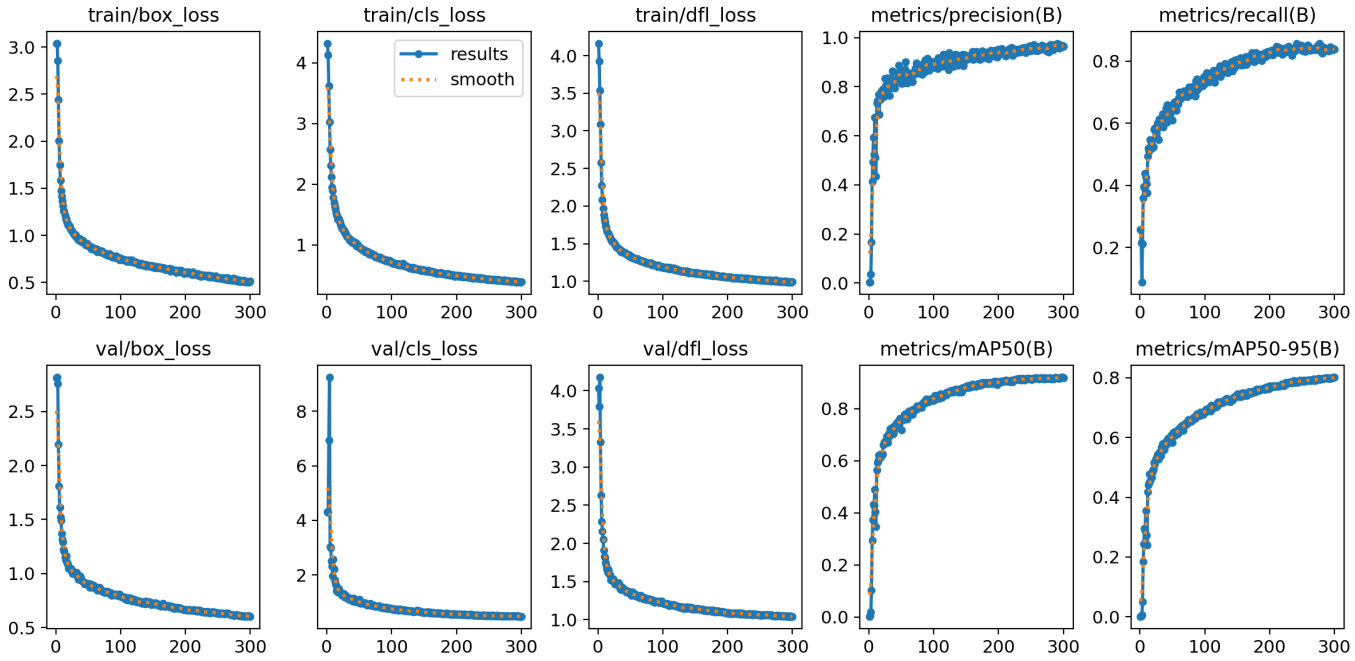
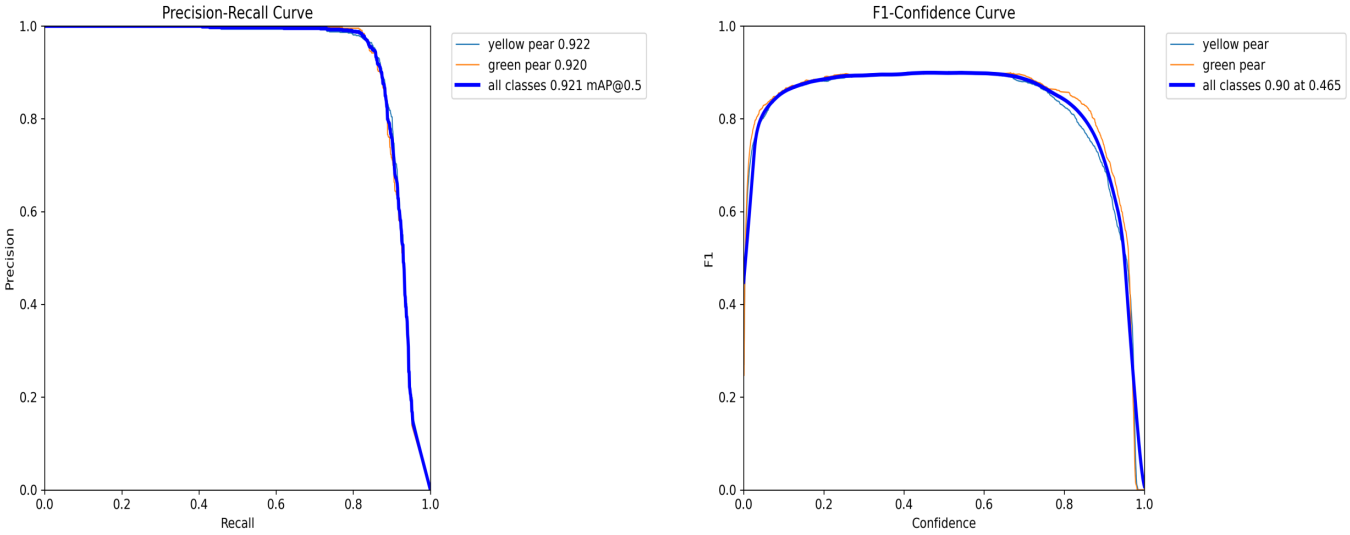


Figure 7. Experimental results of the YOLO11-Pear model.



(a) (b)

Figure 8. (**a**) P–R Curve and (**b**) F1 Score Curve of the YOLO11-Pear model. P–R Curve mostly lies in the upper-right quadrant of the graph and the fluctuation is small. As the confidence increases, the F1 score rises rapidly, and when the confidence reaches 0.8, the F1 score starts to decline.

4.2. Comparison of Different Models Experiment

In order to verify the effectiveness of the improved model, this paper selected multiple state-of-the-art object detection models, based on the same training dataset, for comparative experiments, including SSD [27], Faster R-CNN [28], YOLOv3 tiny [29], YOLOv5n [30], YOLOv6n [31], YOLOv8n [32], YOLOv9t [33], YOLOv11n [34], and the YOLO11-Pear model proposed in this paper. SSD is an early single-stage object detection algorithm that

has a certain balance between detection speed and accuracy; Faster R-CNN is a classic representative of two-stage object detection algorithms and is widely used in many practical applications; and the YOLO series is the most popular single-stage object detection model in recent years. This article provides a more comprehensive evaluation of the performance advantages and disadvantages of the YOLO11-Pear model by comparing these different models. The comparative experimental results of the different models are shown in Table 3, and a comparison of the precision, recall, and mAP results is shown in Figure 9.

Table 3. Comparative experimental results of the different models. The best performance is in bold.

Models	Precision/%	Recall/%	F1 Score/%	mAP50/%	mAP50–95/%	Parameters (m)
SSD [27]	93.3	57.1	70.8	71.1	48.7	3.44
Faster R-CNN [28]	67.8	**84.9**	75.4	84.4	62	41.5
YOLOv3 tiny [29]	93.8	79.5	86.1	86.2	74.1	12.129
YOLOv5n [30]	93.6	81.1	86.9	88.5	75.6	2.503
YOLOv6n [31]	94.1	80.3	86.7	87.4	76	4.234
YOLOv8n [32]	94.6	81.4	87.5	89.4	77.3	3.006
YOLOv9t [33]	92.6	81.8	86.9	89.1	75.9	**1.971**
YOLOv11n [34]	92.7	83.2	87.7	90	77	2.583
YOLO11-Pear	**96.3**	84.2	**89.8**	**92.1**	**80.2**	2.644

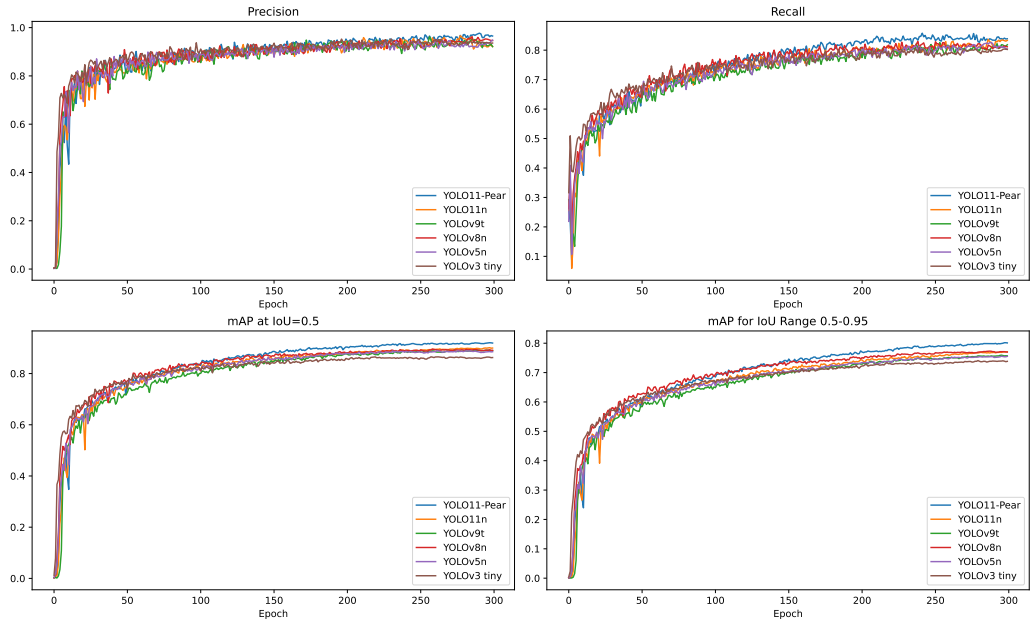


Figure 9. Precision, recall, and mAP results of different YOLO models.

The results indicate that the YOLO11-Pear model outperformed the other eight models. Compared to these models, the YOLO11-Pear model achieved a 3%, 28.5%, 2.5%, 2.7%, 2.2%, 1.7%, 3.7%, and 3.6% increase in precision; a 19%, 14.4%, 3.7%, 2.9%, 3.1%, 2.3%, 2.9%, and 2.1% increase in F1 score; a 21%, 7.7%, 5.9%, 3.6%, 4.7%, 2.7%, 3%, and 2.1% increase in mAP50; and a 31.5%, 18.2%, 6.1%, 4.6%, 4.2%, 2.9%, 4.3%, and 3.2% increase in mAP50–95, respectively. The recall of the YOLO11-Pear model was 27.1%, 4.7%, 3.1%, 3.9%, 2.8%, 2.4%, and 1% higher than SSD, YOLOv3 tiny, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv9t, and YOLO11n, respectively, and 0.7% lower than Faster R-CNN. In terms of model parameters, compared to YOLOv5n, YOLOv9t, and YOLO11n, the parameter count of the YOLO11-Pear model increased by 5.6%, 34.1%, and 2.4%, respectively, while the other five models had higher parameter counts than the YOLO11-Pear model, especially Faster R-CNN and YOLOv3 tiny, which had parameter counts 15 times and 4 times higher than

YOLO11-Pear, respectively. Under the condition of only a slight increase in parameters, the YOLO11-Pear model achieved the best results in terms of precision, F1 score, mAP50, and mAP50–95.

4.3. Comparison of Different Attention Mechanisms Experiment

To further validate the effectiveness of the spatially enhanced attention module (SEAM) in the C2PSS module, SEAM was replaced with different attention mechanisms in the YOLO11-Pear model, including SE, EMA, CBAM, and GAM. These four simple and efficient attention mechanisms have been widely used by researchers in recent years, and comparing them with SEAM provided a more comprehensive evaluation of the advantages and disadvantages of the C2PSS module. The comparison results of YOLO11-Pear using the different attention mechanisms are shown in Table 4.

Table 4. Comparison results of YOLO11-Pear using different attention mechanisms. The best performance is in bold.

Module	Precision/%	Recall/%	mAP50/%	mAP50–95/%	Parameters (m)
SE	93.7	85.8	91.4	79.1	2.67
EMA	94.8	**86.3**	91.7	79.2	**2.62**
CBAM	95.9	83.7	92.1	79.4	2.64
GAM	96.1	84.4	**92.3**	**80.5**	3.03
SEAM	**96.3**	84.2	92.1	80.2	2.64

The precision of SEAM was 96.3%, ranking first among the five attention mechanisms. EMA had the highest recall at 86.3%, while SEAM had 84.2%, indicating that SEAM may have some compromises in comprehensively capturing real targets. The mAP50 and mAP50–95 of GAM were the highest, accounting for 92.3% and 80.5%, respectively, followed by SEAM, with mAP50 and mAP50–95, accounting for 92.1% and 80.2%, respectively, which was not significantly different from GAM. In terms of parameter count, GAM had the highest parameter count, increasing by 14.8% compared to SEAM. Although GAM performed well in multiple performance metrics, a higher number of parameters means higher computational costs and storage requirements, which may not be suitable for resource constrained environments. Therefore, considering all factors, SEAM had the best overall performance.

4.4. Ablation Experiment

The YOLO11-Pear proposed in this paper is based on YOLO11n and incorporates three different improvements. To validate the effectiveness of each improvement in the YOLO11-Pear model, all possible combinations of the three improvement modules were arranged and ablation experiments were conducted on the aforementioned dataset. The experimental results are shown in Table 5.

As shown in Table 5, the model with only the C2PSS module added improved the mAP50 and mAP50–95 by 1.2% and 1.3%, respectively, while the parameter count decreased by 1.2%. The model with only the small target detection layer added improved the mAP50 and mAP50–95 by 1% and 1.4%, respectively, but the parameter count increased by 2.9%. The model with only the DySample module added improved the mAP50 and mAP50–95 by 0.5% and 0.7%, with a 0.5% increase in parameters. In models where two improvement modules were randomly added, the mAP metrics were higher than those of models with only one improvement module, indicating that the improvement modules remained effective when combined. Notably, the model with both the C2PSS and DySample modules showed an improvement in mAP50 and mAP50–95 by 1.5% and 1.3%, respectively, while the parameter count decreased. The results demonstrated that modifying the

YOLO11n model with different module combinations consistently improved the mAP for pear fruit detection.

Table 5. Ablation experiment results. "✓" indicates that the module is used, "-" indicates that the module is not used. The best performance is in bold.

C2PSS	Small Target Detection Layer	DySample	Precision/%	Recall/%	mAP50/%	mAP50–95/%	Parameters (m)
-	-	-	92.7	83.2	90	77	2.583
✓	-	-	94.4	83.7	91.2	78.3	**2.553**
-	✓	-	94.9	83.1	91	78.4	2.659
-	-	✓	93.4	83.7	90.5	77.7	2.595
✓	✓	-	95	84.3	91.6	78.9	2.629
✓	-	✓	93.2	84.9	91.5	78.3	2.565
-	✓	✓	93.7	**85.8**	91.4	79.1	2.673
✓	✓	✓	**96.3**	84.2	**92.1**	**80.2**	2.644

4.5. Evaluation Experiments in Complex Orchard Environments

This paper considered three different factors—color, size, and occlusion—to evaluate the effectiveness of the model in complex orchard environments. A random selection of 15–35 images of each type was made for comparison and analysis. The detection performance of YOLO11-Pear on pear fruit images under different conditions was compared with YOLO11n, YOLOv8n, and YOLOv5n. Light blue boxes indicate the green pears identified by the model, while dark blue boxes indicate the yellow pears identified by the model. By comparing the original labels and the predicted labels, red and yellow boxes were marked on the visualized images (The light blue boxes represent the green pear labels detected by the model, the dark blue boxes represent the yellow pear labels detected by the model, the red boxes represent unrecognized pear labels, and the yellow boxes represent incorrect labels).

4.5.1. Comparison of Detection for Pears of Different Colors

The pear fruits in this dataset were divided into two colors: green and yellow. The detection results of YOLO11-Pear compared with the YOLO11n, YOLOv8n, and YOLOv5n models for pears of different colors are shown in Table 6, and the visualizations are presented in Figure 10.

Table 6. Detection results for pears of different colors.

Models	Green			Yellow		
	Actual	Detected	mAP50–95/%	Actual	Detected	mAP50–95/%
YOLOv5n [30]	190	179	92.9	204	187	83.2
YOLOv8n [32]	190	178	94.1	203	190	85.3
YOLO11n [34]	190	179	92.4	203	181	84.5
YOLO11-Pear	190	188	95.2	203	192	88.3

As shown in Table 6, the YOLO11-Pear model achieved high recognition rates for pears of different colors, with recognition rates of 98.9% for green pears and 94.6% for yellow pears. Additionally, the mAP50–95 for green and yellow pears reached 95.2% and 88.3%, respectively. Compared to yellow pears, the model achieved higher recognition rates and mAP50–95 for green pears, indicating that the improved model could effectively distinguish green pear labels from the background. However, for yellow pears, the model may have mistakenly identified tree branches as yellow pears, leading to a decrease in

both recognition rate and mAP. Compared with the YOLO11n, YOLOv8n, and YOLOv5n models, YOLO11-Pear detected more pears, demonstrating that the improved model had a higher accuracy and better generalization when recognizing pears of different colors. From Figure 10, it can be seen that all three comparison models missed detecting the green pear at the edge of the image when detecting the first image; YOLO11n mistakenly detected the branch as a yellow pear when detecting the second image; YOLO5n missed the small pear when detecting the third image.



Figure 10. Visualization for pears of different colors.

4.5.2. Comparison of Detection for Pears of Different Sizes

Due to the characteristic of appearing larger when closer and smaller when farther away, the pears in images may vary in size. To evaluate the model's detection performance for pears of different sizes, a comparison experiment was conducted using large fruits, small fruits, and fruits of varying sizes. The detection results of YOLO11-Pear and three other network models for pears of different sizes are shown in Table 7, with visualizations provided in Figure 11.

Table 7. Detection results for pears of different sizes.

Models	Big			Small			Varying Sizes		
	Actual	Detected	mAP50–95/%	Actual	Detected	mAP50–95/%	Actual	Detected	mAP50–95/%
YOLOv5n [30]	79	76	95.9	171	154	79.2	213	194	85.7
YOLOv8n [32]	79	76	96.9	171	154	83	213	205	88.5
YOLO11n [34]	79	76	96	171	149	82.1	213	202	86.7
YOLO11-Pear	79	76	96.9	171	157	86.7	213	206	90.9

The recognition rates and mAP50–95 of the four models for large fruits were consistently high, indicating that the YOLO model had a strong detection capability for large objects. Under the condition of small fruits, the YOLO11-Pear, YOLOv8n, and YOLOv5n models achieved relatively high recognition rates. The mAP50–95 of YOLO11-Pear reached 86.7%, which was 4.6%, 3.7%, and 7.5% higher than that of YOLO11n, YOLOv8n, and YOLOv5n, respectively. Under the condition of the different sized fruits, the YOLO11-Pear and YOLOv8n models detected more pears and also achieved higher mAP values. Therefore, it can be concluded that the YOLO11-Pear model exceled in detecting pears

of different sizes. The visualization results show that all four models had outstanding detection capabilities when detecting large pears, and the differences between the different models were also very small. When detecting small pears and pears of different sizes, YOLO11n and YOLOv5n performed poorly, with many small pears missed in the three visualization images. YOLOv8n performed moderately, with fewer small pears missed. YOLO11-Pear performed well, with no missed detections in the three images.

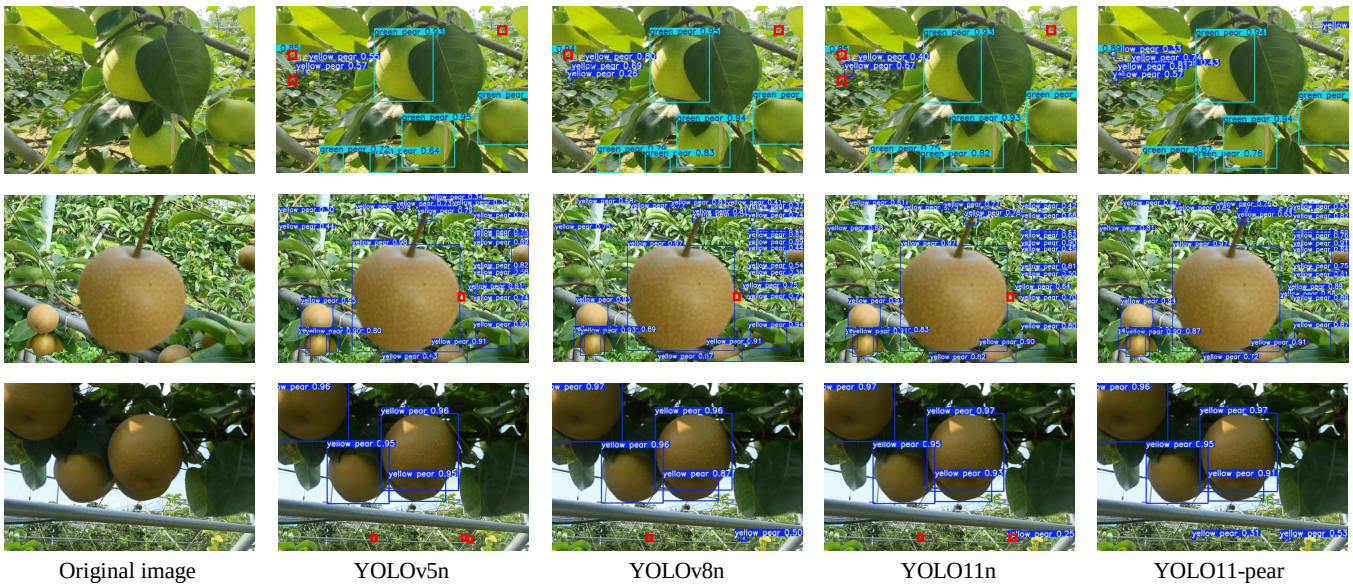


Figure 11. Visualization for pears of different sizes.

4.5.3. Comparison of Detection of Pears with and Without Occlusion

In complex orchard environments, pear fruits are often subject to branch and leaf occlusion, as well as fruit overlap, which can significantly affect fruit detection. This study compared pear detection under two conditions: occlusion and non-occlusion. The detection results of the four models are shown in Table 8, and visualizations are presented in Figure 12.

Table 8. Detection results for pears with occlusion and non-occlusion.

Models	No Occlusion			Occlusion		
	Actual	Detected	mAP50–95/%	Actual	Detected	mAP50–95/%
YOLOv5n [30]	98	94	90.6	161	152	89.8
YOLOv8n [32]	98	96	92.3	161	154	91.1
YOLO11n [34]	98	97	91.9	161	155	89.8
YOLO11-Pear	98	97	94.5	161	155	93

The results show that the YOLO11-Pear and YOLO8n models consistently exhibited high recognition rates when detecting both occluded and non-occluded pears. The YOLO11-Pear model achieved recognition rates and mAP50–95 of 96.3% and 93%, respectively, for detecting occluded pears, which were 2.6% and 1.5% lower than the non-occluded scenario. This decrease was due to the loss of some features caused by occlusion, which affected the detection results. Compared to the other three models, YOLO11-Pear had the highest mAP value, indicating that the improved model enhanced the pear detection performance under various occlusion conditions. From Figure 12, it can be seen that YOLO11n, YOLOv8n, and YOLOv5n all missed the pear at the edge of the image when detecting the first image; YOLO11n missed the pear obscured by the bracket when detecting the second image;

YOLO11n, YOLOv8n, and YOLOv5n all missed the pear obscured by branches and leaves when detecting the third image.

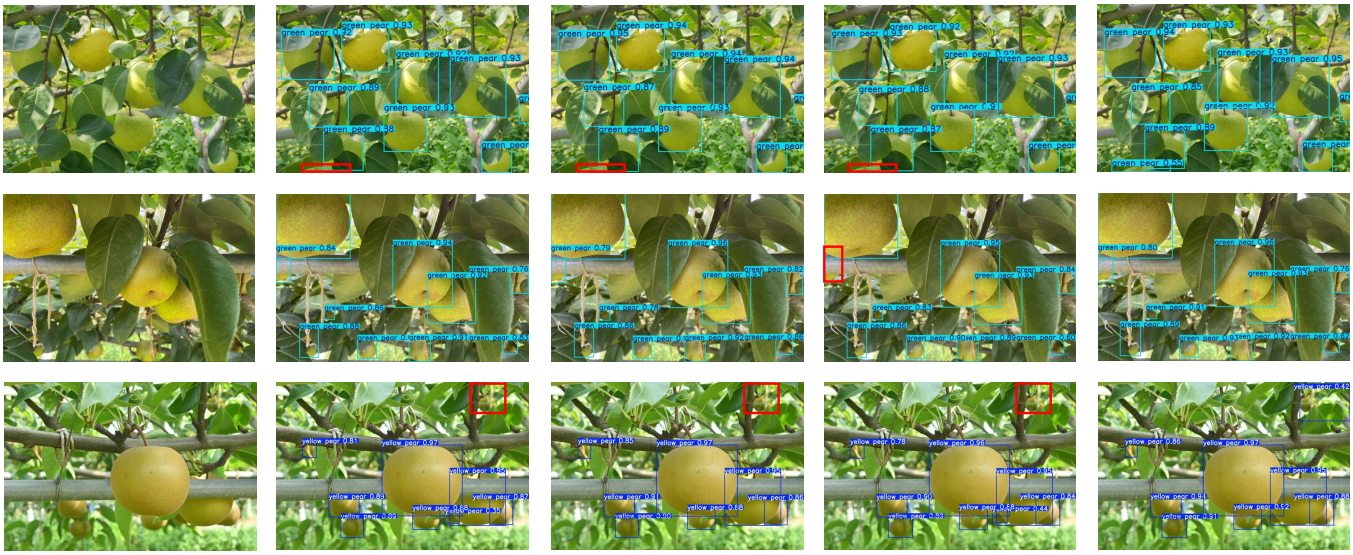


Figure 12. Visualization for pears with occlusion and non-occlusion.

5. Discussion

Although the YOLO11-Pear model performed well in accuracy and other metrics, it may require a trade-off between real-time performance and accuracy on resource-limited devices. If forced to run on agricultural robots, in order to ensure the accuracy of the model, this may result in a slow detection speed and failure to meet the real-time monitoring needs of orchards. On the contrary, if the running accuracy of the model was reduced in order to ensure real-time performance, the significance of the model improvement would be be lost. Therefore, further research is needed on how to optimize the operation of the model under limited resources and find the optimal balance between real-time performance and accuracy.

When studying the YOLO11-Pear model, only two varieties, Qiuyue pear and Cuiguan pear, were selected for testing. This is because it is difficult to collect high-quality datasets covering multiple varieties of pears, and different varieties of pears have differences in appearance, growth environment, etc. Field collection is limited by the season and region, and sample annotation requires a significant amount of manpower and time. Conducting research solely on the Qiuyue pear and Cuiguan pear reduced the complexity of data acquisition and annotation to a certain extent, allowing the study to focus more on model improvement. In future research, we will collect more samples of different pear varieties and retrain the model to learn a wider range of pear features. At the same time, transfer learning and other techniques could be used to utilize the training results of the existing Qiuyue pear and Cuiguan pear data, combined with a small amount of new pear variety data for fine-tuning, in order to improve the model's detection ability for multiple pear varieties and to enhance its portability.

6. Conclusions

This paper proposed an improved pear detection model, YOLO11-Pear, based on YOLO11n, to address the issues of low detection accuracy and poor adaptability in complex orchard environments. First, a pear image dataset for complex orchard environments was constructed. After collecting 503 pear images, data augmentation was performed to

generate a total of 2012 images. To improve the detection capability for occluded pears, the C2PSS module was introduced to replace the original C2PSA module. Additionally, a small object detection layer was added to enhance the detection ability for small pears. Lastly, upsampling was replaced with DySample to improve the model's ability to handle densely packed pears, while maintaining a high efficiency. The experimental results showed that the improved model YOLO11-Pear achieved a mAP50 of 92.1%, which was 2.1%, 2.7%, and 3.6% higher than the YOLOv11n, YOLOv8n, and YOLOv5n models, respectively; and a mAP50–95 of 80.2%, which was 3.2%, 2.9%, and 4.6% higher than the YOLOv11n, YOLOv8n, and YOLOv5n models, respectively. Evaluation experiments demonstrated that the YOLO11-Pear model consistently maintained a high recognition rate and mAP value under various conditions, including different colors, sizes, and occlusion scenarios in complex orchard environments.

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

Funding: This research was funded in part by the National Natural Science Foundation of China under Grants 61901221, and in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant SJCX24_0363 and Grant SJCX24_0364.

Data Availability Statement: Data are contained within the article.

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

References

1. Saquet, A.A. Storage of pears. *Sci. Hortic.* **2019**, *246*, 1009–1016. [CrossRef]
2. Li, J.; Zhang, M.; Li, X.; Khan, A.; Kumar, S.; Allan, A.C.; Lin-Wang, K.; Espley, R.V.; Wang, C.; Wang, R.; et al. Pear genetics: Recent advances, new prospects, and a roadmap for the future. *Hortic. Res.* **2022**, *9*, uhab040. [CrossRef]
3. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight "Yuluxiang" Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [CrossRef]
4. Tang, Y.; Qiu, J.; Zhang, Y.; Wu, D.; Cao, Y.; Zhao, K.; Zhu, L. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review. *Precis. Agric.* **2023**, *24*, 1183–1219. [CrossRef]
5. Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. Object Detection and Recognition Techniques Based on Digital Image Processing and Traditional Machine Learning for Fruit and Vegetable Harvesting Robots: An Overview and Review. *Agronomy* **2023**, *13*, 639. [CrossRef]
6. Wei, J.; Ni, L.; Luo, L.; Chen, M.; You, M.; Sun, Y.; Hu, T. GFS-YOLO11: A Maturity Detection Model for Multi-Variety Tomato. *Agronomy* **2024**, *14*, 2644. [CrossRef]
7. Tan, K.; Lee, W.S.; Gan, H.; Wang, S. Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. *Biosyst. Eng.* **2018**, *176*, 59–72. [CrossRef]
8. Lu, J.; Lee, W.S.; Gan, H.; Hu, X. Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis. *Biosyst. Eng.* **2018**, *171*, 78–90. [CrossRef]
9. Kurtulmus, F.; Lee, W.S.; Vardar, A. Green citrus detection using "eigenfruit", color and circular Gabor texture features under natural outdoor conditions. *Comput. Electron. Agric.* **2011**, *78*, 140–149. [CrossRef]
10. Zhang, W.; Wang, J.; Liu, Y.; Chen, K.; Li, H.; Duan, Y.; Wu, W.; Shi, Y.; Guo, W. Deep-learning-based in-field citrus fruit detection and tracking. *Hortic. Res.* **2022**, *9*, uhac003. [CrossRef]
11. Zhu, A.; Zhang, R.; Zhang, L.; Yi, T.; Wang, L.; Zhang, D.; Chen, L. YOLOv5s-CEDB: A robust and efficiency *Camellia oleifera* fruit detection algorithm in complex natural scenes. *Comput. Electron. Agric.* **2024**, *221*, 108984. [CrossRef]
12. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2023**, *35*, 13895–13906. [CrossRef]
13. Sun, H.; Wang, B.; Xue, J. YOLO-P: An efficient method for pear fast detection in complex orchard picking environment. *Front. Plant Sci.* **2023**, *13*, 18. [CrossRef] [PubMed]

14. Zhao, S.; Zheng, J.; Sun, S.; Zhang, L. An Improved YOLO Algorithm for Fast and Accurate Underwater Object Detection. *Symmetry* **2022**, *14*, 1669. [CrossRef]
15. Hao, X.; Liu, L.; Yang, R.; Yin, L.; Zhang, L.; Li, X. A Review of Data Augmentation Methods of Remote Sensing Image Target Recognition. *Remote Sens.* **2023**, *15*, 827. [CrossRef]
16. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning—Method overview and review of use for fruit detection and yield estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234. [CrossRef]
17. Zhang, Q.; Liu, Y.; Gong, C.; Chen, Y.; Yu, H. Applications of Deep Learning for Dense Scenes Analysis in Agriculture: A Review. *Sensors* **2020**, *20*, 1520. [CrossRef]
18. Zhou, H.; Xia, H.; Fan, C.; Lan, T.; Liu, Y.; Yang, Y.; Shen, Y.; Yu, W. Intelligent Detection Method for Surface Defects of Particleboard Based on Super-Resolution Reconstruction. *Forests* **2024**, *15*, 2196. [CrossRef]
19. Amjoud, A.B.; Amrouch, M. Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access* **2023**, *11*, 35479–35516. [CrossRef]
20. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
21. Wang, X.; Wang, A.; Yi, J.; Song, Y.; Chehri, A. Small Object Detection Based on Deep Learning for Remote Sensing: A Comprehensive Review. *Remote Sens.* **2023**, *15*, 3265. [CrossRef]
22. Ni, J.; Zhu, S.; Tang, G.; Ke, C.; Wang, T. A Small-Object Detection Model Based on Improved YOLOv8s for UAV Image Scenarios. *Remote Sens.* **2024**, *16*, 2465. [CrossRef]
23. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 6027–6037.
24. Lin, Z.; Yun, B.; Zheng, Y. LD-YOLO: A Lightweight Dynamic Forest Fire and Smoke Detection Model with Dysample and Spatial Context Awareness Module. *Forests* **2024**, *15*, 1630. [CrossRef]
25. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review. *Comput. Electron. Agric.* **2024**, *223*, 109090. [CrossRef]
26. Li, Y.; Xia, H.; Liu, Y.; Sun, Q.; Huo, L.; Ni, X. Research on the detection method of phenotypic information of Pinus massoniana Lamb. seedling root system. *Signal Image Video Process.* **2024**, *18*, 6961–6972. [CrossRef]
27. 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 Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
29. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694.
30. Zhao, Y.; Ju, Z.; Sun, T.; Dong, F.; Li, J.; Yang, R.; Fu, Q.; Lian, C.; Shan, P. TGC-YOLOv5: An Enhanced YOLOv5 Drone Detection Model Based on Transformer, GAM & CA Attention Mechanism. *Drones* **2023**, *7*, 446. [CrossRef]
31. Norkobil Saydirasulovich, S.; Abdusalomov, A.; Jamil, M.K.; Nasimov, R.; Kozhamzharova, D.; Cho, Y.I. A YOLOv6-Based Improved Fire Detection Approach for Smart City Environments. *Sensors* **2023**, *23*, 3161. [CrossRef] [PubMed]
32. Chen, H.; Zhou, G.; Jiang, H. Student Behavior Detection in the Classroom Based on Improved YOLOv8. *Sensors* **2023**, *23*, 8385. [CrossRef] [PubMed]
33. Wang, Y.; Rong, Q.; Hu, C. Ripe Tomato Detection Algorithm Based on Improved YOLOv9. *Plants* **2024**, *13*, 3253. [CrossRef]
34. Huang, Y.; Wang, D.; Wu, B.; An, D. NST-YOLO11: ViT Merged Model with Neuron Attention for Arbitrary-Oriented Ship Detection in SAR Images. *Remote Sens.* **2024**, *16*, 4760. [CrossRef]

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