ODL Net: Object Detection and Location Network for Small Pears around the Thinning Period

Yuqi Lu[1], Shuang Du[1], Ze Ji[2], Xiang Yin[3*], Weikuan Jia[1,4*]

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

[2] School of Engineering, Cardiff University, Cardiff CF24 3AA, UK

[3] School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo 255000, China

[4] Key Laboratory of Facility Agriculture Measurement and Control Technology and Equipment of Machinery Industry, Zhenjiang 212013, China

Abstract: In the process of efficient management of intelligent orchards, due to the short cycle and high intensity of fruit thinning, it is urgent to realize the automatic operation of fruit thinning in orchards. However, affected by the complex orchard environment, the color of fruit and the background are similar, and the more important problem is that the fruit is small-scale. These factors bring great challenges to fruit detection before and after the thinning period. For this reason, a detection algorithm for fruits of small green objects is proposed, namely, ODL Net. By integrating the semantic enhancement module and label assignment Center-Box, the small size problem of the target fruit is alleviated. The feature enhancement module and position enhancement module are constructed to enhance the fusion effect of features and improve the detection accuracy. To better verify the performance of the algorithm, this study takes a pear orchard as an example to produce two datasets before and after pear thinning. The experimental results show that the detection accuracy of ODL Net can reach 56.2% and 65.1% before and after the fruit thinning period, respectively, and the recall rate can reach 61.3% and 70.8%, respectively, which are significantly higher than those of other mainstream algorithms at present. The new algorithm can effectively assist the orchard automatic fruit

thinning operation and provide the basis for orchard yield measurement after the fruit thinning period. This study can provide a theoretical basis for the scientific management of intelligent orchards.

Keywords: ODL Net; Fruit thinning; Small fruit detection; Feature fusion

1. Introduction

The development of cutting-edge theories and technologies such as artificial intelligence and 5G communication provides strong support for efficient agricultural production. Intelligent agriculture (Patrício and Rieder, 2018; Wu and Tsai, 2019) and orchards (Xu et al., 2023; Maheswari, et al., 2021) have gradually entered the public view, and agricultural production efficiency has been greatly improved. In the orchard production process, due to the short operation cycle and high labor intensity of fruit thinning, it is urgent to realize automatic fruit thinning in orchards. However, in the fruit thinning period, orchards present a complex environment, and the fruit color is similar to the background and is still small-scale and easily covered by branches and leaves. These factors bring great challenges to the efficient recognition of fruit during this time. The realization of the efficient detection of small fruits can assist automatic fruit thinning operations in orchards when the fruit is clustered at the early stage of fruit thinning. It can also assist the fruit yield measurement to realize the scientific management of the orchard when the fruit is in a single state in the late stage of fruit thinning. In addition, it also helps fruit farmers recalculate irrigation and fertilizer supply due to the change in fruit quantity after thinning. Taking the golden pear orchard as an example, this study focuses on the detection accuracy of small target fruits before and after pear thinning and constructs a high-precision small fruit detection algorithm.

In the orchard environment, object detection has been widely used in orchards (Gongal et al., 2015; Fu et al., 2020; Tang et al., 2023), such as automatic driving (Yang et al., 2021; Tey and Brindal, 2022), pest detection (Ebrahimi et al., 2017; Ngugi et al., 2021), and other operations. Its detection accuracy also restricts the production efficiency of orchards. In complex orchard environments, fruit detection has attracted many scholars' attention and has also achieved gratifying research results. Sa (Sa et al., 2016) proposed a fruit detection algorithm based on Faster RCNN that used images obtained from two modes, color (RGB) and near-infrared (NIR), to compose multi-modal information; in this paper, the algorithm was applied to the detection task of seven kinds of fruits, such as sweet pepper and rock melon. Bargoti (Bargoti, et al., 2017) proposed a tiling method for images containing more than 100 target fruits; combined with image enhancement technology, the F1-score of this new algorithm on apples and mangoes exceeded 0.9. Zhao (Zhao and Yan, 2021) proposed CenterNet for fruit detection, which implemented three backbone networks and finally confirmed CenterNet based on DLA-34 (Yu et al., 2018). In addition, Jia (Jia et al., 2021) presented an algorithm with a transformer structure, which was popular in recent years, to detect green apples in orchards. Hussain (Hussain et al., 2022) proposed a deep learning based framework for automatic detection and recognition of fruits and vegetables in complex scenes. It can help sellers identify vegetables and fruits with high similarity. Although its accuracy is as high as 96%, there is no special design for detecting small-scale fruits. Most of the above algorithms were detection algorithms proposed for specific orchard environments. These algorithms achieved relatively ideal detection results for large-scale fruits, green fruits, etc. However, they ignored the detection effect of small-scale target fruits.

The detection effect of small objects is easily affected by the external environment. For example, the proportion of pixels is small, so the features are difficult to effectively represent. The target itself is small-scale and easily occluded by the background, resulting in missing its recognition. In addition, the color of a small target is similar to the background, leading to incorrect identification. Small object detection is so challenging that it has attracted scholars' attention in many fields. Rabbi (Rabbi et al., 2020) used small objects to over-sample images and enhanced each image by copying and pasting small objects many times to achieve small object detection; the detection accuracy of small objects on the MS COCO dataset was increased by 7.1 percentage points. Yang (Yang et al., 2019) presented a novel multi-category rotation detector for small, cluttered and rotated objects, namely, SCRDet, in which a sampling fusion network was devised that fused multi-layer features with effective anchor sampling to improve the sensitivity to small objects. It was shown on the two remote sensing public datasets and the COCO and VOC 2007 dataset. In the area of remote sensing, Zhang (Zhang et al., 2018) proposed a network with deconvolution layers after the last convolution layer of the basic network for small object detection in remote sensing data; in an experiment on a remote sensing image dataset, the Deconv RCNN reached a much higher mean average precision than the Faster RCNN. Inspired by these different fields, research on small fruit detection has also made significant progress. Mai (Mai et al., 201) presented a multi-classifier fusion strategy for small fruits, which used three different feature levels to learn three classifiers for object classification in the proposal localization phase; at the same time, a new classifier correlation loss term was introduced to improve the detection accuracy of small objects. Tu (Tu et al., 2020) proposed an improved method based on multi-scale Faster RCNN, which used color and depth images acquired by an RGB-D camera; it was improved by combining the feature map of the shallow convolution maps from the region of interest (ROI) pool to detect small passions. Sun (Sun et al., 2022) proposed a balanced feature pyramid network (BFP Net) for small apple detection; the network balanced the information mapped to small apples from two perspectives and was verified on three fruit datasets. The above algorithms achieved ideal results in solving small objects or small target fruit detection. However, these small targets were mostly "small" due to the perspective that they were not as small as the fruit in the fruit thinning period.

At present, there are relatively few studies on fruit recognition during fruit thinning. At this time, the state of fruit appears as the target color is similar to the background, and the real volume of fruit is relatively small and easily blocked. To solve the above problems, this study presents ODL Net, a detection algorithm for small-scale fruit around the pear thinning period. The semantic enhancement module (SEM) and the label assignment Center-Box in this algorithm can deal with small-scale fruit detection well. In addition, the feature enhancement module (FEM) and the positional enhancement module (PEM) for feature fusion also improve the detection accuracy. The following is an explanation of the innovations in this paper:

(1) This study presents ODL Net, a novel detection algorithm for small fruits around the pear thinning period. The detection accuracy in orchards is higher than that of most current detection algorithms.

(2) Two pear datasets are prepared in this study, including image data before and after the pear thinning period. In this way, the detection effect of ODL Net around this period in the orchard can be accurately verified.

(3) Three modules, SEM, FEM, PEM are constructed in the feature fusion network. The modules enhance the information from different angles and provide it to the downstream detection task of the ODL Net.

(4) ODL Net relies on a special label assignment, Center-Box, to accurately locate small fruits.

Center-Box eliminates the influence of object size on positive sample allocation, avoiding ignoring small objects.

This study introduces the ODL Net, which aims to achieve precise detection of pear fruits in real orchard environments both before and after the thinning period. The pre-thinning detection of pears provides valuable insights to fruit farmers, allowing them to monitor early-stage fruit growth. This not only facilitates the determination of optimal irrigation and fertilizer supply for orchards but also guides the thinning process. Similarly, post-thinning detection of pear fruits remains crucial, providing ongoing recommendations for irrigation and fertilizer supply to fruit farmers and enabling scientifically informed yield predictions for orchards. In summary, the primary objective of this study is to enable comprehensive monitoring of fruit growth stages, encompassing both pre and post-thinning stages, thereby achieving intelligent management of orchards.

The organizational structure of this article is as follows: Section 1 describes this research purpose and related work in the current field. The second section is the production process of the two datasets. Section 3 details the composition of the ODL Net, as well as the structure and functions of each component. The experimental details, data and results are shown in Section 4, including contrast and ablation experiments. The summary and expectation of the overall research content is presented in Section 5.

2. Datasets

In this study, two datasets were produced, corresponding to the period before and after pear thinning. The object of the datasets was the golden pear around the fruit thinning period, which was characterized by its small-scale and full green color. The following is an introduction to the data collection and production process.

2.1 Data Acquisition

The objective of this study is to detect the fruit before and after the pear thinning period in orchards to provide thinning guidance for fruit farmers and realize intelligent orchard management. To achieve this goal, two pear datasets were made before and after the pear thinning period to test the feasibility of the ODL Net. The datasets were all taken from the RiSheng Golden Pear Professional Cooperative of Jiaozhou, Qingdao City, Shandong Province. The images taken were saved as.jpg, 24-bit color. As shown in Figure 1, the fruit was characterized by its small-scale and green color around the pear thinning period. It can be seen from the figure that the fruit density before thinning is higher than that after thinning, so detection before thinning is more difficult.

a) images before thinning period   b) images after thinning period

Fig. 1. Images around the pear thinning period in datasets

2.2 Data Processing

LabelMe was used to process the images taken. It used boxes to mark the target, with the marked closed part as the foreground, labeled "pear", and the remaining part as the background. Annotated images were automatically generated into.json files containing coordinates and label information. The final datasets were divided according to a ratio of 7:3. The dataset before pear thinning included 1549 images, with 1084 images in the training set and 465 images in the test set. The dataset after pear thinning included 891 images, with 623 images in the training set and 268 images in the test set. We also calculated statistics on the object scale of the dataset, and the information is shown in Table 1.

Table 1. Statistics of pear fruit scale

	small-scale	middle-scale	large-scale	images
dataset before thinning	4427 (48.41%)	3251 (35.55%)	1466 (16.04%)	1549
dataset after thinning	972 (45.13%)	641 (29.76%)	541 (25.11%)	891

It should be noted that COCO format datasets usually define objects with area pixels less than 32×32 as small-scale targets, objects with area pixels greater than 96×96 as large-scale targets, and objects between them are defined as medium-scale targets. However, the image size of the pear datasets we shot is 3024×4032 pixels, which is bigger than the image size of 640×640 pixels in the COCO dataset. Therefore, this study takes the pixel area as the standard and redefines the scale range according to the multiple relationships of areas. Objects with pixels less than 174×174 are small-scale targets, objects with pixels greater than 523×523 are large-scale targets, and objects in between are defined as medium-scale targets. As seen from Table 1, large-scale objects have the least amount. Intermediate-scale objects are the most numerous, accounting for more than half of the fruit. The fruit coordinates and sizes in the datasets are visualized as shown in Figure 2. The coordinate diagram shows that the fruit density after thinning is much lower than that before thinning. The fruit size before fruit thinning was smaller, concentrated within 0.1.

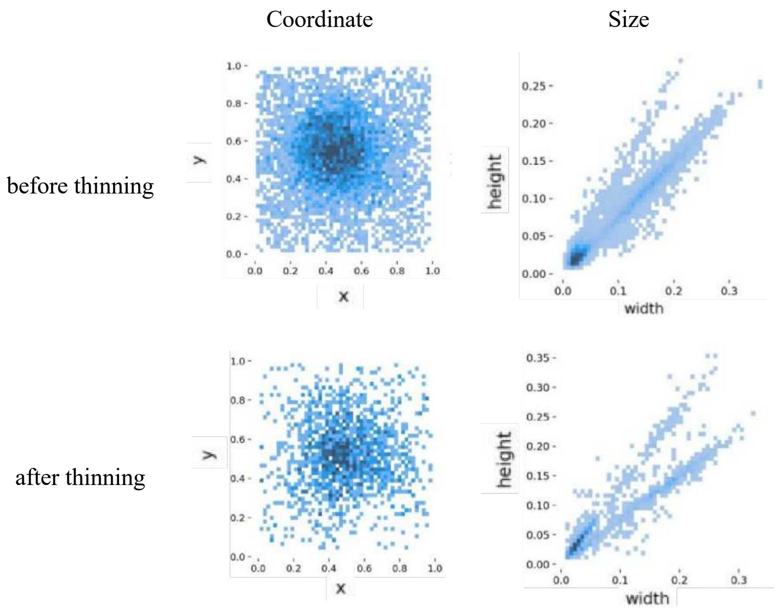


Fig. 2. Statistics of pears in datasets

3. ODL Net Detection Model

ODL Net includes two parts: the backbone and detection head. The backbone network consists of feature extraction (bottom-up) and feature fusion (top-down). Three enhancement modules are built in the backbone for more effective feature fusion and small fruit feature capture. In the detection head, this study uses a label assignment that ignores fruit size to strengthen the detection of small objects. The overall structure of the algorithm is shown in Figure 3.

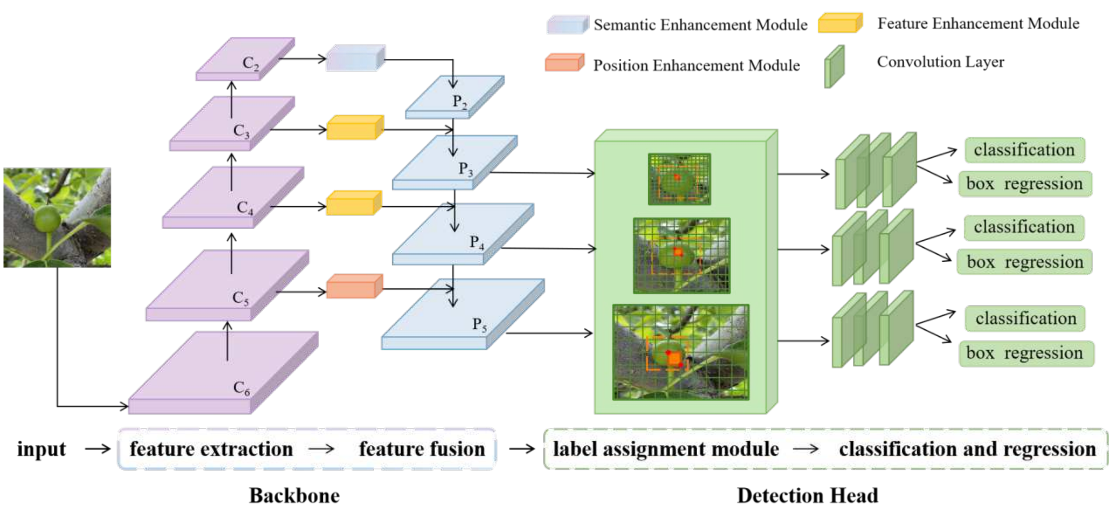


Fig. 3. The overall structure diagram of ODL Net

Note: ODL Net mainly includes two parts: the backbone and the detection head.

3.1 Image Enhancement

To enhance the learning ability of the network, the algorithm enhances the image for the input. Before the input is used for training, they are first scaled to 640×640 pixels. The scaled images are enhanced in four ways: random rotation, saturation transformation, random affine transformation and random stitching. An example of data enhancement is shown in Figure 4. The random rotation operation randomly rotates the original image by 90°. The random saturation transform changes the hue and saturation value of the input to simulate different light conditions in the orchard. Random affine transformation includes random translation, scaling and rotation. The translation, scaling, and rotation factors are set to 0.0625, 0.5, and 45 degrees, respectively. Finally, 4 images are randomly selected for mosaic processing. The image enhancement operation expands the learning range of the neural network to better learn the fruit feature.

a) original image | b) random rotation | c) saturation transformation | d) random affine transformation | e) random stitching

Fig. 4. Image enhancement display

3.2 Feature Extraction

The feature extraction network mainly includes CBS, CSP and SPPF modules, and its architecture is shown in Figure 5. The figure shows the extraction process of feature maps in each layer and the specific structure of each module.

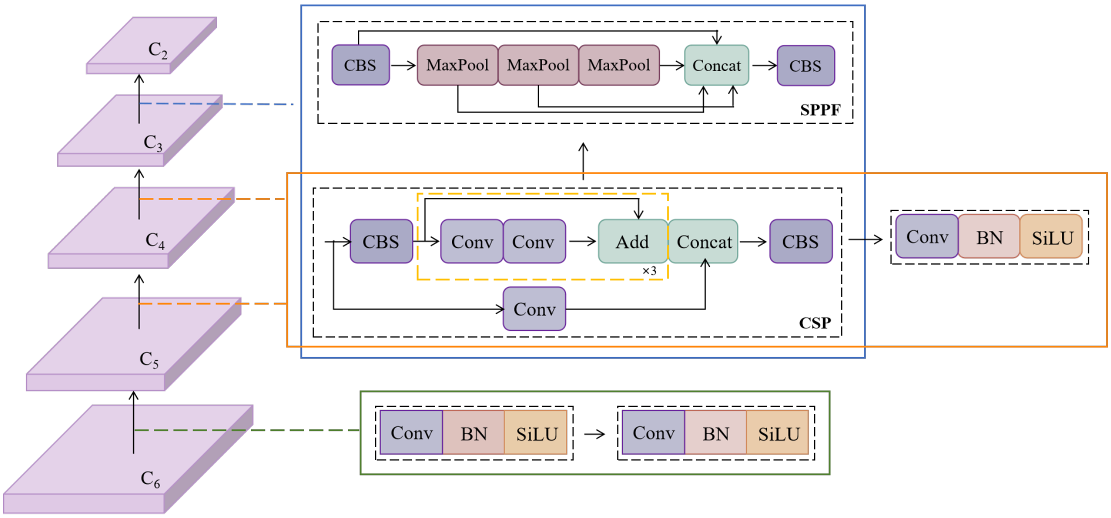


Fig. 5. Diagram of the feature extraction network

The convolution layer, batch normalization layer and activation function leaky ReLU are encapsulated in the CBL of YOLOV5. The ability of the activation function is nonlinear in the neural network, which is replaced by SiLU (Elfwing et al., 2018) in this study, and the module is named CBS. SiLU is defined as the activation of network function approximation in reinforcement learning. It is a weighted linear combination of sigmoid, whose function expression is $\mathrm{SiLU}(x) = \frac{x}{1-e^{x}}$. Leaky ReLU solves the problem of zero ReLU output, but it is still nearly linear. As shown in Figure 6, unlike Leaky ReLU, SiLU is not monotonically increasing but has a minimum value. This makes it self-stable, thus inhibiting the learning of a large number of weights. It can nonlinear neural networks better than Leaky ReLU, thus improving the expression ability of networks to models and solving problems that linear models are not equipped to deal with.

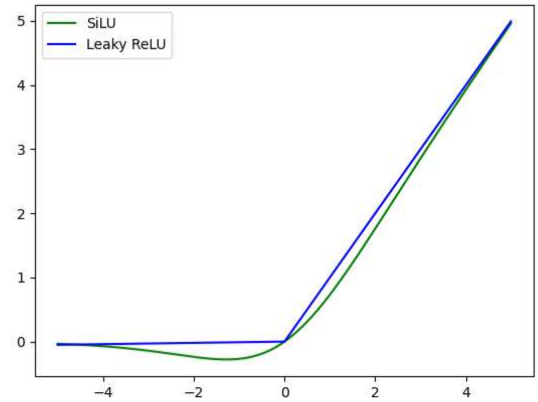


Fig. 6. Contrast graph of SiLU and Leaky ReLU curves

The CSP divides the input into two branches. The number of channels is halved by the convolution operation, and one of the branches is subject to a multi-layer residual operation (i.e., double-layer convolution residual component). Then, two branches are concatenated to make the input and output the same size. Finally, a CBS module is placed to further process feature information, which enables the feature extraction network to learn more fruit features. The SPPF pools the features passing through the CBS three times and concatenates the four groups of features. SPPF specifies one convolution kernel, and the output of each pooling layer is used as the input of the next pooling, which is faster than specifying three. Similar to the CSP, the last step of SPPF is still the CBS module. The SPPF increases the feature representation ability of the feature maps.

As shown in Figure 5, feature map C_5 is processed by C_6 through two stacked CBS modules, whose main step is convolution operations. Both operations of C_4 and C_3 are the same, passing through a CSP and a CBS module. The top-level feature map C_2 is obtained by C_3 through the CSP and SPPF modules in series, which enhances the expression of the algorithm for small objects. The feature extraction network generates six feature layers, denoted from bottom to top by C_6-C_2. However, only the upper five layers are used for feature fusion. The remaining layer is used to deepen the network and obtain richer feature information.

3.3 Feature Fusion

Feature fusion includes horizontal and vertical fusion. Horizontal fusion adds three different enhancement modules, which can also be used in the feature fusion phase of any other algorithm. Top-down fusion combines the CSP and CBS modules in the feature extraction network and uses the sampling and concatenation operations to fuse the adjacent feature maps. The following describes the overall architecture of the three modules and the feature fusion network.

3.3.1 Semantic Enhancement Module (SEM)

Recently, there have been two main structures for image feature processing: Convolutional Neural Network (CNN) and Transformer, which have different core concepts. CNN focuses on the correlation between two-dimensional local data. With the deepening of layers, its focus area will be wider. This makes it suitable for image processing, especially layer-by-layer processing of images (Lin et al., 2017; Liu et al., 2018). However, it cannot capture long-distance information and is limited by the receptive field. A common solution to this problem is to increase the depth of the neural network. This approach can indeed obtain more global information, but it will lead to gradient instability, network degradation and other problems.

At this time, transformers are widely used in the field of computer vision by virtue of their excellent spatial modeling ability (Zhu et al., 2020; Liu et al., 2021; Liu et al., 2022). The multi-head attention mechanism in the visual transformer captures richer information and relationships of features. However, the limitation of Transformer is that it cannot take advantage of the prior knowledge of scale, translation invariance and feature locality of the image itself, which makes it necessary to use a large amount of data for training. In addition, the main reason why the transformer structure cannot replace CNN at present is computational efficiency due to its sequential input format. In natural language processing, the sequence length of the WMT 2014 English-German dataset containing 50 million words and 2 million sentences is only 25. The code length is increased to 3136 when the image resolution of the common ImageNet dataset is 224, and the segmented image block size is defined as 4×4.

In this study, to detect small-scale fruits, the pixels in the dataset will be higher, and the corresponding coding length will be multiplied, which is difficult for computational memory. In consideration of the above factors, this study only constructs a semantic enhancement module with the help of a transformer structure. It does not involve the hierarchical association of feature maps but is applied to a feature map itself to enhance its semantic information. The specific structure of the SEM is shown in Figure 7.

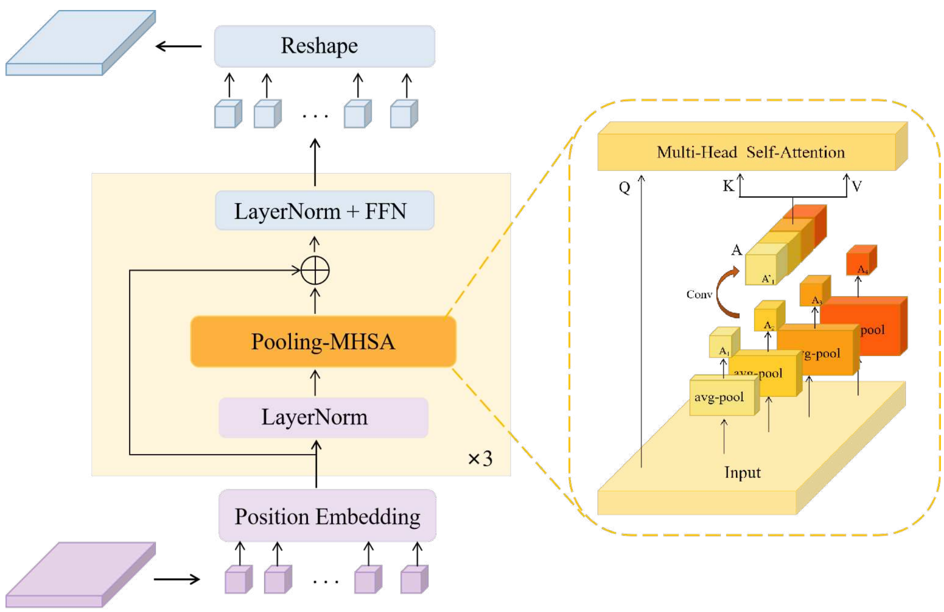


Fig. 7. Diagram of the semantic enhancement module structure

The feature map obtained through the feature extraction network is divided into many patches, generating sequences for position embedding. Then, three structural layers consisting of the norm layer, multihead attention and feed-forward network (FFN) are used to enrich the semantic information. Finally, the sequence is restored to a feature map of the same size through the Reshape operation. In the process above, the traditional multihead attention is replaced by pooling-MHSA, whose structure is shown in the right dotted line box in Figure 7.

The input sequence X is reshaped into the feature map format when it enters the pooling-MHSA. The reshaped feature map is still represented as X for the convenience of understanding the input. Multiple average pooling layers of different sizes are applied to X to generate a feature map $\mathrm{A_i}$ of different contents:

$$\mathrm{A_i} = \mathrm{AvgPool_j(X)} \tag{1}$$

Where i=1, 2, 3, 4. J stands for pool size, $\mathrm{j} = (\frac{\mathrm{H}}{\mathrm{ratios}} \times \frac{\mathrm{W}}{\mathrm{ratios}})$. H, W represents the size of the input feature map, ratios=[1, 2, 5, 10]. Next, the feature map is sent to the convolution layer for relative position coding:

$$\mathrm{A_i'} = \mathrm{Conv(A_i)} + \mathrm{A_i} \tag{2}$$

The encoded feature maps are stacked:

$$\mathrm{A} = \mathrm{LayerNorm(Concat(A_1',\ A_2', A_3', A_4'))} \tag{3}$$

The stacked feature maps A carry more context information in feature map X, which can replace X as the input of the subsequent multihead self-attention. The size of pooled feature maps is smaller, so the generation of K and V matrices is smaller than that of traditional ones, which means that the Pooling-MHSA is more efficient. It can be expressed as Equation 4 and Equation 5:

$$\mathrm{(Q, K, V)} = \mathrm{(XW^q, AW^k, AW^v)} \tag{4}$$

$$\mathrm{X_{Patt}} = \mathrm{Softmax}(\frac{\mathrm{Q \times K^T}}{\sqrt{\mathrm{d\ K}}}) \times \mathrm{V} \tag{5}$$

The feed-forward network is an important part of the SEM. The traditional transformer structure uses the fully connected layer as the feed-forward network. To integrate the nearest neighbor relationships between features, convolution structures are combined to process sequences.

First, the sequence after the cross-layer residual structure is reconstructed into a feature map by the ToImage function:

$$X_{Patt}^{I} = \mathrm{ToImage}(\mathrm{NormLayer}(X_{Patt} + X)) \tag{6}$$

Then, through the two-level convolution matrix in Equation 7 and Equation 8:

$$X' = \mathrm{Hardswish}(X_{Patt}^{I}\, W^{1}) \tag{7}$$

$$X^{2} = \mathrm{Hardswish}(\mathrm{Conv}(X'))W^{2} \tag{8}$$

Where W^{1},W^{2} represents the size of the 1×1 weight matrix, and Hardswish is the activation function. Finally, the feature map through the feed-forward network is converted into a sequence format by the function ToSeq and is given to the structure layer in series or the reshaping layer for subsequent operations:

$$X^{out} = \mathrm{ToSeq}(X^{2}) \tag{9}$$

The semantic enhancement module refers to the Transformer structure and constructs the Pooling-MHSA and a new feed-forward neural network to enhance semantic information. The structure composed of NormLayer, Pooling-MHSA, and FFN is stacked three layers deep. The SEM is applied to the top-level feature map in the ODL Net. This is because the top-level feature map often contains more abstract and semantic features, and applying semantic enhancement operations to it can further extract higher-level semantic features. Moreover, applying SEM to the top-level feature map can expand the receptive field, i.e., increase the observation range of each feature point on the input image. Additionally, the self-attention mechanism in SEM helps the algorithm better understand the contextual information and global structure of the targets. In summary, the application of SEM to the top-level feature map can enhance the receptive field, feature extraction capability, and object localization accuracy of ODL Net, thereby improving the performance and effectiveness of object detection. The experiments also demonstrate that when it is applied to the top-level feature map, ODL Net achieves the highest detection accuracy, as shown in section 3.2.2 of the experimental results. Furthermore, the module can be independently applied and inserted at suitable positions, including downstream tasks, within the neural network.

3.3.2 Feature Fusion Network

The feature fusion network constructs three different enhancement modules to enhance the information of the feature map before fusion in different aspects. In addition, CBS and CSP modules are used for feature integration when adjacent feature maps are fused. Figure 8 shows the structure diagram of the feature fusion network, in which $H \times W$ represents the size of the feature map and C represents the number of channels.

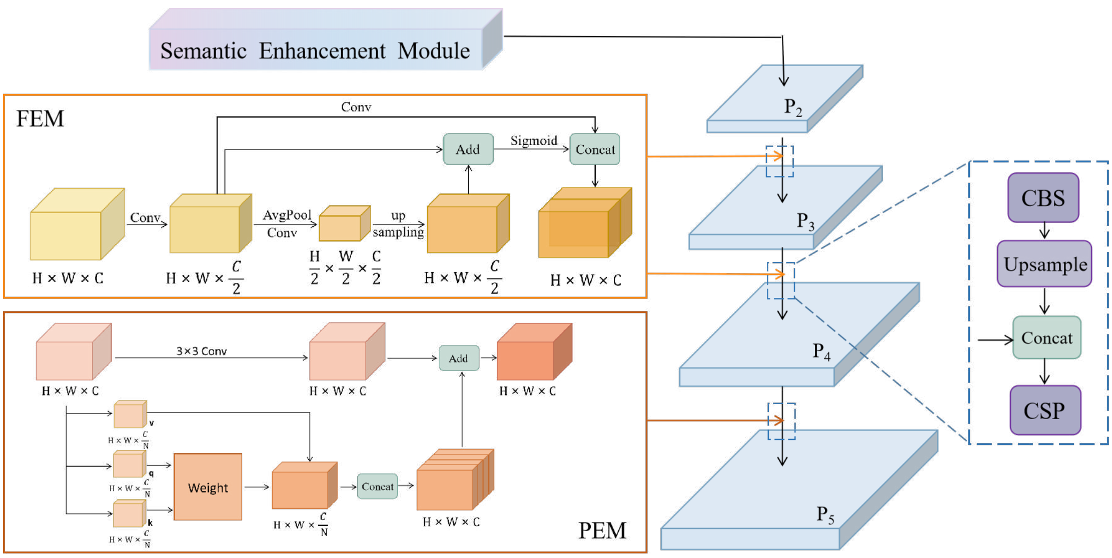


Fig. 8. Diagram of the feature fusion network

As shown in Figure 8 above, the feature enhancement module consists mainly of two nested residual structures. The input feature map is first reduced by a 1×1 convolution to reduce the number of channels to half of the original number and then further processed by average pooling and convolution operations with sizes of 2×2 and 3×3, respectively. This step also makes the feature map size half of the input, as well as the number of channels. Then, it restores the size through the upsampling operation and adds it to the feature map before pooling. It is then restored to size by an upsampling operation and added to the feature map before pooling. The last step is to restore the number of channels by stacking the feature maps after addition and before pooling. In the process of halving the resolution of the feature map, more object features will be amplified and extracted by the network. When the image size is restored, the image information will be updated. The cross-layer addition and stacking operation in the FEM effectively avoids the loss of information during image size changes and achieves the function of feature enhancement as a whole.

There are also two branches in the Position Enhancement Module (PEM). As shown in Figure 8, the upper branch is set with a convolution layer to further extract features on the basis of keeping the size and channel number unchanged. The convolution has a size of 3×3, with a stride and padding of 1. The other branch sets the self-attention mechanism, where N represents the number of attention heads. The PEM mainly enriches the location information through a self-attention mechanism. It provides an effective modeling method through the triplet of Key, Query and Value and obtains greater receptive field and context information by capturing global information. Finally, the captured information will be added with the convolution features to achieve position enhancement.

The following describes the position of modules in the feature fusion network. Obviously, the lower-level feature map brings higher resolution, which means it carries more location information. Therefore, the Position Enhancement Module is added in the fusion process of the lowest feature map to supplement the context information. The feature maps in the middle of the two layers are responsible for extracting features. The fusion of these two layers focuses on whether the features are effectively extracted. Therefore, the Feature Enhancement Module is added to the fusion process of P_2-P_4. Chen (Chen et al., 2021) proved through experiments that the top-level feature map carries the most abundant semantic information among the feature maps generated by the feature extraction network. In addition, the memory requirements of the semantic enhancement module also limit its application scope, so it is added in the process of the top-level feature map C2-P2. To maximize the function of the SEM, this study discusses its reasonable position in the feature fusion network. The experimental data are shown in section 4.3, and the results show that it is best to add SEM to the top layer. The effects of the FEM and PEM are also shown in the same section. The feature fusion network enables ODL Net to fully capture the object features. The enhancement of location and semantic information greatly improved the sensitivity of the algorithm to the feature, which is conducive to the detection of fruit before and after the pear thinning stage.

3.4 Detection Head

In this study, a detection head for small objects is constructed, which mainly relies on a special label assignment to improve the detection accuracy. This assignment eliminates attention to the size and shape of the object so that small-scale fruit will not be ignored. The detection head is mainly composed of the label assignment Center-Box and convolution layers, which are shown in Figure 3. The following is a description of the label assignment, aiming at small-scale objects.

3.4.1 Label Assignment

Yolov5, as the baseline of this study, is an anchor-based algorithm whose sample selection method increases the number of positive samples to a certain extent. In the feature map, the two adjacent grids closest to the center point of the ground truth are selected as the prediction grids. In addition to the grid where the ground truth is located, there are at most nine anchor boxes corresponding to three grids that match it. In the matching process, the aspect ratio between the ground truth and anchor is calculated twice. If the aspect ratio is less than the specified threshold, the anchor is judged as a positive sample; otherwise, it is the background. For example, if the ground truth is matched with the 1:1 and 1:2 size anchors corresponding to the current layer and its own grid, then there are also two sizes of anchors in the nearest two grids. The number of positive samples of this ground truth in the current layer is 6, while the range of possible anchors is [0, 9], and the number of matching three feature maps is [0, 27]. The process above is shown in Figure 9. Although this label assignment is relatively advanced, which increases the number of positive samples to a certain extent, it cannot be used for small object detection.

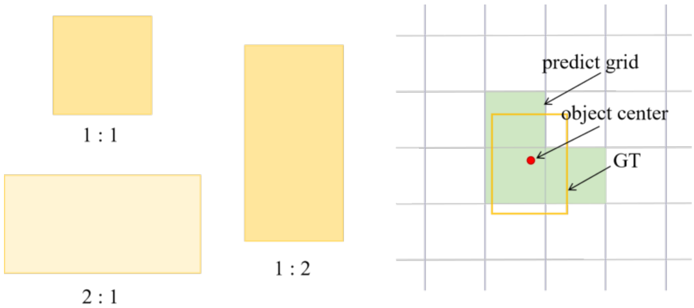


Fig. 9. The label assignment diagram of YOLOV5

Note: The left side is the anchor box of three sizes corresponding to each grid, and the right side is the selection diagram of the prediction grid.

To improve the detection accuracy of small objects, ODL Net uses a label assignment named Center-Box without anchors, which is specifically described in section 3.4.2. The comparison with other label assignments is shown in Figure 10. The dashed box represents the ground truth, and the orange part represents the positive sample. The two columns on the right in Figure 10 show the representative label assignments of the two types of detection algorithms.

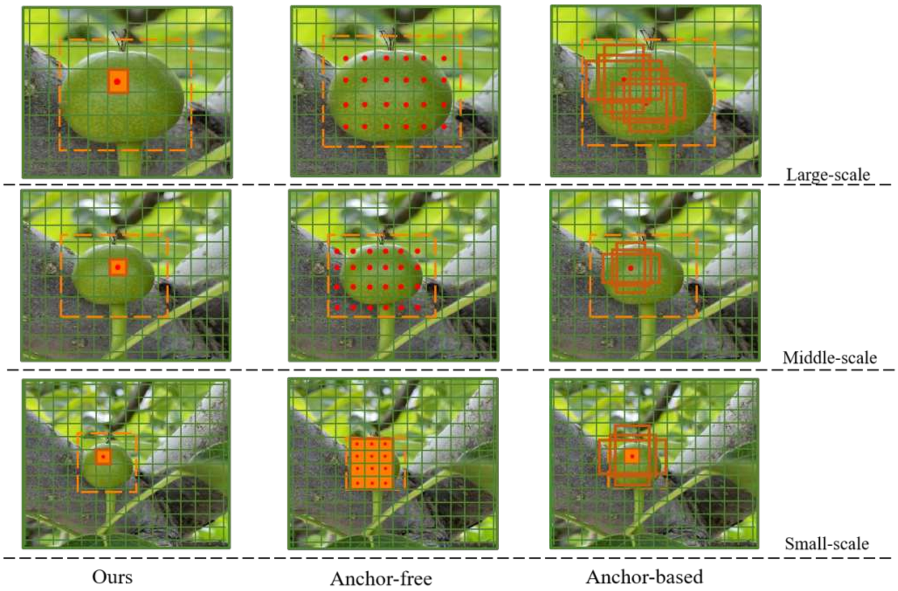


Fig. 10. Comparison of different types of label assignments

The anchor-free algorithms take FCOS as the typical representative and tile the anchor points in the feature map to select positive samples. All anchor points in the ground truth after a feature map is mapped to the original map are selected to calculate the distance from the point to the ground truth: (l^*, r^*, t^*, b^*). FCOS defines the range of max(l^*, r^*, t^*, b^*) on the multi-scale feature maps to determine the scale on which the object is detected. For example, FCOS stipulates $\max(l^*, r^*, t^*, b^*) \in [128, 256]$ in the top feature map, which means that this feature map is used to detect large objects. The fruit in Figure 10 does not meet this range, so there is no positive sample on the large-scale feature map in this case. Correspondingly, the fruit conforms to the detection range in the small-scale feature map, so the grid of all anchor points in the ground truth is determined as a positive sample.

In the classic anchor-based algorithm, the division of positive and negative samples is completed by calculating the IoU between the ground truth and the anchor boxes. When IoU is greater than the specified threshold, this group of anchors will be determined as positive samples. However, it can be seen from the rightmost column in Figure 10 that the IoU of small objects in the large-scale feature map is zero. This is because the anchor is too different from the ground truth or even completely inside it. Therefore, the small-scale fruit can only produce positive samples in the low-level feature map.

These traditional label assignments all define constraints on positive samples, which basically limits the scale range of objects that can be detected at each feature level. The assignments of other algorithms (Kong et al., 2020; Zhu et al., 2019) can also be roughly classified into these two categories. Although these two methods achieve multi-scale detection, larger objects will be allocated more positive samples, and small objects will be easily ignored. This is not conducive to the detection of small objects and makes it difficult to detect fruit around the pear thinning period.

3.4.2 Center-Box

To solve the above problems, ODL Net uses a "fair" label assignment (Zand et al., 2022), Center-Box. As shown on the left of Figure 10, Center-Box cancels the allocation rule of positive and negative samples and directly defines the grid where the object center is located as positive samples (marked in orange) on all levels of feature maps. This strategy prevents the size and shape of objects from dictating the assignment of labels and treats all objects equally at different levels of feature. This means that Center-Box allows the network to learn at all scales of an object, which makes the number of positive samples allocated to small-scale objects and large-scale objects the same. Therefore, the detection will not tend to large-scale objects. To match the positive sample grids, the regression target of Center-Box is defined as the distance from the diagonal vertex of the grid to the ground truth, which is shown in Figure 11 of (L, T, B, R). The coordinates of the upper left corner and the lower right corner of the ground truth are represented as (x_1, y_1) and (x_2, y_2), respectively. The coordinates of the center point are represented as (x, y).

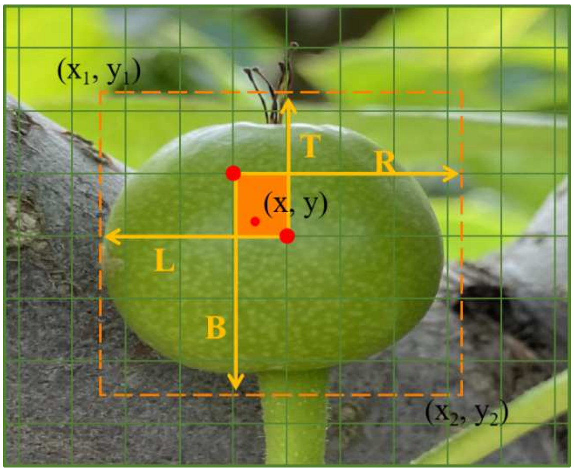


Fig. 11. Diagram of the Center-Box regression

The regression target of Center-Box is the distance between the upper left corner of the grid where the center point is and the right and upper boundaries of the ground truth and the distance between the lower right corner of the grid and the left and lower boundaries of the ground truth. They are represented by (L^*, T^*, B^*, R^*), which is shown in Equation 10:

$$\begin{cases} L^{(i)^*} = (x/s_i + 1) - x_1^{(i)}/s_i \\ T^{(i)^*} = (y/s_i + 1) - y_1^{(i)}/s_i \\ R^{(i)^*} = x_2^{(i)}/s_i - x/s_i \\ B^{(i)^*} = y_2^{(i)}/s_i - y/s_i \end{cases} \tag{10}$$

Where i represents the feature scale of [1, 2, 4]. $(\frac{x}{s_i}, \frac{y}{s_i})$ and $(\frac{x}{s_i} + 1, \frac{y}{s_i} + 1)$ in Equation 10 represent the coordinates in the upper left and lower right corners of the grid, respectively. Further explanation is that $L^{(i)^*} + R^{(i)^*} = (x_2^{(i)} - x_1^{(i)}) + 1$, $T^{(i)^*} + B^{(i)^*} = (y_2^{(i)} - y_1^{(i)}) + 1$, where $x_2^{(i)} - x_1^{(i)} = w^i = \frac{w}{s_i}$, and $y_2^{(i)} - y_1^{(i)} = h^i = \frac{h}{s_i}$. w and h represent the size of the ground truth in the original image, and w^i and h^i represent the width and height of the ground truth on scale i, respectively.

The learning process of the regression target is shown in Equation 11:

$$\begin{cases} L^{(i)} = (\alpha \times \mathrm{Sigmoid}(l))^2 * 2^i \\ T^{(i)} = (\alpha \times \mathrm{Sigmoid}(t))^2 * 2^i \\ R^{(i)} = (\alpha \times \mathrm{Sigmoid}(r))^2 * 2^i \\ B^{(i)} = (\alpha \times \mathrm{Sigmoid}(b))^2 * 2^i \end{cases} \tag{11}$$

Where (l, t, r, b) represent the predicted values in network for the distance in four directions, and their values are controlled between 0 and 1 by the sigmoid function. $i \in \{1, 2, 4\}$, represents the scale of different feature maps, and 2^i is used to distinguish different scales in the learning process.$\alpha$is a range constant used to expand the detection coverage. It is set as 1.0 in the experiment because of the small size of most objects around the pear thinning period, and it can be adjusted according to the size of the object in other studies. $(L^{(i)}, T^{(i)}, R^{(i)}, B^{(i)})$ is the predicted result on the i-th layer of feature map. This predicted distance is used to compare with the real distance and adjust the network parameters according to loss function for learning.

The Center-Box approach does not specifically aim to detect fruits of corresponding sizes on feature maps of different scales, but rather ensures that fruits can be learned on feature maps of all scales where they exist. It directly assigns the grid cell containing the center of the fruit as a positive sample and regresses the distances from the top-left to the bottom-right corners of the grid cell to the true box. This strategy allows for an equal number of positive samples to be assigned to both large-scale and small-scale fruits, enabling the ODL Net to treat the detection of fruits at different scales equally. Consequently, this implicitly enhances the detection capability of the ODL Net for small-scale fruits. In the real working environment of the ODL Net, as supported by the statistical information provided in Section 2, small-scale fruits constitute nearly half of the overall quantity. Hence, the scale-agnostic nature of the Center-Box approach empowers the ODL Net to deliver satisfactory performance in detection tasks before and after thinning in pear orchards.

3.5 Loss Function

The loss function of ODL Net consists of three parts: classification loss, confidence loss and bounding box loss. The network loss is the weighted sum of the above three, which is shown in Equation 12. The impact of each loss can be adjusted by weight λ.

$$\mathrm{Loss} = \lambda_1 L_{cls} + \lambda_2 L_{conf} + \lambda_3 L_{obj} \tag{12}$$

3.5.1 Classification and Confidence Loss

For detection tasks in the pear orchard, only "pear" is the category of prediction tag output from the network. At this point, the common binary cross entropy loss BCE with logit loss is used as the classification loss:

$$y_i = \mathrm{Sigmoid}(x_i) = \frac{1}{1+e^{-x_i}} \tag{13}$$

$$L_{cls} = -\sum_{n=1}^{N} y_i^* \log(y_i) + (1-y_i^*)\log(1-y_i) \tag{14}$$

Where x_i represents the predicted value of the current category. y_i represents the probability of the current category obtained after activating the function. y_i^* is the true value of the class, expressed as 0 or 1.

The confidence level of the prediction box indicates its reliability. The higher the value, the more reliable the prediction box is, and the closer it is to the ground truth. The confidence loss is the same type as the classification loss, using the binary cross entropy loss. It should be noted that the total confidence loss is obtained by weighted addition of the confidence losses on the three prediction branches:

$$L_{conf} = \beta_1 L_{conf}^{L} + \beta_2 L_{conf}^{M} + \beta_3 L_{conf}^{S} \tag{15}$$

Where $\beta_1, \beta_2, \beta_3$ represents the influence of the confidence loss of the feature map with the resolution from high to low. The weight is set to (5.0, 1.0, 0.5) in the experiment to improve the detection accuracy of small fruits. Because small-scale objects are detected on the high-resolution feature map, β_1 is adjusted higher to facilitate small fruit detection.

3.5.2 Bounding Box Loss

The goal in the process of boundary box regression is to minimize the distance between the prediction box and the ground truth. For the relative position of the bounding boxes, the classic method is to calculate the IoU value of the two boxes. IoU is usually used to express the coincident area of two object positions. On this basis, many more advanced methods have been proposed (Rezatofighi et al., 2019; Zheng et al., 2020). For example, DIoU takes into account the distance between the ground truth and the prediction box, the overlap rate and the scale:

$$\mathrm{DIoU} = \mathrm{IoU} - \frac{\rho^2(b, b^{gt})}{c^2} \tag{16}$$

Where b, b^{gt} represents the center of the prediction box and the ground truth, respectively. $\rho(b, b^{gt})$ represents the Euclidean distance between two central points. The symbol c represents the diagonal distance of the smallest area that can contain both boxes.

In this study, to cooperate with the Center-Box, a loss method with scale invariance is proposed. The goal in the regression process is to minimize the distance between the prediction box and the ground truth. As explained in Section 3.4.2, each box is represented by four distances. Therefore, it is the hope of this study that the distance in four directions can be taken into account in the loss of the bounding box, which is shown in Figure 12.

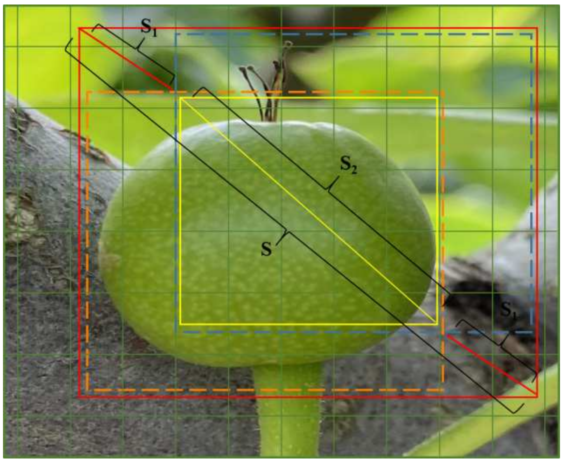


Fig. 12. Calculation diagram of bounding box loss

In the loss of bounding boxes, the overlapping area (yellow box), non-overlapping area and minimum inclusion area (red box) are all considered. They are expressed in square Euclidean Distance as Equation 17:

$$\begin{cases} S_1 = (L^* - L)^2 + (T^* - T)^2 + (R^* - R)^2 + (B^* - B)^2 \\ S_2 = (\min(L^*, L) + \min(R^*, R) - 1)^2 + (\min(T^*, T) + \min(B^*, B) - 1)^2 \\ S = (\max(L^*, L) + \max(R^*, R) - 1)^2 + (\max(T^*, T) + \max(B^*, B) - 1)^2 \end{cases} \quad (17)$$

Where (L^*, T^*, B^*, R^*) and (L, T, B, R) represent the predicted and true values, respectively. The expression of bounding box loss is as Equation 18:

$$L_{obj}(L^*, T^*, R^*, B^*) = 1 - \frac{(S_2 - S_1)}{S} \quad (18)$$

4. Experiments

The experiment is conducted on a server equipped with the Ubuntu 16.04 operating system, which is equipped with four GTX 3090 graphics cards and V11.4 CUDA. During the training, two graphics cards are used, and 16 images are set for each batch. The initial learning rate of the experiment is set to 0.005, and 0.0001 is used as the weight attenuation to prevent over-fitting. To update and calculate the network parameters and minimize the loss function, a random gradient descent (SGD) optimizer with a momentum of 0.9 is used to assist the training. Image enhancement methods are used to enrich the dataset before training to reduce over-fitting. Finally, 300 epochs are trained for the ODL Net.

4.1 Evaluation Index

The average precision (AP) is selected as the evaluation index of algorithm performance in the experiment. It is the area under the PR curve with recall as the horizontal axis and precision as the vertical axis. The calculate method is shown in Equation 19. Other evaluation indicators used in the experiment also belong to the same type: AP_{50} is the measured value of AP when the IOU threshold is 0.5; AP_{75} is the AP measurement value when IOU is 0.75; AP_s, AP_m and AP_l represent AP measurement values of small, medium and large objects, respectively. Fruit with the number of pixels less than 174×174 are defined as small-scale objects, fruit with the number of pixels greater than 523×523 are defined as large-scale objects, and fruit with the number of pixels between them are defined as medium-scale objects. The formulation of the scale range is explained in section 2.2.

$$\mathrm{AP} = \int_0^1 \mathrm{P(R)dR} \tag{19}$$

In this definition of AP, P is represents the proportion of the number of predicted positive samples to the number of real positive samples; R represents the proportion of positive samples correctly predicted by the algorithm in the real positive samples. The calculation equations are shown in Equation 20 and Equation 21, where TP represents the number of detection frames whose intersection to parallel ratio is greater than the set threshold; FP represents the number of detection frames whose intersection ratio is less than the set threshold, or the number of redundant detection frames generated under the same target; FN indicates the number of targets not detected.

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \times 100\% \tag{20}$$

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \times 100\% \tag{21}$$

In addition, the average recall (AR) is also a supplementary evaluation index, although AP is more authoritative. AR refers to the maximum recall in a given number of detection results on each image.

4.2 Comparative Experiments

In this section, a comparison is made between ODL Net and classical CNN-based detection algorithms since 2020. The detection accuracy on the dataset prior to thinning the pears is presented in Table 2. Two key results are emphasized in the experiment: the overall detection accuracy of the algorithm (referred to as AP) and the detection accuracy specifically for small-scale fruits (referred to as APs). Table 2 reveals that ODL Net achieves the highest detection accuracy, reaching 56.2%, surpassing other algorithms by a margin of at least 0.5 percentage points. Among the detection algorithms developed in the past two years, AutoAssign (Zhu et al., 2020) demonstrates the closest accuracy to ODL Net for small fruits, with a mere 0.3 percentage point difference. However, its overall detection accuracy is unsatisfactory. Similarly, the AP of TOOD (Feng et al., 2021) reaches 55.7%, but its performance on small fruits falls significantly behind our algorithm.

From the aforementioned results, it is evident that one of the key advantages of ODL Net lies in its capability to enhance the detection accuracy of small-scale fruits without compromising its overall AP. Building upon the YOLOV5 baseline, ODL Net exhibits an increase of 1.4 percentage points in AP and a 2.1 percentage point increase in APs. In contrast, the other algorithms in Table 2, such as NAS-FCOS (Wang et al., 2020), an enhanced version of FCOS, yield considerably lower accuracy compared to ODL Net, differing by more than 3.0 percentage points. Additionally, ODL Net achieves the highest AR of 61.6% and ARs of 40.0%.

Table 2. Comparative experiments on the pear dataset before thinning

	AP%	AP_{50}%	AP_{75}%	AP_s%	AP_m%	AP_l%	AR%	AR_s%	AR_m%	AR_l%

ATSS (Zhang et al., 2020)	50.3	79.0	50.8	22.6	72.3	85.4	56.3	33.4	77.2	89.1
AutoAssign (Zhu et al., 2020)	52.8	84.2	53.3	29.0	75.6	87.6	60.8	39.3	80.5	91.3
Double-Head RCNN (Wu et al., 2020)	51.8	83.8	54.1	26.6	71.9	79.9	56.7	36.5	75.6	83.3
NAS-FCOS (Wang et al., 2020)	53.1	82.9	54.4	28.3	73.2	86.8	60.0	38.8	79.3	91.0
TOOD (Feng et al., 2021)	55.7	84.7	56.4	28.7	76.8	90.2	61.6	40.0	81.3	93.3
YOLOV5	54.8	81.6	56.3	27.2	75.8	88.9	61.2	39.8	80.5	92.4
ODL Net	56.2	83.0	57.5	29.3	77.0	91.2	61.3	39.0	81.8	93.8

The experimental results on the pear dataset after fruit thinning are presented in Table 3. The dataset exhibits a significantly lower fruit density compared to the pre-thinning dataset, with minimal instances of fruit overlap. In this scenario, ODL Net demonstrates a notable improvement over YOLOV5, with an increase of 2.4 percentage points in both AP and APs. Similarly, for ATSS, which exhibits relatively good performance, there is a 2.0 percentage point increase in both AP and APs. But in terms of detecting small-scale fruits after pear thinning, Double-Head RCNN and NAS-FCOS achieve comparable or slightly higher detection accuracy compared to ODL Net. However, these algorithms tend to focus excessively on small objects and lack sensitivity towards objects of other scales, resulting in an AP that is 3.6-5.0 percentage points lower than ODL Net. They prioritize small object detection at the expense of AP.

The experiments demonstrate that ODL Net enhances the detection accuracy of small objects while also considering the overall detection accuracy (AP) of the algorithm. This is because SEM enlarges the receptive field of the network, enabling the network to perceive fruits at all scales. In addition, effective feature fusion also enables ODL Net to capture more and richer features for accuracy detection.

Table 3. Comparative experiments on the pear dataset after thinning

	AP%	AP$_{50}$%	AP$_{75}$%	AP$_s$%	AP$_m$%	AP$_l$%	AR%	AR$_s$%	AR$_m$%	AR$_l$%
ATSS (Zhang et al., 2020)	63.1	79.5	70.0	36.9	81.7	90.1	69.3	51.6	85.1	92.2
AutoAssign (Zhu et al., 2020)	60.0	76.9	68.0	37.8	77.7	86.3	70.6	55.1	84.6	90.0
Double-Head RCNN (Wu et al., 2020)	61.5	80.5	72.0	40.4	75.7	83.5	66.4	52.9	77.6	85.4
NAS-FCOS (Wang et al., 2020)	60.1	78.1	68.8	38.9	75.5	86.0	67.3	51.6	80.9	88.6
TOOD (Feng et al., 2021)	62.3	79.4	67.6	37.7	61.7	88.3	68.2	48.4	69.3	91.9
YOLOV5	62.7	77.4	68.5	36.5	80.6	88.7	69.3	51.2	86.2	91.0
ODL Net	65.1	78.6	70.4	38.9	81.7	91.2	70.8	53.9	86.1	92.5

4.3 Ablation Experiments

Considering the experimental nature of the dataset, the ablation experiments on the pear dataset after thinning can fully and clearly show the role of each module. The experimental data are shown in Table 4. The addition of the Center-Box focuses on improving the detection accuracy of small-scale objects. At this time, the accuracy of medium-scale and large-scale objects is almost unchanged. It is also the semantic enhancement module for small objects, which further improves the detection accuracy of small fruits. They can be used separately in the detection of other small objects. In addition, the feature enhancement module and the position enhancement module are very helpful in improving the overall detection accuracy. Compared with the SEM, although they are not aimed at small objects, they improve the overall detection accuracy of ODL Net. Finally, the AP and AP_s of ODL Net are 2.4 percentage points higher than those of the baseline algorithm. The detection accuracy of the algorithm for small-scale objects in the dataset is up to 39.3%, although the overall accuracy is not the highest at this time.

Table 4. Ablation experiments on the pear dataset after the thinning period

Structure				AP%	AP_s%	AP_m%	AP_l%
Center-Box	SEM	PEM	FEM				
×	×	×	×	62.7	36.5	80.6	88.7
√	×	×	×	63.8	38.9	80.6	89.3
√	√	×	×	64.1	39.3	80.1	89.7
√	×	√	×	64.8	38.2	81.2	88.1
√	√	√	×	64.9	38.5	82.8	91.0
√	√	√	√	65.1	38.9	81.7	91.2

What needs to be specially explained is the location of the semantic enhancement module in ODL Net. In fact, it is obvious that the top-level feature map brings the richest semantic feature. However, we still confirm the location of SEM through experiments to eliminate the impact of the pear dataset. The experimental data are shown in Table 5, which shows the accuracy of ODL Net when SEM is added at different feature map layers. At this time, ODL Net is only constructed with

Object-Box and SEM, but no other modules. Table 5 shows that the accuracy of ODL Net reaches 64.1% when SEM is added to the top level of the feature fusion network. The detection of small objects reaches 39.3%. Therefore, in this study, SEM is finally added to the top-level feature map C_2 to enhance the detection accuracy of small-scale fruit.

Table 5. Ablation experiments of the layer with SEM

Layer	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
C_2	64.1	77.9	69.7	39.3	80.1	89.7
C_3	62.9	77.6	68.5	36.9	79.4	89.7
C_2, C_3, C_4	63.6	77.4	69.3	38.1	81.0	90.1

4.4 Sample Results

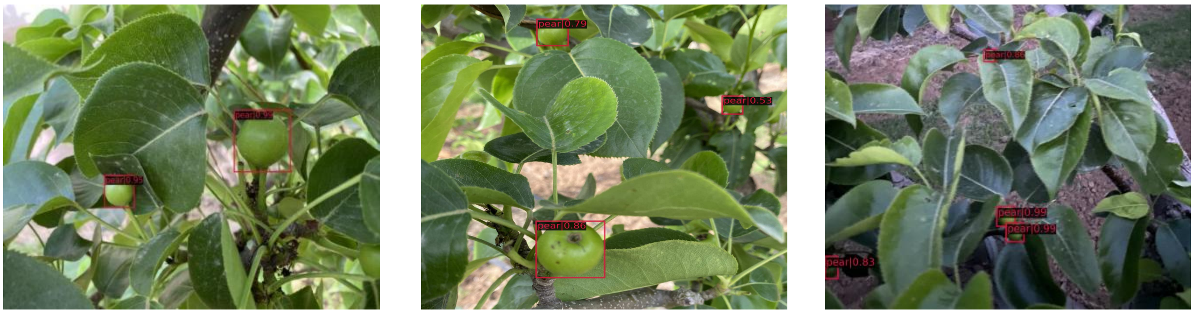
The detection effect of the algorithms on the pear dataset is shown in Figure 13 and Figure 14 before and after fruit thinning, respectively. A representative image is selected to show the effect before the thinning period. The area with objects (marked with orange rectangle in the original image) is enlarged in all of the result images to display the detection results more intuitively. In the image before fruit thinning, there are ten fruits to be detected in the selected image of Figure 13. Box redundancy occurs in the Yolov5, AutoAssign and Double-Head RCNN. That is, there are multiple prediction boxes on a fruit, or nonexistent objects are detected. ATSS and NAS-FCOS miss approximately three numbers of the target fruit. While, TOOD successfully detected all fruits. ODL Net recognizes nine fruits with the highest scores.

original image
ATSS
AutoAssign
Double-Head
RCNN
NAS-FCOS
TOOD

Fig. 13. Comparison images of algorithms on the pear dataset before thinning

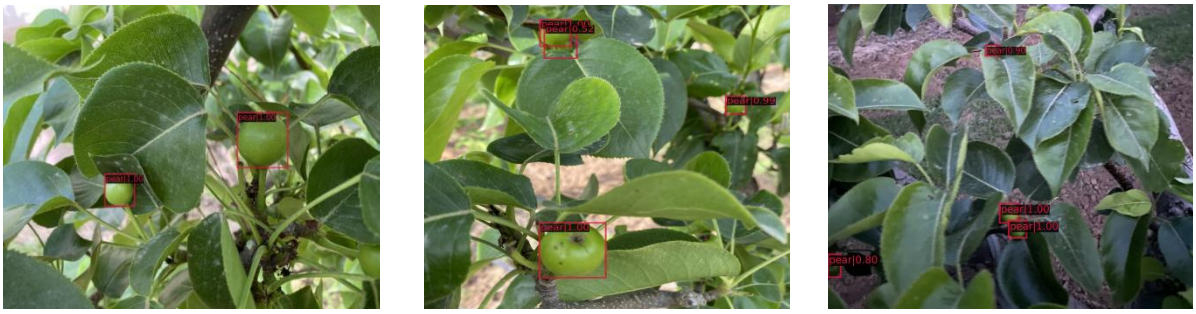
The fruit density after thinning is relatively sparse, and there is basically no problem of box redundancy when detected. The decrease in fruit density also makes the detection easier. However, for the incomplete and fuzzy fruit in the lower right corner in the first image of Figure 14, most of the algorithms fail to detect it. The other two detected images after thinning are also shown in Figure 14.

AutoAssign



Double-Head

RCNN



NAS-FCOS

TOOD

YOLOV5

ODL Net

Fig. 14. Comparison images of algorithms on the pear dataset after thinning

The other detected images before thinning are shown in Figure 15 as examples.

original image

ATSS

AutoAssign

Double-Head RCNN

NAS-FCOS

TOOD

YOLOV5

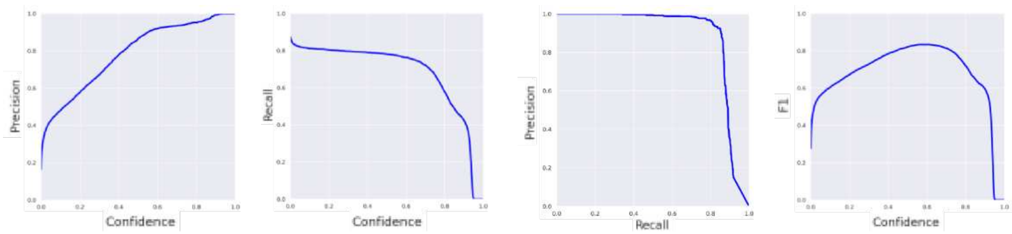
ODL Net

Fig. 15. Comparison images of algorithms on the pear dataset around the thinning period

The curves of ODL Net during training are shown in Figure 16.

P_curve R_curve PR_curve F1_curve

before thinning



after thinning

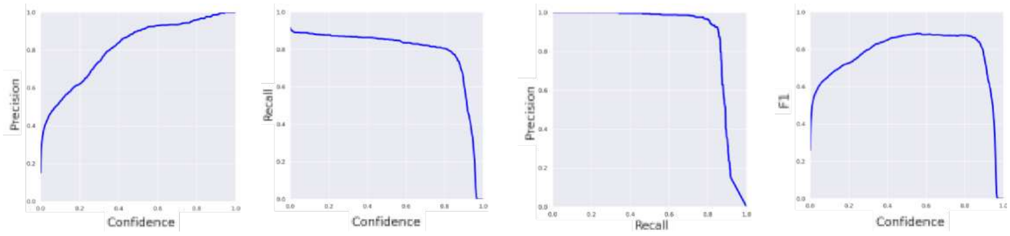


Fig. 16. Curves of ODL Net during training

5. Conclusion and Feature Work

This study proposes a detection algorithm called ODL Net, specifically designed for detecting small-scale fruits before and after thinning in pear orchards. It enhances the detection of small objects through the SEM and the label assignment strategy called Center-Box. Additionally, the modules of FEM and PEM are constructed to further improve the overall detection performance of ODL Net. These enhancement modules can also be used individually.

For fruit varieties that require thinning, the detection of fruits by ODL Net before the thinning stage can guide the thinning process. Moreover, the detection of fruits by ODL Net after thinning enables calculations for irrigation and fertilizer requirements, facilitates scientific yield measurement, and supports intelligent management of orchards. In the case of fruit varieties that do not require thinning, ODL Net provides continuous monitoring throughout the fruit growth period. Particularly in the early stages of fruit growth, where the fruit size is small and detection poses significant challenges, a high-performance detection algorithm is crucial. ODL Net, designed specifically for small-scale fruits, can partially address this issue. And It is precisely because of this characteristic that ODL Net offers significant assistance in intelligent orchard management and fills the research gap in various small-scale fruits detection during the thinning period in orchards.

Although ODL Net has completed the improvement for small-scale objects, the difficulties have not been completely overcome. The complete green appearance of pear during fruit thinning still caused some difficulties in the detection work. In future studies, we hope to propose a detection algorithm for both indistinguishable green color and small size.

Acknowledgments

This work is supported by Natural Science Foundation of Shandong Province in China (No.: ZR2020MF076); Young Innovation Team Program of Shandong Provincial University (No.: 2022KJ250); National Nature Science Foundation of China (No.: 62072289); New Twentieth Items of Universities in Jinan (2021GXRC049); Taishan Scholar Program of Shandong Province of China.

References

Bargoti S, Underwood J. Deep fruit detection in orchards. 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017: 3626-3633.

Chen Q, Wang Y, Yang T, et al. You only look one-level feature. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13039-13048.

Ebrahimi M, Khoshtaghaza M, Minaei S, et al. Vision-based pest detection based on SVM classification method. Computers and Electronics in Agriculture, 2017, 137: 52-58.

Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks, 2018, 107: 3-11.

Feng C, Zhong Y, Gao Y, et al. Tood: Task-aligned one-stage object detection. IEEE/CVF International Conference on Computer Vision (ICCV). IEEE Computer Society, 2021: 3490-3499.

Fu L, Gao F, Wu J, et al. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. Computers and Electronics in Agriculture, 2020, 177: 105687.

Gongal A, Amatya S, Karkee M, et al. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture, 2015, 116: 8-19.

Hussain D, Hussain I, Ismail M, et al. A simple and efficient deep learning-based framework for automatic fruit recognition. Computational Intelligence and Neuroscience, 2022, ID 6538117.

Jia W, Meng u, Ma X, et al.. Efficient detection model of green target fruit based on optimized Transformer network.Transactions of the Chinese Society of Agricultural Engineering, 2021, 37(14): 163-170.

Rabbi J, Ray N, Schubert M, et al. Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network[J]. Remote Sensing, 2020, 12(9): 1432.

Kong T, Sun F, Liu H, et al. Foveabox: Beyound anchor-based object detection. IEEE Transactions on Image Processing, 2020, 29: 7389-7398.

Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.

Liu Z, Hu H, Lin Y, et al. Swin transformer v2: Scaling up capacity and resolution. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 12009-12019.

Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 10012-10022.

Maheswari P, Raja P, Apolo-Apolo O E, et al. Intelligent fruit yield estimation for orchards using deep learning based semantic segmentation techniques—a review. Frontiers in Plant Science, 2021, 12: 684328.

Mai X, Zhang H, Meng M. Faster R-CNN with classifier fusion for small fruit detection. IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 7166-7172.

Ngugi LC, Abelwahab M, Abo-Zahhad M. Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. Information processing in agriculture, 2021, 8(1): 27-51.

Patrício D I, Rieder R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and electronics in agriculture, 2018, 153: 69-81.

Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 658-666.

Sa I, Ge Z, Dayoub F, et al. Deepfruits: A fruit detection system using deep neural networks. sensors, 2016, 16(8): 1222.

Sun M, Xu L, Chen X, et al. Bfp net: balanced feature pyramid network for small apple detection in complex orchard environment. Plant Phenomics, 2022, 2022.

Tang, Y., Zhou, H., Wang, H., et L.. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision, Expert Systems with Applications 2023, 211:118573.

Tey Y S, Brindal M. A meta-analysis of factors driving the adoption of precision agriculture. Precision Agriculture, 2022, 23(2): 353-372.

Tu S, Pang J, Liu H, et al. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precision Agriculture, 2020, 21(5): 1072-1091.

Wang N, Gao Y, Chen H, et al. Nas-fcos: Fast neural architecture search for object detection. proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11943-11951.

Wu H T, Tsai C W. An intelligent agriculture network security system based on private blockchains. Journal of Communications and Networks, 2019, 21(5): 503-508.

Wu Y, Chen Y, Yuan L, et al. Rethinking classification and localization for object detection. Proceedings of the

IEEE/CVF conference on computer vision and pattern recognition. 2020: 10186-10195.

Xu B, Cui X, Ji W, et al. Apple grading method design and implementation for automatic grader based on Improved YOLOv5. Agriculture, 2023,13,124.

Yang L, Chen Y, Tian Z, et al. Field road segmentation method based on improved UNet. Transactions of the Chinese Society of Agricultural Engineering, 2021, 37(09): 185-191. (in Chinese)

Yang X, Yang J, Yan J, et al. Scrdet: Towards more robust detection for small, cluttered and rotated objects. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 8232-8241.

Yu F, Wang D, Shelhamer E, et al. Deep layer aggregation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2403-2412.

Zand M, Etemad A, Greenspan M. Objectbox: From centers to boxes for anchor-free object detection. European Conference on Computer Vision. Springer, Cham, 2022: 390-406.

Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9759-9768.

Zhang W, Wang S, Thachan S, et al. Deconv R-CNN for small object detection on remote sensing images. IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2018: 2483-2486.

Zhao K, Yan W Q. Fruit detection from digital images using CenterNet. International Symposium on Geometry and Vision. Springer, Cham, 2021: 313-326.

Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression. /Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.

Zhu B, Wang J, Jiang Z, et al. Autoassign: Differentiable label assignment for dense object detection. arXiv preprint arXiv:2007.03496, 2020.

Zhu C, He Y, Savvides M. Feature selective anchor-free module for single-shot object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 840-849.

Zhu X, Su W, Lu L, et al. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020.