SOD Head: A Network for Locating Small Fruits from Top to Bottom in Layers of Feature Maps

Yuqi Lu[1], Meili Sun[1], Yujie Guan[1], Jian Lian[2], Ze Ji[3], Xiang Yin[4], Weikuan Jia[1,5*]

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

[2] School of Intelligent Engineering, Shandong Management University, Jinan 250357, China

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

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

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

Abstract: Although object detection technology has been applied in the field of smart orchards, detecting small fruits in real orchard environments is still a great challenge due to the interference of fruit scale issues. In this study, we propose an effective detection head named SOD Head for detecting small-scale fruits in the early growth stage, aiming to enhance the monitoring of fruit growth in the early stages and achieve intelligent management of orchards. SOD Head firstly utilizes the rich semantic information in the top-level feature map to determine the vague feature position, and mapping downward to the next level, achieving layer-by-layer locating and refinement of feature information. This can avoid missing the features of small fruits that are sparse on the high-resolution feature map and reduce the interference brought by information redundancy to small-scale detection. Secondly, SOD Head performs operation of box relocation to make the prediction of the boundary boxes for small-scale fruits more stable. The experimental results show that SOD Head achieves AP_s of 29.5% and 39.6% on the datasets of Gold Pear

before the thinning stage and MinneApple respectively. Overall, SOD Head not only has a higher detection accuracy on small-scale fruits than other algorithms, but also has good generalization and versatility.

Keywords: Locating Network; Box Relocation; SOD Head; Small fruit detection

1. Introduction

With the development of neural networks based on deep learning, object detection has become increasingly mature, mainly including two types: algorithms with anchor-based and anchor-free. The anchor-based object detection model is a conventional approach that relies on a predefined set of anchor boxes (also known as prior boxes or suggestion boxes). These anchor boxes are defined at different scales and aspect ratios, serving as reference regions for potential target areas. The model detects and localizes objects by classifying and regressing these anchor boxes. On the other hand, the anchor-free object detection model represents a relatively new method, where the central idea is to directly predict the object's position in the image without relying on predefined anchor boxes. This approach offers a more concise framework, eliminating the need for anchor box selection and potential hassle. Moreover, it has the advantage of better adaptability to shape and size variations across various targets. Both models possess their distinct advantages and applications in object detection. Regardless of whether people choose to use an anchor-based or anchor-free approach, object detection has found widespread applications across various fields such as autonomous driving (Liu et al., 2020; Xu et al., 2023), pedestrian detection (Dollar et al., 2011; Ge et al., 2021a), and smart orchards (Patrício et al., 2018; Tang et al., 2023a). Among them, the intelligent management of orchards, such as irrigation and fertilizer supply, scientific yield measurement, etc., relies on the

help of object detection technology. Currently, there have been many studies applying object detection to orchard environments, such as disease and pest detection (Ngugi et al., 2021; Singh et al., 2021), fruit counting (Gao et al., 2022; Tang et al., 2023), seed testing (Audu et al., 2021; Pareek et al., 2021) and yield prediction (Tesfaye et al., 2021; Sun et al., 2022c). However, this study aims to use object detection technology to achieve full monitoring of fruit growth stages for orchard management. This can help farmers scientifically plan irrigation and fertilizer supply in the orchard and make scientific yield predictions, thus achieving the goal of intelligent orchard management. In addition, for certain fruit varieties that require thinning, early management of the fruit can help guide the fruit thinning process. During the early stages of fruit growth, the color and size of the fruit are not distinctive, making it difficult to accurately detect early-stage fruit. While in the mature stage, the surface characteristics of the fruit make them easier to detect. Therefore, this paper focuses on the fruit detection of early-stage growth.

During the early growth stage of fruits, their sizes are small and their colors are green, which have low contrast with the background of the orchard. Moreover, the detection of small-scale objects is not well developed in algorithms of object detection. Currently, the detection accuracy of small-scale objects is almost only half that of large-scale objects (Tong et al., 2020). Prior to the thinning period, the green small-scale fruits are densely distributed in the orchard and are heavily occluded, which makes the recognition of them even more challenging using object detection technology in orchards. Many studies have been conducted in the field of object detection to address the detection of small-scale objects (Liu et al., 2021a; Yang et al., 2022).

There are several methods for detecting small objects, including effective feature fusion (Liu et al., 2018; Tan et al., 2020), optimized label assignment strategies (Ge et al., 2021b; Su et al.,

2022), and data augmentation (Bochkovskiy et al., 2020). Since small objects have weak presence in the image, a learnable data augmentation strategy proposed by Zoph (Zoph et al., 2020) could be used for small object detection. This study represented data augmentation as a multi-layer neural network, and optimized the data augmentation strategy by updating network parameters. Experimental results showed that this method improved detection accuracy by at least 2.3 percentage points and had strong scalability. And to enrich the feature representation of objects, most algorithms use the Feature Pyramid Network (FPN) (Lin et al., 2017a) family for feature fusion, which also increases the feature representation of small objects in high-resolution images. Li (Li et al., 2019) proposed the SAT network, which used three parallel convolution networks to obtain feature maps of different scales and then fused them to enhance the ability of networks to detect objects of different scales. The SAT network also introduced a new scale attention mechanism to adaptively adjust the importance of feature maps of different scales. Experimental results showed that the SAT network achieved excellent detection performance on multiple public datasets, especially in small object detection. In terms of defining positive and negative samples in the anchors, Xu (Xu et al., 2022) proposed a label assignment strategy for detecting small objects, called Gaussian Receptive Field based Label Assignment (RFLA). This strategy used the Gaussian response of each pixel which was modeled to calculate the matching degree between candidate boxes and ground truth boxes. And then it selected positive and negative samples based on the matching degree. The RFLA outperformed competitors by four percentage points on the AI-TOD dataset (Wang et al., 2021), and it could also adapt to different datasets and detectors, which was very helpful for detecting small objects. In addition, there are other methods that can help with the detection of small objects. For example, the CASOD proposed by Lim (Lim et al., 2021) introduced

89 a context feature extraction module and attention mechanism into the detector to enhance its
90 response to small objects. The context feature extraction module could increase the feature
91 representation of small objects, while the attention mechanism could adaptively adjust the detector's
92 focus on different parts according to the object size and location. Experimental results showed that
93 CASOD achieved an accuracy of 78.1% on the VOC dataset.
94 With the development of the aforementioned detection algorithms, the application of object
95 detection in fruits has also received widespread attention (Fu et al., 2020; Jia et al., 2020; Sun et al.,
96 2022a). Hussain (Hussain et al., 2022) proposed a deep learning-based framework for automatic
97 detection and recognition of fruits and vegetables in complex scenes. It could help salespeople
98 identify vegetables and fruits with high similarity. Although it achieved an accuracy rate of up to
99 96%, it did not specifically design a detection method for small-scale fruits. Mai (Mai et al., 2018)
100 proposed a multi-classifier fusion strategy and a correlation loss term for the classifiers. The
101 classifiers used features from three different levels to learn three classifiers for object classification.
102 The loss term helped the network better learn the feature differences between fruit targets. Although
103 the detection accuracy was improved, the improvement on small-scale fruit detection was not
104 significant. Koirala (Koirala et al., 2019) proposed MangoYOLO for fast detection of mangoes in
105 tree crown images. The authors accelerated the detection speed while ensuring accuracy, achieving
106 8 ms per pixel image with the size of 512×512. Although both its speed and accuracy were improved,
107 this was mainly due to the distinctive color and shape features of mangoes, which made them easily
108 distinguishable from the background. In contrast, MangoYOLO could not handle small green fruits
109 that were difficult to distinguish from the background during early growth. Sun (Sun et al., 2022b)
110 proposed GHFormer Net for detecting small-scale apples and hawthorn fruits in low-light

conditions. GHFormer Net adopted Transformer-based PVTv2-B1 as the backbone and introduced two loss terms to adapt to low-light conditions. The experimental results showed that it achieved more accurate detection of small-scale fruits. However, it had a relatively large computational cost and requires certain computational resources. Although fruit detection has made some progress in recent years, detecting small-scale fruits in real orchard environments remains challenging. Moreover, there is currently a lack of research on the detection of small-scale green fruits in early growth stages. In this context, this study aims to propose a detection algorithm specifically designed for the features of early-stage fruits in real orchard environments. This algorithm is intended for monitoring the early growth status of fruits to achieve intelligent management of orchards.

This study proposes a universal detection head for detecting small objects named SOD Head. It maps the position information of features from the features of top-level features to bottom-level, and the feature information at these positions in the feature maps is used for classification and regression. This approach largely avoids the interference of redundant information in low-level feature maps on small object features when directly detecting the entire feature map. In addition, when the bounding box is very small, even slight adjustments may greatly affect the overlapping part with the real box, resulting in significant fluctuations in metrics such as Intersection over Union (Rezatofighi, et al., 2019). To address this issue and make predictions for small objects more stable, this study adds the operation of “box relocation” to the regression branch. Overall, the main contributions of this paper are as follows:

(1) This study introduces a novel detection head, specifically designed for small fruits, known as SOD Head. To address the challenge posed by the sparse distribution of small fruits in low-level feature maps, SOD Head strategically maps feature information from the top-level feature map to

lower levels, enabling fruits localization from top to bottom. By adopting this approach, it circumvent the issue of missing small fruit features caused by redundant information while computing the entire low-level feature map.

(2) In order to achieve more precise localization of small fruits, this study introduces an operation called "box relocation." This operation involves a second regression step on the bounding boxes, effectively increasing the learning capacity of the network. By implementing box relocation, the sensitivity to small adjustments in the bounding box is reduced, resulting in more stable and accurate detection of small fruits within the SOD Head.

(3) The experimental results illustrate that SOD Head attains superior detection accuracy for small fruits in both the Gold Pear and MinneApple datasets, achieving accuracy rates of 29.5% and 39.6%, respectively. These accuracy rates surpass those achieved by classical detection algorithms in recent years. Furthermore, SOD Head exhibits remarkable versatility and generalization capabilities, demonstrating its proficiency in complementing various backbones for downstream detection tasks.

The structure of this paper is as follows: Section 2 introduces the dataset used in this study, including the process from acquisition to preparation. Section 3 provides the complete methods, including a detailed description of the SOD Head, which consists of a network for object localization layer by layer, and detection and regression branches with the added operation of box relocation, and losses for the whole algorithm. Section 4 describes the experiments, including the comparison experiments, and ablation studies. Section 5 summarizes and discusses the research findings and conclusions of this study.

2. Datasets

This study proposes SOD Head, a novel approach designed to address the challenges posed by the small scale of fruits during the early growth stage, enabling efficient monitoring of the early growth fruits in orchards. The primary focus of this study is the early growth stage of gold pear, which requires fruit thinning to optimize its development. To create a suitable dataset and simulate real working conditions in orchards, we collected images of gold pears during the thinning period. Statistical analysis revealed that gold pears during this stage exhibit small-scale characteristics, making them an ideal target for precise detection in this study. Conversely, the public dataset MinneApple (Häni et al., 2020) comprises images of mature apples, which are not representative of small-scale fruits during the early growth stage. Nevertheless, due to the small-scale of fruit in the long-distance images, we utilize the MinneApple dataset to evaluate the detection performance of SOD Head on small-scale fruits and demonstrate the algorithm's generalization capabilities in orchard environments. It is essential to highlight that this study adopts the scale division criteria used in the COCO dataset. Accordingly, objects occupying less than 0.25% of the total pixels are classified as small-scale, those covering more than 2.25% are categorized as large-scale, and objects falling between these percentages are considered medium-scale.

2.1 Dataset of Gold Pear

This study aims to achieve effective monitoring of fruit growth throughout all stages by precise detection of small-scale fruits, thereby helping farmers of realize intelligent management of orchards. Based on this, we captured and produced a dataset of gold pears before thinning, which was used to test the feasibility of the algorithm. The fruits during this period are characterized by small size and green color, which can represent the state of other fruits in the early growth stage to some extent. The Golden Pear dataset was captured using a mobile phone, with dimensions of

3024×4032 (width×height). The images were all captured at RiSheng Gold Pear Professional Cooperative in Jiaozhou, Qingdao, Shandong Province. In order to fully simulated the orchard environment for the algorithm, we captured gold pear images at different angles (far and near) and different lighting conditions (front and backlit), as shown in Figure 1. This is to simulate the varying viewpoints of robots operating in the orchard. The intention behind this approach is to ensure that the perspective of the picking robot aligns closely with the fruit, while for yield estimation purposes, a relatively extended perspective is employed to efficiently detect complete orchards.

Fig 1. Images of fruits and their corresponding annotation information (in orange boxes) in Gold Pear.

This study used the tool of LabelMe to process the captured images, which marked fruits using rectangular boxes. The portion inside the box is considered foreground, while the remainder is considered background. The dataset contained only one class, named "pear". Each image was labeled and generated a json format file containing the fruit information. The annotated images were shown in the Figure 1, where the orange rectangle indicated the labeled box and the fruit to be

detected was inside the box. We captured and labeled 1549 images, which were divided into a training set and a validation set in a 7:3 ratio. The training set consists of 1084 images, and the test set consists of 465 images. The number of fruits of different scales in the dataset is shown in Table 1. It can be seen that most of the fruits in the Gold Pear dataset are small in size, accounting for almost half of the total fruit number. The next most common size is medium-sized fruit, accounting for 35.55% of the total, while the number of large-sized fruit is the least.

Table 1. The statistics of fruit quantities with different scales in the Gold Pear dataset.

	small fruits	middle fruits	large fruits	total fruits
train dataset	3122	2307	1001	6430
val dataset	1305	944	465	2714
total	4427 (48.41%)	3251 (35.55%)	1466 (16.04%)	9144

2.2 Public dataset of MinneApple

The publicly available dataset, MinneApple, was used to test the generalization performance of SOD Head. It is an apple dataset for detection and segmentation, which mainly contains distant images as shown in Figure 2. And it is a publicly available dataset with dimensions of 720×1280 (width×height). The fruits in the dataset are of two colors, red and green. There are 670 images used for detection, which were divided into 603 training images and 67 validation images.

Fig 2. Images of fruits and their corresponding annotation information (in orange boxes) in MinneApple .

This study performed multi-scale statistics on the number of fruits in the images used for detection, as shown in Table 2. It can be seen that MinneApple used for detection contains almost exclusively small-scale fruits. It is worth noting that the small-scale fruits in MinneApple are most likely caused by the distant shooting, while the fruit scales in the Gold Pear are real. Therefore, the dataset of Gold Pear is better suited to represent the early growth stages of fruits, while MinneApple is only used to test the detection performance of small-scale fruits in the algorithm.

Table 2. The statistics of fruit quantities with different scales in the MinneApple dataset.

	small fruits	middle fruits	large fruits	total fruits
train dataset	25177	72	0	25249
val dataset	2925	9	0	2934

total	28102 (99.71%)	81 (0.29%)	0	28183

3. Methods

To accurately detect small green fruits in the early growth stages in orchards, a general and state-of-art detection head is proposed in this study, as shown in Figure 3. The detection targets of the SOD head are multi-scale feature maps that have undergone feature extraction and fusion. The SOD head firstly constructs a network for locating small fruits from top to bottom in layers of feature maps, which refines the fuzzy feature content layer by layer based on the rich semantic information of the top-level feature map. The refinement is achieved by mapping the feature position of the upper layer to obtain the feature content of the lower layer (as shown in orange parts in Figure 3), which is used for subsequent classification and regression. In order to predict the boundary box of small-scale fruits more accurately and stably, an operation named “box relocation” is added to the box regression process. It increases the learning contents of the network and achieves second regression of boxes.

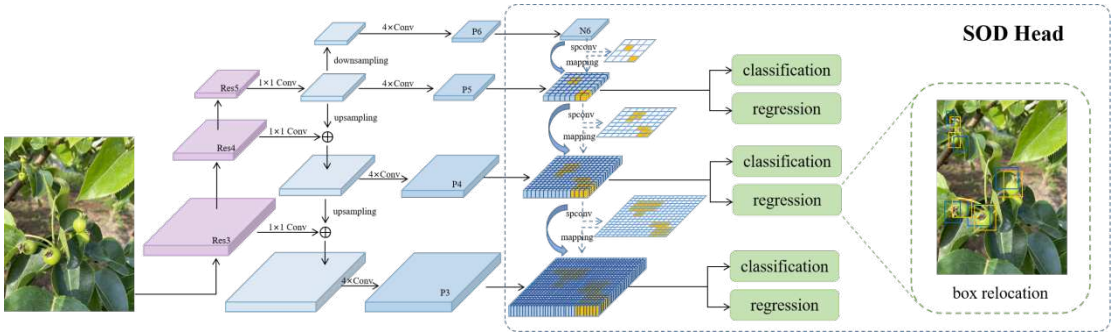


Fig 3. The overall structure diagram of the SOD Head.

3.1 Backbone

In this study, the SOD Head used two classic backbone networks of different types, namely

Resnet50 (He et al., 2016) and Swin Transformer (Liu et al., 2021b), to verify its performance as a downstream task processor. Figure 3 illustrates the process of multi-scale feature maps extraction and fusion using Resnet50 and FPN as an example.

Resnet50 is a classical deep convolutional neural network composed of residual structures. The residual structure consists of two convolutional layers and a skip connection that directly adds the input of the two convolutional layers. If the output of the first convolutional layer is the same as the output of the second, the skip connection will pass a zero vector, which does not affect the subsequent computation. If there is information that needs to be back-propagated, this zero vector will be destroyed, retaining the gradient from the earlier part of the computation to the later part. Resnet50 increases the depth of the network while preventing the problems of gradient vanishing and exploding through skip connections. With the residual structure, Resnet50 can be deeper and more accurate than previous neural networks, making it widely used as a backbone network for feature extraction in fields such as image classification and object detection.

Swin Transformer is different from other transformer-based algorithms applied in computer vision (Dosovitskiy et al., 2020), as it can extract features of different scales for downstream tasks. Specifically, Swin Transformer decomposes the image into different blocks and applies a window-based self-attention mechanism independently on each block. This window-based mode reduces the exponential growth of computational complexity with respect to image size to linear growth, significantly lowering computational requirements. Additionally, it achieves multi-scale feature extraction through patch merging operations. These characteristics make it highly scalable and a useful backbone network for other vision tasks.

The four different scale feature maps output by the backbone network are represented by Res2-

Res5, which are then fed into the FPN for feature fusion. In this study, four additional 3×3 convolutional layers are added after each FPN layer to further extract features for downstream tasks such as classification and regression, without changing the size of the feature maps. Only the architectures of the first four layers are shown in the Figure 3, and the structure of the fifth layer is the same as the previous ones.

3.2 Small Objects Detection Head

The SOD Head mainly consists of two parts: a network for locating small fruits from top to bottom in layers of feature maps and a branch prediction network with box relocation. This section mainly introduces the network, and box relocation is described in detail in Section 3.3.

As is well known, multi-scale detection networks based on deep learning detect small objects in low-level feature maps and large objects in high-level feature maps, because low-level feature maps have higher resolution. However, the distribution of small objects on high-resolution feature maps is sparse due to their small size, which causes redundancy in information. In other words, most of the information on high-resolution feature maps belongs to the background rather than the object to be detected. In this case, down-sampling of the image is a common solution to this problem. The successive down-sampling of feature maps actually enhances the semantic information of maps, but inevitably leads to a loss of some positional information due to the decrease in resolution. This is why feature fusion (Wu et al., 2020) is necessary, which fuses the positional information from low-level feature maps with the semantic information from high-level feature maps. In other words, the top-level feature map contains all the feature information of objects, albeit relatively blurred due to the resolution limitation. Therefore, to avoid the interference caused by the sparse distribution of small object features in low-level feature maps on detection, this study constructs a network in the

detection head that locates objects layer by layer from top to bottom. The core idea is to refine the blurred but complete feature information in the top-level feature map layer by layer downwards for subsequent classification and regression. The layer structure of the network is shown in Figure 4.

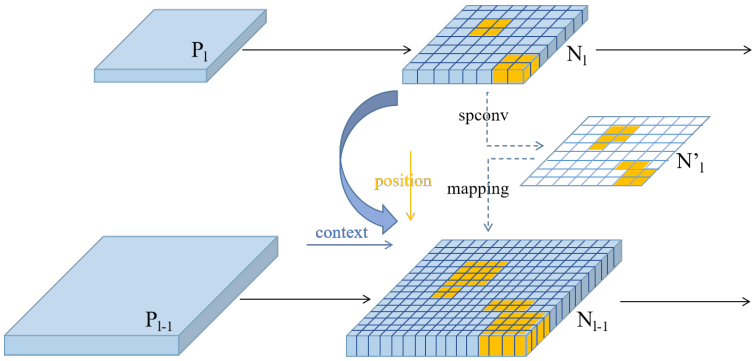


Fig 4. The layer structure of locating network.

Figure 4 presents a layer-by-layer demonstration of the feature localization network, showcasing how the network filters feature regions progressively from top to bottom. N_l represents a layer in a multi-scale feature map. It undergoes a dilated convolution with an output channel number of 1, resulting in the merged values of tensors at each position, forming a score matrix. Strictly speaking, this score matrix determines the position of features and serves as a special feature map $N_l^{'}$ with a channel number of 1. When the score is above the threshold σ (set to 0.5 as a hyper-parameter in the experiment), the position is identified as the presence of an object. These positions are highlighted by the orange grid on $N_l^{'}$: $\{o_l\} = \{(x_l^i, y_l^i)\}$. Here, i represents the number of positions, and l indicates the specified feature layer. These positions are mapped down to the feature map P_{l-1}, producing N_{l-1}. During the mapping process, the feature map $N_l^{'}$ provides positional information, while the feature map P_{l-1} offers the underlying feature content. The mapping rules are depicted in Equation 1:

$$\{0_{l-1}\} = \{(2x_l^i + a, 2y_l^i + b), \forall a, b \in (0,1)\} \quad (1)$$

Where, x_l^i, y_l^i represents the positional information in feature map N_l', and a, b represents the

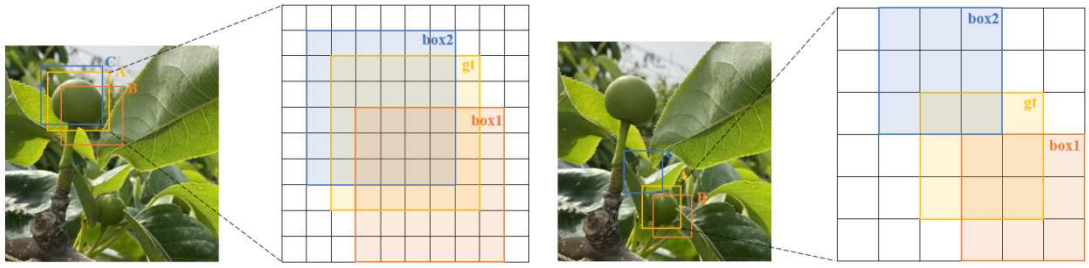
offset weights for mapping in both horizontal and vertical directions. The features at the selected positions are further used for sparse convolution to locate the objects, and map the lower-level features. In other words, this network does not detect the entire feature map using traditional methods, but rather performs classification and regression on the selected positions (marked in orange) that potentially contain objects in a layer-by-layer manner. This approach avoids the interference caused by redundant information when directly computing the low-level feature map for small object detection. The network is characterized by locating objects from top to bottom, from blurry to clear.

The detection head uses an anchor-based approach to perform classification and regression on the positions selected by the top-down localization network in the feature map N_l. Both the classification and regression modules consist of four 3×3 convolutional layers followed by a final prediction layer. The algorithm ultimately performs classification and regression on the five layers of feature maps N2-N6, sharing the same set of detection parameters. The specific training procedure is explained in Section 3.4. Experimental results show that the SOD Head can serve as a universal anchor-based detection head that can be matched with different backbones for downstream detection tasks.

3.3 Box Relocation

During the training process, it fine-tunes the anchor boxes by regressing the offset value between them and the gt boxes. This regression process allows the algorithm to adjust the position of the anchor boxes to align them accurately with the real boxes, facilitating precise object detection during inference. Regression is a crucial step in object detection, as it helps the algorithm learn and generate accurate prediction box coordinates. However, when dealing with small-scale fruits, even

slight movements of the bounding box can lead to significant fluctuations in area-based evaluation metrics like IoU. The reduced number of pixels in small objects amplifies the impact of these fluctuations, often resulting in IoU dropping drastically, even to zero, with minimal changes in the bounding box fine-tuning. Consequently, obtaining accurate bounding boxes for small-scale objects becomes challenging, and it can also affect loss calculations, thereby diminishing the algorithm's detection accuracy for small-scale fruits. While moderate bounding box adjustments prove beneficial in achieving more accurate bounding boxes for normal-scale fruits, the same magnitude of adjustments may not necessarily yield positive results for small-scale fruit detection. The Figure 5 below illustrates this phenomenon, highlighting the intricate challenge posed by fine-tuning bounding boxes for small-scale objects.



a) Normal-scale fruit b) Small-scale fruit

Fig 5. Comparison of boundary box regression for fruits of different scales.

In Figure 5, we observe three representations: A denotes the gt box, B represents the bounding box before movement, and C shows the bounding box after regression. Remarkably, when moving the same number of pixels, the box IoU of the normal-scale fruit changes from 0.38 to 0.52, indicating an improvement in alignment with the position of the gt box. However, the IoU of the small fruit bounding box undergoes a drastic fluctuation from 0.29 to 0.13, which is not conducive to accurate detection of small-scale fruits. To address this issue and enhance the stability of bounding box predictions, this study introduces an additional operation of box relocation during the regression

stage of the detection head, as illustrated in Figure 6. By incorporating this operation, the content of network regression increases, effectively mitigating the fluctuations in bounding box predictions for small-scale fruits.

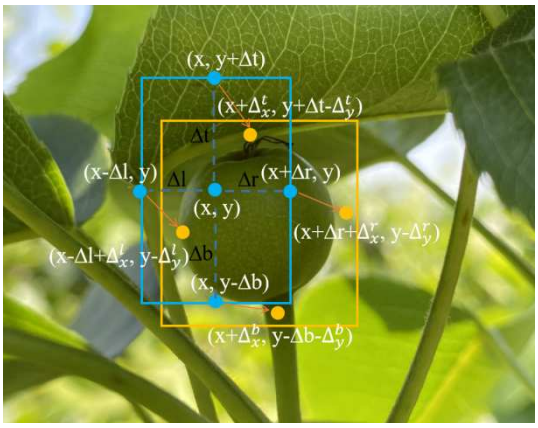


Fig 6. Adjustment content for box location: the blue box to the orange box.

In conventional object detection algorithms, regression is typically performed on each bounding box by pairing anchors and gt boxes based on IoU. The anchor is assigned the label of the corresponding gt box, and the center offset of the anchor relative to the gt box is calculated to train the network using appropriate loss functions. However, this standard regression approach encounters difficulties in accurately detecting small-scale fruits, as slight movements of the bounding box can lead to significant fluctuations in indicators such as IoU. To address this limitation, this study introduces a new regression approach specifically designed for small-scale fruits, ensuring precise positioning of the prediction boxes. The bounding box to be regressed is represented as $k(x, y) = \{\Delta t, \Delta b, \Delta l, \Delta r\}$, where (x, y) denotes the center coordinate of the box, and $\{\Delta t, \Delta b, \Delta l, \Delta r\}$ represents the distances from the center coordinate to the edges of the box in four directions. This representation uniquely characterizes the bounding box, as demonstrated by the blue box in Figure 6. The network employs deformation convolutions to regress the blue bounding box, with the regression content being the four central coordinates located on the box. The network learns a new

set of offsets to predict the updated values of these central coordinates. The Equation 2 is then utilized to determine the regression bounding box based on these updated coordinates:

$$k'(x,y) = \begin{pmatrix} (x+\Delta_x^t, y+\Delta t-\Delta_y^t), (x+\Delta_x^b, y-\Delta b-\Delta_y^b), \\ (x-\Delta l+\Delta_x^l, y-\Delta_y^l), (x+\Delta r+\Delta_x^r, y-\Delta_y^r) \end{pmatrix} \quad (2)$$

Where $\{\Delta_x^t,\ \Delta_x^b,\ \Delta_x^l,\ \Delta_x^r,\ \Delta_y^t,\ \Delta_y^b,\ \Delta_y^l,\ \Delta_y^r\}$ represents the offset learned by the network for different directions, and $\{x, y\}$ is the center coordinate value of the original box. The boundary of the new prediction box is determined by the relocated points, as illustrated by the orange box in Figure 6. The box relocation further regresses to obtain the offset of the box in all four directions. This not only enhances the accuracy of the prediction box, but also mitigates the influence of large-scale detection box regression on small-scale boxes, resulting in improved prediction stability for small-scale bounding boxes.

3.4 Loss

During the training process, the algorithm uses Focal Loss (Lin et al., 2017b) to calculate the classification loss, which can alleviate the problem of class imbalance to some extent. This problem is also a common issue that affects the accuracy of object detection. Focal Loss reduces the weight of samples with clearly classified categories through a scaling factor, while increasing the weight of samples with ambiguous categories. The scaling factor is shown in Equation 3:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (3)$$

Where, p_t represents the confidence of the sample, and $(1-p_t)^{\gamma}$ can reduce the contribution of easily classified samples to the loss. γ represents the weight factor of those difficult-to-classify samples, $\gamma \in [0,5]$. α_t is a balancing factor that can alleviate the imbalance between positive and negative samples. With this scaling factor, samples with excessively high or low confidence will not have a significant impact on the loss. For anchor boxes judged as positive samples, the algorithm

uses SmoothL1Loss to calculate the regression loss, as shown in Equation 4:

$$\mathrm{loss}(x,y)=\frac{1}{n}\sum_{i=1}^{n}\begin{cases}0.5(y_i-f(x_i))^2, & \text{if } |y_i-f(x_i)|<1\\ |y_i-f(x_i)|-0.5, & \text{otherwise}\end{cases} \tag{4}$$

In which, y_i represents the ground truth value, $f(x_i)$ represents the predicted value by the network. SmoothL1Loss belongs to a piecewise function: when the error between the predicted value and the ground truth value is small, it reduces the impact of the error; when the error is large, it makes the gradient value small enough to prevent gradient explosion.

The network used for object localization in the SOD Head is also trained with the Focal Loss function. Prior to computation, the feature maps are re-encoded in the following manner. Firstly, the distance from all positions on the feature map to the ground truth box is calculated: $\mathrm{Distance}((x,y),(x^*,y^*))$. Then, if the distance is less than a hyper-parameter s, then the position is encoded as "1"; if the distance is greater than s, the position is encoded as "0". The Focal Loss calculation is then performed based on this encoding:

$$\mathrm{FL}=\begin{cases}-(1-\hat{p})^{\gamma}\log(\hat{p}), & \text{if } y=1\\ -\hat{p}^{\gamma}\log(1-\hat{p}), & \text{if } y=0\end{cases} \tag{5}$$

In order to better illustrate the calculation process based on this encoding method, the segmented form of Focal Loss is used in Equation 5. Here, y represents the encoding value of each pixel, and s is set to the scale of the smallest anchor box on the feature map of that layer in the experiment.

4. Experiments

4.1 Training

The experiments were conducted on a server equipped with Ubuntu 16.04 operating system. The algorithm was built based on PyTorch and trained on an NVIDIA Tesla V100 GPU. The batch size was set to 8, the initial learning rate was set to 0.001, and a weight decay of 0.0001 was used to prevent over-fitting. The algorithm was trained 4000 iterations.

Average Precision (AP) serves as an indicator to gauge the accuracy of object detection algorithms for different categories, with its calculation method outlined in Equation 6. And AP_s and AP_{75} are widely used evaluation metrics in object detection to assess algorithm accuracy and performance across different target categories. AP_{75} is a specialized version of AP that measures average accuracy at an IoU of 0.75. In other words, when calculating AP_{75}, the detection result is considered correct only when the IoU between the detected box and the real label box is greater than or equal to 0.75. Specifically, AP_s calculates the detection accuracy of small-scale objects, which is of particular significance for SOD Head, as it is designed for detecting small-scale fruits. So evaluating the algorithm's performance on small-scale fruits, denoted as AP_s, is crucial for accurate assessment. The same applies to AP_{50}, AP_m and AP_l.

$$AP = \int_0^1 P(R)dR \tag{6}$$

Where P represents the ratio of predicted positive samples to the number of true positive samples, and R represents the proportion of true positive samples correctly predicted by the algorithm, calculated as shown in Equations 7 and 8.

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{8}$$

In the calculations above, True Positive (TP) represents the number of detection boxes with IoU greater than the set IoU_threshold; False Positive (FP) represents the number of detection boxes with IoU less than the IoU_threshold or the number of redundant detection boxes generated for the same target; and False Negative (FN) represents the number of undetected targets. In this experiment, it sets the IoU_threshold to 0.5, a commonly chosen as it provides a relatively lenient criterion when evaluating object detection algorithms. This means the algorithm can still be considered correct even

with a certain degree of overlap between the predicted bounding box and the actual bounding box. The choice of 0.5 as the threshold is based on empirical observations and is prevalent in classic object detection algorithms like Faster RCNN (Mai et al., 2018), YOLO (Bochkovskiy et al., 2020), and others.

4.2 Ablation Studies

The effect of box relocation is validated on two datasets in this study, as shown in Table 3. The effect is more pronounced on the dataset of Gold Pear. Although there is a slight decrease in AP, which may be due to the decrease in detection accuracy of medium and large-scale fruits, it increases the detection accuracy of small-sized fruits by 0.7 percentage points. However, for the MinneApple, which has more than 99% small-scale fruits, the effect is not very significant, with only about a 0.1 percentage point improvement. We speculate that this is because the fruits to be detected in MinneApple are almost indistinguishable in scale due to the distant shooting, and therefore box relocation does not significantly improve detection accuracy. This ablation study helps us clarify the usage conditions and environment of box relocation, and can provide effective assistance for the detection of small-sized fruits in specific environments.

Table 3. Results of box relocation.

	box relocation	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)
Gold Pear	×	51.60	82.52	53.23	26.70	66.46	80.58
	√	51.26	82.72	53.51	27.42	65.56	80.14
MinneApple	×	39.45	79.84	33.88	39.49	30.92	-1.0
	√	39.56	80.62	34.69	39.55	29.01	-1.0

4.3 Comparison Experiments

The accuracy of the SOD Head is verified on two datasets as follows.

4.3.1 Experiments on Gold Pear

In order to verify the detection performance of the SOD Head, this study firstly compared it with classical and advanced detection algorithms, such as Double-Head (Wu et al., 2020), FoveaBox (Kong et al., 2020), Faster RCNN (Mai et al., 2018), Grid RCNN (Lu et al., 2019), Libra RCNN (Pang et al., 2019), Trident (Li et al., 2019) on the dataset of Gold Pear. It is important to emphasize that SOD Head is specifically designed for detecting small-scale fruits. Therefore, we place particular emphasis on the value of AP_s, as it accurately reflects the algorithm's detection accuracy for small-scale fruits. The experimental results are shown in Table 4.

Table 4. Comparison of algorithms on Gold Pear.

	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)
Double-Head	53.1	83.3	53.6	25.6	70.5	83.2
FoveaBox	52.3	80.9	54.1	24.1	68.6	85.4
Faster RCNN	52.0	80.5	54.2	25.6	69.4	81.9
Grid RCNN	52.3	83.4	54.9	26.2	67.5	81.4
Libra RCNN	53.0	82.8	54.9	27.0	68.4	83.5
Trident	53.0	80.6	53.7	23.6	71.5	86.9
Ours + Resnet	51.3	82.7	53.5	27.4	65.6	80.2
Ours + Swin	52.6	85.9	53.5	29.5	66.2	79.6

This study conducted experiments on two backbones to verify the detection performance of the proposed detection head. The results show that when Swin Transformer is used as the backbone,

446 the SOD Head achieves higher detection accuracy, especially for small-scale fruit with an accuracy
447 of 29.5%. The overall detection accuracy of other algorithms on the Gold Pear dataset is not
448 significantly different from SOD Head, ranging from 52.0% to 53.0%, even higher by 0.2%-0.4%
449 percentage points, such as Grid RCNN. However, SOD Head has the highest detection accuracy for
450 small-scale fruits on both backbones. The AP_s of the proposed SOD Head combines with Swin
451 Transformer outperforms other algorithms by at least 2.5 percentage points. Even when combined
452 with Resnet, our algorithm achieves the highest AP_s at 27.4%, higher than other algorithms. It
453 improves the detection accuracy of small-scale fruits by 3.3 and 1.8 percentage points compared to
454 classic algorithms of FoveaBox and Faster RCNN, respectively. Although Trident achieves the
455 highest detection accuracy at 53.0%, its performance in detecting small-scale fruits is poor, only
456 23.6%. This may be due to its structure that takes a single-scale feature map as input. While Grid
457 RCNN and Libra RCNN achieve higher overall detection accuracy than SOD Head with Resnet,
458 their AP_s are at least 0.4 percentage points lower.

459 In a word, Table 4 clearly shows that regardless of the combined backbone network, SOD Head
460 outperforms other algorithms in terms of detection accuracy for small-scale fruits, with a superiority
461 of at least 0.4% percentage points. The detection accuracy of SOD Head combined with Swin
462 Transformer for small scales can reach an impressive 29.5%, whereas other algorithms achieve only
463 27.0% accuracy. This outstanding performance showcases SOD Head's exceptional ability to detect
464 small-scale fruits. Furthermore, as a general detection head, SOD Head's accuracy continues to
465 improve with the evolution of the backbone network. For instance, when other configurations
466 remain unchanged, using Swin Transformer as the backbone further enhances detection accuracy,
467 demonstrating the adaptability and potential for improvement in SOD Head's performance across

different network configurations.

Figure 7 shows the loss curves of these algorithms above during training on the dataset of Gold Pear. From the change of loss, it can be observed that SOD Head with Swin Transformer not only converges quickly to the minimum value, but also has the smallest oscillation amplitude. Compared with using Resnet as the backbone, it has a faster decrease in loss, especially in the first 500 iterations. This also indicates that when Swin Transformer is used as the backbone on the Gold Pear dataset, SOD Head performs better and is more reliable. In contrast, Double-Head and Trident RCNN have larger oscillation amplitudes and their convergence process is not stable enough. This may be due to the algorithms not being suitable for detecting small objects.

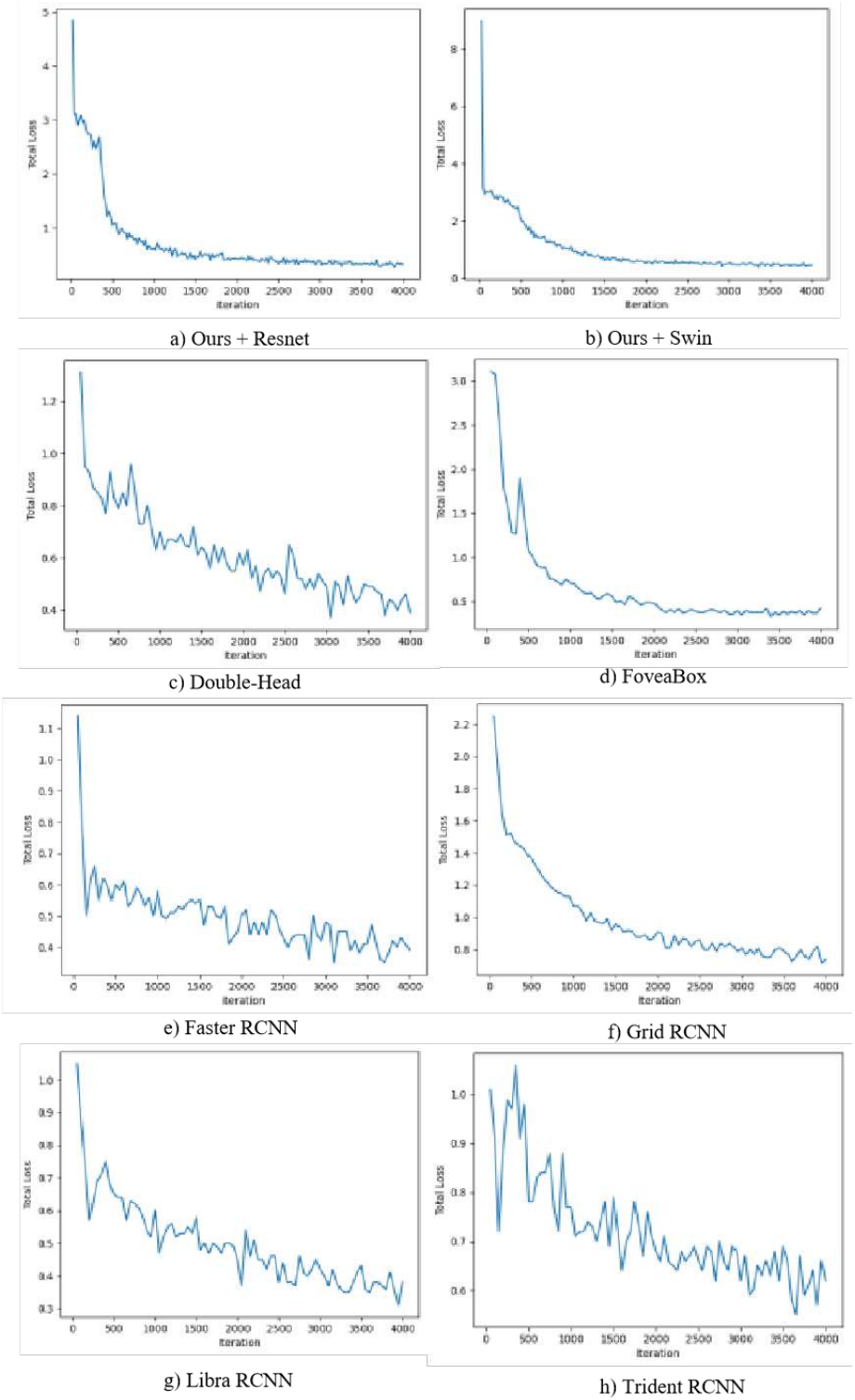


Fig 7. Total_Loss curves of algorithms on Gold Pear dataset.

4.3.2 Experiments on MinneApple

This study also evaluates the generalization performance of SOD Head using a publicly available dataset, MinneApple. The experimental results are shown in Table 5. Since more than 99%

of the fruits in the dataset are small objects, the values of AP_s and AP are similar under such circumstances.

Table 5. Comparison of algorithms on MinneApple.

	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_s (%)
Double-Head	37.8	76.6	32.9	36.6
FoveaBox	32.7	73.7	22.6	32.7
Faster RCNN	37.3	79.8	29.7	35.5
Grid RCNN	38.5	79.9	32.9	37.1
Libra RCNN	36.4	70.5	33.2	36.4
Trident	36.3	77.9	27.6	34.7
Ours + Resnet	39.6	80.6	34.7	39.6
Ours + Swin	37.2	79.6	29.8	37.3

From Table 5, it can be seen that the SOD Head has the highest AP_s, especially when combined with Resnet, achieving a detection accuracy of 39.6% for small-scale fruits, which is at least 2.5 percentage points higher than other algorithms. When using Swin Transformer as the backbone, the detection accuracy of SOD Head for small-scale fruits has slightly decreased to 37.3% (but still higher than AP_s of other algorithms). This is the opposite of the training results on the Gold Pear dataset. We speculate that this is due to the number of small-scale fruits in the dataset and the structural characteristics of the backbone network. Small-scale fruits account for 48.4% of the total fruits in the Gold Pear dataset, while there are more than 99% small-scale fruits in MinneApple. That is, there is almost no difference or division of scales in MinneApple. In addition, Swin

Transformer can handle longer-range dependencies, which may lead to better capture of small-scale fruit features when used as a backbone network on the Gold Pear dataset. On the other hand, Resnet has better shallow feature representation ability, which makes it better at handling more small-scale fruits on the MinneApple dataset. In addition, the detection accuracy of FoveaBox is low on both datasets, which may be due to its anchor generation method not being suitable for detecting small-scale objects.

The loss curves of the algorithms during the training process on the MinneApple dataset are shown in Figure 8.

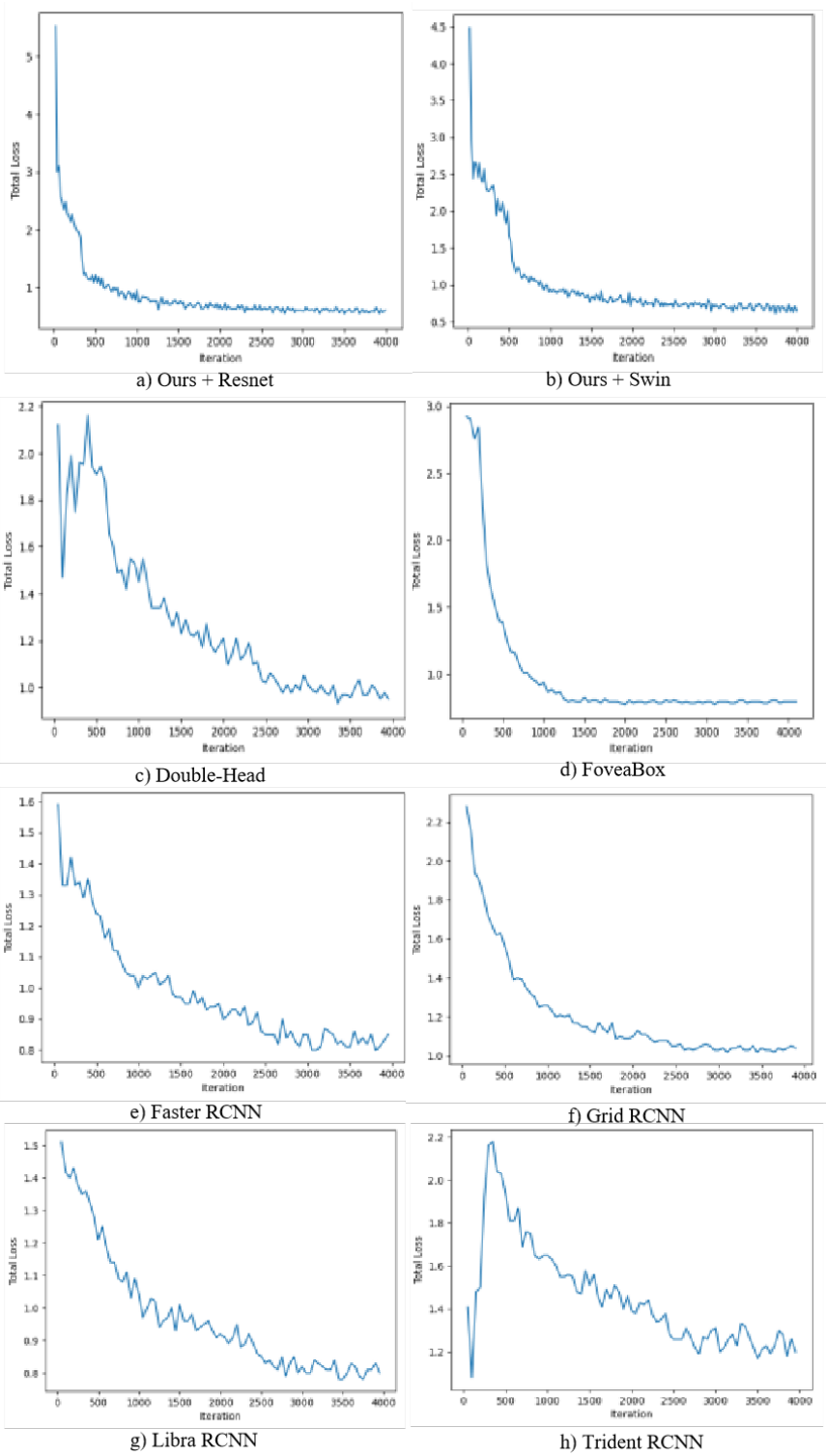


Fig 8. Total_Loss curves of algorithms on MinneApple.

From the loss curves, it can be seen that the SOD Head with Resnet as the backbone has a faster decrease in loss in the first 500 iterations. After 500 iterations, the convergence of the two backbones is comparable. Similar to the performance on the Gold Pear dataset, the oscillation

amplitude of Double-Head and Trident RCNN is still significant.

4.4 Results and Discussion

4.4.1 Visualization of Detection Results

In this section, we selected four images from two datasets to demonstrate the detection performance of the algorithm in real-world scenarios, as shown in Figures 9 and 11. As can be seen from the original image in Figure 9 a), during the early growth stage (before the thinning period) of Gold Pear, the size of the fruit is small. Moreover, the orchard is dense with branches and leaves, and the color and volume of the fruit are not conducive to detection. Most of the pixels in the captured images are redundant information such as branches and leaves, and the distribution of the fruit in the image is very sparse. To address this issue, SOD Head constructs a network that locates feature information layer by layer. It uses a mapping method that combines position and content to obtain the feature positions layer by layer from the topmost feature map, and then performs classification and regression on these positions. Moreover, with the help of box relocation operation, the regression process of small-scale bounding boxes can be more stable.

a) Original images

b) Ours

c) Double-Head

d) FoveaBox

e) Faster RCNN

f) Grid RCNN

g) Libra RCNN

h) Trident

Fig 9. Comparison of detection images on Gold Pear.

Figure 9 b) shows the detection results of our algorithm on the Gold Pear. It can be seen that SOD Head can accurately detect the target fruit with the help of the top-down localization network, and the predicted box scores are mostly above 0.56. Figure 10 shows the enlarged view of the detection results of SOD Head on the fruits that are difficult to detect in the images. However, other algorithms often suffer from missing or redundant detection. For example, Double-Head produces more than one redundant box in the fourth image of Figure 9 c). FoveaBox and Faster RCNN exhibit obvious missing detection in the second image of Figures 9 d) and e), respectively.

Fig 10. A detection image of SOD Head.

Figure 11 shows the detection results of our algorithm on MinneApple. The image contains fruits of two colors, red and green, with similar sizes. Due to the fact that the images of MinneApple image were captured from a distance, it contains more fruits and missed detection are more common among algorithms. Compared to other algorithms, SOD Head shows slightly better detection results. It produces fewer redundant boxes, and the predicted boxes have higher scores and are more accurate in terms of positioning. However, the issue of missed detection for certain types of fruits is discussed in the next section of this study.

a) Original images of MinneApple

b) Ours

c) Double-Head

d) FoveaBox

e) Faster RCNN

f) Grid RCNN

g) Libra RCNN

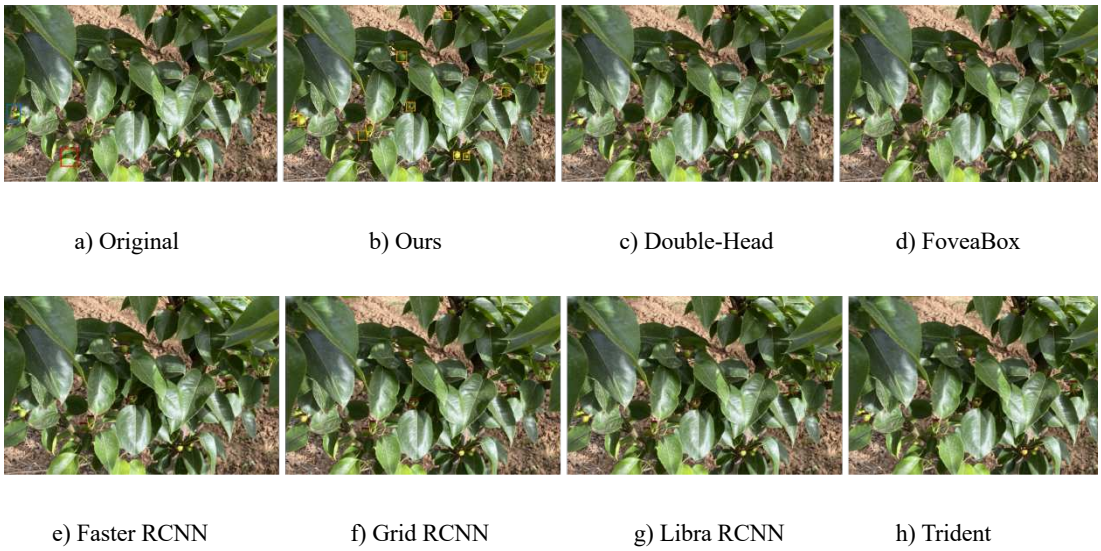
h) Trident

Fig 11. Comparison of detection images on MinneApple.

4.4.2 Discussion

To enhance the clarity of the detection results, we applied a zoomed-in view to the regions containing target fruits in the second image of Figure 9, as illustrated in Figure 12. The evaluation reveals that SOD Head demonstrates exceptional accuracy by detecting all 12 fruits in the image, outperforming other algorithms, which exhibit certain limitations. Specifically, Grid RCNN identifies 10 fruits, while both Libra RCNN and Trident only manage to detect 9 fruits. The most common omission among these algorithms is the fruit located on the far left of the image. This

particular fruit is obscured by leaves or overlapping with neighboring fruits, as indicated by the red box in Figure 12 a). Impressively, SOD Head successfully detects this obscured fruit, setting itself apart from the other methods.



a) Original b) Ours c) Double-Head d) FoveaBox

e) Faster RCNN f) Grid RCNN g) Libra RCNN h) Trident

Fig 12. Comparison of algorithms on enlarged part.

However, it is important to note that none of the algorithms, including our own, are able to detect the fruit in the lower-left corner of the image, as highlighted by the red box in Figure 12 a). This particular fruit poses a considerable challenge for detection due to its small size and the heavy occlusion caused by branches and leaves, making it significantly more difficult to identify. Detecting fruits under such occlusion conditions presents heightened difficulty compared to detecting fully exposed fruits. The obscured fruits often exhibit blurred edges, and SOD Head tends to interpret these features as redundant information when attempting to locate them. Therefore, detecting small fruits under occlusion conditions represents a promising direction for further algorithm optimization.

5. Conclusion

With the rapid development of deep learning, object detection technology has become increasingly mature and widely applied in various fields. In the field of smart orchards, this study

aims to overcome the problem of difficult detection caused by the small size of fruit in the early growth stage using object detection. This will help farmers monitor the growth status of fruit throughout the process, and achieve the goal of scientific yield measurement, fruit thinning guidance, and intelligent management of orchards. This study proposes a universal detection head specifically designed for small-scale objects, named SOD Head. It extracts all the features from the top-level feature map where semantic information is the richest via convolution, even though the information is blurred at this level. During the process of mapping these features down to the lower-level feature maps, the feature localization and refinement from top to bottom are achieved. This can reduce the adverse effects caused by information redundancy when detecting sparsely distributed small object features directly on high-resolution feature maps. In addition, the SOD Head also performs second-stage regression on the bounding boxes, learning a new set of parameters to make the prediction of small-scale object bounding boxes more stable. The experiments were conducted on two datasets of small-sized fruits. One is the dataset of Gold Pear made by us to simulate the working environment of SOD Head in orchards. The dataset of Gold Pear is used to evaluate performance of SOD Head in detecting small-sized fruits in a real orchard environment. The publicly available dataset of MinneApple was also used to demonstrate the generalization ability of SOD Head. The experimental results demonstrate that the SOD Head, as a universal detection head, achieves the highest detection accuracy for small-scale fruits, reaching 29.5% and 39.6% on the Gold Pear before thinning and the MinneApple, respectively. It has a certain competitive edge in detecting small-scale fruits in orchard environments and can meet the needs of intelligent management of orchards.

In addition to fruit size, occlusion and overlapping are also factors that affect fruit detection in real orchard environments. They exist throughout the entire growth cycle of fruits and in all sizes

of fruits. If the problem of fruit occlusion can be overcome, the accuracy of fruit detection in real orchard environments can be further improved. Although SOD Head can handle occlusion to a certain extent, how to address this issue in a targeted manner poses a new direction for our future research.

Acknowledgments

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

References

[1] Audu J, Aremu A K. Development, evaluation, and optimization of an automated device for quality detection and separation of cowpea seeds. Artificial Intelligence in Agriculture, 2021, 5: 240-251.

[2] Bochkovskiy A, Wang C, Liao H. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[3] Dollar P, Wojek C, Schiele B, et al. Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 34(4): 743-761.

[4] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[5] 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.

[6] Gao F, Fang W, Sun X, et al. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Computers and Electronics in Agriculture, 2022, 197: 107000.

[7] Ge Z, Liu S, Li Z, et al. Ota: Optimal transport assignment for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021b: 303-312.

[8] Ge Z, Wang J, Huang X, et al. LLA: Loss-aware label assignment for dense pedestrian detection. Neurocomputing, 2021a, 462: 272-281.

[9] Häni N, Roy P, Isler V. MinneApple: a benchmark dataset for apple detection and segmentation. IEEE Robotics and Automation Letters, 2020, 5(2): 852-858.

[10] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[11] 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, 6538117.

[12] Jia W, Zhang Y, Lian J, et al. Apple harvesting robot under information technology: A review[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 925310.

[13] Koirala A, Walsh K B, Wang Z, et al. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. Precision Agriculture, 2019, 20: 1107-1135.

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

[15] Li Y, Chen Y, Wang N, et al. Scale-aware trident networks for object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6054-6063.

[16] Lim J S, Astrid M, Yoon H J, et al. Small object detection using context and attention. International Conference on Artificial Intelligence in Information and Communication (ICAIIC). IEEE, 2021: 181-186.

[17] 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. 2017a: 2117-2125.

[18] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision. 2017b: 2980-2988.

[19] Liu L, Lu S, Zhong R, et al. Computing systems for autonomous driving: State of the art and challenges. IEEE Internet of Things Journal, 2020, 8(8): 6469-6486.

[20] 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.

[21] Liu Y, Sun P, Wergeles N, et al. A survey and performance evaluation of deep learning methods for small object detection. Expert Systems with Applications, 2021a, 172: 114602.

[22] 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. 2021b: 10012-10022.

[23] Lu X, Li B, Yue Y, et al. Grid r-cnn. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 7363-7372.

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

[25] Ngugi L C, 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.

[26] Pang J, Chen K, Shi J, et al. Libra r-cnn: Towards balanced learning for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 821-830.

[27] 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.

650 [28] Pareek C M, Tewari V K, Machavaram R, et al. Optimizing the seed-cell filling performance of an inclined
651 plate seed metering device using integrated ANN-PSO approach. Artificial Intelligence in Agriculture, 2021,
652 5: 1-12.
653 [29] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding
654 box regression. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:
655 658-666.
656 [30] Singh P, Verma A, Alex J S R. Disease and pest infection detection in coconut tree through deep learning
657 techniques. Computers and Electronics in Agriculture, 2021, 182: 105986.
658 [31] Su H, He Y, Jiang R, et al. DSLA: Dynamic smooth label assignment for efficient anchor-free object detection.
659 Pattern Recognition, 2022, 131: 108868.
660 [32] Sun M, Xu L, Chen X, et al. BFP net: balanced feature pyramid network for small apple detection in complex
661 orchard environment. Plant Phenomics, 2022a, 9892464.
662 [33] Sun M, Xu L, Luo R, et al. GHFormer-Net: Towards more accurate small green apple/begonia fruit detection
663 in the nighttime. Journal of King Saud University-Computer and Information Sciences, 2022b, 34(7): 4421-
664 4432.
665 [34] Sun Z, Li Q, Jin S, et al. Simultaneous prediction of wheat yield and grain protein content using multitask deep
666 learning from time-series proximal sensing. Plant Phenomics, 2022c, ID: 9757948.
667 [35] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection. Proceedings of the IEEE/CVF
668 Conference on Computer Vision and Pattern Recognition. 2020: 10781-10790.
669 [36] Tang Y, Qiu J, Zhang Y, et al. Optimization strategies of fruit detection to overcome the challenge of
670 unstructured background in field orchard environment: A review. Precision Agriculture, 2023a: 1-37.
671 [37] Tang Y, Zhou H, Wang H, et al. 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, 2023b, 211: 118573.

[38] Tesfaye A A, Osgood D, Aweke B G. Combining machine learning, space-time cloud restoration and phenology for farm-level wheat yield prediction. Artificial Intelligence in Agriculture, 2021, 5: 208-222.

[39] Tong K, Wu Y, Zhou F. Recent advances in small object detection based on deep learning: A review. Image and Vision Computing, 2020, 97: 103910.

[40] Wang J, Yang W, Guo H, et al. Tiny object detection in aerial images. 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 3791-3798.

[41] 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.

[42] Xu C, Wang J, Yang W, et al. RFLA: Gaussian receptive field based label assignment for tiny object detection. European Conference on Computer Vision, 2022: 526-543.

[43] Xu X, Zhao S, Xu C, et al. Intelligent mining road object detection based on multiscale feature fusion in multi-UAV networks. Drones, 2023, 7(4): 250.

[44] Yang C, Huang Z, Wang N. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 13668-13677.

[45] Zoph B, Cubuk E D, Ghiasi G, et al. Learning data augmentation strategies for object detection. European Conference on Computer Vision, 2020: 566-583.