

Article

APEIOU Integration for Enhanced YOLOV7: Achieving Efficient Plant Disease Detection

Yun Zhao, Chengqiang Lin, Na Wu and Xing Xu *

College of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; yunzhao@zust.edu.cn (Y.Z.); chengqianglin@zust.edu.cn (C.L.); nwu@zust.edu.cn (N.W.)

* Correspondence: xuxing@zust.edu.cn

Abstract: Plant diseases can severely hamper plant growth and yield. Currently, these diseases often manifest diverse symptoms, characterized by small targets and high quantities. However, existing algorithms inadequately address these challenges. Therefore, this paper proposes improving plant disease detection by enhancing a YOLOV7-based model. Initially, we strengthen multi-scale feature fusion using the fourth prediction layer. Subsequently, we reduce model parameters and the computational load with the DW-ELAN structure, followed by optimizing the downsampling process using the improved SPD-MP module. Additionally, we enhance the Soft-SimAM attention mechanism to prioritize crucial feature components and suppress irrelevant information. To distinguish overlapping predicted and actual bounding box centroids, we propose the APEIOU loss function and refine the offset formula and grid matching strategy, significantly increasing positive samples. We train the improved model using transfer learning. The experimental results show significant enhancements: the mAP, F1 score, Recall, and Precision are 96.75%, 0.94, 89.69%, and 97.64%, respectively. Compared to the original YOLOV7, the improvements are 5.79%, 7.00%, 9.43%, and 3.30%. The enhanced model outperforms the original, enabling the more precise detection of plant diseases.

Keywords: disease detection; YOLO v7; loss function; attention mechanism



Citation: Zhao, Y.; Lin, C.; Wu, N.; Xu, X. APEIOU Integration for Enhanced YOLOV7: Achieving Efficient Plant Disease Detection. *Agriculture* **2024**, *14*, 820. <https://doi.org/10.3390/agriculture14060820>

Academic Editor: David Guest

Received: 25 April 2024

Revised: 20 May 2024

Accepted: 23 May 2024

Published: 24 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Plants play a crucial role in global food security, but they often face diseases due to environmental factors, significantly impacting their growth and yield. Historically, manual inspection was relied upon to manage foliar diseases in plants, yet this approach is laborious and time-consuming. Determining disease types accurately and their locations at different stages is challenging and may lead to misdiagnosis or delayed prevention, causing losses. Scholars have turned to advanced technologies like deep learning for early disease detection, aiming to overcome these challenges. Before early convolutional neural networks emerged, studies used conventional methods such as decision trees and support vector machines. For instance, LIBS technology compared healthy and unhealthy citrus leaves, combined with quadratic discriminant analysis and support vector machine models, aiding in citrus orchard assessment [1]. Also, a method combining FZM and SVM has been proposed for grape leaf disease identification [2]. However, SVMs face challenges in handling large datasets and addressing issues like multiclass and class imbalance. Moreover, numerous scholars have deliberated on the applicability of formal methods in artificial intelligence applications, yielding significant insights [3,4].

With the advancement of convolutional neural networks (CNNs), they have been widely employed in plant disease detection tasks, transitioning from two-stage to one-stage development. Early two-stage approaches included R-CNN [5], Fast R-CNN [6], and Faster R-CNN [7], but they struggled to meet detection timeliness. One-stage algorithms offer advantages in detection speed, with prominent ones such as SSD, RetinaNet, and the YOLO series [8–17]. Building upon this foundation, researchers have conducted various

studies. For instance, to address the limitations of complex background environments and sparse features in apple leaf disease detection, some proposed an improved Faster R-CNN method integrating advanced Res2Net and feature pyramid network architectures for reliable multidimensional feature extraction [18]. Another group introduced MFaster R-CNN, employing a hybrid loss function constructed using a central cost function and four pre-trained structures to enhance maize disease detection in real-world environments [19]. Additionally, enhancements to the SSD algorithm led to the VMF-SSD method for more reliable multiscale feature representation in detecting apple leaf diseases [20]. Furthermore, researchers proposed the CAHA-AXRNet method, based on AX-RetinaNet, for rice disease detection using cross-augmented artificial bee colony optimization [21]. Improvements in various versions of YOLO are also ongoing. For example, ALAD-YOLO, based on YOLO-V5s, offers a precise lightweight model for apple leaf disease detection [22], while YOLO-Tobacco, an enhanced YOLOX-Tiny network, integrates hierarchical mixed-scale units to improve dense spot detection capability for tobacco brown spot disease in open-field scenarios [23].

Most of the improvements mentioned earlier focus on enhancing the overall detection accuracy or making models lightweight and deployable. However, there are also scholars who have made improvements in other directions. For instance, addressing the challenge of detecting small targets, which often have a lower detection accuracy compared to larger or medium-sized targets in object detection tasks. Some diseases manifest with small lesions in their early stages, making them difficult to identify, posing a significant challenge to detection tasks. In response to this challenge, some scholars have proposed a novel method called Multi-Scale Dense YOLO (MD-YOLO) [24]. This method is designed for detecting three typical small targets of Lepidopteran pests on sticky insect boards. Additionally, to address the detection of small and high-density pests in unstructured natural environments, researchers have proposed the SSV2-YOLO model [25]. This model reconstructs the backbone network using Stem and ShuffleNet V2 and adjusts the width of the neck network to achieve this goal. Additionally, researchers have significantly enhanced the detection effectiveness of various-sized tomato diseases in complex environments by proposing improved detection methods through data balancing [26]. Moreover, they have applied dual-path attention gate modules to agricultural tomato automatic detection methods, enhancing the detection of small targets [27].

On another front, the accurate localization of targets in object detection tasks relies on effective bounding box regression. Currently, there are two main categories of bounding box regression loss functions: those based on l_n -norm norms and those based on Intersection over Union (IOU). Initially, the former was widely used in bounding box regression due to its simplicity. However, it is highly sensitive to different scales. In YOLO V1 [10], the square root was employed to mitigate this issue, but the localization performance remained suboptimal. Subsequently, in YOLO V2 [11] and YOLO V3 [12], the authors continuously enriched the prior boxes to adapt to different sizes of detection targets. Later, in order to better measure the difference between the predicted boxes and ground-truth boxes, *IOU* was introduced [28]. To ensure stable loss computation, an upper bound was added to *IOU* later [29]. For deep learning models in object detection, scholars believe that metrics based on *IOU* are more suitable than those based on l_n -norm norms [30]. The original *IOU* represents the ratio of the intersection area of the predicted bounding box and the ground-truth bounding box to the union area. The *IOU* Loss can be represented by the following Formula (1):

$$L_{IOU} = 1 - IOU = 1 - \frac{B_{gt} \cap B_{prd}}{B_{gt} \cup B_{prd}} \quad (1)$$

The notation B_{gt} denotes the ground truth bounding box, while B_{prd} represents the predicted bounding box. However, its computation sometimes fails to reflect the actual overlap situation, and when the predicted box and the ground truth box do not intersect, i.e., when $IOU = 0$, a gradient vanishing issue arises, making it difficult to optimize the predicted box. Subsequently, *GIOU* was proposed [30], as shown in Formula (2) of the *GIOU* Loss.

It introduces a minimum enclosing rectangle C that encompasses both the predicted and ground truth bounding boxes. Here, C in the formula represents the minimum bounding rectangle area of the two boxes, resolving the issue of $IOU = 0$. However, $GIOU$ still has its drawbacks; when the predicted bounding box coincides with the ground truth bounding box, $GIOU$ degenerates to IOU and fails to accurately reflect the actual degree of overlap.

$$L_{GIOU} = 1 - IOU + \frac{|C - B \cup B^{gt}|}{|C|} \quad (2)$$

In contrast, in [31], $DIOU$ and $CIOU$ were introduced. $DIOU$ incorporates a center point penalty term based on the Euclidean distance to address the issue of indistinguishable loss between the predicted and ground truth boxes when they overlap in $GIOU$. The $DIOU$ Loss is formulated as shown in Formula (3):

$$L_{DIOU} = 1 - IOU + \frac{p^2(b, b^{gt})}{c^2} \quad (3)$$

The term $p^2(b, b^{gt})$ represents the square of the Euclidean distance between the center points of the predicted and ground truth bounding boxes, while c^2 denotes the square of the diagonal distance between the smallest enclosing intervals of the predicted and ground truth bounding boxes. Although $DIOU$ enables the rapid convergence and better representation of the overlap between the ground truth and predicted boxes, the absence of aspect ratio consideration may lead to situations where the $DIOU$ loss values remain identical when the center points of the ground truth and predicted boxes coincide, but their aspect ratios differ. Therefore, $CIOU$ incorporates three geometric factors: the intersection area, center point distance, and aspect ratio. The formula for $CIOU$ loss is as follows in Formulas (4) and (5):

$$L_{CIOU} = 1 - IOU + \frac{p^2(b, b^{gt})}{c^2} + \alpha v \quad (4)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2, \quad \alpha = \frac{v}{(1 - IOU) + v} \quad (5)$$

In the formula, w^{gt} and h^{gt} represent the width and height of the ground truth box, while w and h denote the width and height of the predicted box, respectively. α is a weight parameter. The introduction of $CIOU$ addresses the shortcomings of $DIOU$. However, in $CIOU$, it was observed that the gradients of w and h relative to v are a pair of opposites, indicating that w and h cannot simultaneously increase or decrease. Additionally, the term v in the $CIOU$ Loss formula reflects the difference in the aspect ratio rather than the differences in width and height with their respective confidences, which may sometimes impede the effective optimization of model similarity.

In recent years, $EIOU$ [32] has been proposed as an enhancement over $CIOU$. It decomposes the influence factor of the aspect ratio of the predicted and ground truth boxes separately, based on the penalty term in $CIOU$, to address its limitations. $EIOU$ Loss primarily comprises IOU loss, distance loss, and aspect ratio loss, as shown in Formula (6).

$$L_{EIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \frac{\rho^2(w, w^{gt})}{(w^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2} \quad (6)$$

Here, w^c and h^c represent the width and height of the minimum enclosing rectangle of the predicted and ground truth bounding boxes, respectively. $\rho^2(w, w^{gt})$ denotes the square of the Euclidean distance between the width of the predicted box and the width of the ground truth box, while $\rho^2(h, h^{gt})$ represents the square of the Euclidean distance between the height of the predicted box and the height of the ground truth box.

The main contributions of this paper can be summarized as follows:

1. We have enhanced the model structure of YOLO V7 to address the issue of increasing receptive fields during the downsampling process, which often leads to missing small objects. To mitigate this problem, we redesigned the fourth prediction head and proposed the SPD-MP structure to optimize the downsampling process in the backbone and neck sections. Additionally, we employed the lightweight DW-ELAN structure to achieve model lightweighting without significant performance loss.
2. We introduced the improved lightweight attention mechanism, Soft-SimAM, during the backbone and upsampling stages. This mechanism enhances focus on high-weight regions while suppressing low-weight regions through the introduction of a softening threshold. As a result, feature maps before entering the fusion process pay more attention to detailed features.
3. We proposed APEIOU loss and designed an auxiliary penalty term to address situations where the predicted box and the ground truth box have the same aspect ratio but different width and height values, potentially resulting in identical loss values. This innovation aids in achieving higher precision in box regression.
4. We significantly increased the maximum number of positive samples that the lead head in YOLO V7 can match by adding prediction layers and increasing the number of grid cells matched with ground truth boxes. This enhancement provides more choices for the subsequent SimOTA fine screening. Additionally, we investigated the impact of the dynamic_k parameter on the training speed during the positive sample screening process in SimOTA under different settings.

2. Materials and Methods

2.1. Materials

2.1.1. Image Dataset

The images used in this study are sourced from the publicly available datasets PlantVillage [33] and PlantDoc [34]. Most samples in the PlantVillage dataset were captured in laboratory settings. To increase the diversity of the data and enhance the robustness of model training, we introduced a portion of samples captured in natural backgrounds from the PlantDoc dataset. From these two datasets, we selected five common categories of plant diseases. For each category, we chose 200 images, totaling 811 images from PlantVillage and 189 images from PlantDoc. Subsequently, we manually annotated 1000 images using the LabelImg image annotation tool. To prevent issues such as overfitting or convergence difficulties due to limited training data, we further applied data augmentation techniques, including Gaussian noise, random brightness adjustment, mirroring, and motion blur, expanding the dataset to 5000 images.

The expanded dataset was divided into training, validation, and test sets in an 8:1:1 ratio. The categories and quantities of the labeled dataset are shown in Table 1.

Table 1. Type and number of diseases in the labeled datasets.

Type of Diseases	Number
Apple scab	1000
Corn leaf blight	1000
Grape black rot	1000
Potato late blight	1000
Tomato early blight	1000

2.1.2. Image Augmentation

During the training process, we primarily employed two methods of data augmentation. One is the commonly used Mixup data augmentation method [35], which involves blending two different images at a certain ratio to generate a new image. This approach enhances model robustness and reduces the risk of overfitting. The other method is the Mosaic data augmentation method proposed in YOLO V4 [13]. It involves randomly cropping four different images and then stitching them together to form a new image. This

technique increases the diversity of training data, enabling the model to adapt better to various environments. Both data augmentation methods are illustrated in Figure 1.

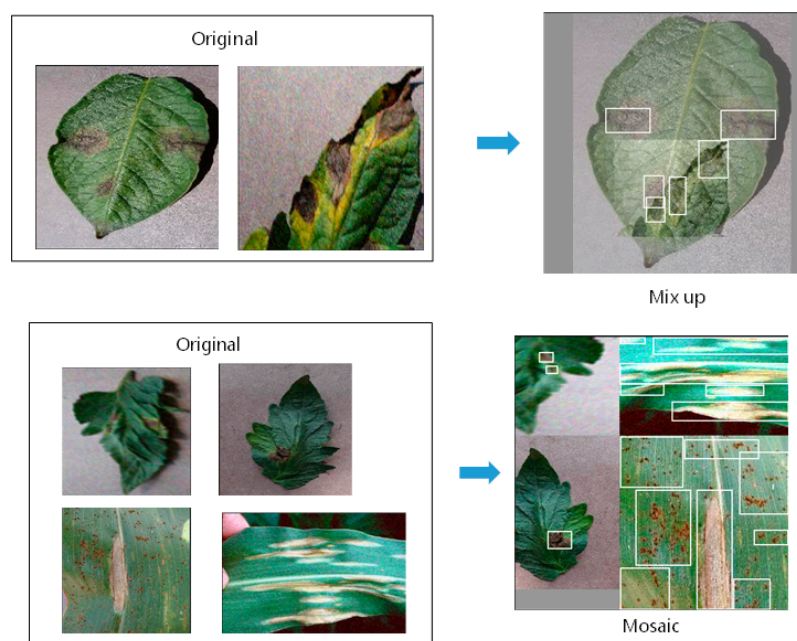


Figure 1. Data enhanced effect display.

While mosaic augmentation can enhance data diversity, it may also deviate from the true distribution of natural images. We tested the effect of using mosaic augmentation at different ratios on the improved model's mAP values, with intervals of 10%, as shown in Figure 2. From the graph, it can be observed that not using mosaic augmentation or applying it in all epochs yields poor results. This indicates the need to select an appropriate ratio for this augmentation method. The optimal outcome was achieved when using a 40% mosaic augmentation ratio. Therefore, in our experiments, mosaic augmentation was employed in the first 40% of training epochs and omitted in the remaining 60% of epochs.

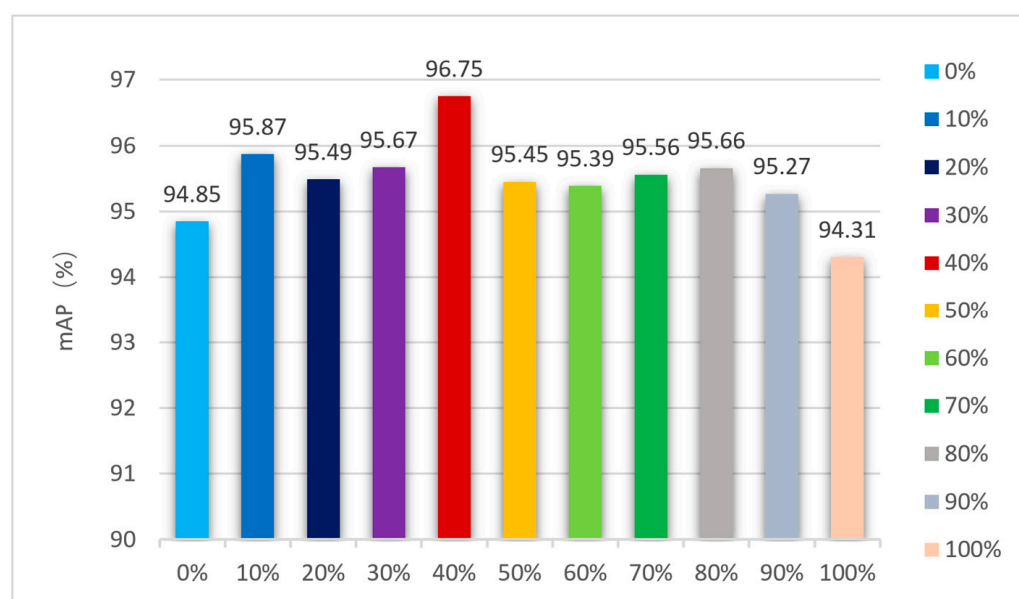


Figure 2. Different rates of mosaic data enhancement methods.

2.2. Methods

2.2.1. Improvement of the YOLO V7 Structure

In the task of plant disease detection, early-stage disease areas are relatively small and difficult to detect. Therefore, we consider how to design a framework that maintains high detection accuracy while also significantly enhancing small target detection. The original YOLO V7 consists of three classification and localization prediction heads. Each scale's features correspond to different feature map sizes. For detection tasks, multi-scale feature maps are crucial. In this paper, after preprocessing, the image size is set to 640×640 , and the feature map sizes of the classification and localization prediction layers are 80×80 , 40×40 , and 20×20 , respectively. Through multiple stages of downsampling, the feature maps continuously decrease in size, corresponding to larger receptive fields in the original image. However, at this point, they may easily lose feature information about small objects. Lower-level feature semantic information is not as rich as that of higher levels, but is more accurate for the position of small targets. Therefore, we add an additional prediction head based on this foundation, as illustrated in Figure 3. The four prediction heads simultaneously utilize the high resolution of low-level features and the rich semantic information of high-level features for multi-scale feature fusion and prediction.

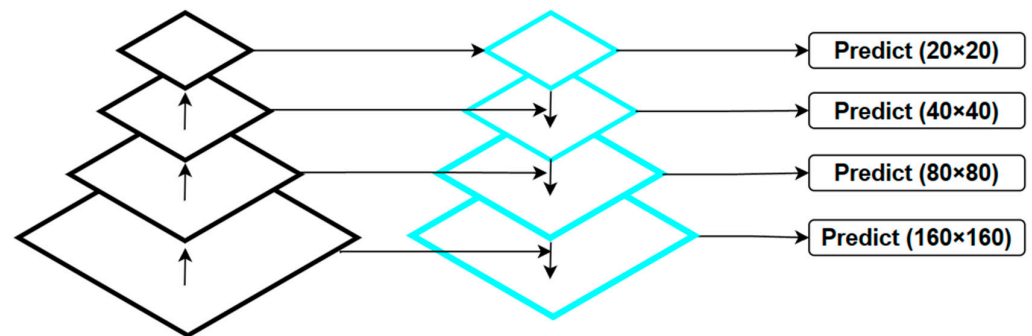


Figure 3. Prediction heads of different sizes.

The original YOLO V7 utilizes the MP module for downsampling in both the backbone and neck sections. It consists of two branches: one branch performs max-pooling and a CBS module. CBS comprises convolutional, BN, and SiLU activation layers. The other branch undergoes a 1×1 convolutional layer for channel adjustment, followed by a 3×3 convolutional block with a stride of 2. Finally, the results of both branches are concatenated to achieve downsampling. However, using convolutions or pooling with a stride greater than 1 can result in a loss of fine-grained information and suboptimal learning, particularly when there are many small objects in the detection scene [36]. Therefore, we employed the SPD-Conv to optimize downsampling in our structure, leading to the development of the new downsampling module, SPD-MP, as shown in Figure 4.

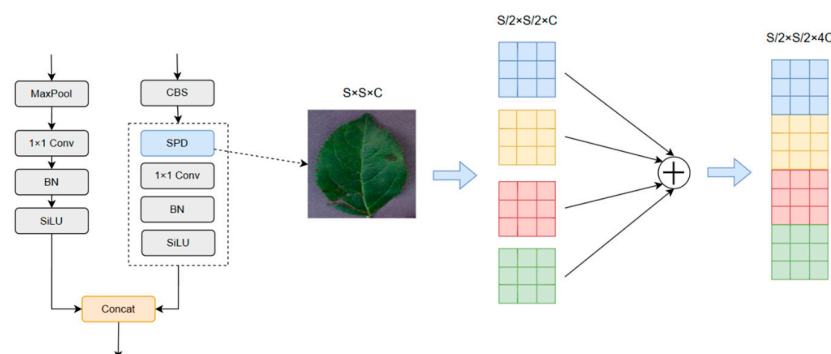


Figure 4. SPD-MP structure.

Specifically, the input feature map first passes through the SPD layer, which reduces the spatial dimensions of the input features to the channel dimension. During this process, the spatial dimensions decrease while the channel dimension increases. Then, a convolutional operation is applied. This combination can reduce the spatial dimensions without losing information and retain information within channels, thereby optimizing our model for the detection of small target diseases.

In order to ensure the lightweight and deployable nature of the model, we have streamlined the original ELAN architecture by incorporating depthwise separable convolutions in lieu of conventional convolutions. The parameter count associated with depthwise separable convolutions is significantly lower than that of standard convolutions, enabling a substantial reduction in both parameter quantity and computational workload, while only marginally sacrificing accuracy. The specific structure is illustrated in Figure 5. Such a design is capable of reducing the original computational workload by one-third.

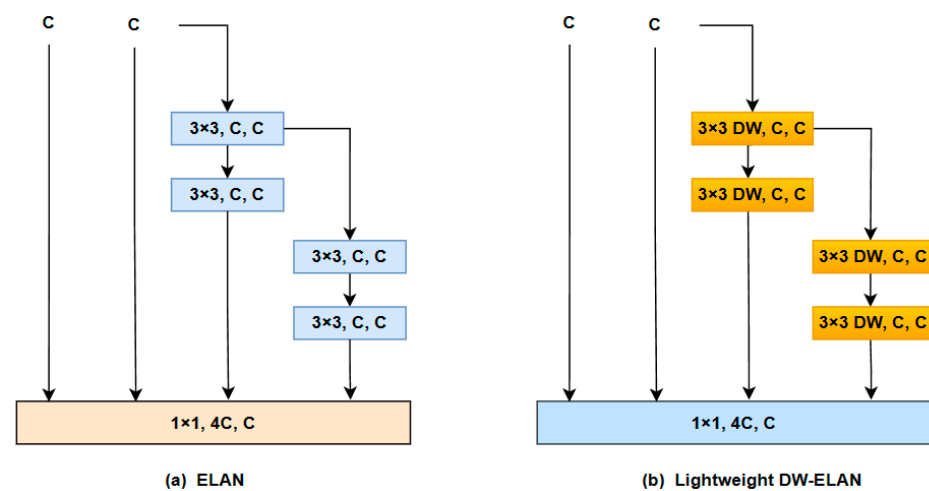


Figure 5. Different ELAN module designs.

2.2.2. Improvement of Attention Mechanism

Our task involves a plethora of detailed texture information that requires learning. Hence, employing attention mechanisms to enhance feature learning effectiveness is necessary. We also considered the positions where attention mechanisms should be applied. Since the image abstraction is low in shallow networks, we chose not to incorporate attention in the backbone section. As the network deepens, with more texture detail feature information present in the image, the obtained feature maps become increasingly abstract. Therefore, we opted to place the improved Soft-SimAM attention in the middle of the feature transfer from the backbone to the neck section, as illustrated in Figure 6. Additionally, we incorporated this attention during the upsampling process in the neck section. This structural arrangement facilitates a better fusion between high-dimensional and low-dimensional features, making the obtained feature maps more sensitive to detail information and thus enhancing the detection performance.

SimAM, recognized for its lightweight nature, was selected due to its minimal computational and parameter overhead. SimAM attention is derived through the formulation of an energy equation, generating corresponding weights for each neuron. Lower energy values indicate greater dissimilarity with surrounding neurons, thereby signifying increased importance for visual processing and warranting higher weights. Inspired by [37] and in order to further reinforce high-weight feature information while suppressing low-weight feature information, we have introduced the concept of a soft-thresholding mechanism. This enhancement, termed Soft-SimAM attention, is defined by Formula (7) as follows:

$$X^* = X \times \text{sigmoid}(\sigma \times E - \text{mean}(E)) \quad (7)$$

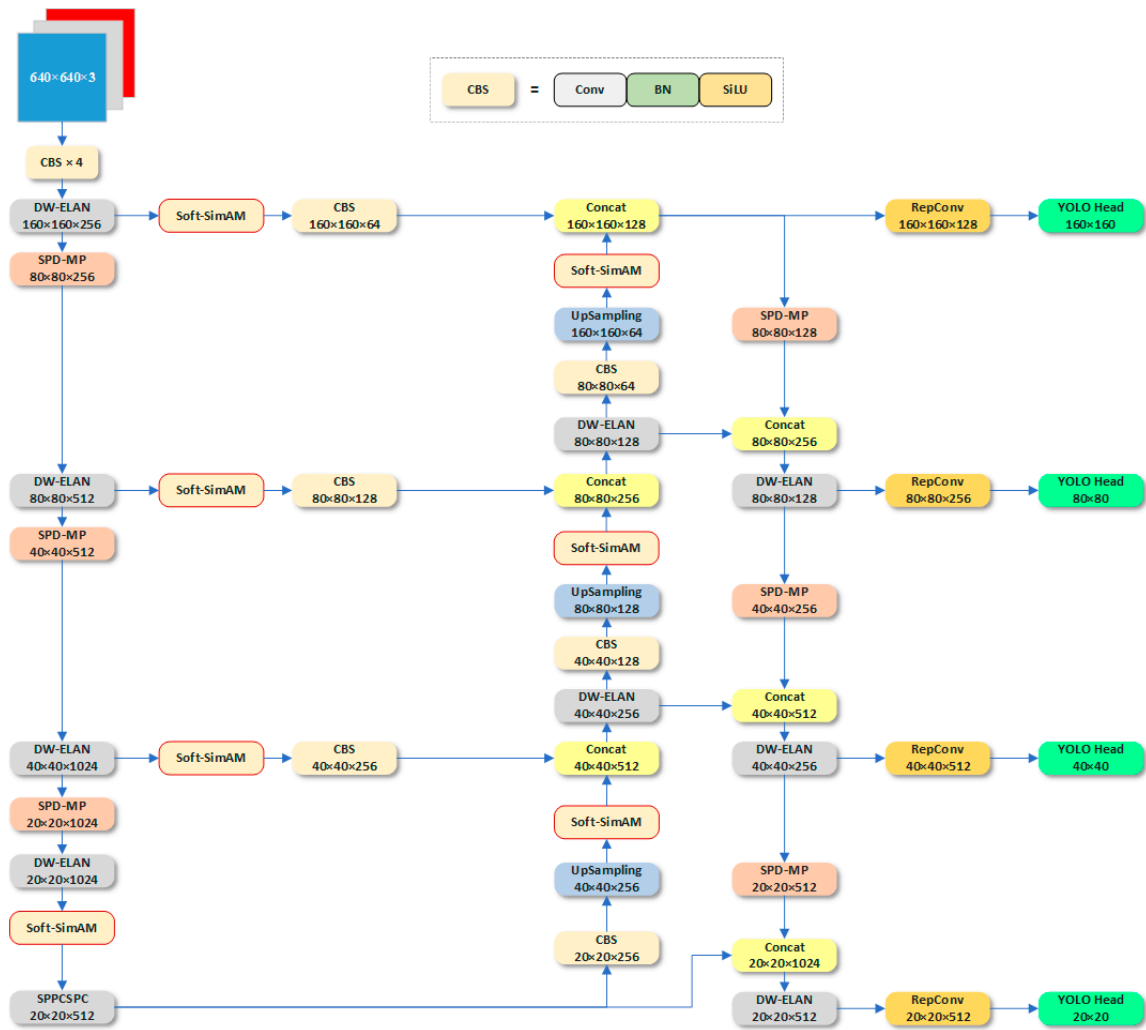


Figure 6. Model overall architecture.

The energy formula E , derived from [38], represents the importance of each neuron. The $mean(E)$ signifies the average weight of neurons, and by subtracting the mean weight, we center the weights around zero. The parameter σ in the soft-thresholding operation is initialized at 0.5 and optimized gradually by the optimizer. Ultimately, this module enhances attention to areas with higher neuron weights while suppressing areas with lower neuron weights, thereby preserving high-weight information and inhibiting low-weight information. X represents the input feature, and X^* denotes the feature after weighting. Incorporating a monotonically increasing smooth function such as the sigmoid() function maps the weight of each neuron to the range (0, 1), restricting excessively large input values.

2.2.3. Improvement of the Loss Function

The YOLO V7 loss comprises three parts: classification loss, confidence loss, and localization loss. Both confidence loss and classification loss are computed using Binary Cross-Entropy (BCE) Loss, as shown in the Formula (8), where y_i represents the true class, typically taking values of 0 or 1, and p_i denotes the corresponding predicted value.

$$BCE\ Loss = -\frac{1}{N} \times \sum_{i=1}^N [y_i \times \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (8)$$

The coordinate loss adopts the Formula (4) of CIoU loss. However, as previously mentioned, the issue with CIoU loss is the existence of parameter gradients being opposites, which may hinder precise coordinate regression. EIOU Loss, building upon the penalty term of CIoU Loss, separates the influence factors of the aspect ratio between the predicted and ground truth boxes, as shown in Formula (6). Calculating the lengths and widths of both predicted and ground truth boxes separately can assist in better fitting the predicted boxes to the ground truth ones. However, when the predicted and ground truth bounding boxes have the same aspect ratio but differ in width and height values, as depicted in Figure 7, where the green box represents the ground truth bounding box and the red box represents the predicted bounding box, the loss values of IOU Loss, GIOU Loss, DIOU Loss, and CIoU Loss in (a) and (b) are all 0.75, while the loss values of EIOU Loss in (a) and (b) are both 1.25. In other words, EIOU Loss cannot distinguish whose fitting is better at this moment, which will limit the convergence speed and accuracy. Therefore, we attempt to design a new loss function for bounding box regression that simultaneously achieves higher efficiency and accuracy in bounding box regression.

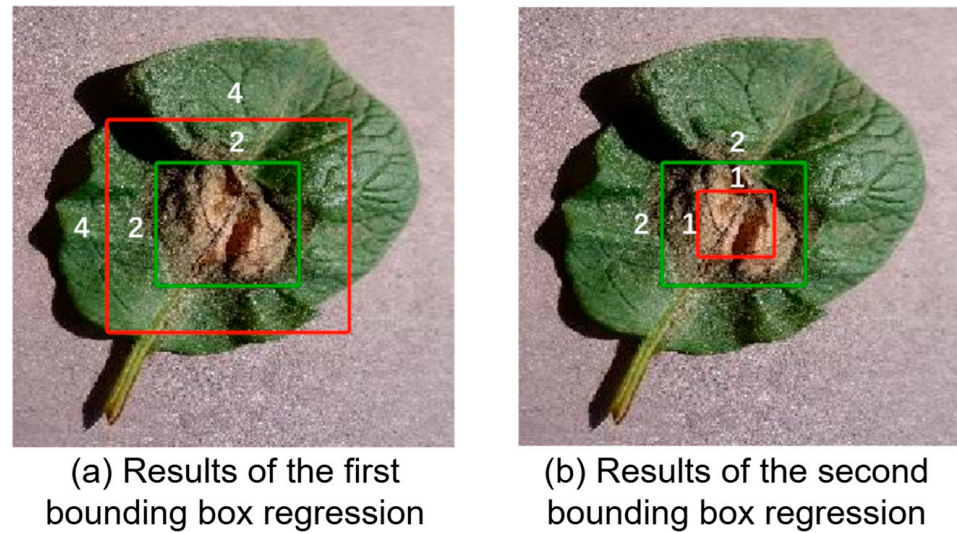


Figure 7. Two cases of different bounding boxes regression results.

Inspired by the geometric properties of bounding boxes, as mentioned in [39], where a unique rectangle can be defined by the coordinates of its top-left and bottom-right points, we propose incorporating a penalty term based on the distances between the top-left and bottom-right coordinates of the bounding box and the predicted box. Therefore, we introduce an additional “auxiliary penalty term” to the original EIOU loss, resulting in APEIOU loss (Auxiliary Penalty-Enhanced Efficient IOU Loss), defined as shown in Formulas (9) and (10):

$$L_{APEIOU} = 1 - IOU + \frac{p^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \frac{p^2(w, w^{gt})}{(w^c)^2} + \frac{p^2(h, h^{gt})}{(h^c)^2} + \frac{\sqrt{p^2(l, l^{gt}) + p^2(r, r^{gt})}}{1 + e^\varphi + p^2(b, b^{gt})} \quad (9)$$

$$\varphi = \frac{\sqrt{p^2(l, l^{gt}) + p^2(r, r^{gt})}}{(w^c)^2 + (h^c)^2} \quad (10)$$

The term $\frac{\sqrt{p^2(l, l^{gt}) + p^2(r, r^{gt})}}{1 + e^\varphi + p^2(b, b^{gt})}$ serves as our auxiliary penalty term. Here, $p^2(b, b^{gt})$ represents the squared Euclidean distance between the center points of the predicted and ground truth boxes, w^c and h^c denote the width and height of the minimum bounding rectangle of the predicted and ground truth boxes, respectively. $p^2(w, w^{gt})$ represents the squared Euclidean distance between the widths of the predicted and ground truth

boxes, while $p^2(h, h^{gt})$ represents the squared Euclidean distance between the heights of the predicted and ground truth boxes. $p^2(l, l^{gt})$ and $p^2(r, r^{gt})$ are the squared Euclidean distances between the top-left and bottom-right corners of the predicted and ground truth boxes, respectively.

As the distance between the predicted and ground truth boxes increases, the denominator's growth rate outpaces that of the numerator, causing the penalty term to tend towards 0. This ensures that the penalty term minimally impacts the loss value when the two boxes have no intersection. Moreover, as the center points of the predicted and ground truth boxes become closer, $p^2(b, b^{gt})$ tends towards 0. However, in cases where the center points coincide but the distances between the top-left and bottom-right corners of the predicted and ground truth boxes are large, the auxiliary penalty term assigns a larger loss value.

To demonstrate our loss function's ability to address the aforementioned issues, we generalize numerically, as depicted in Figure 8. In this illustration, the green box represents the ground truth box B_{gt} , with width w_{gt} and height h_{gt} , while the red boxes represent predicted bounding boxes with the same aspect ratio but different values. We define the larger red predicted box as B_{prd1} , with width $k \times w_{gt}$ and height $k \times h_{gt}$, and the smaller red predicted box as B_{prd2} , with width $\frac{1}{k} \times w_{gt}$ and height $\frac{1}{k} \times h_{gt}$, where $k > 1$ and $k \in \mathbb{R}$. In this figure, the center points of the predicted and ground truth boxes coincide.

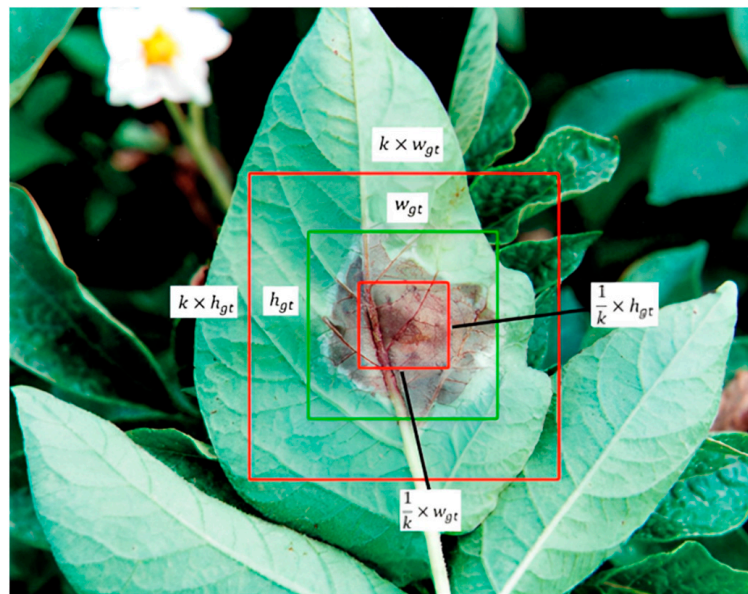


Figure 8. Examples of predicted bounding boxes and ground truth box with the same aspect ratio but different width and height.

$$\text{Proof. } \therefore \text{IOU}(B_{gt}, B_{prd1}) = \frac{w_{gt} \times h_{gt}}{w_{prd1} \times h_{prd1}} = \frac{w_{gt} \times h_{gt}}{k \times w_{gt} \times k \times h_{gt}} = \frac{1}{k^2},$$

$$\text{IOU}(B_{gt}, B_{prd2}) = \frac{w_{gt} \times h_{gt}}{w_{prd2} \times h_{prd2}} = \frac{\frac{1}{k} \times w_{gt} \times \frac{1}{k} \times h_{gt}}{w_{gt} \times h_{gt}} = \frac{1}{k^2}$$

$$\therefore \text{IOU}(B_{gt}, B_{prd1}) = \text{IOU}(B_{gt}, B_{prd2}) = \frac{1}{k^2}$$

$\therefore$ predicted bounding boxes coincide with the center of the ground truth bounding box

$$\therefore p^2(B_{prd1}, B^{gt}) = 0 \text{ and } p^2(B_{prd2}, B^{gt}) = 0$$

In the calculation of prediction box 1 versus the ground truth box :

$$p^2(w_{prd1}, w^{gt}) = (k \times w_{gt} - w_{gt})^2 \text{ and } (w^c)^2 = (k \times w_{gt})^2$$

$$p^2(h_{prd1}, h^{gt}) = (k \times h_{gt} - h_{gt})^2 \text{ and } (h^c)^2 = (k \times h_{gt})^2 \quad (11)$$

$$\begin{aligned}
\therefore EIOU(B_{gt}, B_{prd1}) &= IOU(B_{gt}, B_{prd1}) - \frac{p^2(B_{prd1}, B^{st})}{(w^c)^2 + (h^c)^2} - \frac{p^2(w_{prd1}, w^{st})}{(w^c)^2} - \frac{p^2(h_{prd1}, h^{st})}{(h^c)^2} \\
&= IOU(B_{gt}, B_{prd1}) - 0 - \frac{(k \times w_{gt} - w_{gt})^2}{(k \times w_{gt})^2} - \frac{(k \times h_{gt} - h_{gt})^2}{(k \times h_{gt})^2} \\
&= \frac{1}{k^2} - \frac{(k-1)^2}{k^2} - \frac{(k-1)^2}{k^2} = \frac{4k-2k^2-1}{k^2},
\end{aligned}$$

In the calculation of prediction box 2 versus the ground truth box:

$$p^2(w_{prd2}, w^{st}) = \left(w_{gt} - \frac{1}{k} \times w_{gt}\right)^2 = (w_{gt})^2 \times \left(1 - \frac{1}{k}\right)^2 \text{ and } (w^c)^2 = (w_{gt})^2$$

$$p^2(h_{prd2}, h^{st}) = \left(h_{gt} - \frac{1}{k} \times h_{gt}\right)^2 = (h_{gt})^2 \times \left(1 - \frac{1}{k}\right)^2 \text{ and } (h^c)^2 = (h_{gt})^2$$

$$\begin{aligned}
EIOU(B_{gt}, B_{prd2}) &= IOU(B_{gt}, B_{prd2}) - \frac{p^2(B_{prd2}, B^{st})}{(w^c)^2 + (h^c)^2} - \frac{p^2(w_{prd2}, w^{st})}{(w^c)^2} - \frac{p^2(h_{prd2}, h^{st})}{(h^c)^2} \\
&= IOU(B_{gt}, B_{prd2}) - 0 - \frac{(w_{gt})^2 \times (1 - \frac{1}{k})^2}{(w_{gt})^2} - \frac{(h_{gt})^2 \times (1 - \frac{1}{k})^2}{(h_{gt})^2} \\
&= \frac{1}{k^2} - \frac{(k-1)^2}{k^2} - \frac{(k-1)^2}{k^2} = \frac{4k-2k^2-1}{k^2}
\end{aligned}$$

$$\therefore EIOU(B_{gt}, B_{prd1}) = EIOU(B_{gt}, B_{prd2})$$

The following loss calculation is performed using the loss function proposed in this paper.

In the calculation of prediction box 1 versus the ground truth box:

$$p^2(B_{prd1}, B^{st}) = 0$$

$$p^2(l_{prd1}, l^{st}) = \frac{1}{4}(k-1)^2 \times (w_{gt}^2 + h_{gt}^2) \text{ and } p^2(r_{prd1}, r^{st}) = \frac{1}{4}(k-1)^2 \times (w_{gt}^2 + h_{gt}^2),$$

$$\begin{aligned}
\therefore APEIOU(B_{gt}, B_{prd1}) &= EIOU(B_{gt}, B_{prd1}) - \frac{\sqrt{p^2(l_{prd1}, l^{st}) + p^2(r_{prd1}, r^{st})}}{1 + e^{\varphi_1} + p^2(B_{prd1}, B^{st})} \\
&= EIOU(B_{gt}, B_{prd1}) - \frac{\sqrt{\frac{1}{2}(k-1)^2 \times (w_{gt}^2 + h_{gt}^2)}}{1 + e^{\varphi_1} + 0} \\
&= EIOU(B_{gt}, B_{prd1}) - \frac{\frac{\sqrt{2}}{2}(k-1)\sqrt{w_{gt}^2 + h_{gt}^2}}{1 + e^{\varphi_1}} \\
\text{where } \varphi_1 &= \frac{\sqrt{p^2(l_{prd1}, l^{st}) + p^2(r_{prd1}, r^{st})}}{(w^c)^2 + (h^c)^2} = \frac{\sqrt{\frac{1}{2}(k-1)^2 \times (w_{gt}^2 + h_{gt}^2)}}{(k \times w_{gt})^2 + (k \times h_{gt})^2} \\
&= \frac{\frac{\sqrt{2}}{2}(k-1)\sqrt{w_{gt}^2 + h_{gt}^2}}{k^2 \times (w_{gt}^2 + h_{gt}^2)} = \frac{\sqrt{2}}{2} \times \frac{k-1}{k^2 \times \sqrt{w_{gt}^2 + h_{gt}^2}}
\end{aligned}$$

In the calculation of prediction box 2 versus the ground truth box:

$$p^2(B_{prd2}, B^{st}) = 0$$

$$p^2(l_{prd2}, l^{st}) = \frac{1}{4}\left(1 - \frac{1}{k}\right)^2 \times (w_{gt}^2 + h_{gt}^2) \text{ and } p^2(r_{prd2}, r^{st}) = \frac{1}{4}\left(1 - \frac{1}{k}\right)^2 \times (w_{gt}^2 + h_{gt}^2),$$

$$\begin{aligned}
APEIOU(B_{gt}, B_{prd2}) &= EIOU(B_{gt}, B_{prd2}) - \frac{\sqrt{p^2(l_{prd2}, l_{gt}^t) + p^2(r_{prd2}, r_{gt}^t)}}{1 + e^{\varphi_2} + p^2(B_{prd2}, B_{gt}^t)} \\
&= EIOU(B_{gt}, B_{prd2}) - \frac{\sqrt{\frac{1}{2}(1-\frac{1}{k})^2(w_{gt}^t + h_{gt}^t)^2}}{1 + e^{\varphi_2} + 0} \\
&= EIOU(B_{gt}, B_{prd2}) - \frac{\frac{\sqrt{2}}{2}(1-\frac{1}{k})\sqrt{w_{gt}^t + h_{gt}^t}}{1 + e^{\varphi_2}} \\
&= EIOU(B_{gt}, B_{prd2}) - \frac{\frac{\sqrt{2}}{2}(k-1)\sqrt{w_{gt}^t + h_{gt}^t}}{k(1 + e^{\varphi_2})}
\end{aligned}$$

where $\varphi_2 = \frac{\sqrt{p^2(l_{prd2}, l_{gt}^t) + p^2(r_{prd2}, r_{gt}^t)}}{(w_{gt}^t)^2 + (h_{gt}^t)^2} = \frac{\sqrt{\frac{1}{2}(1-\frac{1}{k})^2(w_{gt}^t + h_{gt}^t)^2}}{(w_{gt}^t)^2 + (h_{gt}^t)^2}$

$$\begin{aligned}
&= \frac{\frac{\sqrt{2}}{2}(1-\frac{1}{k})\sqrt{w_{gt}^t + h_{gt}^t}}{w_{gt}^t + h_{gt}^t} = \frac{\sqrt{2}}{2} \times \frac{k-1}{k \times \sqrt{w_{gt}^t + h_{gt}^t}} \\
&\because k > 1 \text{ and } k \in \mathbb{R} \\
&\therefore \varphi_1 < \varphi_2 \text{ and } 0 < \frac{1}{k} < 1 \\
&\therefore y = \frac{1}{1+e^x} \text{ is a monotonically decreasing function} \\
&\therefore \frac{1}{1+e^{\varphi_1}} > \frac{1}{1+e^{\varphi_2}} \\
&\therefore \frac{1}{1+e^{\varphi_1}} > \frac{1}{k(1+e^{\varphi_2})} \\
&\therefore APEIOU(B_{gt}, B_{prd1}) < APEIOU(B_{gt}, B_{prd2}) \\
&\therefore L_{APEIOU(B_{gt}, B_{prd1})} = 1 - APEIOU(B_{gt}, B_{prd1}), L_{APEIOU(B_{gt}, B_{prd2})} = \\
&1 - APEIOU(B_{gt}, B_{prd2}) \\
&\therefore L_{APEIOU(B_{gt}, B_{prd1})} > L_{APEIOU(B_{gt}, B_{prd2})}. \square
\end{aligned}$$

Through the aforementioned demonstration, we deduce that under our APEIOU formulation, it is possible to successfully distinguish cases where the center points of the predicted and ground truth boxes coincide, as well as cases where two predicted boxes have the same aspect ratio, but differ in width and height values, resulting in identical losses. Furthermore, our added auxiliary penalty term function ensures that L_{APEIOU} remains non-negative.

2.2.4. Improved Grid Matching Strategy

In the YOLO series, the quantity and quality of positive samples during detection are crucial factors that determine the effectiveness of model training. YOLO V7, being anchor-based, adopts a positive sample matching strategy that combines elements from YOLO V5 and YOLO X. Initially, it employs the positive sample filtering strategy from YOLO V5, followed by the use of the SimOTA strategy from YOLO X for fine-grained positive sample screening. The YOLO V7 comprises both a Lead head and an Aux head. The Aux head is designed to assist the network in learning additional information and, according to the original text, places greater emphasis on the recall metric, making its selection of positive samples more lenient. YOLO V7 consists of three prediction layers, each equipped with three anchors. Furthermore, each ground truth (GT) can be matched with three grids. Thus, in the original lead head of the YOLO V7, each GT can potentially be matched with up to 27 positive samples. The original YOLO V7 employs offset calculation formulas for prediction boxes, as described in Formulas (11) and (12):

$$b_x = 2 \times \sigma(t_x) - 0.5 + c_x \quad (12)$$

$$b_y = 2 \times \sigma(t_y) - 0.5 + c_y \quad (13)$$

In the YOLO V7 Lead Head, b_x and b_y represent the predicted box's center position, and $\sigma(\cdot)$ denotes the Sigmoid function. The use of the Sigmoid function aims to confine

the offset within the range of 0 to 1, preventing the predicted center point from exceeding the grid's responsible range due to excessively large offsets. t_x and t_y represent the offsets of the target center x and y relative to the top-left corner of the grid. c_x and c_y are the x and y coordinates of the corresponding grid's top-left corner. When the true target center falls within a grid, the YOLO V7 Lead head first determines in which quadrant of the grid the point lies. It then considers the two adjacent grids as matching grids, as depicted in Figure 9A. If the presumed true target center falls within the first quadrant of the grid, besides the current grid, the Lead head also considers the grid above and to the right as positive samples, resulting in three gray grids being treated as positive samples. In contrast, the Aux head's selection of positive samples is more lenient. It selects both the gray and red grids as matching grids because its main purpose is to assist the Lead head in learning more information, focusing on recall.

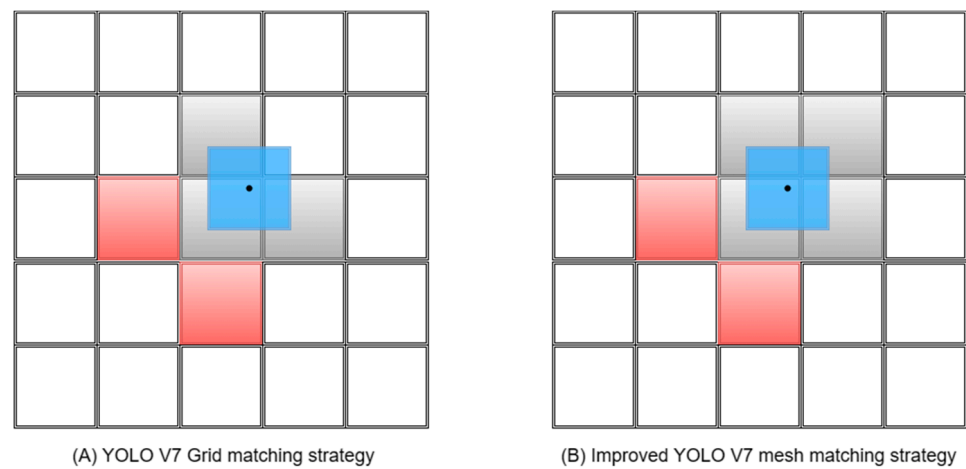


Figure 9. Yolo V7 positive sample matching strategy: (A) represents the positive sample matching strategy of YOLO V7, and (B) represents the positive sample matching strategy of YOLO V7 after the improvement in this paper.

In [13], the authors previously mentioned that the offset formulas designed for YOLO V2 and V3 initially struggled to attain extreme values. Similarly, this issue arises here. Taking Figure 9A as an example, according to Formulas (11) and (12), we can derive that b_x falls within the range of $(-0.5 + c_x, 1.5 + c_x)$, and b_y falls within the range of $(-0.5 + c_y, 1.5 + c_y)$. However, in practice, when predicting the target using the grid above, it requires $b_y = 1.5 + c_y$, implying that $\sigma(t_y) = 1$, indicating that t_y tends towards positive infinity. Similarly, when predicting the target using the grid to the right, it requires $b_x = -0.5 + c_x$, indicating that $\sigma(t_x) = 0$, implying that t_x tends towards negative infinity. These scenarios represent the boundary values of the grid. In practice, it is challenging for the network training to achieve these extreme values. Therefore, to mitigate the sensitivity of the grid, we appropriately scale and restrict the formula. The modified formulas are presented as follows in Formulas (13)–(18):

$$x_1 = 2 \times \sigma(t_x) - 0.5 \quad (14)$$

$$x_2 = 2.2 \times \sigma(t_x) - 0.6 \quad (15)$$

$$y_1 = 2 \times \sigma(t_y) - 0.5 \quad (16)$$

$$y_2 = 2.2 \times \sigma(t_y) - 0.6 \quad (17)$$

$$b_x = \begin{cases} -0.5 + c_x, & \frac{x_1+x_2}{2} < -0.5 \\ \frac{x_1+x_2}{2} + c_x, & -0.5 \leq \frac{x_1+x_2}{2} \leq 1.5 \\ 1.5 + c_x, & \frac{x_1+x_2}{2} > 1.5 \end{cases} \quad (18)$$

$$b_y = \begin{cases} -0.5 + c_y, & \frac{y_1+y_2}{2} < -0.5 \\ \frac{y_1+y_2}{2} + c_y, & -0.5 \leq \frac{y_1+y_2}{2} \leq 1.5 \\ 1.5 + c_y, & \frac{y_1+y_2}{2} > 1.5 \end{cases} \quad (19)$$

To address the sensitivity issue of boundary values, we calculate the ranges of x_1 , x_2 , y_1 , and y_2 separately to obtain different scaling degrees. Then, we obtain $(x_1 + x_2)/2$ and $(y_1 + y_2)/2$ by averaging these values. After scaling and averaging, the range of $(x_1 + x_2)/2 + c_x$ falls within $(-0.55 + c_x, 1.55 + c_x)$, and the range of $(y_1 + y_2)/2 + c_y$ falls within $(-0.55 + c_y, 1.55 + c_y)$. This approach helps mitigate the sensitivity issue of boundary values.

To ensure that the original ranges of b_x and b_y are not affected, we further restrict them using piecewise functions. Our optimized bounding box offset formula helps the prediction box fit more accurately to the ground truth box. Moreover, by increasing the number of positive samples matched by the Lead head, we significantly enhance the maximum number of positive samples each ground truth can be matched with. As shown in Figure 9B, we assign an additional neighboring grid to be responsible for prediction. Additionally, the improved YOLO V7 now comprises four prediction layers, each equipped with three anchors. Furthermore, each ground truth can be matched with four adjacent grids. Thus, the Lead head can now match up to 48 positive samples for each ground truth, significantly increasing the number of positive samples during training and providing more choices for the fine-grained positive sample screening in the second step of the SimOTA strategy [14].

3. Results

3.1. Equipment Setting

3.1.1. Equipment and Experiment Parameter

The experiments were conducted on the following hardware configuration: Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz, 128 GB memory, NVIDIA GeForce RTX 2080 Ti graphics card, operating system Windows 10. The software used includes PyCharm 2021, Python 3.8, and PyTorch 1.10.1.

For model training, we utilized the SGD optimizer, known for its good convergence properties. To expedite the learning process, we set the momentum parameter to 0.937 and the initial learning rate to 0.01. However, as we progressed towards finding more precise, optimal solutions, we gradually adjusted the learning rate. The minimum learning rate was not set lower than 0.0001.

The training images were resized to 640×640 pixels. The training process lasted for 300 epochs, and to accelerate model convergence, transfer learning was employed. The batch size was set to 8.

3.1.2. Evaluation Metrics

In order to better evaluate the performance of the model detection, we utilized several metrics including the *mAP*, average precision (*AP*), *precision*, *recall*, *F1 score*, network parameters quantity, computational complexity, and detection speed. We set the IoU threshold to 0.5. The formulas for calculating the *AP*, *mAP*, *precision*, *recall*, and *F1 score* correspond to Formulas (19)–(23), respectively:

$$AP = \int_0^1 P(R) dR \quad (20)$$

$$mAP = \frac{\sum_{i=1}^Q AP_i}{Q} \quad (21)$$

$$precision = \frac{TP}{TP + FP} \quad (22)$$

$$recall = \frac{TP}{TP + FN} \quad (23)$$

$$F1score = \frac{2 \times precision \times recall}{precision + recall} \quad (24)$$

where TP represents the number of plant diseases correctly detected by the model, FP represents the number of plant diseases incorrectly detected by the model, and FN represents the number of plant diseases not detected by the network. The precision metric measures the proportion of correctly detected positive samples out of all samples detected as positive, while recall measures the proportion of truly positive samples that are correctly detected. Under each plant disease detection scenario, we can plot a Precision–Recall (P-R) curve based on these two metrics. Average precision (AP) is the area under the P-R curve, with values closer to 1 indicating better model performance. To balance precision and recall, we additionally utilize the F1 score, which represents the weighted average of precision and recall. The mAP is the average of AP across multiple categories, making it the most commonly used evaluation metric in object detection. In the formula, Q represents the number of categories in the dataset. FPS is used to evaluate the model's detection speed, where a higher FPS indicates higher detection efficiency.

3.2. Comparative Experiments and Deployment

3.2.1. Comparison of Different Attention Mechanisms

The algorithm proposed in this paper is based on an improved YOLO V7 model, which includes enhancements to the “neck” and “backbone” parts of the YOLO V7 model through the MP module. We introduced an SPD-MP downsampling module that reduces feature loss and benefits the detection of small targets. Additionally, we redesigned the fourth prediction head to improve the detection of small targets, thereby enhancing the overall detection performance. We employed our improved Soft-SimAM attention mechanism to enhance the focus on fine-grained features. To validate its effectiveness, we tested the effects of incorporating different attention mechanisms [40–44] solely into the original YOLO V7 model. The results, as shown in Table 2, indicate that both ECA attention and SimAM attention are lightweight attention mechanisms, with SimAM attention demonstrating a superior performance. In contrast, SE attention, CBAM attention, CA attention, and GAMA attention introduce additional parameters and slightly lower detection performances compared to SimAM attention. Furthermore, by setting a softening threshold, Soft-SimAM attention allows the model to focus more on important fine-grained details while maintaining lightweight characteristics, thereby increasing the mAP value.

Table 2. Comparative effect of different attention.

Different Attention	mAP (%)	mAP (%) Small Obj.	Params (M)	FLOPs (G)
SE	92.97	47.3	37.52	104.84
CBAM	92.39	46.7	37.82	104.85
ECA	92.61	47.0	37.21	104.84
CA	92.93	45.7	37.67	104.87
GAMA	93.24	46.5	37.22	104.85
SimAM	93.00	47.2	37.21	104.83
Soft-SimAM (our)	93.59	47.9	37.21	104.83

3.2.2. Comparative Experiments of Different Loss Functions

After completing both overall model improvements and enhancements to the attention mechanism, we conducted comparative experiments using several classical loss functions in detection along with the loss function proposed in this paper. The experimental results, as depicted in Figure 10, demonstrate the performance of the improved loss function.

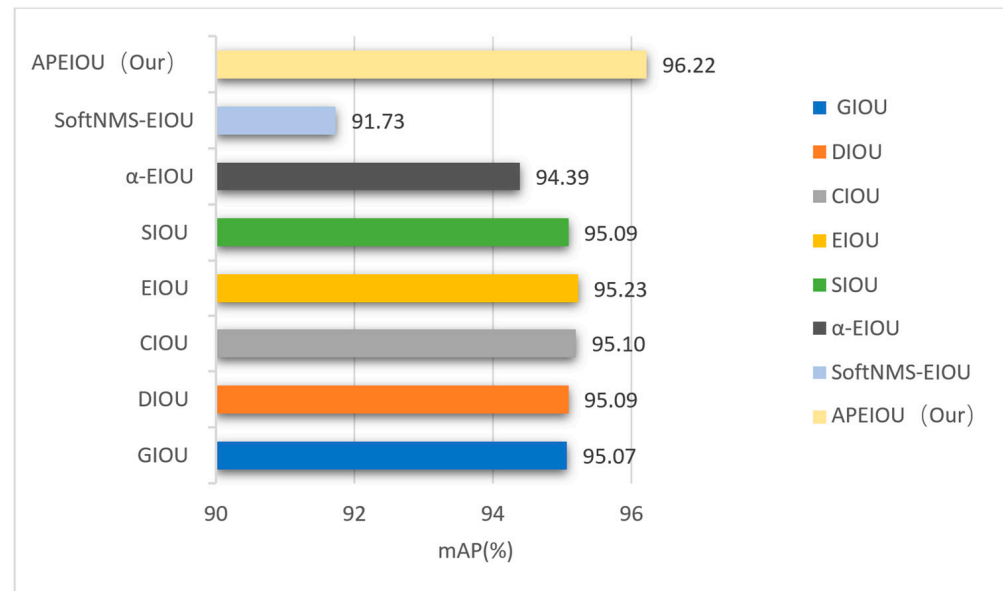


Figure 10. Comparison effect of different loss functions.

In addition to the EIOU loss, we incorporated SoftNMS, a soft form of non-maximum suppression, to retain candidate boxes with high overlap but low confidence scores. We also applied a unified formulation of α -IOU (with $\alpha = 3$) to strengthen the learning weights for challenging samples. However, we observed that these adjustments did not yield significant improvements. In contrast, the proposed improved loss function achieved a higher mAP value, increasing by approximately 1% compared to other loss functions. This outcome suggests that the inclusion of auxiliary penalty terms is beneficial for enhancing the alignment between predicted bounding boxes and real bounding boxes.

3.2.3. Comparative Experiments of Each Improvement Part

To validate the effectiveness of each improvement, we partitioned the augmented dataset into training, validation, and test sets. We then trained the model separately by incorporating each improvement. The comparative results between each improvement and the original model are illustrated in Figure 11.

In the Figure 11, A represents the YOLO V7 model with the addition of the fourth prediction head, the improved SPD-MP module, and the lightweight DW-ELAN module. B represents the YOLO V7 model incorporating the Soft-SimAM attention mechanism. C represents the YOLO V7 model utilizing the improved loss function proposed in this paper. D represents the YOLO V7 model employing the improved grid matching strategy proposed in this paper. “Ours” represents the final integrated effect after combining all four components. It can be observed that each part of the original model improved mAP values compared to the baseline YOLO V7, confirming the effectiveness of each improvement. Further testing revealed that the final improved model achieved a 5.79% increase in mAP compared to the original YOLO V7. Here mAP refers to the average AP value for all categories with a set IoU threshold of 0.5.

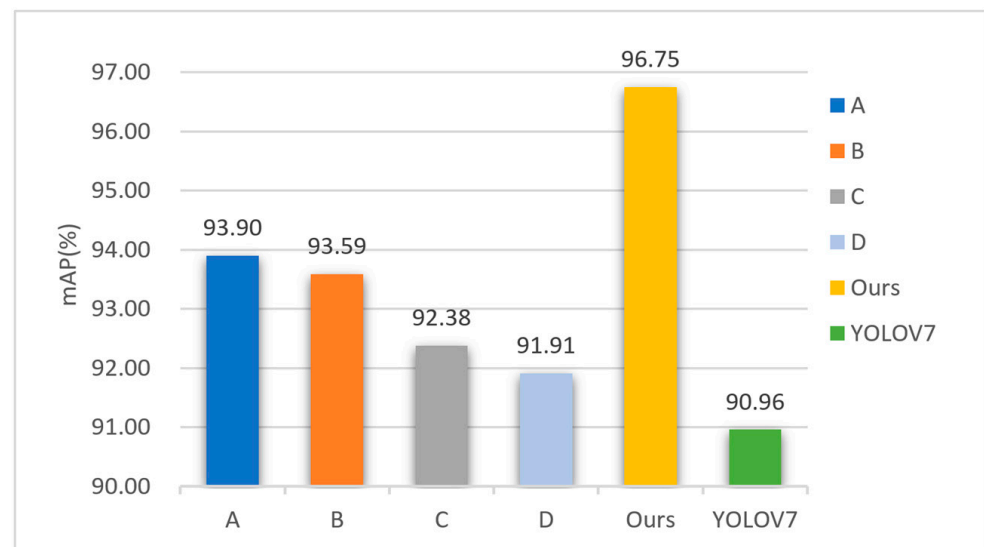


Figure 11. Comparison results of each improved part of the model.

Further, we recorded the specific performance of each improvement, as shown in Table 3. Here, we found that the highest improvement in the overall accuracy and accuracy for small targets came from the structural enhancement of our model. Additionally, we evaluated the FPS for each improvement on our device, all of which yielded around 30 FPS. After incorporating Soft-SimAM attention and the improved loss function, the FPS of our model remained similar to that of the original YOLO V7. However, when combining all improvements, there was a slight decrease in FPS, indicating that a reduction in parameters and computational complexity does not necessarily lead to an increase in FPS. Nevertheless, the overall and small target detection accuracy increased by 5.79% and 10.1%, respectively. Thus, we achieved a higher detection performance at a relatively low cost.

Table 3. The index comparison results of each improved part of the model.

Each Improvement Part	mAP (%)	mAP (%) Small Obj.	Params (M)	FLOPs (G)	FPS
YOLO V7	90.96	48.0	37.21	104.83	30.6
YOLO V7_A	93.90	53.1	26.35	74.24	30.1
YOLO V7_B	93.59	52.7	37.21	104.83	30.2
YOLO V7_C	92.38	51.7	37.21	104.83	30.8
YOLO V7_D	91.91	51.2	37.21	104.83	29.5
Improved-YOLO V7	96.75	58.1	26.35	74.24	28.8

The dataset used in this study consists of five disease categories. The experimental results comparing the improved model with the original model for each disease category are presented in Table 4. It can be observed that our model showed significant improvements in AP for each disease category. Specifically, Apple scab, Corn leaf blight, Grape black rot, Potato late blight, and Tomato early blight saw improvements in AP by 7%, 10.89%, 6.32%, 1.88%, and 2.85%, respectively. Upon closer examination, we found that the increase in Precision was generally greater than the increase in Recall. This suggests that our model is more inclined to detect more targets, confirming that the improvements in our model make it more sensitive to targets that may not have been detected previously, thereby improving recall. Consequently, the overall detection performance is also enhanced.

Table 4. Comparative results of five disease types.

Type of Diseases	Model	AP(%)	F1 Score	Recall (%)	Precision (%)
Apple scab	YOLO V7	89.12	0.84	79.96	91.31
	Improved-YOLO V7	96.12	0.92	87.68	95.9
Corn leaf blight	YOLO V7	84.59	0.76	64.76	91.9
	Improved-YOLO V7	95.48	0.91	85.36	97.18
Grape black rot	YOLO V7	92.3	0.9	85.32	95.23
	Improved-YOLO V7	98.62	0.97	95.35	97.71
Potato late blight	YOLO V7	96.43	0.96	94.32	97.3
	Improved-YOLO V7	98.31	0.98	96.07	99.1
Tomato early blight	YOLO V7	92.37	0.87	79.95	95.99
	Improved-YOLO V7	95.22	0.91	84.01	98.32

3.2.4. Comparative Experiments of Different Target Detection Models

To enhance the persuasiveness of the results, in this section, we compare commonly used object detection algorithms including Faster R-CNN, RetinaNet, SSD, YOLO V5, YOLO X, YOLO V7, YOLO V8, and the latest version, YOLO V9. The experimental results are presented in Table 5. In this comparison, we define small targets based on an absolute scale, following the definition of the COCO dataset, where objects smaller than $32 \text{ pixels} \times 32 \text{ pixels}$ are considered small targets. All comparative models were evaluated against this criterion.

Table 5. Comparison of detection effect of different models.

Model	mAP (%)	mAP (%) Small obj.	Params (M)	FLOPs (G)	F1	Recall (%)	Precision (%)
Faster R-CNN	79.61	27.9	136.77	370.01	0.65	84.73	52.94
RetinaNet	85.21	43.9	55.40	201.16	0.83	74.40	94.17
SSD	80.96	32.4	26.28	62.80	0.71	59.69	92.44
YOLO V5	92.34	47.6	46.65	114.30	0.86	79.50	95.23
YOLO X	95.80	55.7	54.15	155.38	0.92	89.58	95.21
YOLO V7	90.96	48.0	37.21	104.83	0.87	80.26	94.34
YOLO V8	95.82	58.0	25.85	79.07	0.93	89.61	96.63
YOLO V9	97.06	58.6	25.44	102.8	0.94	90.20	96.91
Improved-YOLO V7	96.75	58.1	26.35	74.24	0.94	89.69	97.64

From the table, it can be observed that although SSD has relatively fewer parameters and computational requirements, its mAP and small-target mAP are both very low, with a recall of only 59.69%. On the other hand, Faster R-CNN, a two-stage method that employs RPN for candidate box generation, exhibits a large parameter and computational overhead, making it less deployable and also ineffective for detecting small target diseases.

Additionally, we compared our model with several classic YOLO variants. The YOLO V7 model we improved upon is the original version described in reference [16], which corresponds to the YOLO V7 model presented in the experimental tables of the original paper. Specifically selecting the large versions of YOLO V5 and YOLO X as comparison benchmarks. The large version of YOLO X achieved a mAP of 95.80%, which is close to our model's performance. However, our model's size is approximately half that of YOLO X, enabling us to achieve higher accuracy with fewer parameters and computational requirements. Furthermore, we compared the YOLO V8 m model and the YOLO V9 C model, which have similar parameter counts. The overall performance of V8 is slightly lower than ours. Although the latest version, YOLO V9, has a 0.31% higher mAP than ours, its computational cost is 28.56G more, indicating that our model demonstrates superior efficiency.

Compared to the original YOLO V7, our model achieved a 5.79% increase in mAP and a 10.1% increase in small-target mAP while reducing parameters by 10.86 M and computational requirements by 30.59 G. Furthermore, the model's F1 score, recall, and precision improved by 7%, 9.43%, and 3.3%, respectively.

To provide a more visual comparison of detection performance, we selected an image containing multiple instances of small target diseases for detection comparison, as shown in Figure 12. This image displays the detection results of various object detection models for Grape Black Rot disease.

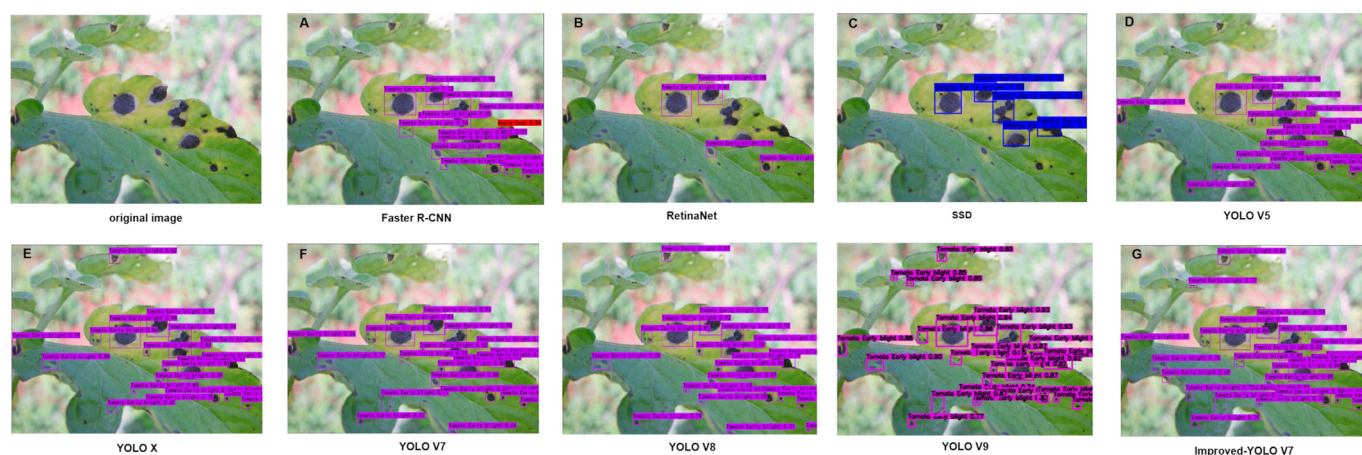


Figure 12. Image of detection effect of different target detection models on grape black rot. (A–G) represents different target detection models.

From the figure, we can observe that Faster R-CNN correctly detected 15 disease instances but made 1 false detection. RetinaNet and SSD only detected four or five relatively large disease portions. YOLO V5 correctly detected 17 disease instances, YOLO X detected 19 disease instances, and YOLO V7 detected 20 disease instances. The YOLO V8 correctly detected 21 instances of disease, while the YOLO V9 correctly detected 25 instances. Our improved model also correctly detected 25 instances, while maintaining relatively high detection accuracy. Overall, the performance of our improved model is comparable to that of YOLO V9, indicating that our model can accurately detect the location of diseases while effectively avoiding missed and false detections for small and multiple targets.

In the comparison of the model's performance data, we observed that the YOLO series performs well overall. To provide a clearer visualization of the detection performance of different YOLO models, we generated heatmaps for different types of diseases detected by the YOLO series models, as shown in Figure 13. In this figure, A represents Apple Scab Disease, B represents Grape Black Rot Disease, C represents Potato Late Blight Disease, D represents Tomato Early Blight Disease, and E represents Corn Leaf Blight Disease.

In our selected disease samples, some exhibit numerous small target diseases, while others are densely clustered or scattered. The heatmaps reveal areas of greater model attention, represented by darker regions. Upon analysis, we note that the YOLO V5 model's attention span appears excessively broad, encompassing non-disease areas and showing limited focus on small targets. Conversely, the YOLO X model excels in detecting small targets, yet it may overlook slightly larger diseases, resulting in smaller attention regions compared to the actual disease area. Similarly, YOLO V7 often overlooks or allocates less attention to smaller target diseases.

There are also the more powerful V8 and V9 versions, both of which have a more comprehensive focus on diseases, but in comparison, our model has a more comprehensive focus on disease coverage as well as the more effective detection of small diseases. It shows that our model can focus on more disease areas and also has a better detection effect on small target diseases. In summary, the improved YOLO V7 model in this paper is more effective in detection.

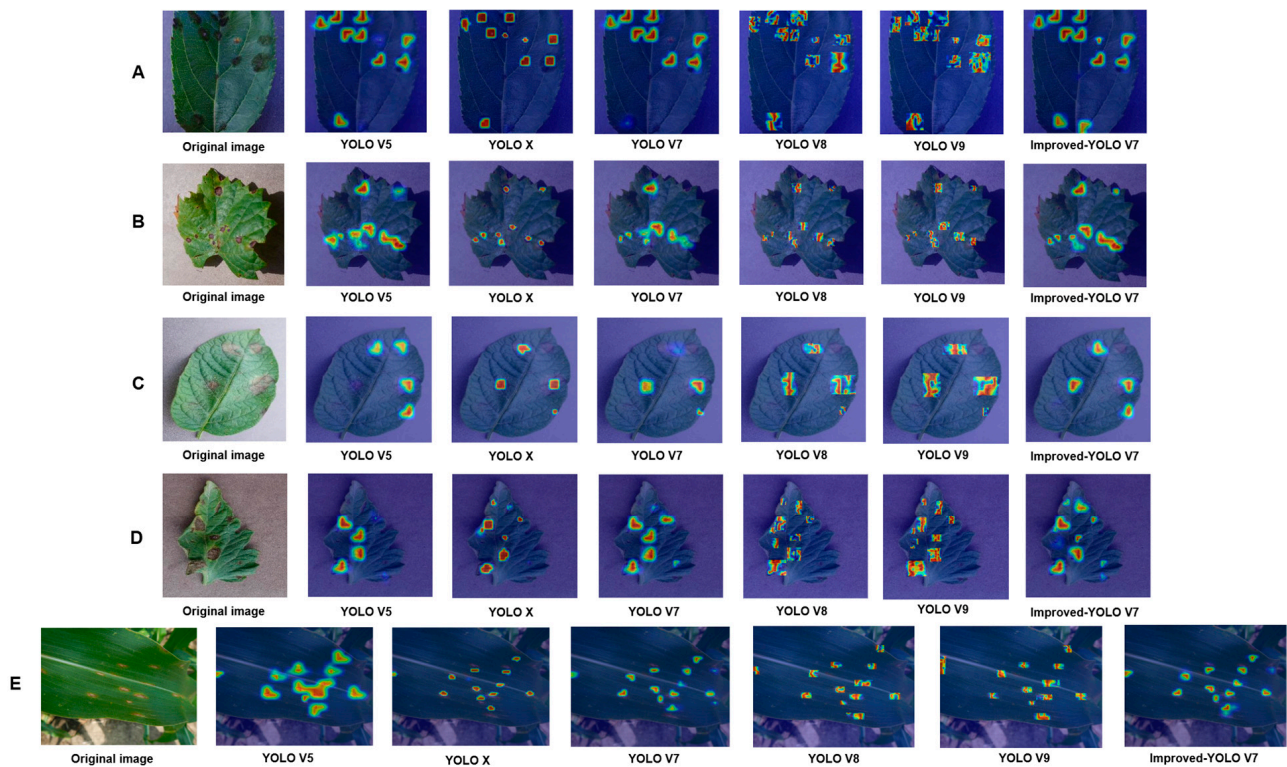


Figure 13. The effect of disease heat map detection under different target detection models. (A) represents Apple Scab Disease, (B) represents Grape Black Rot Disease, (C) represents Potato Late Blight Disease, (D) represents Tomato Early Blight Disease, and (E) represents Corn Leaf Blight Disease.

3.2.5. Comparison of Models under Different Anchors and Dynamic k Values

After the initial screening of positive samples, YOLO V7 utilizes the SimOTA strategy to further refine these samples. Since each ground truth (GT) box requires a different number of positive samples, the model calculates the cost for each sample against each GT box. It then selects the top 10 predicted boxes with the highest Intersection over Union (IOU) with each sample. If there are fewer than 10 such boxes, it includes as many as available, with a minimum of 1. This quantity is referred to as the dynamic k value. Subsequently, the model selects the top dynamic k anchors based on the cost for each sample as the positive samples used for calculating the final loss.

During this process, due to the low IOU values in the early stages of model training, the dynamic k value is typically close to 1. Therefore, we investigated whether different minimum settings for a dynamic k would affect the speed of model training. The results are presented in Table 6.

Table 6. The influence of different K values on model training.

Dynamic k Minimum	mAP (%)	Train Time
min = 1	96.75%	1 d 22 h 38 m
min = 2	96.95%	2 d 19 h 21 m
min = 3	96.77%	2 d 21 h 47 m

Through multiple tests, we found that setting the minimum value of dynamic k to greater than 1 may slightly improve the mAP, but it also leads to an increase in the training time. Therefore, in the experiments conducted in this study, we opted to maintain the same minimum value of 1 as in the original configuration.

On the other hand, traditional YOLO anchor box generation primarily relies on the K-means clustering algorithm to obtain varying anchor box sizes across different datasets. High-quality anchor boxes facilitate better bounding box regression, thereby enhancing model performance. However, the K-means algorithm is susceptible to sensitivity issues regarding cluster centers; the improper initialization of cluster centers may lead to sub-optimal clustering. Hence, we opted for the K-means++ algorithm, which introduces a probabilistic selection method on top of the K-means algorithm to mitigate the aforementioned sensitivity issue. In essence, after initializing the cluster centers, K-means++ computes the shortest distance between each sample and the existing cluster centers. Samples farther away from the current cluster centers are more likely to be selected as the next cluster center. Consequently, K-means++ prevents the clustering centers from being too close together, thus mitigating the generation of anchor box sizes that are highly similar. This approach enhances the rationality of anchor box size settings. Given that our model consists of four prediction layers, each requiring three anchors, we need to generate 12 sets of recommended anchor box sizes.

The anchor distributions generated by different algorithms are illustrated in Figure 14a,b. From the visual representation, it is evident that due to the prevalence of small-to-medium-sized target diseases in our dataset, there is a higher concentration of clusters within the 100×100 -pixel range. Furthermore, it is apparent that the clustering produced by the K-means++ algorithm is more rational. It delineates cluster centers with greater granularity, preventing them from being too closely positioned, which in turn mitigates the production of similar anchor sizes. For our dataset, the 12 sets of anchors obtained from the K-means algorithm are as follows: 20, 21, 30, 30, 43, 38, 65, 35, 35, 68, 53, 50, 83, 50, 68, 78, 120, 78, 83, 149, 203, 145, 123, and 415. Meanwhile, the 12 sets of anchors derived from the K-means++ algorithm are: 25, 25, 42.5, 37.5, 32.5, 57, 52.5, 57.5, 72.5, 45, 52.5, 112.5, 80, 75, 132.5, 72.5, 105, 136.2, 215, 145, 95, 382.5, 180, and 437.5. Finally, we conducted accuracy comparison experiments on our improved model using the anchors obtained from the two different algorithms, as depicted in Table 7.

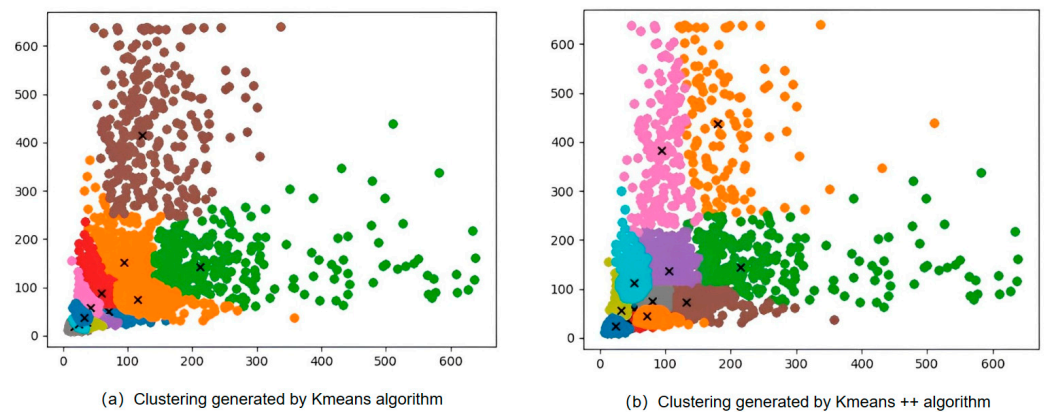


Figure 14. Anchor distributions generated by different algorithms. Different colors represent different clusters.

Table 7. Comparative experiments of anchor with different algorithms.

Different Anchor	mAP (%)
Anchor generated by Kmeans	96.75%
Anchor generated by Kmeans++	96.82%

Through multiple comparisons, it has been observed that the anchor boxes generated by the K-means++ algorithm slightly improve the accuracy of model regression. Therefore, in the final deployment stage, we opt to train the model using the anchor values obtained from the K-means++ algorithm.

3.2.6. Deployment and Application

Building upon the aforementioned experiments, we conducted deployment testing by deploying the improved algorithm onto the SCOUT 2.0 mobile robot. SCOUT 2.0 is a multifunctional modular industrial application mobile robot development platform characterized by its modular and intelligent design. As depicted in Figure 15, the SCOUT 2.0 mobile robot possesses specific mechanical parameters, as outlined in Table 8.

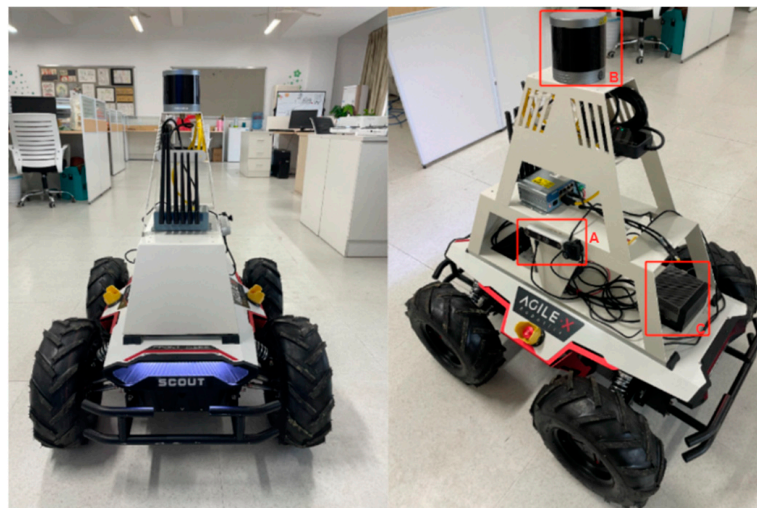


Figure 15. SCOUT 2.0 Mobile Robot Front and Side View.

Table 8. Mechanical parameters of mobile robots.

Parameter Type	Parameter Value
Length, width, and height (mm)	930 × 699 × 349
Wheelbases (mm)	498
Front/Rear Wheelbase	583
Total weight	67 ± 1 kg
Battery Type	li-ion battery
Battery parameters	24 V 30 Ah
Power Drive Motor	DC brushless 4 × 400 W
Steering Forms	four-wheel differential steering

The red box labeled A contains two expanded RGB cameras primarily used for identifying and detecting plant diseases along the robot's movement path. The red box labeled B houses a LiDAR sensor, while the red box labeled C houses the NVIDIA Jetson AGX Orin Developer Kit, which serves as the main control unit for the mobile robot. Detailed specifications for the NVIDIA Jetson AGX Orin Developer Kit are provided in Table 9.

Table 9. NVIDIA Jetson AGX Orin Developer Kit Parameter Sheet.

Major Parameter	Parameter Values
CPU	12-core Arm Cortex-A78AE v8.2 64-bit processor (3 MB L2 + 6 MB L3)
GPU	2048 NVIDIA CUDA cores and 64 Tensor cores @ 1 GHz
Memory	32 GB 256-bit LPDDR5 @ 204.8 GB/s
Reservoir	64 GB eMMC 5.1
USB	3 × USB 3.2, 4 × USB 2.0
Vision accelerator	PVA v2.0
Power mode	15 W, 30 W or 50 W
Size	100 × 87 mm

We simulated the task of plant disease detection in a laboratory environment by affixing multiple color-printed images containing plant diseases to the side of a laboratory table. Subsequently, we maneuvered the mobile robot remotely to positions where the plant diseases were present. Then, we utilized the side-mounted expanded RGB cameras to identify and detect the diseases in the images. The recognition and detection results are displayed on the mobile robot platform, as illustrated in Figure 16.

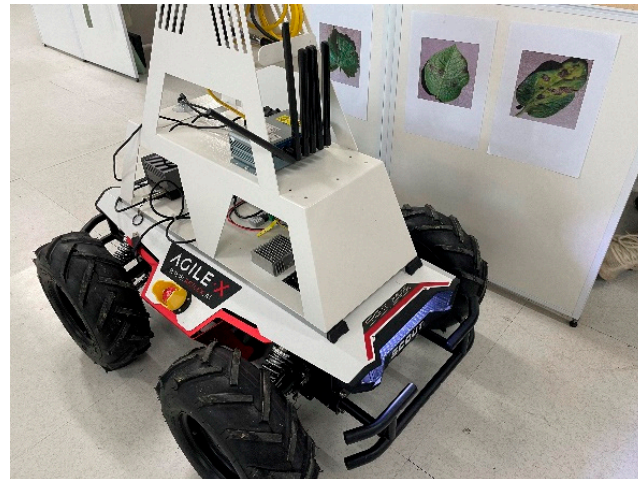


Figure 16. Mobile Robot Detects Plant Disease Processes.

The recognition and detection results of the SCOUT 2.0 mobile robot are illustrated in Figure 17. Panel (a) depicts the detection image of apple black spot disease, while panel (b) shows the detection image of grape black rot disease. In panel (c), we observe the detection image of corn leaf spot disease, and in panel (d), we see the detection image of tomato early blight disease. From the figures, it is evident that when the mobile robot reaches the location of plant diseases, it is capable of detecting the category and location of the diseases. It achieves precise localization for large and medium-sized disease targets and also demonstrates a satisfactory detection performance for small disease targets.

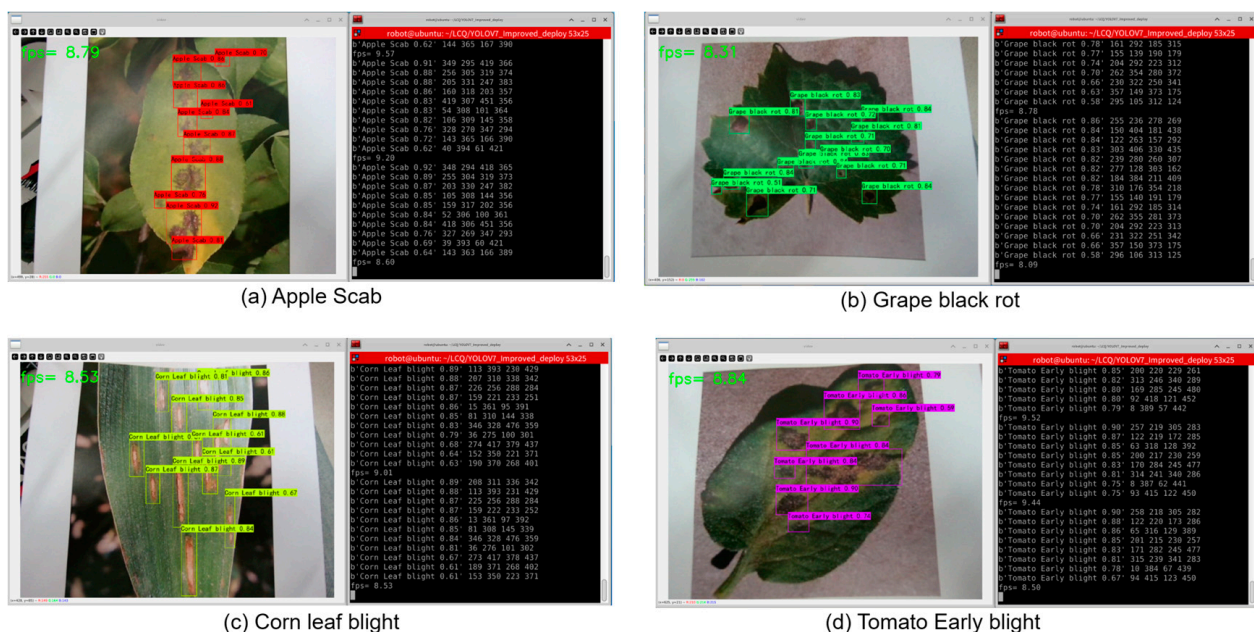


Figure 17. Mobile robot actual detection effect.

4. Discussion

In this study, we employed several commonly used data augmentation methods to process and train the dataset. Among them, Mixup data augmentation enhances model robustness while reducing the risk of overfitting, while mosaic data augmentation increases data diversity. In this paper, we explored the effects of using different proportions of mosaic and Mixup data augmentation and found that using excessively large or small proportions may lead to a decrease in model performance. This is because such training methods may deviate from the true distribution of natural images. Therefore, we adopted a 40% ratio for data augmentation training rounds, with the remaining rounds trained using the normal method.

Subsequently, we augmented the original YOLO V7 model by adding a fourth prediction head and incorporating our improved downsampling SPD-MP module in the backbone and neck parts. During this process, the SPD-MP module slightly improved the model's mAP value while reducing some parameters. The addition of the fourth prediction head significantly increased the model's mAP value, albeit at the cost of an increased computational overhead. To ensure model lightness and deployability, we employed the improved lightweight DW-ELAN module. After these overall model structure improvements, the modified model reduced 30% of the computational load and parameter count compared to the original YOLO V7. Additionally, the inclusion of the SPD-MP module and the fourth prediction head further improved the detection performance for small disease targets.

Given the plethora of texture features to be learned in plant disease detection, we first compared a series of classic attention mechanisms and found that SimAM attention was effective and lightweight. Building upon this, we introduced a softening threshold operation based on this attention. We set the initial threshold to 0.5, which automatically updates during loss iteration, resulting in the improved Soft-SimAM attention mechanism. It focuses more on important neurons, and we strategically positioned this attention in the backbone-neck connection and upsampling processes in the neck to facilitate the better fusion of contextual features, enhancing the network's global information extraction capability.

Subsequently, we experimented with a series of different loss functions and found that EIOU loss performed well. However, it faced issues when the center points of the predicted and ground truth boxes overlapped, but had different aspect ratios and sizes, leading to identical losses for both boxes. Therefore, we further improved the loss function, designing a new one for bounding box regression to achieve higher efficiency and accuracy.

Finally, we replaced the original YOLO V7's CIOU loss with the APEIOU loss obtained from our improvement. It builds upon EIOU Loss by incorporating an auxiliary penalty term function, successfully differentiating cases where the center points of predicted and ground truth boxes overlap, but have different aspect ratios and sizes. To increase the number of positive samples, we modified the grid matching mechanism, increasing the number of adjacent grid matches in YOLO V7's lead head and adjusting the offset formula to mitigate boundary issues. This also allows each ground truth to match up to 48 positive samples, providing more options for the subsequent SimOTA fine screening of positive samples.

We demonstrated, in the experimental section of this paper, that each improvement is meaningful and results in significant enhancements compared to the original YOLO V7. However, there are limitations, as indicated by the need for improved FPS in the deployment section due to hardware and computational constraints. Therefore, in the future, we will attempt deployment acceleration using TensorRT and continue researching how to create a lighter and more efficient model for the real-time detection of a broader range of plant diseases, enabling real-time detection in botanical garden environments.

5. Conclusions

This study introduces an enhanced version of the YOLO V7 model, which utilizes five common plant diseases sourced from the publicly available datasets PlantVillage

and PlantDoc. The dataset was manually annotated and subsequently partitioned into training, validation, and test sets at an 8:1:1 ratio for subsequent experimentation. Subsequent adjustments were made to the overall architecture of YOLO V7, including the integration of a fourth prediction head to improve the detection performance for small targets. Additionally, the SPD-MP subsampling module was proposed and implemented in both the backbone and neck sections to mitigate information loss during subsampling, particularly when dealing with a high volume of small targets. Furthermore, a lightweight DW-ELAN module was employed to streamline the model. Simultaneously, the SimAM attention mechanism was fine-tuned by softening its threshold, resulting in the Soft-SimAM attention mechanism, which was then fused with the features output by each backbone layer and utilized in the upsampling process of the neck section to enhance the learning of detailed features. This facilitated further emphasis on crucial features while suppressing less important ones.

Subsequently, a novel loss function termed APEIOU Loss was developed based on the EIOU Loss, incorporating an auxiliary penalty term to better distinguish between different fitting scenarios of predicted and ground truth bounding boxes. Considering the impact of the quantity and quality of positive samples on model training, modifications were made to the grid matching quantity and offset formulas to allow each ground truth to match up to 48 positive samples, providing more options for the subsequent SimOTA fine-screening of positive samples. Additionally, the influence of different dynamic k minimum values on the model training speed was explored, revealing that setting the minimum value greater than 1 may slightly enhance the model accuracy, but also increases the training time. Finally, both the Kmeans and Kmeans++ algorithms were employed to recluster the dataset, resulting in recommended anchor box values. Training and testing with anchor boxes generated by the Kmeans++ algorithm led to a slight improvement in regression accuracy. This study concludes by deploying the improved model onto the SCOUT 2.0 mobile robot, achieving the precise identification and detection of plant diseases, thus demonstrating the model's applicability.

The experimental results confirm that the improved YOLO V7 model proposed in this study exhibits enhanced feature extraction capabilities, leading to higher accuracy in plant disease detection, particularly for detecting small-scale diseases.

Author Contributions: Y.Z. and N.W.: Writing—original draft, Conceptualization. C.L.: Writing—original draft, Formal analysis. X.X.: Data curation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2019YFE0126100); the Key Research and Development Program in the Zhejiang Province of China (2019C54005); the National Natural Science Foundation of China (61605173) and (61403346); and the Natural Science Foundation of the Zhejiang Province (LY16C130003).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We extend our gratitude to all authors for their technical assistance in this study, and we also wish to thank the anonymous reviewers for their valuable suggestions and critical comments.

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

References

1. Sindhuja, S.; Ehsani, R.; Morganc, K.T. Detection of anomalies in citrus leaves using laser-induced breakdown spectroscopy (LIBS). *Appl. Spectrosc.* **2015**, *69*, 913–919.
2. Parminder, K.; Pannu, H.S.; Malhi, A.K. Plant disease recognition using fractional-order Zernike moments and SVM classifier. *Neural Comput. Appl.* **2019**, *31*, 8749–8768.

3. Kim, L.; Legay, A.; Nolte, G.; Schlüter, M.; Stoelinga, M. Formal methods meet machine learning (F3ML). In *International Symposium on Leveraging Applications of Formal Methods*; Springer Nature: Cham, Switzerland, 2022.
4. Moez, K.; Mihoub, A.; Alzahrani, M.Y.; Adoni, W.Y.H.; Nahhal, T. Are formal methods applicable to machine learning and artificial intelligence? In *Proceedings of the 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Riyadh, Saudi Arabia, 9–11 May 2022.
5. Ross, G.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 23–28 June 2014.
6. Ross, G. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [[CrossRef](#)] [[PubMed](#)]
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In *Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; *Proceedings, Part I 14*. Springer International Publishing: Cham, Switzerland, 2016.
9. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 22–29 October 2017.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 27–30 June 2016.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 June 2017.
12. Redmon, J.; Farhadi, A. YoloV3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YoloV4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YoloX: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
15. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
16. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada, 17–24 June 2023.
17. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
18. Gong, X.; Zhang, S. A high-precision detection method of apple leaf diseases using improved faster R-CNN. *Agriculture* **2023**, *13*, 240. [[CrossRef](#)]
19. Lee, S.-H.; Gao, G. A Study on Pine Larva Detection System Using Swin Transformer and Cascade R-CNN Hybrid Model. *Appl. Sci.* **2023**, *13*, 1330. [[CrossRef](#)]
20. Tian, L.; Zhang, H.; Liu, B.; Zhang, J.; Duan, N.; Yuan, A.; Huo, Y. VMF-SSD: A Novel v-space based multi-scale feature fusion SSD for apple leaf disease detection. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *20*, 2016–2028. [[CrossRef](#)] [[PubMed](#)]
21. Sankarashwaran, S.P.; Jayaraman, G.; Muthukumar, P.; Krishnan, A. Optimizing rice plant disease detection with crossover boosted artificial hummingbird algorithm based AX-RetinaNet. *Environ. Monit. Assess.* **2023**, *195*, 1070. [[CrossRef](#)] [[PubMed](#)]
22. Xu, W.; Wang, R. ALAD-YOLO: An lightweight and accurate detector for apple leaf diseases. *Front. Plant Sci.* **2023**, *14*, 1204569. [[CrossRef](#)] [[PubMed](#)]
23. Lin, J.; Yu, D.; Pan, R.; Cai, J.; Liu, J.; Zhang, L.; Wen, X.; Peng, X.; Cernava, T.; Oufensou, S.; et al. Improved YOLOX-Tiny network for detection of tobacco brown spot disease. *Front. Plant Sci.* **2023**, *14*, 1135105. [[CrossRef](#)] [[PubMed](#)]
24. Tian, Y.; Wang, S.; Li, E.; Yang, G.; Liang, Z.; Tan, M. MD-YOLO: Multi-scale Dense YOLO for small target pest detection. *Comput. Electron. Agric.* **2023**, *21*, 108233. [[CrossRef](#)]
25. Xu, W.; Xu, T.; Thomasson, J.A.; Chen, W.; Karthikeyan, R.; Tian, G.; Shi, Y.; Ji, C.; Su, Q. A lightweight SSV2-YOLO based model for detection of sugarcane aphids in unstructured natural environments. *Comput. Electron. Agric.* **2023**, *211*, 107961. [[CrossRef](#)]
26. Solimani, F.; Angelo, C.; Giovanni, D.; Angelo, P.; Stephan, S.; Francesco, C.; Vito, R. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [[CrossRef](#)]
27. Yang, G.; Wang, J.; Nie, Z.; Yang, H.; Yu, S. A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. *Agronomy* **2023**, *13*, 1824. [[CrossRef](#)]
28. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 15–19 October 2016.
29. Lachlan, T.-S.; Petersson, L. Improving object localization with fitness NMS and bounded IOU loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018.
30. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June 2019.
31. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IOU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on Artificial Intelligence*, New York, NY, USA, 7–12 February 2020; Volume 34.

32. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [[CrossRef](#)]
33. David, H.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
34. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 5–7 January 2020; pp. 249–253.
35. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
36. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2022.
37. Zhao, M.; Zhong, S.; Fu, X.; Tang, B.; Pecht, M. Deep residual shrinkage networks for fault diagnosis. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4681–4690. [[CrossRef](#)]
38. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
39. Ma, S.; Yong, X. MPDIoU: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv:2307.07662.
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
41. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
42. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
43. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
44. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561.

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