symmetry

MDPI

Article

Multifactorial Tomato Leaf Disease Detection Based on Improved YOLOV5

Guoying Wang [1,†], Rongchang Xie [1,†], Lufeng Mo [1,2,*], Fujun Ye [3], Xiaomei Yi [1] and Peng Wu [1]

[1] College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China; wgy@zafu.edu.cn (G.W.); 2021611011064@stu.zafu.edu.cn (R.X.); yxm@zafu.edu.cn (X.Y.); wp@zafu.edu.cn (P.W.)
[2] Information and Education Technology Center, Zhejiang A&F University, Hangzhou 311300, China
[3] Network and Data Center, Communication University of Zhejiang, Hangzhou 310018, China; yefuj@cuz.edu.cn
* Correspondence: molufeng@zafu.edu.cn
† These authors contributed equally to this work.

Abstract: Target detection algorithms can greatly improve the efficiency of tomato leaf disease detection and play an important technical role in intelligent tomato cultivation. However, there are some challenges in the detection process, such as the diversity of complex backgrounds and the loss of leaf symmetry due to leaf shadowing, and existing disease detection methods have some disadvantages in terms of deteriorating generalization ability and insufficient accuracy. Aiming at the above issues, a target detection model for tomato leaf disease based on deep learning with a global attention mechanism, TDGA, is proposed in this paper. The main idea of TDGA includes three aspects. Firstly, TDGA adds a global attention mechanism (GAM) after up-sampling and down-sampling, as well as in the SPPF module, to improve the feature extraction ability of the target object, effectively reducing the interference of invalid targets. Secondly, TDGA uses a switchable atrous convolution (SAConv) in the C3 module to improve the model's ability to detect. Thirdly, TDGA adopts the efficient IoU loss (EIoU) instead of complete IoU loss (CIoU) to solve the ambiguous definition of aspect ratio and sample imbalance. In addition, the influences of different environmental factors such as single leaf, multiple leaves, and shadows on the performance of tomato disease detection are extensively experimented with and analyzed in this paper, which also verified the robustness of TDGA. The experimental results show that the average accuracy of TDGA reaches 91.40%, which is 2.93% higher than that of the original YOLOv5 network, which is higher than YOLOv5, YOLOv7, YOLOHC, YOLOv8, SSD, Faster R-CNN, RetinaNet and other target detection networks, so that TDGA can be utilized for the detection of tomato leaf disease more efficiently and accurately, even in complex environments.

Keywords: disease detection; tomato leaf images; object detection; attention mechanism

Citation: Wang, G.; Xie, R.; Mo, L.; Ye, F.; Yi, X.; Wu, P. Multifactorial Tomato Leaf Disease Detection Based on Improved YOLOV5. *Symmetry* **2024**, *16*, 723. https://doi.org/10.3390/sym16060723

Academic Editors: João Ruivo Paulo, Cristina P. Santos and Gabriel Pires

Received: 11 May 2024
Revised: 2 June 2024
Accepted: 5 June 2024
Published: 11 June 2024

1. Introduction

Effective and accurate detection of crop diseases is essential to minimize damage. Usually, farmers have to consult experts, which requires a lot of time. So, a fast, accurate, and less costly technique is needed to identify diseases in crop leaves. Image processing [1,2] and machine learning techniques [3,4] can meet the requirements of early detection of disease [5–7] of crop leaves for phenotyping of crop disease disorders to reduce the use of pesticides.

Research on target detection of tomato leaf diseases is divided into two symmetrical classes of methods: one is based on manual extraction of features [8,9], and the other is based on convolutional neural networks [10,11]. In terms of manual-based feature extraction, Sabrol and Satish [12] segmented tomato leaf pests and diseases by the ostu thresholding method, extracted color, shape, and texture features after removing the

background influence of leaf images and inputted them into a decision tree to obtain the final classification. Jaisakthi et al. [13] used the GrabCut algorithm with a support vector machine for the classification of grape leaf pests and diseases for classification in order to remove the region of interest other than pests and diseases. As for convolutional neural network methods, Zu et al. [14] used Mask R-CNN [15] for the detection and segmentation of ripe tomatoes; Xie et al. [16] proposed a deep learning-based faster DR-IACNN model with enhanced feature extraction capability for detecting grape leaf disease; Syed-Ab-Rahman and Gong [17,18] used Faster R-CNN [19]-based models to detect citrus and apple leaf pests. As a one-stage target detection algorithm, YOLO was proposed by Redmon et al. [20], which has excellent performance on classical datasets (COCO datasets) and is widely used for various agricultural target detection tasks. Qi et al. [21]; Wang et al. [22]; and Liu et al. [23] introduced the concept of sysmmetry to improve the YOLO series [24,25] for target detection of leaf diseases of tomato.

In real complex environments, diversity and leaf shading in complex backgrounds can lead to loss of leaf symmetry. In addition, tomato leaf images may contain multiple leaves, and there may be weeds, and other interfering factors. These symmetry losses may be similar to tomato leaf disease symptoms and can easily lead to disease detection algo-rithms misclassifying tomato leaf disease. areas. At the same time, there are also some diseases with small target areas that are difficult to detect. Wang et al. [26] proposed a tomato leaf disease detection method based on the fusion of attentional mechanisms and multiscale features, which can achieve an average accuracy of 92.9% in the detection of tomato leaf diseases. However, it is not effective in dealing with small disease spots with similar symptoms under complex backgrounds; Liu and Wang [27] proposed a target detection method for tomato diseases by fusing an a priori knowledge attention mechanism, multiscale features, a unique prediction layer, and loss function ASIOU. The tomato leaf disease detection in a complex context possesses 91.96% accuracy. However, the proposed model lacks the ability to autonomously acquire tacit knowledge (e.g., the precise location and shooting angle of tomato diseases), and the model operation is relatively more than cumbersome.

To address the above problems, a target detection model for tomato leaf disease using deep learning with global attention, TDGA, is proposed in this paper, which is constructed on the basis of YOLOv5 [28]. The main idea of TDGA includes three aspects. Firstly, to improve the model's extraction ability, we incorporate the attention mechanism GAM [29] after up-sampling and down-sampling and in the SPPF module to reduce the interference of invalid targets. Second, to address the disease's occlusion and shadowing due to multiple leaves, we use SAConv [30] in the C3 module to improve the model's ability to detect multi-scale targets. Finally, the loss function EIoU [31] is used to solve the ambiguity of aspect ratio definition and sample imbalance brought by the loss function CIoU [32].

In addition, in order to verify the robustness of the proposed method, TDGA, the influences of factors such as the count of leaves and the area of shadows [33] in real complex environments on the effectiveness of the detection of tomato leaf disease are also examined in this paper. The experimental results show that TDGA is able to meet the requirements in terms of detection accuracy in each dataset.

The remaining parts of the paper are organized as follows. Section 2 describes in detail the main idea of TDGA, the model proposed in this paper, and the related detail methods involved; Section 3 describes the experimental setup and the experimental procedure; Section 4 carries out the analysis of the comparative experimental results; and Section 5 concludes and summarizes the paper.

2. Materials and Methods

2.1. Main Ideas

The target detection model for tomato leaf disease using deep learning with global attention, TDGA, is proposed in this paper, and the structure of TDGA is shown in Figure 1.

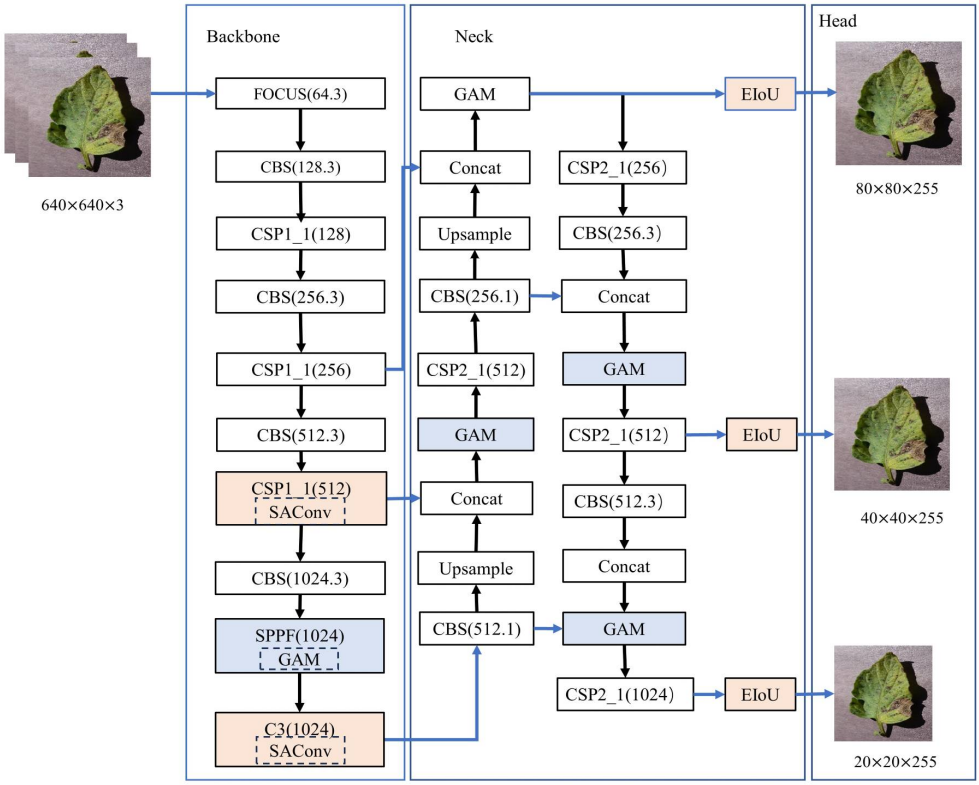


Figure 1. Structure of TDGA. The black and blue arrows show the processing direction of the TDGA model for image detection.

According to Figure 1, the main ideas of TDGA include three aspects, which are listed below.

(1) Importing the GAM. A GAM is imported after up-sampling and down-sampling as well as in the SPPF module to reduce the loss of accuracy, train a more accurate detection model, and improve the accuracy of the model in detecting tomato leaf disease images.

(2) Transplanting SAConv to replace ordinary convolution. A SAConv is transplanted into the C3 module to replace the ordinary convolution, because the ordinary convolution uses a fixed convolution kernel size, which limits the range of the sensory field. The SAConv expands the receptive field and increases the perceptual ability of the network by changing the dilation rate of the convolution kernel, which improves the model's ability to detect and recognize objects at different scales and improves its robustness to scale changes.

(3) Adopting EIoU to replace CIoU. EIoU solves the problem of the penalty failure for an equal proportional change of the aspect ratio in the CIoU, which accelerates the convergence speed and improves the regression accuracy. Meanwhile, focal EIoU loss is introduced to reduce the optimization contribution of anchor frames with low overlap with the target frame to BBox regression, so that the regression process focuses on high-quality anchor frames.

2.2. Importing GAM

Attention is pivotal in human perception, as individuals employ localized observations and selectively concentrate on salient aspects to more effectively discern visual structures. Similarly, numerous researchers have enhanced the efficacy of convolutional neural networks [34] in large-scale classification tasks by integrating attentional mechanisms.

In this paper, the TDGA integrates a GAM following the up-sampling, down-sampling, and SPPF modules, primarily composed of channel attention and spatial attention mecha-

nisms [35]. Initially, the global dependency between features is captured in both the spatial and channel dimensions, enhancing the representation of contextual feature information. The output from the channel attention module is combined with the original image features to produce F_2, which is further refined by summing it with the output from the spatial attention module. The final feature map is F_3. This iterative enhancement results in a more precise feature representation, leading to more accurate detection outcomes. The structure of this module is depicted in Figure 2.

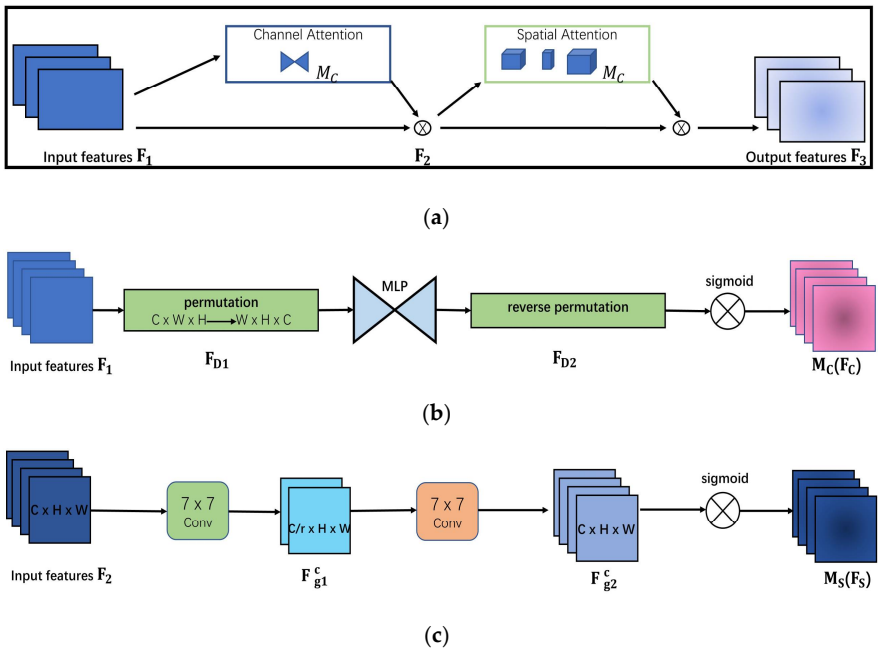


Figure 2. GAM structure diagram. (**a**) Structure of GAM. (**b**) Channel attention submodule. (**c**) Spatial attention submodule.

The channel attention mechanism selectively highlights interrelated channel graphs by integrating pertinent features across all channel graphs. This study explicitly models the interdependencies between channels by incorporating a channel attention mechanism module. Given an input feature map $F \in R(C \times W \times H)$, where C represents the number of channels, and W and H denote the width and height of the feature map, respectively, the input feature map F_1 is initially organized in 3D using a permutation module to retain the 3D information, transforming it into $W \times H \times C$. Subsequently, a two-layer multilayer perceptron (MLP) [36] is employed to enhance the channel-space dependency across dimensions. The reverse permutation module reverts the arrangement to the original 3D format. Then, the shared network composed of multilayer perceptron is computed, and the Sigmoid function is summed to finally obtain the channel-attention mechanism mapping feature map $F_{Cout} \in R^{C \times W \times H}$. Finally, the original input feature map is multiplied element by element to obtain the channel attention weighted map $F_{Cout} \in R^{C \times W \times H}$. The specific calculation process is shown below.

$$M_C = \sigma\left(F_{D2}\left(MLP\left(F^1_{D1}\right)\right)\right) \tag{1}$$

$$F_{Cout} = F \otimes F_c(M_c) \tag{2}$$

In the formula, σ denotes the Sigmoid activation function, F_{D1} denotes permutation, F_{D2} denotes reverse permutation, $\otimes$ denotes the multiplication between elements, and F_c denotes the replication of M_c along the spatial dimensions to obtain the $C \times W \times H$ feature vector.

The spatial attention mechanism selectively aggregates features at each location by computing a weighted sum of features across all locations, thereby associating similar features irrespective of their distances. A spatial attention mechanism module must be introduced to establish more prosperous contextual relationships among local features. Given an input feature map $F \in R^{C\times W\times H}$, where *C* denotes the number of channels, and W and *H* represent the width and height of the feature map, respectively, the input feature map (F) is initially reduced to a single channel through a 7×7 convolution operation, yielding the background description F^c_{g1}. Subsequently, F^c_{g1} is expanded back to C channels via another 7×7 convolution operation, resulting in F^c_{g2}. The spatial attention mechanism mapping feature map $M_s \in R^{C\times W\times H}$ is then obtained through the application of the Sigmoid function. Finally, the original input feature map undergoes element-wise multiplication to produce the spatial attention-weighted map $F_{Sout} \in R^{C\times W\times H}$. The detailed calculation process is illustrated below.

$$M_s = \sigma\left(f^{7\times7}\left(f^{7\times7}\left(F^2_g\right)\right)\right) \tag{3}$$

$$F_{Sout} = F \otimes F_s(M_s) \tag{4}$$

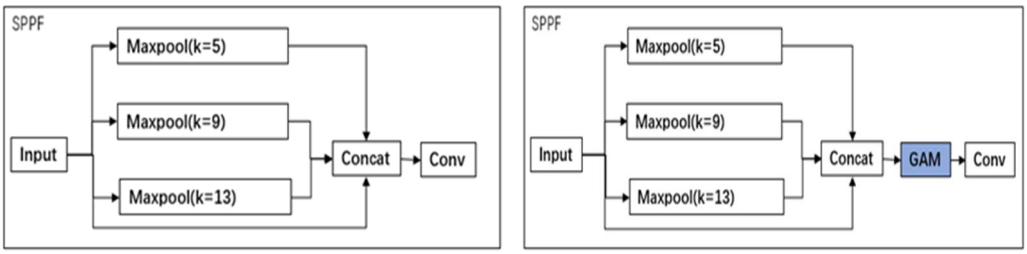
In the formula, σ denotes the Sigmoid activation function, $f^{7\times7}$ denotes the convolution operation, the size of the convolution kernel is 7×7, $\otimes$ denotes the multiplication between the elements, and F_s denotes the replication of M_s along the channel direction to obtain $C \times W \times H$ feature vector.

Thus, the expression for the GAM can be derived as shown below.

$$F_2 = F_{Cout} \otimes F_1 \tag{5}$$

$$F_3 = F_{Sout} \otimes F_2 \tag{6}$$

In the TDGA proposed in this paper, the GAM is added to the SPPF module of YOLOv5 and up-sampling and down-sampling. The improved structure of the SPPF module is shown in Figure 3.



(**a**) (**b**)

Figure 3. SPPF module added to the GAM structure. (**a**) Original structure. (**b**) Diagram of the improved structure.

2.3. Using Switchable Atrous Convolution (SAConv) to Replace Ordinary Convolution

The main role of the convolutional layer is feature extraction. The SAConv module has three main components: two global context modules and an SAC component. The goal of SAConv is to roughly detect objects at different scales of the same object by implementing the computation of convolution using the same convolutional weights between different dilation rates. Compared to atrous convolution [37], SAConv adds a weight-locking mechanism, which allows for a more flexible and efficient choice of scale during network training, and without changing any pre-trained models. Thus, SAConv is a plug-and-play module for many pre-trained backbone networks. Also, SAConv uses global context information to stabilize the switching mechanism. The structure of the SAConv module is shown in Figure 4.

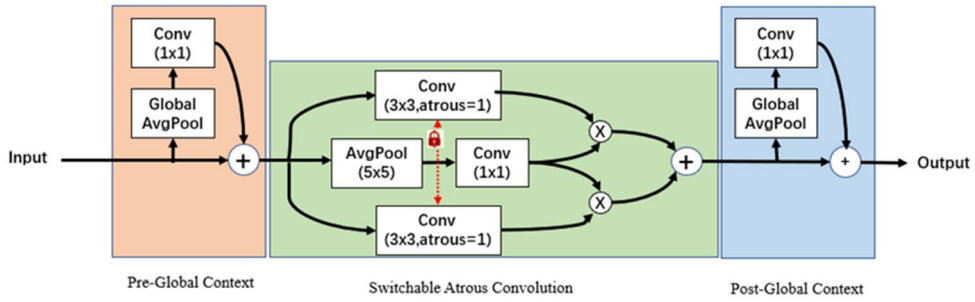


Figure 4. Structure of SAConv. The plus sign in the figure indicates that the feature maps are superimposed on each other; the multiplication sign indicates that the feature maps from different layers are multiplied element by element to realize the fusion of features; the red lock indicates the closure mechanism, which controls the weight of the feature maps for better-extracted feature maps information; and the red arrows indicate that the closure mechanism allocates the weights to the upper and lower layers of the maps.

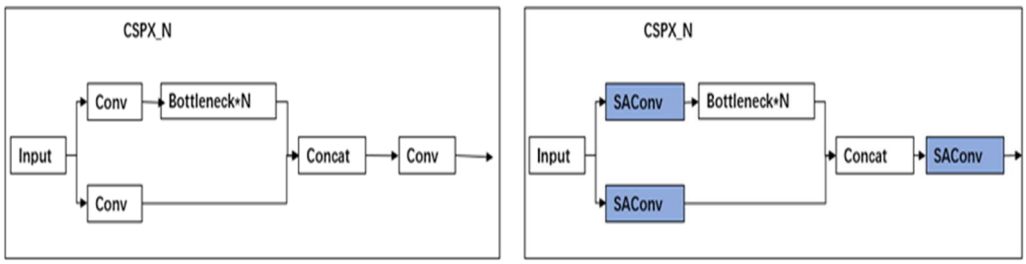
Target detection algorithms usually use pre-trained weights to initialize the network weights. However, for SAConv layers converted from standard convolutional layers, the weights for larger dilation rates are usually missing. Since it is possible to detect the same object at different scales using the same convolutional weights and at different dilation rates, it is possible to initialize these missing weights with the weights of the pre-trained model. The blocking mechanism in SAConv requires that one weight is set to be w and the other weight is set to be $\mathrm{w} + \Delta\mathrm{w}$ and $\mathrm{w} + \Delta\mathrm{w}$ is used as these missing weights, where w comes from the weights of the pre-trained model, and $\Delta\mathrm{w}$ is initialized to 0.

$$\mathrm{Conv}(\mathrm{x}, \mathrm{w}, 1) \overset{\text{Convert to SAC}}{\rightarrow} \mathrm{S}(\mathrm{x}) \cdot \mathrm{Conv}(\mathrm{x}, \mathrm{w}, 1) + (1 - \mathrm{S}(\mathrm{x}) \cdot \mathrm{Conv}(\mathrm{x}, \mathrm{w} + \Delta_{\mathrm{w}}, \mathrm{r})) \quad (7)$$

In the formula, $\mathrm{y} = \mathrm{Conv}(\mathrm{x}, \mathrm{w}, \mathrm{r})$ denotes the convolution operation, y is the output, x is the input, w is the weight, and r is the dilation rate (r is a hyperparameter of SAConv).

In SAConv's global context information module, the input features first go through a global average pooling layer for model compression, followed by a 1×1 convolutional layer, and this output is summed with the input features to obtain the output of the module, which is very similar to SENet [38], except that there is only one convolutional layer in the global context information module, and there are not any other nonlinear layers; the output of the global context module is summed with the backbone paths instead of being multiplied after a sigmoid. Therefore, the global context information module added to SAConv can make stable switching predictions after the switch function uses the global information, which has a positive impact on the detection performance.

This method replaces the ordinary convolution in the C3 module in YOLOv5 with SAConv. The improved C3 structure is shown in Figure 5.



(**a**) (**b**)

Figure 5. Replacing the ordinary convolution in the C3 module with SAConv. (**a**) Original C3 structure. (**b**) Improved C3 structure.

2.4. Adopting EIoU to Replace CIoU

Target detection encompasses two primary subtasks: target classification and target localization. It stands as one of the most pivotal challenges in computer vision. The contemporary leading-edge target detectors, including Cascade R-CNN, Mask R-CNN, Dynamic R-CNN, and DETR, hinge on bounding box regression (BBR) modules for accurate target localization. Within this framework, the design of a practical loss function is paramount to the success of BBR. However, previous IoU-based loss functions, such as CIoU and GIoU, must accurately quantify the disparity between the target frame and the anchor, resulting in sluggish convergence and imprecise localization during BBR model optimization.

The loss function utilized in YOLOv5 is CIoU. CIoU considers the distance between the target and the anchor box, the overlap, the scale, and the penalty term, which enhances the stability of target-box regression and avoids issues such as dispersion during the training process, commonly observed with IoU and GIoU. However, the penalty factor in CIoU, which aims to adjust the predicted box aspect ratio to fit the target box, is represented by v in Equation (9). This factor reflects the difference in aspect ratio rather than the actual differences in width, height, and confidence level. Consequently, this sometimes impedes the model's ability to optimize similarity effectively. EIoU comprises three components: overlap loss, center distance loss, and width and height loss. The first two elements follow the methodology of CIoU, while the penalty term in EIoU refines the penalty term from CIoU by separating the aspect ratio's influence factors to calculate the length and width of the target and anchor boxes individually. This approach directly minimizes the discrepancies in width and height between the target and anchor boxes, thus accelerating convergence. In box regression, the quantity of high-quality anchor frames with minimal regression errors for a single image is significantly less than the number of low-quality samples with large errors. These lower-quality samples generate excessive gradients that negatively impact the training process, leading to an imbalance in training samples. To address this issue, EIoU proposes focal EIoU loss, which combines EIoU with focal loss to distinguish high-quality anchor frames from low-quality ones based on gradient perspective. The formula is as follows:

$$\mathrm{CIoU} = 1 - \mathrm{IOU} + \frac{\rho^2\left(b, b^{gt}\right)}{C^2} - \alpha v \tag{8}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\left(\frac{w}{h}\right)\right)^2 \tag{9}$$

$$L_{EIOU} = 1 - \mathrm{IoU} + \frac{\rho^2\left(b, b^{gt}\right)}{C^2} + \frac{\rho^2(w, w^{gt})}{C_w^2} + \frac{\rho^2\left(h, h^{gt}\right)}{C_h^2} \tag{10}$$

In the formula, b, b^{gt} represent the centroids of the prediction box and the real box, respectively, and ρ represents the computation of the Euclidean distance between the two centroids, C represents the diagonal length of the smallest outer rectangle that can contain both the prediction box and the real box, α is a weight function, v represents the difference between the aspect ratios of the prediction box and the real box, respectively, C_w and C_h are the widths and heights of the smallest outer box that covers the two BOX, and γ is the parameter controlling the degree of outlier suppression.

The focal EIoU loss in this loss is somewhat different from the traditional focal loss The traditional focal loss mediates positive and negative samples through the hyperparameter α_t for the unbalanced proportion of sample sizes. p_t size actually reflects the degree of difficulty in categorizing the samples, and the larger p_t is the more correct the prediction is, and $(1 - p_t)^\gamma$ is used to adjust the weight of the difficult to categorize samples. α_t adjusts again for the loss that is after the attenuation of the $(1 - p_t)^\gamma$ coefficient. And α_t interacts with γ. As γ increases, α_t should correspondingly decrease. Consequently, the traditional focal loss assigns a more significant loss to more challenging samples, effectively serving as a mechanism for difficult sample mining. According to the formula below, it is evident

that in focal EIoU loss, a higher IoU results in a more significant sample loss. This effect resembles a weighting mechanism, where better regression targets incur higher losses, enhancing regression accuracy. The pertinent formula is presented as follows.

$$L_{Focal-EIOU} = IoU^{\gamma} L_{EIoU} \tag{11}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{12}$$

$$\text{Focal Loss} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{13}$$

In the formula, α represents the weight of the positive samples, p represents the output value of the Sigmoid activation function, y represents the actual labels, 1 for the positive samples and 0 for the negative samples, and γ is a parameter controlling the degree of outlier suppression.

3. Experiments

This section performs the experimental design in order to test the performance of the TDGA model proposed in this paper. First is the hardware and software equipment configuration, followed by the dataset production and preprocessing required for this experiment and the experimental network hyper-parameter settings. Finally, the robustness test and ablation experiment were performed.

3.1. Hardware and Software Configuration

The experiments in this paper used the deep learning framework PyTorch to train and test the performance of the TDGA method. The specific configuration of the experiments is shown in Table 1.

Table 1. Experimental software and hardware configuration.

Item	Detail
CPU	AMD Ryzen 5 5600X 6-Core Processor @3.70 GHz
GPU	RTX3060Ti(8G)
RAM	16 GB
Operating system	64-bit Windows 11
CUDA	CUDA12.2
Python	Python 3.7

3.2. Datasets

3.2.1. Data Acquisition and Preprocessing

The original images of the dataset used for the experiments in this paper were obtained from the Internet, Kaggle [39] and image synthesis, totaling 4578 images. Inspired by the concept of symmetry, we increase the sample capacity and improve the generalization ability by augmenting the dataset with a combination of operations, in-cluding random inversion, adding noise, zooming in and out, cropping, and mirroring. Finally, the tomato images in the dataset were labeled using the labeling software LabelImg to generate XML files. Although the YOLO series has provisions for the dataset labeling file, the dataset is uniformly stored in the PASCAL VOC [40] data format for better comparison experiments of various methods and experimental efficiency.

In order to test the effect of factors such as single leaf, multiple leaves, and shadows on the target detection performance of tomato leaf disease images, as well as to verify the robustness of the TDGA method used in this paper, three types of tomato leaf disease datasets such as single leaf, multiple leaves and shadows were constructed in this paper.

3.2.2. Image Dataset of Tomato Leaf Diseases in a Single Leaf

This dataset is mainly based on the presence of only one leaf in the image. Also, based on the type of disease in the image, the tomato disease images in this dataset are divided into healthy, early blight, late blight, and leaf mold datasets, and the images are divided into training set, testing set, and validation set according to 8:1:1, and some of the data are shown in Figure 6.

Figure 6. Images with different diseases in single leaf and labeling. (**a**) Healthy. (**b**) Early blight. (**c**) Late blight. (**d**) Leaf mold. Red boxes are disease-labeling boxes.

Among them, 3086 images of single leaves were available, of which the number of images of healthy, early blight, late blight, and leaf mold tomato leaves was 951, 970, 501, and 664, respectively. This dataset was expanded to 7838 images by image enhancement. See Table 2 for details.

Table 2. Single leaves dataset.

Disease	Raw Data	Enhanced Data
Healthy	951	1902
Early blight	970	1940
Late blight	501	2004
Leaf mold	664	1992
Total	3086	7838

3.2.3. Dataset of Tomato Leaves with Diseases in Multiple Leaves

This dataset mainly consists of tomato leaf images with two or more leaves. Due to the small number of tomato leaf disease images in the natural environment, this dataset not only collects disease images in the natural environment, but also utilizes image synthesis technology to synthesize tomato leaf disease image to generate different numbers of leaf disease images used to investigate the effectiveness of TDGA in detecting tomato leaf images under multi-leaf conditions. During training, the images were divided into training sets, test sets, and validation sets according to 8:1:1. The training set used the synthesized images, while the validation set was manually screened with images from a natural environment to obtain the real data results. Some of the data are shown in Figure 7.

Among them, 1168 images of multiple leaves. The number of tomato leaf images containing healthy, early blight, late blight, and leaf mold were 403, 405, and 360, respectively. This dataset was expanded to 5840 images by image enhancement, as detailed in Table 3.

Figure 7. Images with different diseases in multiple leaves and labeling. (**a**) Early blight. (**b**) Late blight. (**c**) Leaf mold. Red boxes are disease-labeling boxes.

Table 3. Dataset of tomato leaves with diseases in multiple leaves.

Disease	Raw Data	Enhanced Data
Early blight	403	2015
Late blight	405	2004
Leaf mold	360	1800
Total	1168	5840

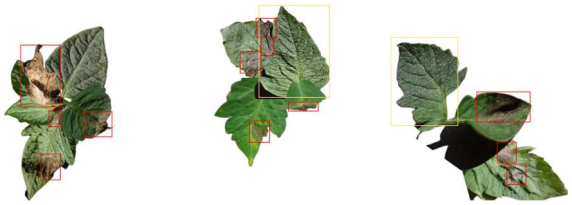
The real environment images were used as the validation set for the multi-leaf dataset without data enhancement. The 324 images of the real environment, which contained healthy, early blight, late blight, and leaf mold tomato leaf images, were 135, 91, and 98, respectively. as detailed in Table 4.

Table 4. Dataset of tomato leaves with diseases in multiple leaves in real environment.

Disease	Raw Data
Early blight	135
Late blight	91
Leaf mold	98
Total	324

3.2.4. Dataset of Tomato Leaves with Diseases in Multiple Leaves with Shadow

This dataset division is based on the presence or absence of shadows in the images under multiple leaves. The number of shadows present in the images of tomatoes under multiple leaves in natural environment is less. Therefore, the image synthesis technique is utilized to synthesize the shadow position of tomato leaves so as to generate tomato images with different shadow positions and areas for experimental investigation. Based on the ratio of the shadowed portion of the image to the area of the entire tomato image. Since the tomato leaves accounted for less than 50% of the 640 × 640 image ratio, the shadow area is too large, and it is easy to over-affect the effective target for disease detection. Therefore, the proportion of shadow area of tomato leaves in this dataset is divided into three sub-datasets according to 0, (0, 5%], (5%, 10%], and the images are divided into training set, testing set, and validation set according to 8:1:1. Part of the data are shown in Figure 8.



(**a**) (**b**) (**c**)

Figure 8. Images of shadow tomato leaves and labeling. (**a**) No shading. (**b**) Shading of 0–5%. (**c**) Shading of 5–10%. Red boxes are disease-labeling boxes.

Of these, 1168 images have shadows. The number of images where the percentage of shadows is 0, the number of images where the percentage of shadows is (0, 5], and the number of images where the percentage of shadows is (5–10] are 488, 396, and 284, respectively. Healthy leaves were present in other diseased images. This dataset was expanded to 4587 images by image enhancement, as detailed in Table 5.

Table 5. Shaded leaf dataset.

Shadow Percentage	Disease	Raw Data	Enhanced Data
0	Healthy	142	426
	Early blight	188	564
	Late blight	181	543
	Leaf mold	119	476
Subtotal		488	1583
(0, 5]	Healthy	121	484
	Early blight	118	472
	Late blight	141	564
	Leaf mold	137	548
Subtotal		396	1584
(5, 10]	Healthy	62	310
	Early blight	92	485
	Late blight	82	410
	Leaf mold	105	525
Subtotal		284	1420
Total		1168	4587

3.3. Evaluation Indicators

In this paper, average precision (AP) and frames per second (FPS) are used as important evaluation metrics for the detection of tomato leaf disease to analyze the network detection performance.

(1) Average Precision (AP)

AP is the area enclosed by the PR curve and the coordinate axis and is calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{AP} = \int_0^1 \text{p}(\text{r})\text{dr} \tag{16}$$

In the formula, TP is the number of predicted bounding boxes that are correctly categorized and have the correct coordinates of the bounding box, FP is the number of predicted bounding boxes that are incorrectly categorized, FN is the number of bounding boxes that are not predicted, p (Precision) is the precision rate, and r (Recall) is the recall rate.

(2) Frames Per Second (FPS)

FPS is the number of frames per second transmitted, which indicates the number of images that can be processed in a second or the time it takes to process an image to evaluate the detection speed, the shorter the time, the faster the speed. The calculation formula is as follows:

$$\text{FPS} = 1/\text{Latency} \tag{17}$$

3.4. Experimental Scheme

3.4.1. Determination of Training Parameters

The original YOLOv5 model, under the premise of the initial learning rate of 0.0001 and batch-size of 16, the model on the PASCAL VOC2012 and COCO datasets achieved good detection results. On this basis, according to the commonly used empirical values of network training hyperparameters, the hyperparameters of the TDGA network are finally determined after repeated tests, as shown in Table 6.

Table 6. Single leaves dataset.

Epoch	Batch	Lr	Input-Shape
100	16	0.0001	640 × 640

3.4.2. Test Scheme

In order to test the performance of the model TDGA proposed in this paper in tomato disease image detection task, comparative experiments are conducted with TDGA with traditional target detection methods such as Faster R-CNN, SSD [41], YOLOv7 [42], YOLOHC, YOLOv8, RetinaNet and YOLOv5. The experiment divides the total dataset into the training sets, testing sets, and validation sets according to the ratios of 80%, 10%, and 10%, which are used to train the model and conduct the test, and AP is selected as the index to test the detection performance of this paper's method, and FPS, training time, and single-image prediction time are selected as the indexes to verify the detection efficiency, TDGA, is proposed in this paper.

In addition, detection comparison experiments are also conducted on the construction of a divided tomato single-leaf dataset, tomato multiple-leaf dataset, and shadow dataset. To test the generalization ability of the model and verify the robustness, TDGA is proposed in this paper.

3.4.3. Ablation Experiments

To verify the effectiveness of changing the YOLOv5 model CIoU to EIoU, the normal convolution in the C3 module to SAConv, and the GAM is added after up-sampling and down-sampling and in the SPPF module, eight sets of ablation experiments were performed on the total dataset.

YOLOE: On the basis of the original YOLOv5 network, the CIoU was changed to EIoU.

YOLOS: Based on the original YOLOv5 network, the ordinary convolution in the C3 module was changed to SAConv.

YOLOA: Based on the original network, the GAM was added to after the up-sampling and down-sampling and in the SPPF module.

YOLOES: On the basis of YOLOE, the ordinary convolution in the C3 module was changed to SAConv.

YOLOEA: On the basis of YOLOE, the GAM was added after the up-sampling and down-sampling and in the SPPF module.

YOLOSA: On the basis of YOLOS, the GAM was added after the up-sampling and down-sampling and in the SPPF module.

$TDGA_{1:}$ Based on YOLOES, only the GAM was added after the up- and down-sampling.

$TDGA_2$: Based on YOLOES, only the GAM was added to the SPPF module.

TDGA: Based on YOLOES, the GAM was added after the up-sampling and down-sampling and in the SPPF module, which reflected the method proposed in this paper.

4. Results and Analysis

4.1. Overall Detection Performance

In order to verify the detection performance of the TDGA model, the training set, test set and validation set are randomly divided according to the ratio of 80%, 10%, and 10% on the total dataset. Models such as Faster R-CNN, SSD, YOLOv5, RetinaNet and YOLOv7 are selected for the experiments and compared with the method in this paper. Some of the detection results are shown in Figure 9.

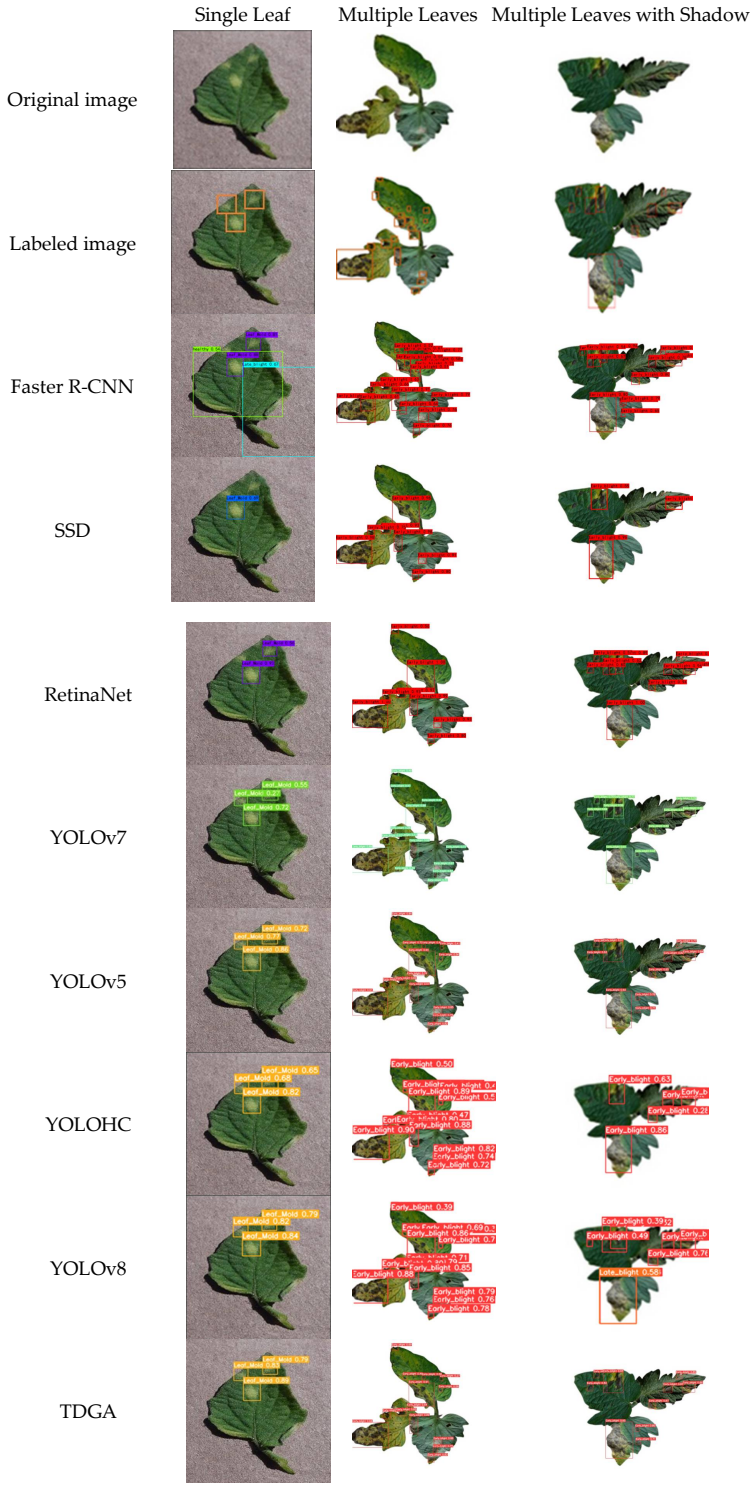


Figure 9. Effect of detection using different algorithms.

From Figure 9, it can be seen that TDGA has a higher detection accuracy for tomato leaf diseases, multiple-leaf, and shadow conditions compared to the other five methods, and there are fewer misses and misdetections for diseases with small targets, which proves that the method in this paper has a high detection performance.

The performance of the detection of tomato leaf disease using TDGA and the other five models was compared using mAP, FPS, training time, and single image prediction time as metrics. The results are shown in Tables 7–9.

Table 7. Five-fold cross-validation.

Experimental	mAP/%
Experiment 1	91
Experiment 2	91.5
Experiment 3	90.9
Experiment 4	92.30
Experiment 5	91.30
Average	91.4

Table 8. Performance comparison of different methods for detection of tomato leaf disease.

Method	mAP/%
Faster R-CNN	75.20
SSD	83.33
YOLOv5	88.80
RetinaNet	80.53
YOLOv7	80.00
YOLOHC	87.80
YOLOv8	88.90
TDGA	91.40

Table 9. Comparison of the efficiency of different methods for detection of tomato leaf disease.

Method	Training Time/h	Single Image Prediction Time/ms	FPS
Faster R-CNN	12.5	69.4	14.40
SSD	9.13	21.3	46.85
YOLOv5	6.89	11.8	84.74
RetinaNet	7.72	32.2	30.98
YOLOv7	8.97	21.2	47.17
YOLOHC	7.76	20.6	48.54
YOLOv8	8.05	20.9	47.84
TDGA	7.69	20.5	48.78

The average accuracy of TDGA in the five-fold cross-validation experiments was 91.4%. The maximum difference in the accuracy of the model in the five cross-validation experiments does not exceed 0.9%, indicating that TDGA has some stability in tomato leaf disease detection and the model has strong generalization ability.

As can be seen from Table 8, the mAP of TDGA is 91.40%, which is 9.68%, 21.54%, 2.93%, 13.49%, 14.25%, 4.1%, and 2.81% higher than that of SSD, Faster R-CNN, YOLOv5, RetinaNet, YOLOv7, YOLOHC and YOLOv8, respectively. The experimental data prove that the addition of the GAM improves the feature extraction ability of the model, and the detection accuracy of various tomato leaf disease image datasets is higher, which further proves the performance of the method in this paper.

From Table 9, we can see that the training time of TDGA is 7.69 h, which is 64.25%, 19.97%, 1.44%,17.87%, 0.91%, and 4.68% faster than Faster R-CNN, SSD, RetinaNet, YOLOv7, YOLOHC, and YOLOv8, respectively; the single-image prediction time of TDGA is 20.5 ms, which is faster than Faster R-CNN, SSD, RetinaNet, YOLOv7, YOLOHC, and YOLOv8 are 48.9 ms, 0.8 ms, 11.7 ms, 0.7 ms, 0.1 ms, and 0.4 ms faster, respectively; the FPS of TDGA

with batch size = 1 is 48.78, which is 239%, 4.12%, 57.46%, 0.49%, and 1.96% faster than Faster R-CNN, SSD, RetinaNet, YOLOv7, YOLOHC, and YOLOv8, respectively, 3.41%.

Compared with the YOLOv5 model, the training time of TDGA is 11.42% more, and the single-image prediction time is increased by 9.2 ms, which is due to the fact that the YOLOv5 model itself has a simpler model structure and a smaller number of parameters, which reduces the model training time as well as the single-image prediction time. However, as can be seen from Figure 9 and Table 8, its detection accuracy is not as good as that of this paper's method TDGA. Overall, in this paper's method, by replacing the CIoU with the EIoU, adding the GAM, and replacing the ordinary convolution in the C3 module with the SAConv, the detection time is increased compared to the YOLOv5 model, but the model detection accuracy is also improved.

4.2. Performance of Disease Detection for Single Tomato Leaf

Detection experiments were conducted on single tomato leaf image datasets of different diseases using AP as an indicator and the comparison results are shown in Table 10.

Table 10. Comparison of detection performance of disease detection in single tomato leaf.

Method	Health	Early Blight	Late Blight	Leaf Mold	mAP/%
Faster R-CNN	99.01	45.63	67.79	76.55	72.24
SSD	99.02	49.79	77.00	78.04	76.19
YOLOv5	99.40	91.70	97.20	91.50	94.90
RetinaNet	99.99	58.62	73.78	89.42	80.45
YOLOv7	99.40	73.40	94.20	33.90	75.20
YOLOHC	99.30	84.80	94.40	84.60	90.80
YOLOv8	99.50	91.40	96.80	91.30	94.75
TDGA	99.50	93.30	99.20	92.50	96.10

As can be seen from Table 10, the AP of this paper's method for the healthy, early blight, late blight, and leaf mold tomato leaf image datasets are 99.5%, 93.3%, 99.2%, and 92.5%, respectively, and the mAP for the overall dataset is 96.1%. It was 33.02%, 26.13%, 1.26%, 19.45%, 27.79%, 5.84%, and 1.42% higher than Faster R-CNN, SSD, YOLOv5, RetinaNet, YOLOv7, YOLOHC, and YOLOv8, respectively, and TDGA was the most effective for detection.

It can also be seen in Table 10 that the average accuracy of the tomato leaf image dataset for healthy and late blight is higher than that of the early blight view and the leaf mold view due to the similar presence of pathological features of the three diseases, which can have a negative impact on detection.

4.3. Performance of Disease Detection for Multiple Tomato Leaves

Detection experiments were conducted on the tomato multi-leaf dataset using AP as an indicator, and the comparison results are shown in Table 11.

Table 11. Comparison of detection performance of disease detection in multiple tomato leaves.

Method	Health	Early Blight	Late Blight	Leaf Mold	mAP/%
Faster R-CNN	97.87	59.28	80.20	62.15	74.87
SSD	98.69	58.17	79.09	56.99	73.23
YOLOv5	97.80	79.30	69.70	58.00	76.20
RetinaNet	94.66	66.41	79.77	63.52	76.09
YOLOv7	96.70	72.10	63.70	54.60	71.80
YOLOHC	98.10	79.80	75.30	60.00	78.30
YOLOv8	97.50	80.45	69.50	66.55	78.50
TDGA	98.50	81.00	70.20	68.30	79.50

As can be seen from Table 11, the AP of TDGA in the three tomato leaf datasets of healthy, early blight, late blight, and leaf mold were 98.5%, 81%, 70.2%, and 68.3%, respectively, and the mAP of the overall dataset was 79.5%, which was higher than that of Faster R-CNN, SSD, YOLOv5, RetinaNet, YOLOv7, YOLOHC and YOLOv8 methods, respectively 6.18%, 8.56%, 4.33%, 4.48%, 10.72%, 1.53% and 1.27%, and TDGA has the best detection effect.

In addition, it can also be seen from Tables 10 and 11 that there is a significant difference in the average accuracy AP of the same detection method for both single and multiple leaves, indicating that multiple leaves have a large impact on the detection results.

4.4. Influence of Shadow on the Performance of Disease Detection

Based on the above experiments, all the methods in this paper are better than other methods, so the only selected experimental control model is the YOLOv5 model, using AP as an indicator. The comparison results are shown in Table 12.

Table 12. Influence of shadows on the performance of disease detection.

Method	Percentage of Shadows/%	Health	Early Blight	Late Blight	Leaf Mold	mAP/%
YOLOv5	0	97.6	81.1	71.7	60.4	77.7
	(0, 5]	95.5	79.8	70.8	59.9	76.5
	(5, 10]	97.2	80.9	71.3	60.6	77.5
TDGA	0	98.7	82.7	73.5	63.4	79.6
	(0, 5]	97.1	81.1	71.6	62.6	78.1
	(5, 10]	98.6	82.4	72.8	63.8	79.4

As can be seen from Table 12, the AP of TDGA in healthy, early blight, late blight, and leaf mold three tomato leaves in the three datasets were 98.7%, 82.7%, 73.5%, and 63.4% in shadow percentage of 0; 97.1%, 81.1%, 71.6%, and 62.6% in shadow percent-age of (0, 5]; and (5%, 10%] in shadow percentage of 98.6%, 82.4%, 72.8%, and 63.8%, which are 2.45%, 2.09%, and 2.45% higher than the YOLOv5 method, respectively. TDGA has the best detection effect.

In addition, it can also be seen from Table 12 that the average accuracy AP of the same detection method for the detection of leafy diseases with different shadow percentages are all significantly different. At shadowing of (0, 5%], the effect of time in healthy images is relatively large, while at (5%, 10%], the AP is not very different from that of images without shadowing. This is because, at a shadow percentage of (0, 5%], the shadow interfered with the model's discrimination, resulting in healthy leaves being determined as other diseases. The large effect on healthy leaves, early blight, and late blight, and the smallest effect on leaf mold are due to the fact that the pathological features of leaf mold are quite different from those of the shadows. And when the percentage of the shadow is (5, 10%], because the shadow area is larger, it is easier to recognize the relatively small area of shadow, and the model misjudgment probability is even smaller. Therefore, it shows that the multilobed shadows at (0, 5%] have a certain influence on the detection effect.

After comparing the experimental results of each method on each dataset, the AP of TDGA is higher than the other comparison models, which proves that the method of this paper has improved the detection accuracy and effect on the three categories of datasets, and also verifies that the method of this paper has a strong generalization ability, which successfully verifies the robustness of this paper's method.

4.5. Results of Ablation Experiments

Ablation experiments were carried out based on the ablation experiment scheme described in Section 3.4.3, and mAP was used as an indicator. The results of the experiments are shown in Table 13.

Table 13. Performance of different ablation methods.

Method	mAP/%
YOLOv5	88.80
YOLOE	89.10
YOLOS	89.60
YOLOA	90.30
YOLOSA	90.60
YOLOES	89.70
YOLOEA	90.90
$TDGA_1$	90.20
$TDGA_2$	90.40
TDGA	91.40

From the results of the ablation experiments in Table 13, it can be seen that the mean average precisions (mAP) of the methods YOLOE, YOLOES, YOLOEA, $TDGA_1$, $TDGA_2$, and TDGA are all improved compared with that of YOLOv5, which illustrates that replacing the CIoU with the EIoU can improve the accuracy of the model detection to a certain extent.

SAConv is added on top of YOLOE and YOLOv5. It can be seen that the accuracy is improved relative to YOLOE and YOLOv5, which illustrates that replacing the ordinary convolution in the C3 module with SAConv expands the sensory field and increases the perceptual ability of the network, improving the model's ability to detect and recognize objects at different scales.

Meanwhile, YOLOEA, YOLSA, $TDGA_1$, $TDGA_2$, and TDGA have better evaluation indexes than YOLOv5, YOLOE, and YOLOS, which indicates that the model detection accuracy can be improved by adding the GAM after the up-sampling and down-sampling module or SPPF module. TDGA improves more than $TDGA_1$ and $TDGA_2$, which indicates that the model detection accuracy can be improved more obviously by adding the GAM after the up-sampling and down-sampling module and the SPPF module at the same time, while TDGA has the highest mAP in this paper's method, which proves the validity of this paper's method.

The training detection efficiency of each ablation experiment model was also compared, and the results are shown in Table 14.

Table 14. Efficiency of different ablation methods.

Method	Training Time/h	Single Image Prediction Time/ms
YOLOv5	6.89	11.8
YOLOE	6.83	11.3
YOLOS	7.22	16.2
YOLOA	8.10	16.4
YOLOSA	8.27	20.4
YOLOES	6.60	15.9
YOLOEA	7.49	19.2
$TDGA_1$	7.19	14.8
$TDGA_2$	7.56	18.4
TDGA	7.69	20.5

As can be seen from Table 14, compared with the original YOLOv5, YOLOE reduces the training time and prediction time relative to the other models, indicating that replacing CIoU with EIoU optimizes the computational structure of the model, balances the samples, and improves the computational speed. TDGA, YOLOEA reduces the training time relative to the YOLOS and YOLOA models, and the single-image prediction. There is a small increase in time, indicating that replacing the ordinary convolution in the C3 module with SAConv and adding the GAM will increase the number of parameters of the model, resulting in an increase in training time, while in the EIoU role, which optimizes the model's computational method, the training time is reduced. While YOLOSA, YOLOEA, $TDGA_1$,

$TDGA_2$, and TDGA model training time, as well as single-image prediction time are increased, because of the addition of the GAM in the up-sampling and down-sampling and SPPF modules and the attention mechanism, the complex structure of the GAM increases the number of parameters, resulting in a decrease in the speed of computation. It can be seen from Table 13 that the addition of the EIoU and the GAM into the network model can bring about a substantial improvement in accuracy and reduction in training time.

5. Discussion

The TDGA model, using image processing techniques, provides farmers with an accurate disease detection tool. This tool not only strengthens the accuracy of disease identification but also enables timely and effective control measures, reducing damage to crop health. In addition, the TDGA model significantly improves crop yields through early disease detection and treatment, while it can help agribusinesses make more scientific management decisions. The integration and application of the model can enhance the level of intelligence and automation in agriculture, thus improving the overall efficiency of agricultural production.

The main idea of TDGA includes adopting a dual attention mechanism GAM, switchable atrous convolution SAConv, and loss function EIoU. In Section 3.4.3 ablation experiments, all these aspects are experimented with compared to other complementary approaches. For applying the GAM, TDGA adds the GAM in up-sampling, down-sampling, and SPPF, and the complementary approach consists of adding the GAM in some of the three positions mentioned above. For SAConv, the complementary approach is the normal convolution under the C3 module in YOLOv5. For EIOU, the complementary approach is the CIOU used in YOLOv5. Based on the results of the ablation experiments, the TDGA proposed in this paper outperforms all of these complementary approaches in terms of overall performance.

The results and analysis section analyzes the performance of different detection methods under various environmental factors (single leaf, multiple leaves, and shadows). A comparison of Tables 10–12 shows that TDGA performs differently in different environments. The detection performance is highest in the single-leaf case, and the performance is relatively poor in the case of multiple leaves and a large proportion of shadows. Therefore, in practical applications, before utilizing TDGA for disease monitoring on tomato leaf images, two approaches can be used to improve the overall performance of the TDGA method. (1) Reducing the proportion of shadows in tomato leaf images by shadow detection and removal methods. (2) Converting multi-leaf images into single-leaf images for disease detection using image semantic or instance segmentation methods.

In practical applications, the efficiency of the TDGA model for tomato disease detection needs to be considered, which is related to the scalability of the whole system and the response speed when the application scale is large. As seen from Table 9, under the experimental configuration conditions shown in Table 1, the time required for detecting a single picture with TDGA is about 20 ms, which can meet the corresponding real-time requirements of the system. The TDGA model can detect about 50 tomato leaves per second and complete the detection of about 180,000 pictures per hour, which can satisfy the detection needs of large-scale farmland applications. In addition, the system's response speed can be improved by upgrading the hardware and software configuration of the system.

The tomato disease detection solution proposed in this paper requires capturing and analyzing images of tomatoes in agricultural fields. The collection, transmission, and storage of similar crop images may pose security and privacy issues for agricultural information. Therefore, when deploying the TDGA model into the farmland, it is recommended that a local area network (LAN) be constructed for data transmission and storage. Suppose the data are to be transmitted over the Internet. In that case, encryption of the images and disease-related data and the deployment of network facilities such as firewalls should be conducted.

6. Conclusions

A network target detection method based on deep learning, TDGA, was proposed in this paper. The model uses SAConv to replace the ordinary convolution in the C3 module, which expands the sensing field and increases the perceptual ability of the network, improves the detection and recognition ability of the model for objects of different scales, and improves the robustness to scale changes. GAM was added after up-sampling and down-sampling as well as the SPPF module, which can better focus on the image information, which improves the feature extraction ability of the model, therefore causing less omission and misclassification; EIoU replaces CIoU to solve the problems of ambiguous definition of aspect ratio and sample imbalance.

The method proposed in this paper, TDGA, was shown to more effectively detect tomato leaf diseases in images than other test methods. The mAP of the whole dataset of tomato disease images reached 91.40%, 96.10% on the image dataset of tomato leaf diseases in a single leaf, 79.50% on the dataset of tomato leaves with diseases in multiple leaves, and 78.10% on the dataset of tomato leaves with diseases in multiple leaves with shadow. All of these are better than the target detection methods such as Faster R-CNN, SSD, YOLOv5, RetinaNet, YOLOv7, YOLOHC, YOLOv8, etc., and realize more efficient and accurate detection on tomato leaf images, especially on multiple-leaf images and shadow images.

The robustness of the method in this paper is successfully verified through comparison experiments with tomato single-leaf, multiple-leaf, and shadow datasets. It has also been demonstrated that TDGA outperforms other methods under the influence of single leaves, multiple leaves, and shading.

In this paper, although we achieved more accurate detection results for single-leaf, multiple-leaf, and shadow images. However, there is a diversity of complex backgrounds and a loss of leaf symmetry due to leaf shadowing, and tomato leaves in complex backgrounds with multiple leaves are interfered with by shadows, colors, and occlusions, resulting in low detection accuracy, and further experimental studies are needed to investigate these related influencing factors. In addition, tomato leaves or branches can be damaged during the image collection process, so we need to improve our image collection method. Future research work includes the following: (1) we will improve the model for tomato plant leaf images acquired in natural environments. (2) We will extend our work with this method for disease detection on other parts of tomato plants, such as roots, stems and flowers.

In this paper, although we achieved more accurate detection results for single-leaf, multiple-leaf, and shadow images. However, there are still some unresolved issues. Relevant future research directions include how to detect and eliminate shadows in tomato leaf images, how to accurately detect diseases when they are similar in color to the leaves, and how to identify diseases when obstacles such as other leaves or stalks are present. Thus, shadows, colors, and occlusions would be reduced in the detection of tomato pests in the complex background of real environments. Of course, as automated farms evolve, agribusinesses can integrate TDGA models into drones or robotic systems for automated tomato management and fully automated disease detection. To investigate the key features and patterns TDGA learns, interpretable machine learning techniques will be used to explore the model's decision-making process.

Author Contributions: Conceptualization, R.X. and G.W.; methodology, R.X., G.W., L.M., F.Y., P.W. and X.Y.; software, R.X.; validation, R.X.; formal analysis, R.X., F.Y., P.W. and X.Y.; investigation, R.X., G.W., L.M., F.Y., P.W. and X.Y.; data curation, R.X.; writing—original draft. R.X.; resources, G.W. and L.M.; supervision: G.W.; writing—review and editing, G.W.; funding acquisition, L.M. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Key Research and Development Program of Zhejiang Province (Grant number: 2021C02005) and Zhejiang Provincial Commonweal Projects (Grant number: LGG21F020001).

Informed Consent Statement: Not applicable.

Data Availability Statement: The code for our proposed model TDGA and dataset used in the experiments can be found on GitHub: https://github.com/zafucslab/TDGA (accessed on 11 May 2024).

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

References

1. Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; Gao, W. Pre-trained image processing transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12299–12310.
2. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. Maxim: Multi-axis mlp for image processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5769–5780.
3. Mazhar, S.; Sun, G.; Bilal, A.; Li, Y.; Farhan, M.; Awan, H.H. Digital and Geographical Feature Detection by Machine Learning Techniques Using Google Earth Engine for CPEC Traffic Management. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1192752. [CrossRef]
4. Tăbăcaru, G.; Moldovanu, S.; Răducan, E.; Barbu, M. A Robust Machine Learning Model for Diabetic Retinopathy Classification. *J. Imaging* **2023**, *10*, 8. [CrossRef] [PubMed]
5. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
6. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant disease detection and classification by deep learning. *Plants* **2019**, *8*, 468. [CrossRef] [PubMed]
7. Mo, L.; Xie, R.; Ye, F.; Wang, G.; Wu, P.; Yi, X. Enhanced Tomato Pest Detection via Leaf Imagery with a New Loss Function. *Agronomy* **2024**, *14*, 1197. [CrossRef]
8. Acharya, S.; Kar, T.; Samal, U.C.; Patra, P.K. Performance comparison between svm and ls-svm for rice leaf disease detection. *EAI Endorsed Trans. Scalable Inf. Syst.* **2023**, *10*. [CrossRef]
9. Narla, V.L.; Suresh, G. Multiple feature-based tomato plant leaf disease classification using SVM classifier. In *Machine Learning, Image Processing, Network Security and Data Sciences: Select, Proceedings of the 3rd International Conference on MIND 2021, Raipur, India, 11–12 December 2021*; Springer Nature: Singapore, 2023; pp. 443–455.
10. Liang, J.; Jiang, W. A ResNet50-DPA model for tomato leaf disease identification. *Front. Plant Sci.* **2023**, *14*, 1258658. [CrossRef] [PubMed]
11. Lv, M.; Su, W.H. YOLOV5-CBAM-C3TR: An optimized model based on transformer module and attention mechanism for apple leaf disease detection. *Front. Plant Sci.* **2024**, *14*, 1323301. [CrossRef] [PubMed]
12. Sabrol, H.; Satish, K. Tomato plant disease classification in digital images using classification tree. In Proceedings of the 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, Tamilnadu, India, 6–8 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1242–1246.
13. Jaisakthi, S.M.; Mirunalini, P.; Thenmozhi, D. Grape leaf disease identification using machine learning techniques. In Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Gurugram, India, 6–7 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
14. Zu, L.; Zhao, Y.; Liu, J.; Su, F.; Zhang, Y.; Liu, P. Detection and segmentation of mature green tomatoes based on mask R-CNN with automatic image acquisition approach. *Sensors* **2021**, *21*, 7842. [CrossRef]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Xie, X.; Ma, Y.; Liu, B.; He, J.; Li, S.; Wang, H. A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. *Front. Plant Sci.* **2020**, *11*, 751. [CrossRef]
17. Syed-Ab-Rahman, S.F.; Hesamian, M.H.; Prasad, M. Citrus disease detection and classification using end-to-end anchor-based deep learning model. *Appl. Intell.* **2022**, *52*, 927–938. [CrossRef]
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. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
20. 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; pp. 779–788.
21. Qi, J.; Liu, X.; Liu, K.; Xu, F.; Guo, H.; Tian, X.; Li, M.; Bao, Z.; Li, Y. An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. *Comput. Electron. Agric.* **2022**, *194*, 106780. [CrossRef]
22. Wang, Y.; Wang, Y.; Zhao, J. MGA-YOLO: A lightweight one-stage network for apple leaf disease detection. *Front. Plant Sci.* **2022**, *13*, 927424. [CrossRef] [PubMed]
23. Liu, J.; Wang, X. Tomato disease object detection method combining prior knowledge attention mechanism and multiscale features. *Front. Plant Sci.* **2023**, *14*, 1255119. [CrossRef] [PubMed]
24. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.

25. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
26. Wang, Y.; Zhang, P.; Tian, S. Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion. *Front. Plant Sci.* **2024**, *15*, 1382802. [CrossRef] [PubMed]
27. Liu, J.; Wang, X.; Zhu, Q.; Miao, W. Tomato brown rot disease detection using improved YOLOv5 with attention mechanism. *Front. Plant Sci.* **2023**, *14*, 1289464. [CrossRef]
28. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Tao, X.; Michael, K.; Fang, J.; Imyhxy; Lorna, W.; et al. ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo* **2022**. [CrossRef]
29. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561.
30. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10213–10224.
31. 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]
32. 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, pp. 12993–13000.
33. Liu, Z.; Yin, H.; Wu, X.; Wu, Z.; Mi, Y.; Wang, S. From shadow generation to shadow removal. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4927–4936.
34. Latif, J.; Tu, S.; Xiao, C.; Rehman, S.U.; Sadiq, M.; Farhan, M. Digital forensics use case for glaucoma detection using transfer learning based on deep convolutional neural networks. *Secur. Commun. Netw.* **2021**, *2021*, 4494447. [CrossRef]
35. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
36. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24261–24272.
37. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
38. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
39. Available online: https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf (accessed on 1 June 2022).
40. Li, Q.; Xie, B.; You, J.; Bian, W.; Tao, D. Correlated logistic model with elastic net regularization for multilabel image classification. *IEEE Transac-Tions Image Process.* **2016**, *25*, 3801–3813. [CrossRef]
41. 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: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
42. Wang, C.Y.; Bochkovskiy, A.; Liao HY, 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; pp. 7464–7475.