scientific reports

OPEN

DGS-Yolov7-Tiny: a lightweight pest and disease target detection model suitable for edge computing environments

Ping Yu[1,3,4], Baoshu Zong[2,4], Xiaozhong Geng[1,4], Hui Yan[5]✉, Baijin Liu[4,6], Cheng Chen[4,6], Hupeng Liu[2,4] & Xiaoqing Xu[2,4]

Pest detection is vital for maintaining crop health in modern agriculture. However, traditional object detection models are often computationally intensive and complex, rendering them unsuitable for real-time applications in edge computing. To overcome this limitation, we proposed DGS-YOLOv7-Tiny, a lightweight pest detection model based on YOLOv7-Tiny that was specifically optimized for edge computing environments. The model incorporated a Global Attention Module to enhance global context aggregation, thereby improving small object detection and increasing precision. A novel fusion convolution, DGSConv, replaced the standard convolutions and effectively reduced the number of parameters while retaining detailed feature information. Furthermore, Leaky ReLU was replaced with SiLU, and CIOU was substituted with SIOU to improve the gradient flow, stability, and convergence speed in complex environments. The experimental results demonstrate that DGS-YOLOv7-Tiny performs excellently on the tomato leaf pest and disease dataset, with 4.43 million parameters, 10.2 GFLOPs computational complexity, and an inference speed of 168 FPS, achieving 95.53% precision, 92.88% recall, and 96.42% mAP@0.5. The model delivered faster inference and reduced computational requirements while maintaining competitive performance, offering an efficient and effective solution for pest detection in smart agriculture with substantial theoretical and practical value.

Keywords Lightweight model, Depthwise separable convolution (DSC), Object detection, Mobile edge computing (MEC), Artificial Intelligence(AI)

Smart agriculture at the forefront of modern agricultural development is increasingly incorporating advanced technologies, such as the Internet of Things (IoT), big data, edge computing, cloud computing, and artificial intelligence, to enable intelligent, precise, and efficient production. Among these, object detection technology plays a critical role by facilitating the real-time identification of key targets, such as crops, pests, and agricultural machinery, providing accurate data to support agricultural operations. However, traditional object detection models are computationally intensive and complex, making it challenging to balance the real-time performance with accuracy. This challenge is particularly acute in edge computing environments, where efficient utilization of limited computational resources and power is essential for effective object detection.

Edge computing minimizes data transmission latency and enhances system response speed by transferring data processing tasks from the cloud to the network edge. This approach is particularly well-suited for smart agriculture, where real-time performance is essential[1–8]. In such environments, lightweight models reduce the computational load and storage demands while preserving detection accuracy, making them ideal for deployment and operation on edge devices[9–16].

The lightweight object detection in edge computing environments has emerged as a significant research focus in recent years. Integrating deep learning and edge computing technologies enables efficient and accurate object

[1]School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China. [2]School of Energy and Power Engineering, Changchun Institute of Technology, Changchun 130012, China. [3]School of Computer Science and Technology, Jilin University, Changchun 130015, China. [4]Jilin Province S&T Innovation Center for Physical Simulation and Security of Water Resources and Electric Power Engineering, Changchun Institute of Technology, Changchun 130012, China. [5]School of Computer Science and Technology, Suqian University, Suqian 223800, China. [6]School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China. ✉email: 17151@squ.edu.cn

detection on resource-constrained devices, thereby facilitating various practical applications. Given the limited computational resources and storage capacity of edge devices, the development of lightweight object detection models and algorithms is essential to achieve real-time object detection in such environments.

This study primarily focuses on tomato diseases and pests within agricultural pest management. Traditionally, the identification and diagnosis of tomato diseases and pests have relied mainly on manual inspection by agricultural experts. However, this approach is not only time-consuming and labor-intensive but also prone to errors due to individual experience and subjective judgment. In recent years, deep learning technologies, particularly convolutional neural networks (CNNs) and object detectors such as the YOLO series, have significantly advanced the development of pest and disease recognition techniques.

Zarboubi[17] et al. proposed a lightweight model, CustomBottleneck-VGGNet, based on transfer learning for tomato leaf disease image recognition. The model achieved a recognition accuracy of 99.12% with only 1.4 million parameters, demonstrating strong deployment capability on mobile devices and suitability for resource-constrained agricultural environments. Bellout[18] et al. evaluated various YOLO-based deep learning models for tomato leaf disease detection. By integrating the PlantDoc and PlantVillage datasets, the optimized YOLOv5 achieved a mean Average Precision (mAP) of 93.1% and was successfully deployed on smartphones, enhancing the system's practicality in agricultural scenarios. Zarboubi[19,20] et al. also developed an innovative pest management solution combining deep learning and the Internet of Things (IoT). They designed a portable pest counting device based on YOLOv8, deployed on a Raspberry Pi 4B and connected to the Firebase platform for real-time monitoring, aiding farmers in optimizing pest control, increasing yields, and reducing environmental impact. Zeng[21] et al. introduced a lightweight improved YOLOv5 algorithm that integrates MobileNetV3 modules and channel pruning, effectively reducing model size while improving detection speed and accuracy. This algorithm realized real-time localization and ripeness detection of tomato fruits, with the optimized model quantized via the NCNN framework and successfully deployed on Android mobile devices, demonstrating strong practicality and scalability. Wang[22] et al. proposed a lightweight detection framework, TomatoGuard-YOLO, based on an improved YOLOv10 architecture. The framework incorporates the Multi-Path Inverted Residual Unit (MPIRU) and Dynamic Focusing Attention Framework (DFAF) to enhance feature extraction and region focus. Utilizing the Focal-EIoU loss function, the model achieved 94.23% mAP50, an inference speed of 129.64 FPS, and a compact size of 2.65 MB, exhibiting both high efficiency and practical applicability.

This study addressed the application of crop pest and disease target detection, taking tomato leaf diseases and pests as an example, focusing on certain challenges, such as limited edge device resources, the need for improved lightweight models, and the difficulty of balancing accuracy and speed. To address these issues, we proposed DGS-YOLOv7-Tiny as a lightweight pest and disease detection model optimized for edge computing environments that could offer an effective solution for pest and disease monitoring. The main contributions of this study are summarized as follows.

(1) This study introduced DGS-YOLOv7-Tiny as an efficient and lightweight object detection model. The model incorporated a Global Attention Module (GAM) to enhance global contextual information aggregation, improved the detection of small objects, and increased accuracy. The innovative DGSConv convolution replaced the traditional convolutions, retaining detailed information and strengthening feature extraction. The SiLU activation function optimized the nonlinear mapping, enhanced the gradient flow, and mitigated the gradient vanishing. Furthermore, the SIOU Loss localization function accelerated the training convergence and improved the detection precision. Together, these advancements enabled DGS-YOLOv7-Tiny to perform object detection tasks efficiently and effectively.

(2) This study presented a lightweight design for an object detection model optimized for resource-constrained mobile devices and embedded systems. The DGS-YOLOv7-Tiny model employed the innovative DGSConv convolution, along with depthwise separable convolutions and ghost convolutions to replace traditional convolutions. These optimizations reduced the model parameters, computational load, and memory usage, thereby effectively lowering the computational complexity. Consequently, the model achieved real-time object detection on resource-limited devices while maintaining high performance.

(3) The performance of the DGS-YOLOv7-Tiny model was thoroughly analyzed, demonstrating the robust generalization and reliability of the agricultural pest detection dataset. The experimental evaluations optimized the relationship between confidence, F1 score, precision, recall, and accuracy, with an optimized balance between precision and recall observed at confidence levels of 0.4–0.6. Ablation experiments further confirmed the effectiveness of the model. Compared with the other models, the proposed DGS-YOLOv7-Tiny achieved superior lightweight performance while maintaining excellent accuracy, demonstrating that its overall performance exceeded that of the comparison models.

(4) The DGS-YOLOv7-Tiny model had significant potential for real-time object detection applications. Its efficiency, accuracy, and real-time capabilities made it a valuable technological asset in advancing smart agriculture. With continued technological progress and the expansion of application scenarios, DGS-YOLOv7-Tiny is expected to further drive the development of smart agriculture.

The remaining sections of this paper are organized as follows: The "Related Technologies" section introduces the architecture of YOLOv7-Tiny. The "Method" section details the optimization modules of the model. The "Experiments and Results" section presents the experimental parameters, model evaluation, ablation experiments, and comparison experiments. The "Conclusion" section discusses the results and provides the final conclusions.

Related technologies

YOLOv7-Tiny algorithm

YOLOv7-Tiny, a lightweight variant of the YOLOv7 series, was designed to minimize computational load and memory usage while preserving high detection accuracy[23]. The architecture of the YOLOv7-Tiny algorithm is illustrated in Fig. 1.

The YOLOv7-Tiny model comprised four main components: Input, Backbone, Neck, and Head. The Input section handled image resizing and data augmentation using techniques such as Mosaic augmentation and adaptive anchor box computation to process images of varying sizes, enhancing the generalization and detection accuracy. The Backbone included CBL convolution, ELAN layers, and MP layers. The CBL convolution extracted features, the ELAN layers improved the network's learning capacity by concatenating multiple CBL convolution, and the MP layers reduced the feature map size by pooling, decreasing the computational load, and enhancing the inference speed. Neck combined Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) to effectively integrate low- and high-level features, thereby improving multi-scale object detection accuracy. Finally, the Head processed the features from the neck, performing object classification, bounding box regression, and confidence prediction to deliver precise detection results.

Methodology

Global attention module

To address complex visual tasks such as object detection, this study incorporated a Global Attention Module (GAM) before the first two branches of the backbone network and three detection heads. The GAM enhanced the aggregation of global context information, enabling the model to perceive the relationships between distant pixels and improved its ability to accurately identify object positions and contours. Global feature extraction is essential for detecting objects in scenes with diverse backgrounds and complex structures[24].

The GAM was integrated into the first two branches of the backbone network to enhance the global context modeling of low-level features. This module enabled the network to capture the overall structure and critical information of the image better, thereby improving its understanding of local details. Consequently, it strengthened the feature representation and enhanced the discriminative ability.

Integrating the GAM before each detection head in the YOLOv7-Tiny model enabled the network to effectively utilize global information, thereby enhancing the target localization and classification accuracy. This could be particularly beneficial for detecting small or occluded targets, as GAM improved positional accuracy and ensured stable and reliable performance in complex scenarios. By reducing information redundancy and noise, GAM allows the lightweight model to focus selectively on key areas while minimizing the influence of irrelevant or noisy information, thereby enhancing the detection precision. The detailed structure of the GAM module is illustrated in Fig. 2.

The GAM applied the global average pooling to the input feature map $X \in \mathbb{R}^{B \times C \times H \times W}$, extracted the global features for each channel, and compressed the spatial dimensions $H \times W$. This process distilled global information for each channel (Eq. (1)).

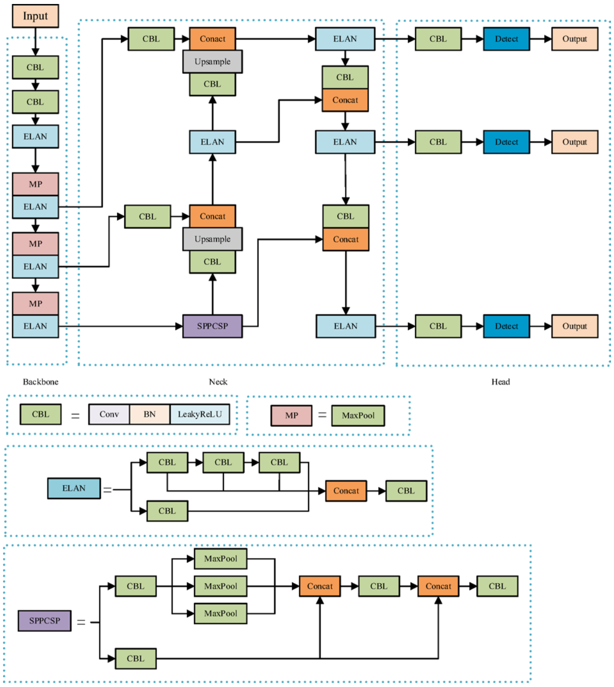


Fig. 1. YOLOv7-tiny algorithm architecture.

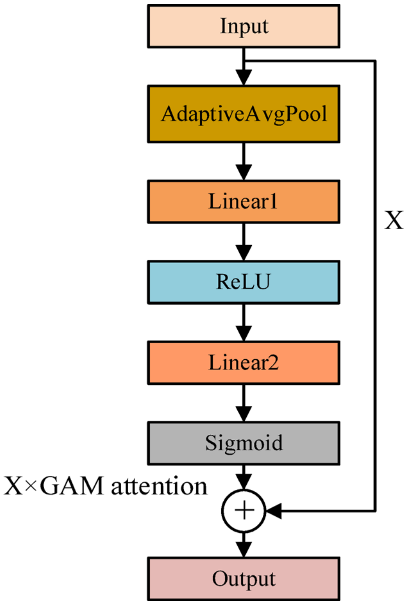


Fig. 2. Structure of the GAM module.

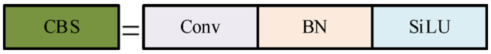


Fig. 3. Structure of the CBS convolution.

$$g = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X\left(i,j\right), g \in \mathbb{R}^{B\times C} \tag{1}$$

where B represents the number of data samples input into the network, and C denotes the number of feature dimensions in the input. The global feature vector g undergoes dimensionality reduction via the first fully connected layer, which applies a reduction weight matrix $W_1 \in \mathbb{R}^{C\times(C/16)}$, reducing the channel number to $\frac{C}{16}$ to decrease the computational complexity and parameter count. The ReLU activation function then introduced nonlinearity. The second fully connected layer adopted an weight matrix $W_2 \in \mathbb{R}^{C\times(C/16)}$ to restore the channel number to C, and the Sigmoid function generated the normalized weights, yielding the channel attention weights a (Eq. (2)):

$$a = \sigma\left(W_2 \cdot ReLU\left(W_1 \cdot g\right)\right), a \in \mathbb{R}^{B\times C} \tag{2}$$

where σ is the Sigmoid activation function, which adaptively generates the attention weights to significantly enhance the expressive power of key channels. Attention weights were applied to the input features X through element-wise multiplication for weighting, whereas residual connections preserved the original input features. This combination enhanced the robustness and discriminative power of the features. The overall formula of this module is shown in Eq. (3)

$$\tilde{X} = X \cdot \sigma\left(W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot g\right)\right) + X \tag{3}$$

SiLU activation function

The Leaky ReLU activation function in all CBL convolution was replaced with SiLU, converting all CBL convolution into CBS convolution[25], as shown in Fig. 3. In general, the CBS convolution adopts either a 3 × 3 or 1 × 1 convolutional kernel, a stride of 1, and utilizes the SiLU activation function to enhance nonlinearity.

The SiLU activation function offered superior adaptability compared with Leaky ReLU. It retained effective activation for positive values while dynamically adjusting the activation based on the input magnitude. Its smooth curve facilitated seamless information flow during forward and backward propagation, avoiding abrupt changes in the negative value range. This enhanced the feature representation of small objects and complex scenes. The smoothness of the function stabilized the gradient flow, mitigating the risks of gradient explosion or vanishing and improving the overall stability of model training.

Replacing Leaky ReLU with SiLU significantly enhanced the nonlinear expression capability of the model, alleviated the gradient vanishing problem, and improved robustness. This substitution also optimized feature

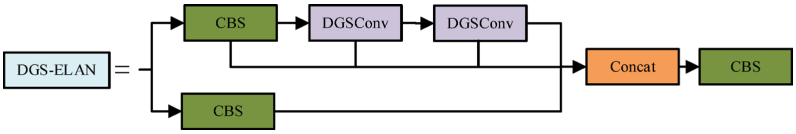


Fig. 4. Structure of the DGS-ELAN module.

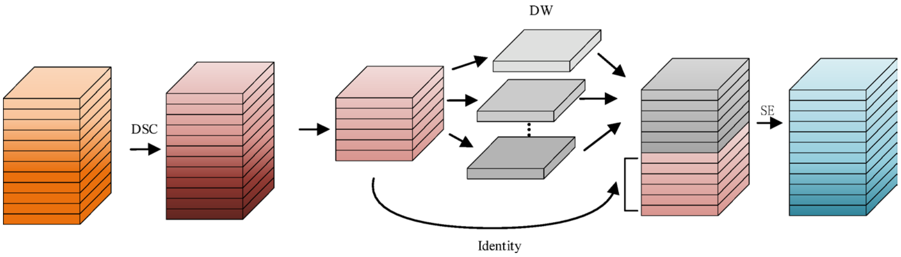


Fig. 5. Structure of the DGSConv convolution.

fusion, accelerated model convergence, and increased recognition accuracy. In complex scenarios, SiLU exceled at capturing subtle features, thereby enhancing the detection performance and enabling the DGS-YOLOv7-Tiny model to achieve superior results in object detection tasks.

DGSConv convolution

To develop a lightweight model, reduce parameters and computational complexity, and enhance the efficiency of the convolution layers, this study introduced a novel fusion convolution, DGSConv. DGSConv integrated Depthwise Separable Convolution (DSC), Ghost modules, and SE attention mechanisms. By replacing the standard convolutions with DGSConv, the ELAN module in the YOLOv7-Tiny model was transformed into a DGS-ELAN module. This modification achieved a balance between retaining detailed information and reducing the parameter count and computational complexity.

The proposed DGS-ELAN module processes the input feature map through two parallel branches. The upper branch sequentially consists of a CBS convolutional layer with a 1 × 1 kernel size and a stride of 1, followed by two stacked DGSConv layers. Each DGSConv layer employs a 3 × 3 kernel with a stride of 1 and maintains the same number of input and output channels. These three layers are connected in a cascaded manner to construct a hierarchical feature extraction pathway. The lower branch contains a single CBS convolutional layer with a 1 × 1 kernel size and a stride of 1. The outputs from the four convolutional layers across both branches are concatenated along the channel dimension. Finally, a CBS convolutional layer with a 1 × 1 kernel, stride of 1, and an output channel number twice that of the input is applied to fuse the concatenated feature maps, thereby enhancing feature representation capability. The structure of the DGS-ELAN module is illustrated in Fig. 4.

The DGSConv convolution was integral to the DGS-ELAN module, enhancing both the efficiency and accuracy of the DGS-YOLOv7-Tiny model. The structure of the DGSConv convolution is shown in Fig. 5.

Depthwise Separable Convolution (DSC) was employed to process the input feature maps[26]. DSC consisted of two operations: depthwise convolution and pointwise convolution. In this study, the depthwise convolution is implemented with a 3 × 3 kernel and a stride of 1, while maintaining identical input and output channel dimensions and omitting any activation function. This operation performs spatial convolution independently on each input channel, thereby preserving channel-specific information and effectively reducing computational complexity. The pointwise convolution utilizes a 1 × 1 kernel with a stride of 1 and does not incorporate an activation function. It integrates cross-channel information by combining features from different channels while preserving the spatial resolution of the feature map. This approach reduced the parameter count and computation complexity while maintaining robust feature extraction capabilities, making the model lightweight and suitable for resource-constrained environments.

Following DSC, the Ghost convolution was applied to further reduce the computational costs while preserving most of the feature information[27]. In the Ghost convolution mechanism, a subset of the feature maps—typically half—obtained from depthwise separable convolution is designated as base features (also referred to as cheap features). The remaining ghost features are then generated through lightweight operations, specifically 3 × 3 depthwise convolutions with a stride of 1. This design significantly reduces computational cost while preserving the representational capacity of the network. Finally, the expanded features are processed by batch normalization and the SiLU activation function to improve non-linear representation and network stability. By generating the "pseudo" features through low-cost operations, the Ghost module reduced the computational overhead and parameter count while maintaining model performance. This approach decreased complexity and memory usage, making the model lightweight and suitable for resource-constrained mobile devices and embedded systems.

In the final stage of the DGS-ELAN module, the SE attention mechanism was introduced to capture the dependencies among spatial elements within the feature map. This mechanism generated attention weights to adjust the importance of each position dynamically. A 1×1 convolution is applied to compress the channel dimension of the input feature map to one-sixteenth of its original size, followed by rectified linear unit (ReLU) activation to extract the critical features. Another 1×1 convolution then restored the original channel size, with the sigmoid activation function generating the attention weights to dynamically adjust the weight of each channel. This allowed the network to focus adaptively on relevant information while ignoring irrelevant or redundant data. The SE attention mechanism enhanced the performance in complex backgrounds, improving the model's ability to capture key information and increasing the accuracy and robustness of object detection.

SIOU loss function

In object detection tasks, the loss function can be a critical metric for evaluating the difference between the predictions of the model and ground truth. Although the Complete IoU (CIOU) loss function in YOLOv7-Tiny accounted for overlap, center point distance, and aspect ratio, it did not fully address the spatial alignment between objects. The scalable intersection over union (SIOU) loss function, an enhancement of IoU, optimized the bounding box regression by incorporating the bounding box scale and angle information[28]. This refined approach considered the shape, orientation, and distance differences between the predicted and ground truth boxes. The SIOU loss function enhanced the accuracy and robustness of the model, particularly in complex scenes, thereby improving its overall performance in object detection tasks.

The SIOU loss function consisted of four components: IOU, distance, aspect ratio, and shape losses. The IOU loss as the fundamental term evaluated the overlap between the predicted and ground truth boxes by calculating the ratio of their intersection area to their union area. This metric assessed the degree of overlap and the quality of the match between boxes (Eq. (4)).

$$\mathrm{IOU} = \frac{\mathrm{Area\,(Intersection)}}{\mathrm{Area\,(Union)}} \tag{4}$$

The distance loss measured the Euclidean distance between the center points of the predicted and ground-truth boxes, aiming to minimize this distance. By constraining the predicted box center to align closely with the ground truth box center, the distance loss enhanced the localization accuracy and improved the precision in predicting the target locations. The distance loss, v_{dis}, is defined in Eq. (5).

$$v_{dis} = \frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2} \tag{5}$$

where (x_1, y_1) represents the center coordinates of the predicted box, and (x_2, y_2) represents the center coordinates of the ground truth box. The minimal enclosing box is a rectangular box that contains both predicted and ground truth boxes. Moreover, (x_{min}, y_{min}) and (x_{max}, y_{max}) represent the coordinates of the bottom-left and top-right corners of the minimal enclosing box, respectively.

In real-world object detection applications, target shape information is crucial for accurate recognition. To address this, the SIOU loss function incorporates the consistency of bounding box aspect ratios in addition to the center point distance. By accounting for these geometric properties, the SIOU loss function is better suited for handling targets with varying shapes.

The aspect ratio loss adjusted the difference between the aspect ratios of the predicted and ground truth boxes, aligning them more closely to improve the localization accuracy. The aspect ratio loss, v_{asp}, is defined in Eq. (6).

$$v_{asp} = \frac{4}{\pi^2}\left(\arctan\frac{w^g}{h^g} - \arctan\frac{w}{h}\right)^2 \tag{6}$$

where w and h are the width and height of the predicted box, and w^g and h^g are the width and height of the ground truth box. The aspect ratio loss was normalized using the arctan function to mitigate the influence of extreme box ratios on the loss. Its squared form ensured a smoother optimization process and minimized the gradient oscillations. To balance its contribution within the overall loss function, a weight adjustment factor λ_{asp} was introduced, as shown in Eq. (7).

$$\lambda_{asp} = \frac{v_{asp}}{1 - \mathrm{IOU} + v_{asp}} \tag{7}$$

Shape loss can quantify the alignment between the shapes of the predicted and ground truth boxes by focusing on the width-height differences. This study examined the effects of these differences on object detection performance. The shape loss, v_{shape}, was calculated by measuring the width-height disparity between the boxes, as defined in Eq. (8):

$$v_{shape} = 2\lambda_{asp} \cdot \frac{|w^g - w|}{w^g} \cdot \frac{|h^g - h|}{h^g} \tag{8}$$

The shape alignment loss improved upon CIOU by optimizing both the area ratio and angle difference, addressing the issues of inadequate shape matching and direction consistency. It comprised two components: area-ratio loss and angle-alignment loss. The area-ratio loss measures the area difference between the predicted and ground-truth boxes, as defined in Eq. (9).

$$v_{shape\_area} = \left(\sqrt{\frac{\text{Area(pred)}}{\text{Area(gt)}}} - 1\right)^2 \tag{9}$$

In summary, the overall SIOU loss function is expressed by Eq. (10):

$$\mathcal{L}_{SIOU} = \text{IOU} - v_{dis} - \lambda_{asp} \cdot v_{asp} - v_{shape} \tag{10}$$

The SIOU Loss enhanced the robustness of the network by incorporating shape information and reducing the degrees of freedom in the loss function. This reduced the risk of overfitting during training, improved the generalization n of the test set, and minimized false negatives and false positives. With smoother gradients and faster convergence, the SIOU Loss significantly accelerated the training. Replacing CIOU Loss with SIOU Loss improved the DGS-YOLOv7-Tiny model's detection accuracy, robustness, and training efficiency.

DGS-YOLOv7-Tiny model

This study introduced DGS-YOLOv7-Tiny serving as a lightweight object detection model designed for edge-computing environments. The structure of the DGS-YOLOv7-Tiny model is shown in Fig. 6.

This study incorporated the GAM into the first two branches of the backbone network and the front part of the three detection heads in the new model. This addition enhanced global context information aggregation, improved small-object detection, and reduced redundancy and noise.

The ELAN modules in the Backbone and Neck were partially replaced with DGS-ELAN modules, and DGSConv convolutions substituted certain CBL convolution. These changes enhanced feature extraction, reduced the model parameters and computational load, and improved the generalization capability.

The Leaky ReLU activation function in the Neck was replaced with SiLU, enhancing the model's nonlinear expression capability, mitigating gradient vanishing, improving the robustness, accelerating the convergence, and increasing the recognition accuracy.

The CIOU Loss function was replaced with SIOU Loss, reducing the degrees of freedom and enhancing robustness. This substitution decreased the risk of overfitting, improved the generalization and training speed, and increased the detection accuracy in complex scenarios.

Experiment and results

A description of the experimental development environment and tools is shown in Table 1.

Dataset and experimental setup

The dataset used in this study is the Tomato Leaf Disease Detection Dataset published on the Roboflow platform. The dataset was split into training, validation, and test sets in a 7:2:1 ratio. All images have a resolution of 640 × 640, with a total of 3,826 images covering seven disease categories: Tomato Early Blight Leaf, Tomato Septoria Leaf Spot, Tomato Leaf Bacterial Spot, Tomato Leaf Late Blight, Tomato Leaf Mosaic Virus, Tomato Leaf Yellow Virus, and Tomato Mold Leaf, along with one healthy class (Tomato Leaf) as a control category.

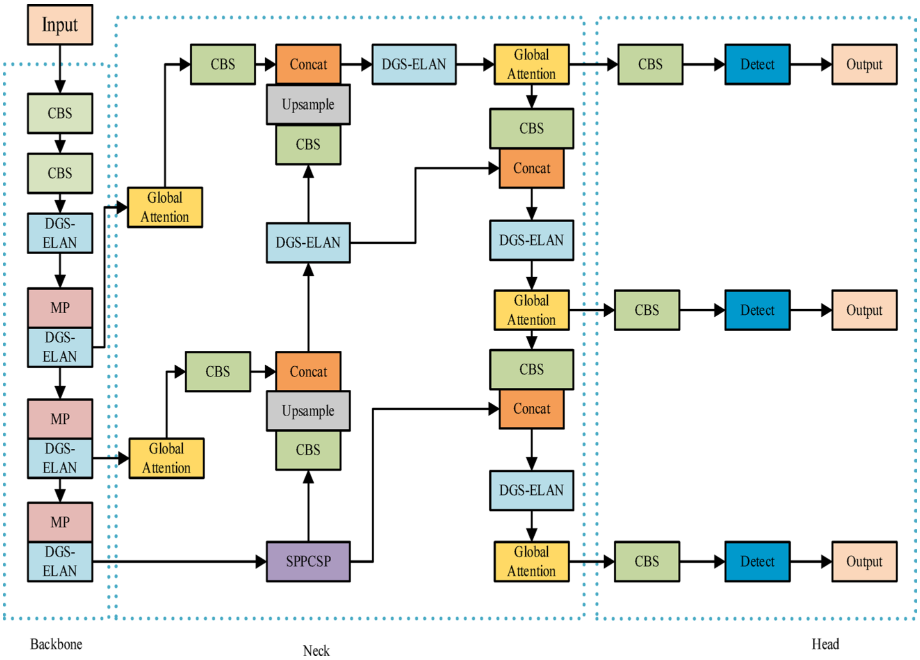


Fig. 6. DGS-YOLOv7-tiny model structure.

Component	Specification
CPU	12th Gen Intel(R) Core(TM) i7-12700F
GPU	GeForce RTX 3060
Operating System	Windows 11
Framework	Python-3.10.15+PyTorch-2.1.0+CUDA12.1
Development Software	Visual Studio Code 2023

Table 1. Development environment and tools experiment parameter configuration.

Category name	Original sample count	Augmented training samples	Validation sample count	Test sample count
Tomato early blight leaf	328	1014	88	44
Tomato Septoria leaf spot	475	1425	142	70
Tomato leaf bacterial spot	480	1440	133	76
Tomato leaf	254	762	77	35
Tomato leaf late blight	408	1224	111	56
Tomato leaf mosaic virus	153	459	46	25
Tomato leaf yellow virus	299	897	66	34
Tomato mold leaf	326	978	117	51
All	2680	8040	764	382

Table 2. Table of category distribution for tomato pest and disease dataset.

Considering the relatively limited number of samples in the original dataset, which may restrict the model's ability to learn effective feature representations, data augmentation was applied only to the training set using three automated techniques provided by the Roboflow platform: horizontal flipping, ± 25% saturation adjustment, and the addition of 0.1% random noise. As a result, the training set was expanded from 2680 to 8040 images, while the validation and test sets remained unaugmented to ensure an objective performance evaluation. The final dataset comprises a total of 9186 images at 640 × 640 resolution. The category distribution of tomato pest and disease samples is summarized in Table 2. These augmentations significantly improved the diversity of training data and enhanced the model's generalization ability, thereby strengthening its applicability in real-world scenarios. Sample images from the dataset are shown in Fig. 7.

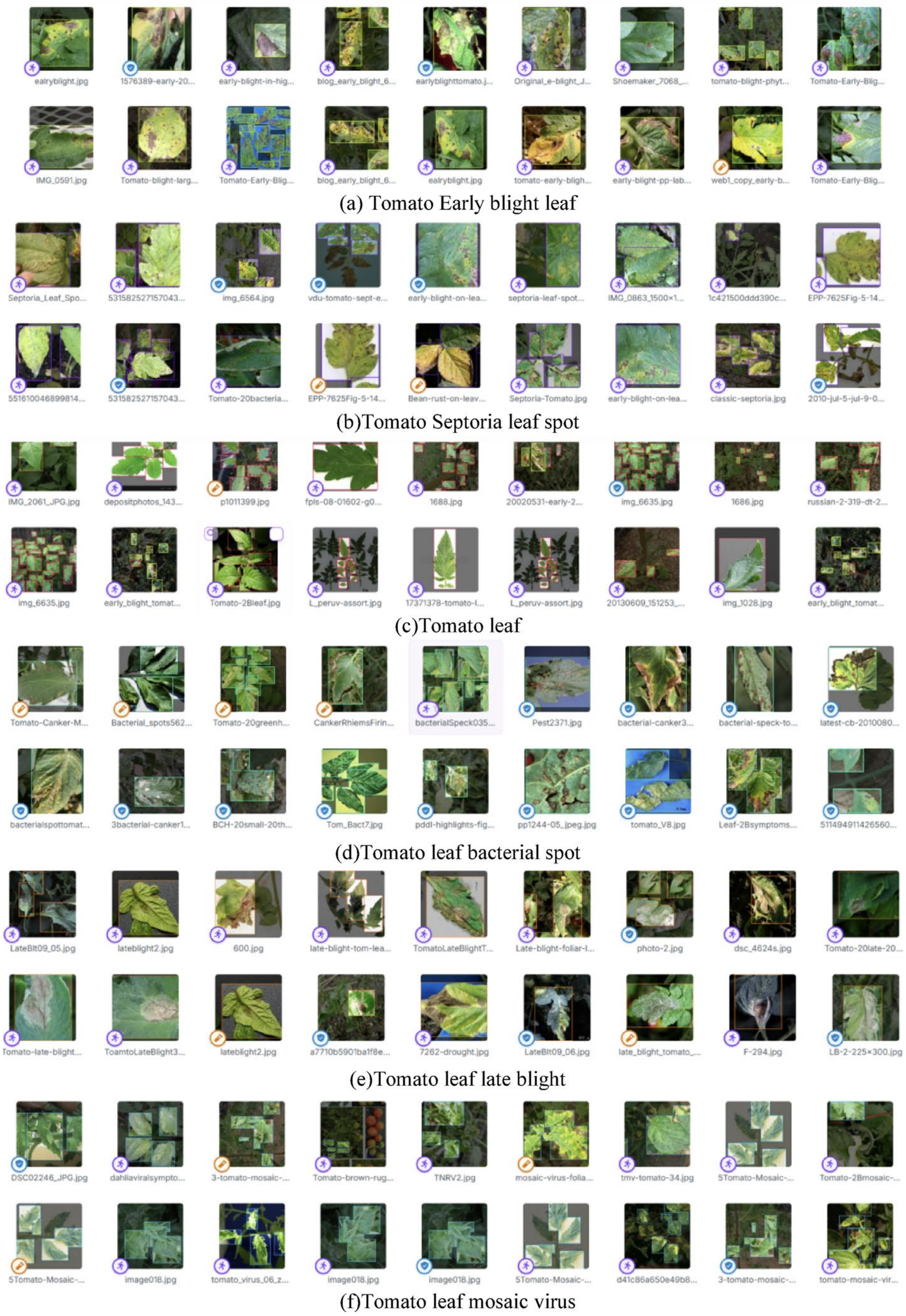
The batch size was set to 16, and the model was trained for 300 epochs. Stochastic Gradient Descent (SGD) was adopted as the optimizer with an initial learning rate of 0.01 and a momentum parameter of 0.937 to enhance the training performance. The learning rate gradually decayed to 0.001 over the epochs, effectively updating the model parameters and improving performance and accuracy. The detailed parameters are listed in Table 3.

Performance testing

Figure 8 presents the confusion matrix used to evaluate the performance of various commonly used models, illustrating its object detection accuracy across different categories and providing an intuitive view of the classification accuracy and error analysis. For clarity, the following abbreviations were employed for tomato leaf disease types: Tomato Early Blight Leaf (TEBL), Tomato Septoria Leaf Spot (TSLS), Tomato Leaf (TL), Tomato Leaf Bacterial Spot (TLBS), Tomato Leaf Late Blight (TLLB), Tomato Leaf Mosaic Virus (TLMV), Tomato Leaf Yellow Virus (TLYV), and Tomato Mold Leaf (TML). In addition, BF represents background false positives.

The results shown in Fig. 8 indicate that the DGS-YOLOv7-Tiny model demonstrates excellent performance in detecting various categories of tomato leaf diseases. Compared to the latest YOLOv11s and YOLOv12s models, DGS-YOLOv7-Tiny achieves higher accuracy in detection tasks for most categories. Similarly, when compared to YOLOv8 and YOLOv7, the DGS-YOLOv7-Tiny model also attains superior accuracy across multiple pest and disease categories. Relative to the original YOLOv7-Tiny model, DGS-YOLOv7-Tiny exhibits comparable detection accuracy in most categories, and given the optimization in parameter count and computational complexity, the slight differences in accuracy are negligible. Furthermore, compared with older models such as YOLOv5s and YOLOv3-Tiny, DGS-YOLOv7-Tiny clearly outperforms them in detection accuracy for the majority of pest and disease categories. The detection accuracy for the TLMV category is relatively low across all models, likely due to the similarity of its symptoms to other disease classes and the high overlap of characteristic features. Regarding missed detection rates, most categories perform well; however, the TLYV category exhibits a higher missed detection rate, possibly because the symptoms mainly manifest as mild leaf yellowing and curling, with inconspicuous lesion features, making it difficult for the models to effectively distinguish diseased areas from the background, leading to misclassification. Overall, the DGS-YOLOv7-Tiny model demonstrates strong robustness and generalization capability, delivering outstanding detection performance across the vast majority of disease categories.

In image object detection, the interplay between confidence, F1 score, precision, and recall can be critical for evaluating and optimizing model performance. A comparison of these metrics may help identify areas for improvement. For instance, if the precision is low, the confidence threshold can be raised or false positives



(a) Tomato Early blight leaf

(b)Tomato Septoria leaf spot

(c)Tomato leaf

(d)Tomato leaf bacterial spot

(e)Tomato leaf late blight

(f)Tomato leaf mosaic virus

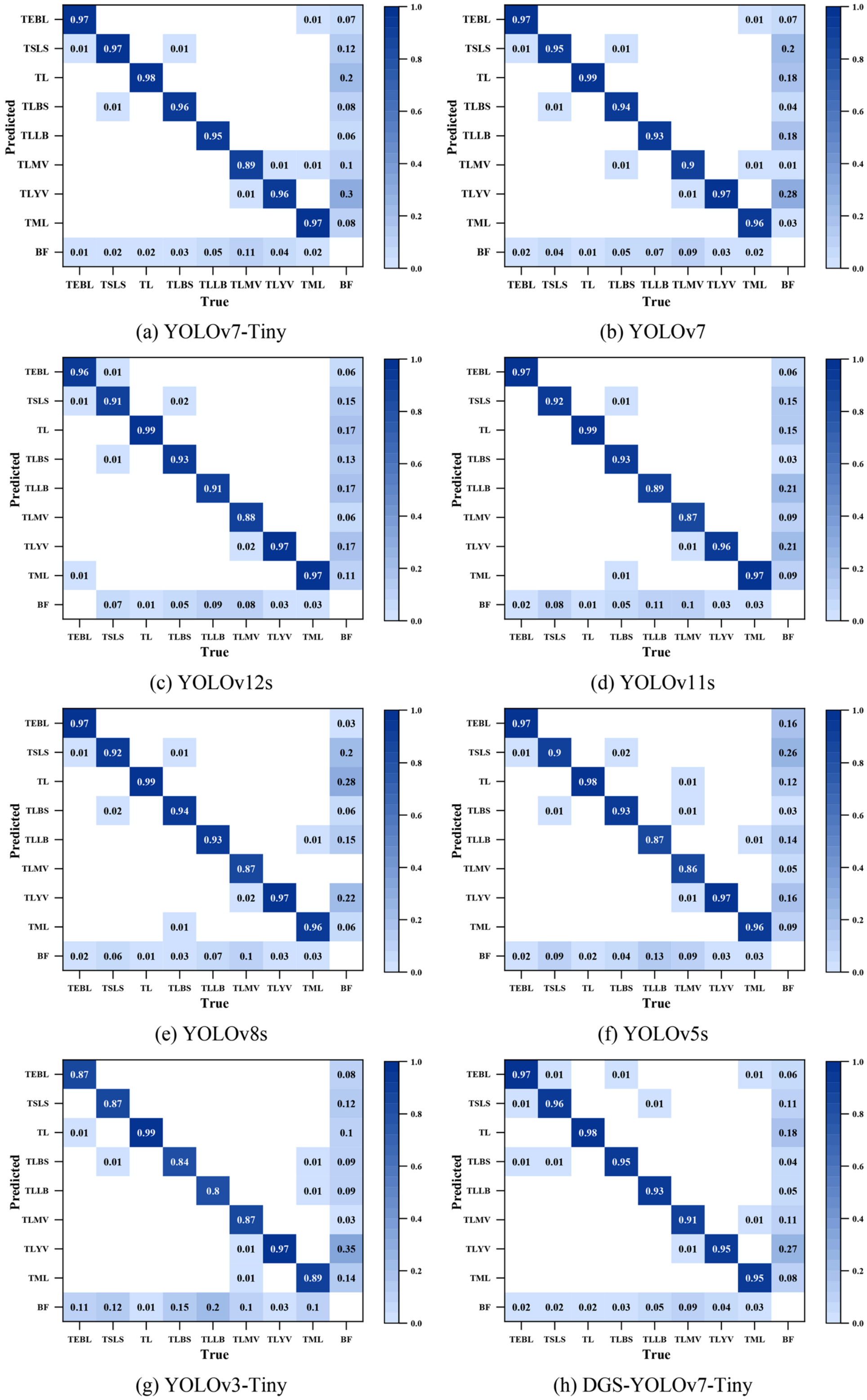
Fig. 7. Sample images from the tomato leaf diseases detection dataset.

Batch size	Epoch	Lr0	Lrf	Opt
16	300	0.01	0.1	SGD

Table 3. Experiment parameter configuration.

minimized. If recall is low, the threshold can be lowered to detect more targets. In practical applications, appropriate metrics and threshold settings should be selected based on specific requirements to achieve optimal detection results.

Precision quantifies the proportion of true-positive instances among all the instances predicted as positive. The calculation formula is shown in Eq. (11).



(a) YOLOv7-Tiny

(b) YOLOv7

(c) YOLOv12s

(d) YOLOv11s

(e) YOLOv8s

(f) YOLOv5s

(g) YOLOv3-Tiny

(h) DGS-YOLOv7-Tiny

Fig. 8. Confusion matrices corresponding to each evaluated model.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

where TP denotes the true positives, and FP denotes the false positives.

Confidence reflects the model's certainty regarding the presence of a target in the detection box, with the confidence threshold directly influencing the precision. Raising the confidence threshold typically increases

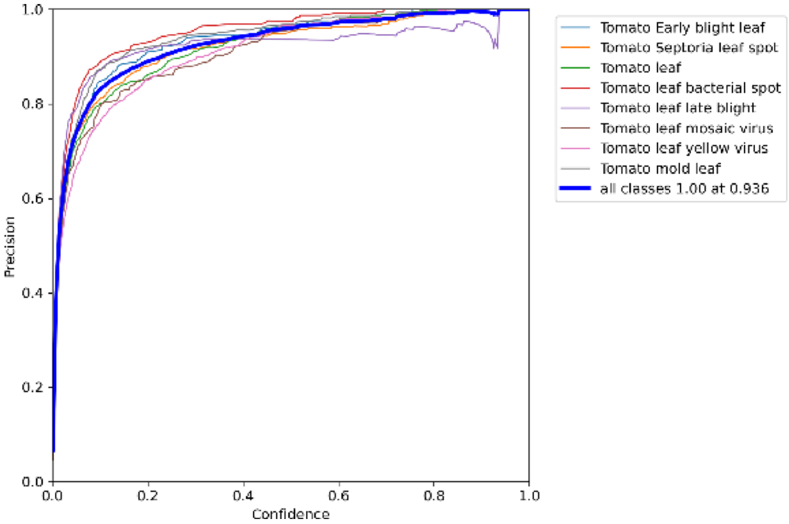


Fig. 9. Precision-confidence curve.

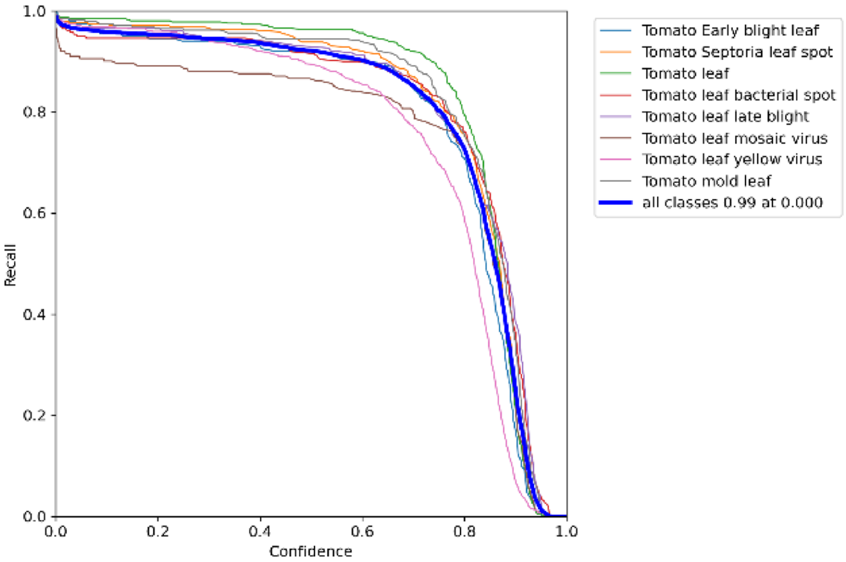


Fig. 10. Recall-confidence curve.

precision but may reduce recall by excluding lower-confidence detections. Conversely, lowering the threshold increases the number of positive predictions, increasing both true positives (TP) and false positives (FP), although FP often increases more rapidly, resulting in reduced precision.

Figure 9 displays the precision-confidence curve of the DGS-YOLOv7-Tiny model. Increasing the confidence threshold reduces false positives, thereby enhancing precision.

Recall quantifies the proportion of true positive instances correctly predicted among all actual positive instances.The calculation formula is shown in Eq. (12).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

where FN denotes a false negative.

Unlike precision, the denominator of recall (TP + FN) represents the total number of positive samples, which is fixed. Recall changes only when TP changes. Lowering the confidence threshold classifies more low-confidence samples as positive, increasing TP and recall, but also increasing FP, which reduces precision. Conversely, raising the threshold improves precision but may decrease recall.

Figure 10 shows the recall confidence curve of the DGS-YOLOv7-Tiny model. Lowering the confidence threshold increased the number of detected objects, thereby improving recall. This approach could be beneficial in scenarios requiring high recall to ensure the detection of as many objects as possible.

The F1 score, calculated as the harmonic mean of precision and recall, can provide a balanced and comprehensive evaluation of the model's performance.The calculation formula is shown in Eq. (13).

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{13}$$

Adjusting the confidence threshold can filter the detection results at varying confidence levels, affecting the F1 score. Analyzing the relationship between confidence and F1 score may help determine the optimal threshold to balance precision and recall, maximizing the F1 score. This optimization could be essential for enhancing model performance, improving detection accuracy, and minimizing false positives.

Figure 11 illustrates the F1 score versus confidence curve for the DGS-YOLOv7-Tiny model. In object detection, a higher F1 score reflects better model performance. The confidence threshold filtered the detection results at various confidence levels. A higher threshold reduces the number of detections but increases precision while lowering recall. Conversely, a lower threshold increases the number of detections but may introduce more false positives.

Figures 9, 10 and 11 illustrate the systematic relationships between confidence, precision, recall, and F1 scores. As confidence increases, precision improves, whereas recall declines. The F1 score peaks when confidence is between 0.4 and 0.6, indicating that the DGS-YOLOv7-Tiny model achieves the optimal balance between precision and recall within this range.

Precision and recall often exhibit a trade-off: increasing recall may decrease precision and vice versa. Adjusting the model parameters or training strategies can improve recall while maintaining precision, or enhance precision while preserving recall. Depending on the application scenario, prioritizing precision may be necessary to ensure result reliability, whereas prioritizing recall may be essential for maximizing target detection.

Figure 12 depicts the precision-recall curve of the DGS-YOLOv7-Tiny model. As recall increased, precision decreased slightly, and the overall performance remained stable. However, when recall approaches 0.9, the precision drops significantly, suggesting that achieving a higher recall may introduce more false positives. Despite this, the DGS-YOLOv7-Tiny model demonstrated excellent performance within a typical range.

Figure 13 shows the training results for the DGS-YOLOv7-Tiny model. During the first 200 epochs, the loss function value decreased rapidly, whereas precision, recall, and mean average precision (mAP@0.5) increased significantly, indicating a rapid improvement in model performance. Between 200 and 250 epochs, the loss reduction slowed, and the growth of precision, recall, and mAP@0.5 stabilized. Beyond 250 epochs, the training set loss curve exhibited a minimal further decrease, and the other metrics remained stable, suggesting that the model was close to convergence. Although the loss in the validation set was slightly higher than that in the training set, both remained within the same order of magnitude, indicating a strong generalization ability and the absence of significant overfitting.

Parameter comparison experiment

As shown in Table 4, a comparative analysis of the number of parameters between YOLOv7-Tiny and DGS-YOLOv7-Tiny was conducted. In DGS-YOLOv7-Tiny, the module format is denoted as CBS [3, 32, 3, 2], which indicates a convolutional block with the SiLU activation function, an input channel count of 3, an output channel count of 32, a kernel size of 3 × 3, and a stride of 2. In contrast, YOLOv7-Tiny uses CBL [3, 32, 3, 2], where the block adopts the Leaky ReLU activation function, while the other parameters remain the same. In the DGS-ELAN module, the notation “1or3” refers to the use of a 1 × 1 convolution kernel in the CBS block and a 3 × 3

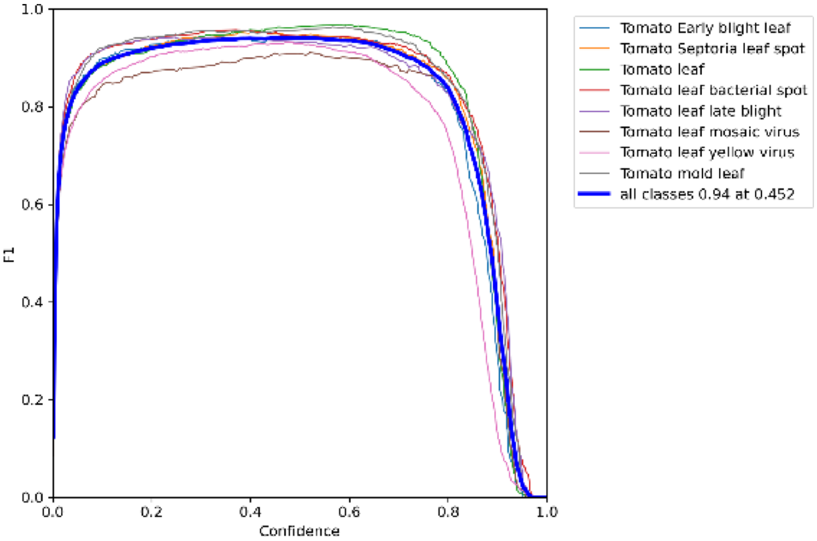


Fig. 11. F1 Score-confidence curve.

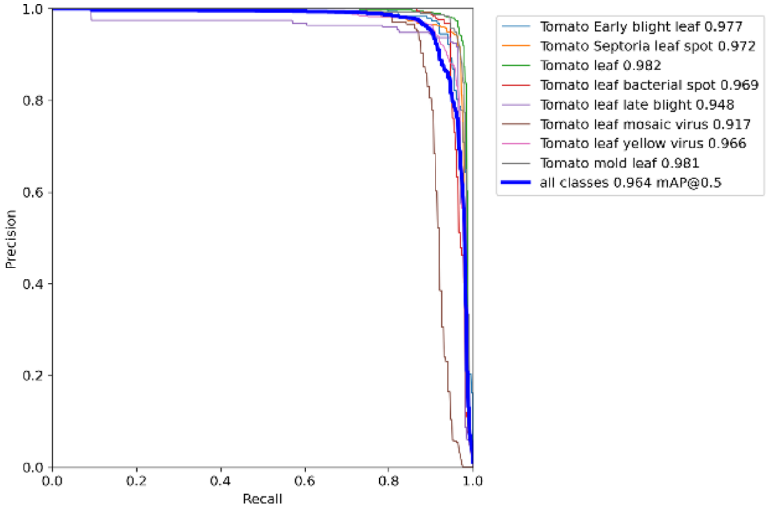


Fig. 12. Precision-recall curve.

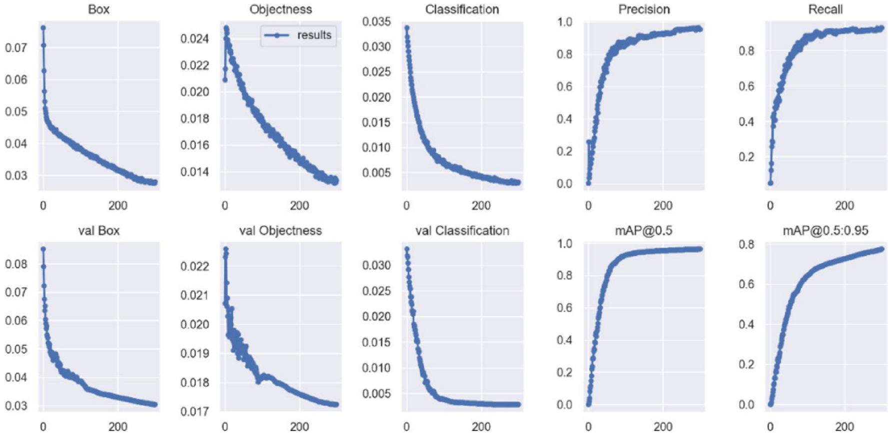


Fig. 13. DGS-YOLOv7-tiny training results.

convolution kernel in the corresponding DGSConv block. The term "Global Attention [256]" denotes a global attention module with 256 input channels.

As shown in the table, DGSConv within the DGS-ELAN module significantly reduces the number of parameters across various output channel configurations. Specifically, when the input and output channels are 256 and 512, respectively, DGS-ELAN reduces the parameter count from 1,838,080 (in the conventional ELAN) to 813,344, achieving a reduction of 1,024,736 parameters, or 55.75%. In the same configuration, the DGSConv module reduces the parameters from 590,336 (in CBL) to 77,968, representing a reduction of 512,368 parameters, or 86.82%. When the input and output channels are 128 and 256, respectively, the parameter count of DGS-ELAN decreases from 460,288 to 205,968, a reduction of 254,320 parameters (55.25%). The corresponding DGSConv module reduces parameters from 147,712 to 20,552, a reduction of 127,160 parameters (86.10%). Similarly, under the configuration of 64 input channels and 128 output channels, the parameters of DGS-ELAN decrease from 115,456 to 52,808, resulting in a reduction of 62,648 parameters (54.26%), while DGSConv reduces the parameter count from 36,992 to 5668, achieving a reduction of 31,324 parameters (84.68%). In summary, DGSConv consistently demonstrates substantial parameter compression across different channel configurations, highlighting its effectiveness in lightweight model design. Moreover, the inclusion of the Global Attention module introduces only a minimal increase in parameters, which is nearly negligible compared to the significant savings achieved by DGSConv.

Ablation experiment

The DGS-YOLOv7-Tiny model demonstrated significant performance improvements across multiple metrics compared with the YOLOv7-Tiny model. A comparison of the evaluation metrics is presented in Table 5.

Replacing the Leaky ReLU activation function with SiLU improved the DGS-YOLOv7-Tiny model's performance compared with the original model, with precision increasing by 1.10% and mAP@0.5 increasing

YOLOv7-Tiny	Params	DGS-YOLOv7-Tiny	Params
CBL [3,32,3,2]	928	CBS [3,32,3,2]	928
CBL [32,64,3,2]	18,560	CBS [32,64,3,2]	18,560
ELAN [64,64,1or3,1]	31,104	ELAN [64,64,1or3,1]	31,104
MP [64,64,2,2]	0	MP [64,64,2,2]	0
ELAN [64,128,1or3,1]	115,456	DGS-ELAN [64,128,1or3,1]	52,808
MP [128,128,2,2]	0	MP [128,128,2,2]	0
ELAN [128,256,1or3,1]	460,288	DGS-ELAN [128,256,1or3,1]	205,968
MP [256,256,2,2]	0	MP [256,256,2,2]	0
ELAN [256,512,1or3,1]	1,838,080	DGS-ELAN [256,512,1or3,1]	813,344
SPPCSP [512,256,1,1]	657,408	SPPCSP [512,256,1,1]	657,408
CBL [256,128,1,1] + Upsample	33,024	CBS [256,128,1,1] + Upsample	33,024
		Global Attention [256]	8192
CBL [128,64,1,1] + Concat	33,024	CBS [128,64,1,1] + Concat	33,024
ELAN [256,128,1or3,1]	140,032	DGS-ELAN [256,128,1or3,1]	77,384
CBL [128,64,1,1] + Upsample	8320	CBS [128,64,1,1] + Upsample	8320
		Global Attention [128]	2048
CBL [128,64,1,1] + Concat	8320	CBS [128,64,1,1] + Concat	8320
ELAN [128,64,1or3,1]	35,200	DGS-ELAN [128,64,1or3,1]	20,004
		Global Attention [64]	512
CBL [64,128,3,2] + Concat	73,984	CBS [64,128,3,2] + Concat	73,984
ELAN [256,128,1or3,1]	140,032	DGS-ELAN [256,128,1or3,1]	77,384
		Global Attention [128]	2048
CBL [128,256,3,2] + Concat	295,424	CBS [128,256,3,2] + Concat	295,424
ELAN [512,256,1or3,1]	558,592	DGS-ELAN [512,256,1or3,1]	172,688
		Global Attention [256]	8192
CBL [64,128,3,1]	73,984	CBS [64,128,3,1]	73,984
CBL [128,256,3,1]	295,424	CBS [128,256,3,1]	295,424
CBL [256,512,3,1]	1,180,672	CBS [256,512,3,1]	1,180,672
Detect	36,074	Detect	36,074

Table 4. Parameter comparison between YOLOv7-Tiny and DGS-YOLOv7-Tiny.

YOLOv7-Tiny	SiLU	SIOU	Global attention	DGSConv	Precision (%)	mAP@0.5 (%)	Parameters (M)
√					94.37	96.48	6.03
√	√				95.41	96.56	6.03
√	√	√			96.94	96.66	6.03
√	√	√	√		96.82	97.02	6.05
√	√	√	√	√	95.53	96.42	4.43

Table 5. Comparison of key model metrics.

by 0.08%. This improvement was attributed to the SiLU's enhanced nonlinear expression capability, which increased its stability during training.

Changing the loss function from CIOU to SIOU improved the DGS-YOLOv7-Tiny model's performance, with precision increasing by 2.72% and mAP@0.5 increasing by 0.19% compared with the original model. This enhancement was primarily owing to the improved object localization accuracy of the SIOU loss function, particularly for targets with uneven scales, where SIOU demonstrated superior performance.

The addition of the Global Attention module improved the DGS-YOLOv7-Tiny model's performance compared with the original model, with precision increasing by 2.60% and mAP@0.5 rising by 0.56%. This improvement was attributed to the module's ability to capture global features more effectively, particularly in complex scenarios, thereby enhancing recognition performance.

Replacing the ELAN module with the DGS-ELAN module resulted in a 26.53% reduction in the parameter count and a 1.23% improvement in precision, whereas mAP@0.5 decreased slightly by 0.06%. This modification demonstrated that the model could maintain high detection accuracy while reducing computational costs. In resource-constrained environments, the DGS-ELAN module significantly enhanced efficiency with a minimal impact on performance.

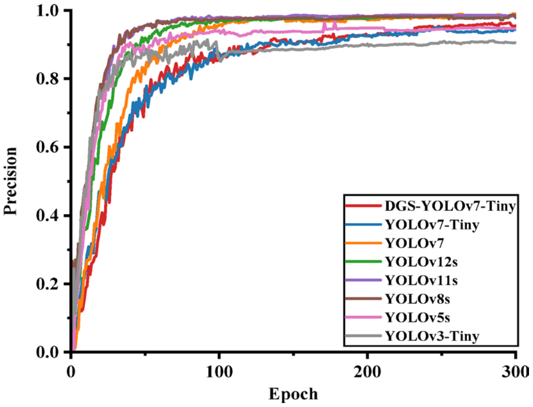


Fig. 14. Comparison of precision results for each model.

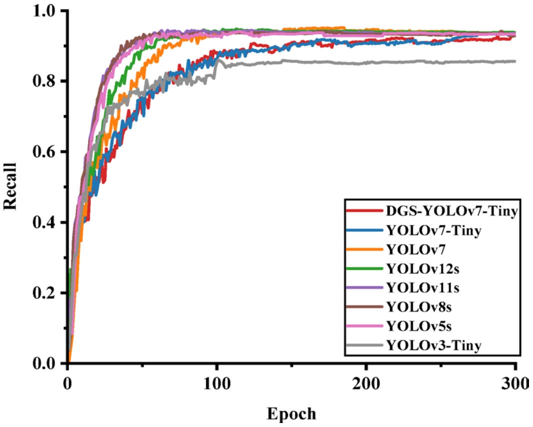


Fig. 15. Comparison of recall results for each model.

In summary, the slight decrease in mAP@0.5 was outweighed by the improvement in precision and the significant reduction in the parameter count, rendering the change negligible. Therefore, the DGS-YOLOv7-Tiny model demonstrated exceptional performance in agricultural pest and disease detection tasks, achieving a high detection accuracy while maintaining low model complexity.

Comparative experiment

To evaluate the performance of the DGS-YOLOv7-Tiny model for image classification, it was compared with several classic object detection models, including YOLOv3-Tiny, YOLOv5s, YOLOv8s, YOLOv11s, YOLOv12s, YOLOv7, and YOLOv7-Tiny. The experiments plot convergence curves and final detection results using training epochs on the x-axis and mAP@0.5, precision, and recall on the y-axis. The experimental results are shown in Figs. 14, 15, and 16.

Figures 14, 15, and 16 demonstrate that the improved DGS-YOLOv7-Tiny model achieved a comparable or even equivalent performance to the other models in terms of mAP@0.5, precision, and recall. This confirmed that the model sustained high detection performance while delivering greater efficiency.

Table 6 presents a comparison of the evaluation metrics, including precision, recall, mAP@0.5, parameter count, and GFLOPs, for the DGS-YOLOv7-Tiny model and the other models.

Params and GFLOPs effectively measured the complexity and computational load of the DGS-YOLOv7-Tiny model. As shown in Table 6, the DGS-YOLOv7-Tiny model exceled in both parameter and computational efficiency, with only 4.43 M parameters, 10.2 GFLOPs, and 168 FPS. Compared with YOLOv12s, the model reduces the number of parameters by 51.21%, lowers GFLOPs by 47.15%, and improves inference speed by 64.71%. Compared with YOLOv11s, the parameter count is reduced by 52.97%, GFLOPs decrease by 52.11%, and inference speed increases by 5.00%. Compared with YOLOv8s, the parameter count is reduced by 60.19%, GFLOPs by 64.21%, and inference speed is increased by 15.86%. Compared with the original YOLOv7, the number of parameters is reduced by 88.24%, GFLOPs by 90.74%, and inference speed is improved by 102.41%. Compared with the baseline YOLOv7-Tiny, the model achieves a 26.53% reduction in parameters, a 22.73% decrease in GFLOPs, and a 3.07% improvement in inference speed. Compared with YOLOv5s, the parameters are reduced by 36.99%, GFLOPs by 35.44%, and inference speed is improved by 6.33%. Compared with

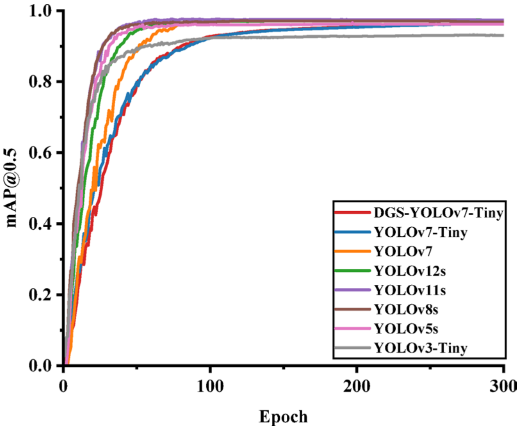


Fig. 16. Comparison of mAP results for each model.

Method	Params(M)	GFLOPs	FPS	Precision (%)	Recall (%)	mAP@0.5 (%)
YOLOv7-Tiny	6.03	13.2	163	94.37	93.45	96.48
YOLOv7	37.68	110.2	83	99.04	93.03	97.07
YOLOv12s	9.08	19.3	102	98.29	93.82	97.31
YOLOv11s	9.42	21.3	160	98.33	93.53	97.33
YOLOv8s	11.13	28.5	145	97.85	93.45	96.76
YOLOv5s	7.03	15.8	158	94.75	93.31	96.19
YOLOv3-Tiny	8.68	12.9	138	90.57	85.59	93.04
DGS-YOLOv7-Tiny	4.43	10.2	168	95.53	92.88	96.42

Table 6. Comparison of key metrics for different models.

YOLOv3-Tiny, the model achieves a 48.96% reduction in parameters, a 20.93% reduction in GFLOPs, and a 21.74% increase in inference speed.

Although the DGS-YOLOv7-Tiny model exhibits slightly lower performance in detection accuracy metrics such as Precision, Recall, and mAP@0.5 compared to the more recent YOLOv11s and YOLOv12s models, it achieves approximately 50% reductions in both parameter count and computational complexity, along with a notable improvement in inference speed. Given the emphasis on lightweight design and real-time performance, the slight degradation in Precision, Recall, and mAP@0.5 is considered acceptable.

Presentation of detection results by DGS-YOLOv7-Tiny

To better demonstrate the performance of the proposed DGS-YOLOv7-Tiny model, detection results under the same experimental settings are presented on the validation set of the tomato leaf disease detection dataset, as shown in Fig. 17. Figure 17a, b, c, d, e, f, g and h illustrate the detection results for: Tomato Early Blight Leaf, Tomato Septoria Leaf Spot, Tomato Leaf Bacterial Spot, healthy Tomato Leaf (normal category), Tomato Leaf Late Blight, Tomato Leaf Mosaic Virus, Tomato Leaf Yellow Virus, and Tomato Mold Leaf, respectively. The proposed DGS-YOLOv7-Tiny model exhibits outstanding performance in the field of tomato leaf disease detection, achieving detection accuracy and reliability comparable to human-level diagnosis.

Discussion

Although the proposed model was trained and evaluated on a tomato leaf dataset, its lightweight architecture and efficient feature extraction modules demonstrate strong generalization capabilities. These characteristics suggest that the model can potentially be adapted to other crops exhibiting similar disease patterns. In future work, we plan to incorporate additional datasets from crops such as potatoes, peppers, and cucumbers, in order to further enhance the model's robustness and applicability in broader agricultural contexts.

While DGS-YOLOv7-Tiny significantly reduces parameter count and inference time compared to baseline models, a slight decrease in detection accuracy (e.g., mAP@0.5) is observed when compared with larger models such as YOLOv12s, YOLOv11s, and YOLOv8s. This trade-off is expected, as model compression and lightweight design often come at the cost of marginal performance degradation. However, in practical applications—particularly those involving edge computing and real-time field deployment—the benefits of improved speed and reduced resource consumption often outweigh the minor loss in accuracy. Therefore, the proposed model achieves a favorable balance between efficiency and precision, making it highly suitable for deployment on resource-constrained agricultural devices.

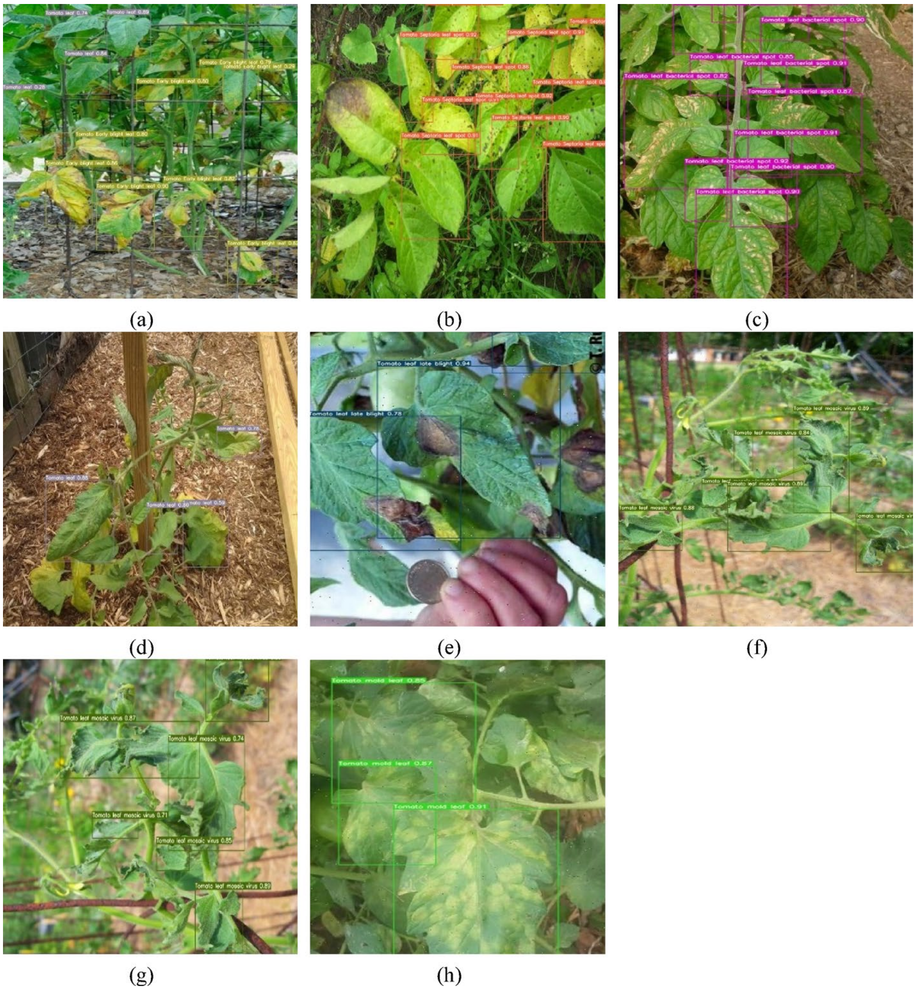


Fig. 17. Detection results of tomato leaf diseases.

Conclusion

To mitigate the harmful effects of pests and diseases on plant health, this study proposes a lightweight pest detection model based on YOLOv7-Tiny, named DGS-YOLOv7-Tiny. The model addresses the challenges of large parameter sizes and high computational requirements that existing object detection models face when applied in real-time edge computing environments. To enhance performance, the model introduces a Global Attention Module before the first two branches of the backbone network and the three detection heads, which improves the aggregation of global contextual information and thereby enhances the accuracy and capability of small object detection.

Additionally, the model employs a novel fusion convolution, DGSConv, to replace the standard convolution, successfully reducing the model's parameter count while retaining detailed feature information. Furthermore, the Leaky ReLU activation function is replaced with SiLU, and CIOU loss is replaced with SIOU loss, improving gradient flow, stability, and convergence speed in complex environments. The model has 4.43 M parameters, 10.2 GFLOPs, 168 FPS, an accuracy of 95.53%, a recall rate of 92.88%, and a mAP@0.5 of 96.42%.

Compared with the original model, DGS-YOLOv7-Tiny is more suitable for deployment on resource-constrained edge devices and embedded systems, demonstrating higher feasibility and practical value in real-world applications. Its lightweight design and efficient architecture make it an ideal candidate for intelligent agricultural scenarios that require real-time detection under limited computational budgets. In future work, we will continue to explore optimization strategies tailored to real agricultural environments, aiming to further improve the model's inference efficiency, adaptability, and robustness on low-power devices, and to better meet the practical needs of large-scale deployment in smart agriculture.

Data availability

The public dataset for Tomato leaf pest and disease detection used in this experiment is sourced from the Roboflow website, available at https://universe.roboflow.com/plantdoc-xztat/proposed. Due to the limited size of the dataset, we utilized Roboflow's built-in image augmentation techniques to expand it. The augmented dataset can be accessed at https://github.com/zongbaoshu/Tomato-leaf-pest-and-disease-detection-data-set.git. The experi mental code for the model used in this study has been open-sourced and is available at: https://github.com/zongbaoshu/DGS-Yolov7-Tiny.

Received: 16 January 2025; Accepted: 23 July 2025
Published online: 14 August 2025

References

1. Kim, Y., Hwang, W. In Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, Revised Selected Papers, Part I 5. 653–662. Springer (2019).
2. Mao, D. et al. Real-time fruit detection using deep neural networks on CPU (RTFD): An edge AI application. *Comput. Electron. Agric.* **204**, 107517. https://doi.org/10.1016/j.compag.2022.107517 (2023).
3. Kolosov, D., Kelefouras, V., Kourtessis, P. & Mporas, I. Anatomy of deep learning image classification and object detection on commercial edge devices: A case study on face mask detection. *IEEE Access* **10**, 109167–109186. https://doi.org/10.1109/ACCESS.2022.3214214 (2022).
4. Bonam, J., Kondapalli, S. S., Prasad, L. & Marlapalli, K. Lightweight CNN models for product defect detection with edge computing in manufacturing industries. *J. Sci. Ind. Res.* **82**, 418–425. https://doi.org/10.56042/jsir.v82i04.72390 (2023).
5. Xu, S. et al. Airborne small target detection method based on multi-modal and adaptive feature fusion. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2024.3443856 (2024).
6. Alqahtani, D.K., Cheema, M.A., Toosi, A.N. In: International Conference on Service-Oriented Computing, 142–150. Springer.
7. Fan, Y., Zhang, L. & Li, P. A lightweight model of underwater object detection based on YOLOv8n for an edge computing platform. *J Mar. Sci. Eng.* **12**, 697. https://doi.org/10.3390/jmse12050697 (2024).
8. Yu, P. et al. An optimized heterogeneous multi-access edge computing framework based on transfer learning and artificial internet of things. *Alex. Eng. J.* **109**, 176–190. https://doi.org/10.1016/j.aej.2024.08.105 (2024).
9. Patrikar, D. R. & Parate, M. R. Anomaly detection using edge computing in video surveillance system. *Int. J. Multimed. Inf. Retr.* **11**, 85–110. https://doi.org/10.1007/s13735-022-00227-8 (2022).
10. Cheng, Q., Wang, H., Zhu, B., Shi, Y. & Xie, B. A real-time uav target detection algorithm based on edge computing. *Drones* **7**, 95. https://doi.org/10.3390/drones7020095 (2023).
11. Yu, C., Zhang, Y., Mukherjee, M. & Lloret, J. Augmented semi-supervised learning for salient object detection with edge computing. *IEEE Wirel. Commun.* **29**, 109–114. https://doi.org/10.1109/MWC.2020.2000351 (2022).
12. Zhou, X., Xu, X., Liang, W., Zeng, Z. & Yan, Z. Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT. *IEEE Internet Things J.* **8**, 12588–12596. https://doi.org/10.1109/JIOT.2021.3077449 (2021).
13. Geng, X. et al. A motor imagery EEG signal optimized processing algorithm. *Alex. Eng. J.* **101**, 38–51. https://doi.org/10.1109/JIOT.2021.3077449 (2024).
14. Deng, X., Li, J., Ma, Y., Guan, P. & Ding, H. Allocation of edge computing tasks for UAV-aided target tracking. *Comput. Commun.* **201**, 123–130. https://doi.org/10.1016/j.comcom.2023.01.021 (2023).
15. Wu, Y. et al. Edge computing driven low-light image dynamic enhancement for object detection. *IEEE Trans. Netw. Sci. Eng.* **10**, 3086–3098. https://doi.org/10.1109/TNSE.2022.3151502 (2022).
16. Setyanto, A., Sasongko, T. B., Fikri, M. A. & Kim, I. K. Near-edge computing aware object detection: A review. *IEEE Access* **12**, 2989–3011. https://doi.org/10.1109/ACCESS.2023.3347548 (2023).
17. Zarboubi, M., Bellout, A., Chabaa, S. & Dliou, A. CustomBottleneck-VGGNet: Advanced tomato leaf disease identification for sustainable agriculture. *Comput. Electron. Agric.* **232**, 110066. https://doi.org/10.1016/j.compag.2025.110066 (2025).
18. Bellout, A., Zarboubi, M., Dliou, A., Latif, R. & Saddik, A. Advanced YOLO models for real-time detection of tomato leaf diseases. *Math. Model. Comput.* **11**, 1198–1210. https://doi.org/10.23939/mmc2024.04.1198 (2024).
19. Zarboubi, M., Bellout, A., Chabaa, S. & Dliou, A. Revolutionizing tomato pest management: Synergy of deep learning, IoT, and precision agriculture. *Math. Model. Comput.* **11**, 1167–1178 (2024).
20. Zarboubi, M., Chabaa, S., Dliou, A. In: 2023 IEEE International Conference on Advances in Data-Driven Analytics and Intelligent Systems (ADACIS). 1–6 (IEEE).
21. Zeng, T., Li, S., Song, Q., Zhong, F. & Wei, X. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. *Comput. Electron. Agric.* **205**, 107625. https://doi.org/10.1016/j.compag.2023.107625 (2023).
22. Wang, X. & Liu, J. TomatoGuard-YOLO: A novel efficient tomato disease detection method. *Front. Plant Sci.* **15**, 1499278. https://doi.org/10.3389/fpls.2024.1499278 (2025).
23. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 7464–7475.
24. Liu, Y., Shao, Z., Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv preprint* arXiv:2112.05561 (2021) https://doi.org/10.48550/arXiv.2112.05561.
25. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint* arXiv:2004.10934 (2020) https://doi.org/10.48550/arXiv.2004.10934.
26. Chollet, F. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 1251–1258.
27. Luo, K. et al. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 1590–1599.
28. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv preprint* arXiv:2205.12740 (2022) https://doi.org/10.48550/arXiv.2205.12740.

Acknowledgements

The authors extend their sincere gratitude to the anonymous reviewers for their insightful comments, which significantly improved the quality of this manuscript. This research was supported by grants from the Science and Technology Department of Jilin Province (Grant Nos. 20240404058ZP)

Author contributions

P.Y. and B.Z. designed the research framework and proposed the DGS-YOLOv7-Tiny model. X.G. and H.Y. were responsible for model implementation and conducting the experiments. B.L. and C.C. handled data collection and preprocessing, including the construction of the tomato leaf pest and disease dataset. H.L. and X.X. performed performance analysis and ablation studies. P.Y. and B.Z. wrote the main manuscript text, and H.Y. prepared Figs.1-17 and Tables 1–6. All authors contributed to manuscript revision and reviewed and approved the final version for submission.

Funding

Supported by Jilin Provincial Science and Technology Department, 20240404058ZP.

Declarations

Competing interests

The authors declare that they have no conflicts of interest.

Additional information

Correspondence and requests for materials should be addressed to H.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.