Research on the Strawberry Disease and Pest Identification Algorithm Based on the YOLOv8 Framework

Teng Zhang, Jing Tao

School of Mechanical Engineering, Sichuan University of Science & Engineering, Yibin 644000, China

Abstract: As an important economic crop, strawberries are highly susceptible to various pests and diseases during their growth, seriously affecting yield and quality. Traditional manual detection methods are time-consuming, labor-intensive, and difficult to apply promptly and comprehensively. To promote the development of intelligent pest and disease detection systems, improve strawberry cultivation efficiency, and ensure strawberry quality, a lightweight Yolo-CAD model based on Yolov8n was proposed, designed to address issues such as multi-labeling and inaccurate detection. The model uses ADown[10] as the downsampling module to reduce feature loss during pooling and optimizes Yolov8s Neck based on the Cross -Scale Feature Fusion Module (CCFM), integrating multi-scale features and contextual information to enhance the models adaptability to scale variations and improve its ability to detect small objects. Experimental results show that the enhanced Yolo-CAD model achieves a mean accuracy (mAP50) of 76.33%, a precision of 76.19%, and a recall of 71.20% on the validation set. Compared to SSD, Retinanet, Yolov3, Yolov5n, Yolov8n, Yolov9t, and Yolov10n, the average precision (mAP50) was improved by 11.58, 6.22, 7.22, 6.25, 5.08, 5.44, and 6.21 percentage points, respectively. Moreover, the size of Yolo-CAD model is only 3.37MB, 43.65% smaller than Yolov8n, making it more suitable for deployment on mobile platforms with limited computational power. The proposed Yolo-CAD model has shown promising results in identifying and detecting strawberry pests and diseases, and can provide technical support for pest prevention and automated harvesting.

Keywords: Strawberry Pest; Machine vision; Deep learning; Cross-scale feature fusion.

1. Introduction

China stands as the worlds largest strawberry producer, demonstrating an overall upward trend in both cultivation area and yield. In 2022, the strawberry cultivation area in China reached 2.2118 million mu, with a yield approaching 4 million tons. Due to the warm and humid growth environment and their proximity to the ground, strawberries are inevitably susceptible to infestation by various diseases and pests. Common afflictions include strawberry anthracnose, powdery mildew, gray mold, and angular leaf spot. These diseases and pests lead to a decline in strawberry quality and yield, causing significant economic losses for growers. However, in natural environments, factors such as occlusion, varying lighting conditions, and inconspicuous disease features pose significant challenges to the object detection of strawberry diseases and pests.

With the advancement of computer technology, deep learning has demonstrated significant advantages in the field of object detection [1]. Compared with traditional detection methods, Convolutional Neural Networks (CNNs) are capable of extracting multi-level, computer-recognizable features from datasets, thereby substantially improving the detection performance and generalization ability of models [2]. For instance, Xiang Xinjian et al. proposed an AM-YOLOX algorithm for strawberry disease and pest detection, which achieved a precision of 97.17% on the validation set [3]. Wang Xingwang et al. focused on the region-based Active Contour Model (CV) without edges; by adding an energy function and utilizing the gray correlation degree as image edge information, they established a Gray Correlation Degree-based Energy Function CV model (GCD-EF-CV) that combines global terms with boundary information terms [4]. Similarly, Wang Weixing et al. constructed a lightweight YOLOv4-G model by substituting traditional convolution with the lightweight Ghost Module, which effectively suppressed interference from complex backgrounds and achieved accurate and rapid detection of lychee pests [5]. Shi Jie et al. replaced the convolutional layers in the feature extraction and fusion networks of YOLOv5s with the lightweight GhostNet and substituted the CIOU loss function with EIOU, significantly improving detection precision while reducing model complexity for successful mobile deployment [6]. Furthermore, Yan Yuncai et al. compared K-means clustering analysis with the Otsu threshold segmentation algorithm, determining that the latter offered simpler operation and superior performance [7]. Yang Kun et al. designed the YOLOv5-VE, which was modified by incorporating the visual attention-based feature enhancement module (CBAM) and the bounding box localization loss function (DIoU), thereby enhancing the models feature extraction capability and mitigating the impact of target overlap and occlusion [8]. Additionally, Xing Weiyin et al. combined DenseNet to reconstruct the feature extraction network of RetinaNet, reinforcing feature reuse [9]. Although object detection has yielded promising results in the field of agricultural product inspection, there remains a need for improvement in both detection precision and speed, particularly for real-time detection involving multi-label tasks and complex scenarios.

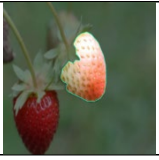
To address existing issues in strawberry disease and pest detection, such as poor precision in multi-label classification and occluded environments, this paper proposes an improved Yolo-CAD model. In this model, the lightweight ADown module [10] is utilized to replace the original down-sampling module of YOLOv8n. Furthermore, the Neck section is modified based on the Cross-Scale Feature Fusion Module (CCFM) [11] to integrate features across different scales and contextual information, thereby enhancing the models adaptability to scale variations and its capability in detecting small-scale objects. Additionally, the original CIoU is replaced with DIoU [12] to guide the model in rapidly finding the optimal solution.

2. Experimental Data and Platform

2.1. Dataset

The study utilized the Strawberry Disease Dataset V13, which was uploaded by MarStrawberry to the RobFlow platform in 2024. This dataset comprises 1,944 images with an input resolution of 640×640 pixels and encompasses 11 distinct categories: Angular Leaf Spot, Anthracnose Fruit Rot, Blossom Blight, Gray Mold, Healthy Strawberry Leaf, Healthy Strawberry, Leaf Spot, Mulch, Inedible Strawberry, Powdery Mildew on Fruit, and Powdery Mildew on Leaf. regarding the dataset partitioning, the images were divided into training, validation, and testing sets, accounting for 74%, 23%, and 3% of the total dataset, respectively.

Table 1. Data set categories

name (of a thing)	Angular Leafspot	Anthracnose Fruit Rot	Blossom Blight	Gray Mold	Healthy-Leaf-Strawberry	Healthy-Strawberry
photograph						
name (of a thing)	Leaf Spot	Mulch	non-edible-Strawberry	Powdery Mildew Fruit	Powdery Mildew Leaf	
photograph						

2.2. Experimental environment and evaluation

The experiments were conducted on the Windows operating system, utilizing the PyTorch deep learning framework with GPU acceleration via CUDA. The specific configuration parameters are presented in Table 2.

Table 2. Experimental training environment configuration

software and hardware platform	Model parameters
operating system	Win11 22H2
processing unit	Intel(R) Xeon(R) w9-3495X 1.9GHz
display card (computer)	NVIDIA RTX A5500
organizing plan	Pytorch 1.12.1 + CUDA 11.6.0
Programming Environment	Python 3.9
display memory	24GB
random access memory (RAM)	64GB×12

To enhance the training efficiency of the model, the initial learning rate (lr0) was set to 0.01, and the Stochastic Gradient Descent (SGD) algorithm was employed as the optimizer. Additionally, the batch size was configured to 16, and the training process was executed for 200 epochs.

3. Research on the Optimization of Object Detection Algorithms

3.1. The YOLOv8 Object Detection Algorithm

Launched by Ultralytics in 2023, YOLOv8 stands as a state-of-the-art object detection algorithm. In comparison with its predecessors, such as YOLOv3 [13], YOLOv5, and YOLOv7 [14], YOLOv8 incorporates a lightweight network architecture and leverages TensorRT engine acceleration. Consequently, a substantial enhancement in inference speed has been achieved while maintaining high detection precision. The YOLOv8 model family comprises five distinct variants: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Among these, YOLOv8x represents the largest model in terms of parameters and complexity, whereas YOLOv8n is the most compact and lightweight version.

The architecture of the YOLOv8 model comprises three primary components: the Backbone, the Neck, and the Head. Furthermore, the model incorporates the Adaptive Spatial Feature Fusion (ASFF) module to accommodate input images of varying resolutions. This integration significantly strengthens feature extraction capabilities and anchor-free point detection performance, thereby enhancing the models proficiency in detecting small-scale objects.

However, during the growth cycle, strawberry leaves frequently overlap. This issue is particularly pronounced when strawberries exhibit a prostrate growth habit, leading to feature occlusion and interference, which subsequently impede accurate identification. Such scenarios exacerbate the complexity of feature detection, necessitating an improvement in the models detection capabilities within complex and dense environments. Consequently, this paper introduces an improved approach that utilizes ADown as the down-sampling module to mitigate the loss of feature information. Additionally, the Neck architecture is optimized to bolster the models robustness and precision in detecting features across varying scales.

Table 3. Yolov8 model

	depth	width	max_channels	layers	parameters	GFLOPs
Yolov8n	0.33	0.25	1024	225	3157200	8.9
Yolov8s	0.33	0.50	1024	225	11166560	28.8
Yolov8m	0.67	0.75	768	295	25902640	79.3
Yolov8l	1.00	1.00	512	365	43691520	165.7
Yolov8x	1.00	1.25	512	365	68229648	258.5

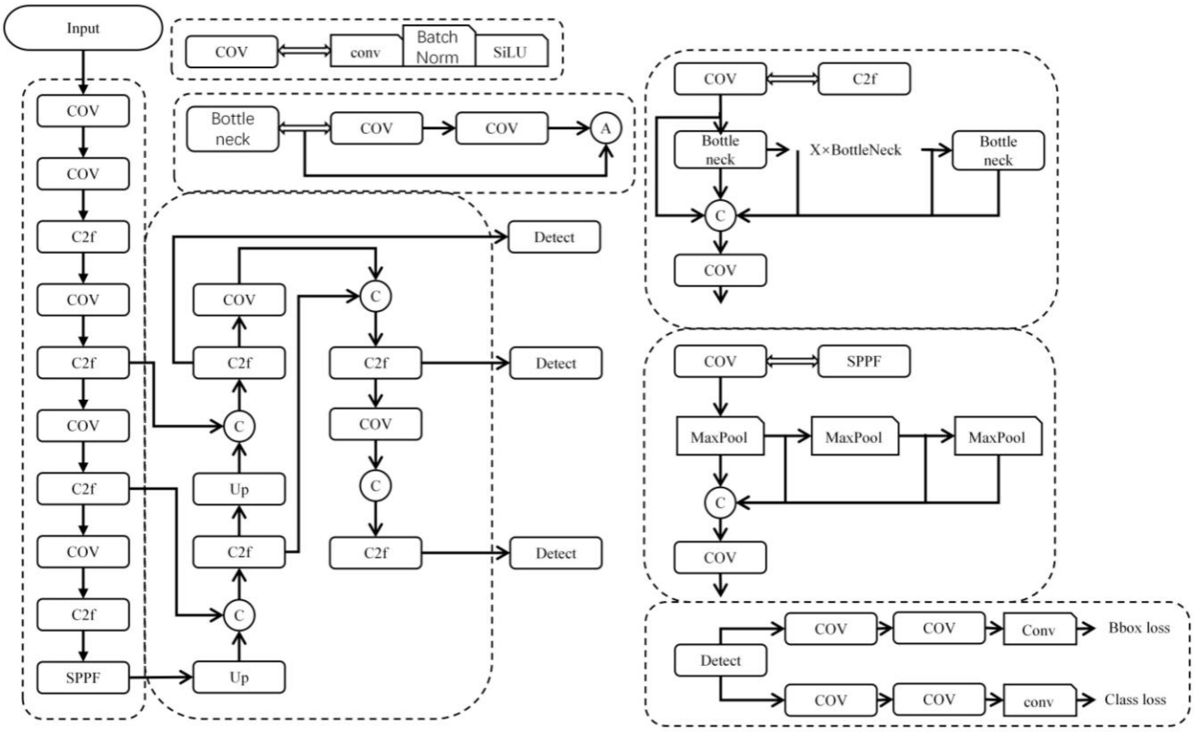


Fig. 1 Yolov8 model structure

3.2. Construction of the Yolo-CAD Model

3.2.1. Integration of the ADown Down-sampling Mechanism

To further optimize the performance and efficiency of the object detection model, the design philosophy of the efficient AConv (Adaptive Convolution) module from YOLOv9 has been incorporated into the down-sampling stage of YOLOv8. Consequently, an enhanced down-sampling module, designated as the ADown module, has been developed. This innovation aims to achieve a simultaneous enhancement in both parameter efficiency and detection precision through the implementation of refined feature extraction and fusion strategies.

The ADown module integrates the strengths of various down-sampling techniques. Initially, Average Pooling (AvgPool) is performed on the input feature maps from the preceding layer, resulting in an output four-dimensional tensor with dimensions of.

$$out(N_i, C_j, h, w) = \frac{1}{kH \times kW} \sum_{m=1}^{kH-1} \sum_{h=1}^{kW-1} input(N_i, C_j, stride[0] \times h + m, stride[1] \times w + n) \tag{1}$$

Where N, C, H, and W represent the batch size, channel count, image height, and image width, respectively, while "stride" denotes the number of pixels by which the convolution kernel shifts during each operation.

This procedure facilitates feature smoothing and the reduction of spatial dimensions while preserving global contextual information. Subsequently, the features are distributed into two parallel processing pathways:

The x1 Convolution Pathway: A 3×3 convolution kernel (k=3) is employed with a stride of 2 (s=2) and padding of 1 (p=1). By expanding the receptive field and reducing spatial resolution, this pathway further extracts local detailed features.

The x2 Max Pooling and Convolution Pathway：A max pooling operation is utilized to capture salient feature points within the image, which are subsequently fed into a convolution module. This module employs a 1×1 convolution kernel (k=1) with a stride of 1 (s=1) and padding of 0 (p=0). This configuration enhances feature representational capability through dimensional transformation without inducing additional reduction in spatial dimensions.

$$out(N_i, C_j, h, w) = \frac{max}{m=0,\dots,kH-1} \frac{max}{n=0,\dots,kW-1} input(N_i, C_j, sride[0] \times h + m, stride[1] \times w + n) \tag{2}$$

Finally, a Concatenation (Concat) operation is employed to merge complementary feature information derived from distinct pathways. This process generates a richer and more comprehensive feature representation, thereby facilitating the execution of subsequent detection tasks.

Extensive training and validation procedures have demonstrated that substituting the original Conv module with the ADown module as the down-sampling unit in YOLOv8 not only significantly reduces the parameter count and alleviates the computational burden but also yields a discernible improvement in detection precision. Consequently, this optimization effectively strikes a balance between model complexity and performance, rendering the realization of real-time, high-efficiency object detection systems more attainable.

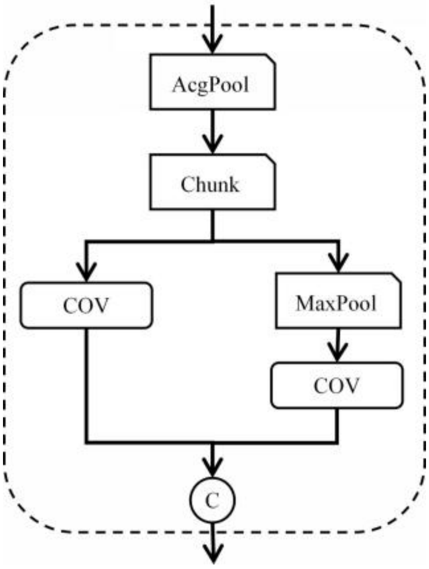


Fig. 2 Structure of ADown module

3.2.2. CCFM Enhanced Neck Performance Optimization

The RT-DETR architecture comprises a backbone network, a hybrid encoder, and a transformer decoder equipped with auxiliary prediction heads. An overview of the model architecture is illustrated in the figure below. In this framework, the output features {S3,S4,S5} derived from the final three stages of the backbone network are utilized as inputs for the encoder. Through mechanisms of intra-scale interaction and cross-scale fusion, the hybrid encoder transforms these multi-scale features into a sequence of image features.

Subsequently, IoU-aware query selection is employed to extract a fixed number of image features from the encoder output sequence, which serve as the initial object queries for the decoder. Finally, the decoder, augmented with auxiliary prediction heads, iteratively optimizes these object queries to generate bounding boxes and confidence scores. This process can be mathematically formulated as follows:

$$Output = CCFM(\{S_3, S_4, S_5\}) \tag{3}$$

$$Q = K = V = Flatten(S_5) \tag{4}$$

$$F_5 = Reshape(Attn(Q, K, V)) \tag{5}$$

In the equation, Q, K, V denote the Query, Key, and Value matrices, respectively, which are derived from the self-attention mechanism of the Transformer model. The term Flatten signifies the process of transforming multi-dimensional inputs into a one-dimensional array, whereas Attn represents Multi-Head Attention. Additionally, Reshape indicates the operation of restoring feature dimensions to align with those of S_5. Finally, S3, S4 and S5 correspond to the output features extracted from the final three stages of the backbone network.

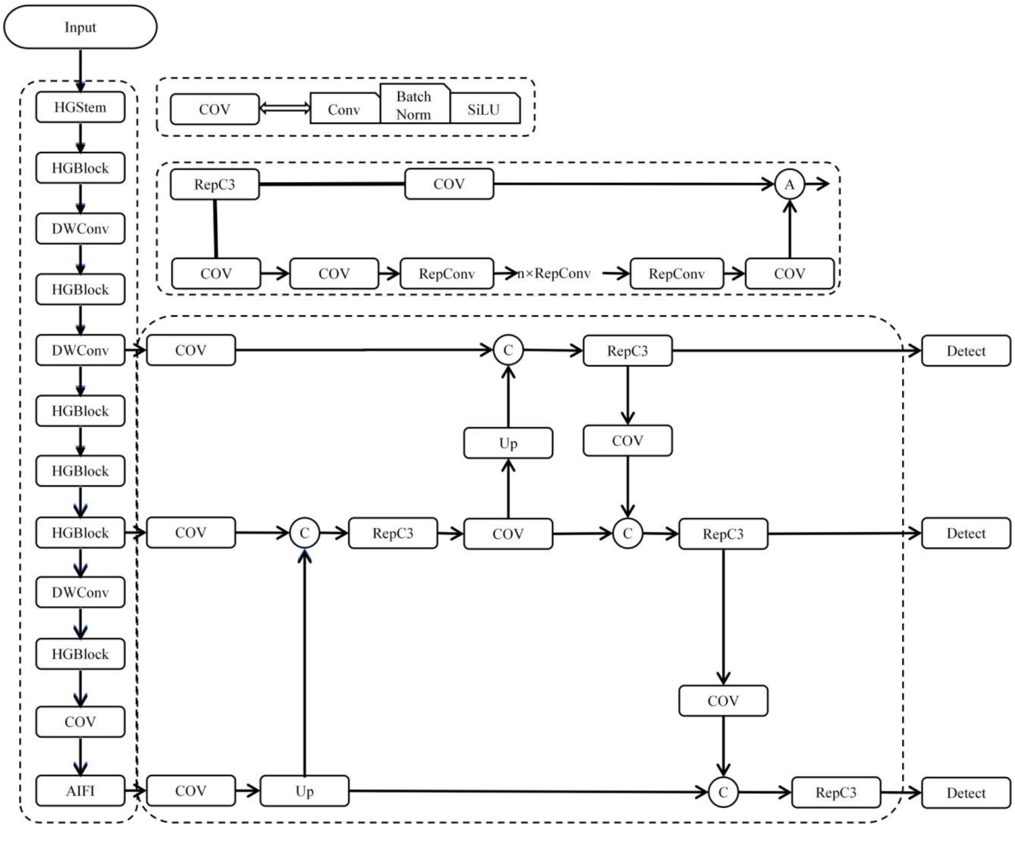


Fig. 3 DT-DERT model structure

In this study, the Neck architecture of YOLOv8 is modified by incorporating the proposed Cross-Scale Feature Fusion Module (CCFM) to enhance its detection capabilities. Based on Convolutional Neural Networks (CNNs), the CCFM structure has demonstrated significant efficacy in handling complex visual tasks, particularly in improving model robustness against scale variations and augmenting detection precision for small-scale objects. This architecture effectively integrates features across various scale levels with contextual information, thereby empowering the model to comprehend multi-level detailed information within images.

Through specific fusion pathways, the CCFM achieves deep interaction and integration of information derived from multi-level feature maps. During this process, N Repetitive Blocks (RepBlocks) are embedded within the fusion pathways. Each RepBlock comprises multiple convolutional layers, activation functions, and potential normalization layers, which are utilized for the deep extraction and transformation of input features. These RepBlocks not only enhance feature representational capability but also facilitate the effective fusion of cross-scale features.

Subsequently, feature maps from adjacent scales are initially fed into their corresponding fusion blocks. Within these fusion blocks, a series of convolution operations are performed; while preserving spatial information, the channel dimensions of the feature maps are adjusted to accommodate the requirements of subsequent fusion. Thereafter, the adjusted feature maps are merged via Element-wise Addition to generate new feature maps that encapsulate multi-scale information. This fusion strategy not only maintains computational efficiency but also promotes the linear combination of features. Consequently, the model is enabled to learn more complex and rich feature representations, thereby enhancing feature discriminability through complementarity. The architecture of the Yolo-CAD model is illustrated in Figure 4.

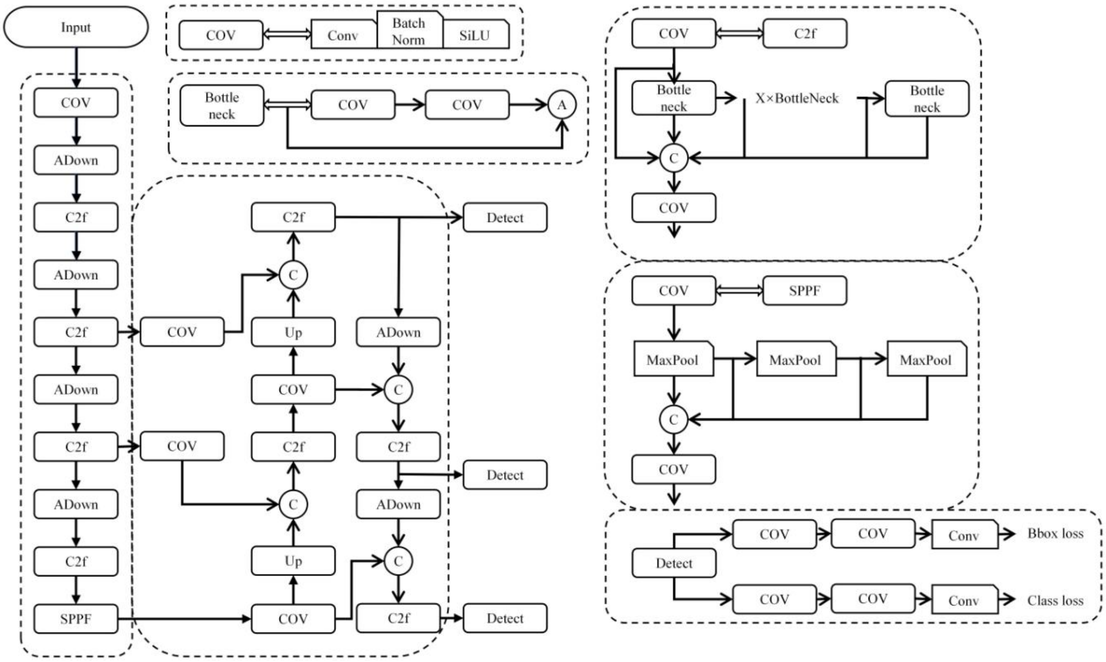


Fig. 4 Yolo-CAD model structure

3.3. Introduction of the DIoU Loss Function

The default YOLOv8 architecture utilizes the Complete Intersection over Union (CIoU) as the loss function for bounding box regression. This mechanism evaluates the positional discrepancy between the predicted box and the ground truth box by analyzing their aspect ratios. However, the CIoU metric fails to account for the directional alignment between the predicted and target boxes. Consequently, the Distance Intersection over Union (DIoU) has been introduced in this study as the regression loss function for detection boxes.

The DIoU metric incorporates not only the ratio of the intersection area to the union area (i.e., the IoU value) between the two bounding boxes but also the distance between the center points of the predicted and ground truth boxes (denoted as distance1) relative to the diagonal length of the smallest enclosing box covering both boxes (denoted as distance2).

$$DIoU = IoU - \frac{Distance\ \ from\ \ the\ \ centre\ \ of\ \ the\ \ bounding\ \ box^2}{Minimum\ \ closed\ \ frame\ \ area} \quad (6)$$

By simultaneously accounting for both the degree of overlap and the distance between center points of bounding boxes, the DIoU metric enables a more precise evaluation of bounding box similarity. This capability facilitates the rapid guidance of the model toward the optimal solution, thereby enhancing object detection accuracy and accelerating the model convergence process.

4. Experimental results and analysis

4.1. Comparison of detection performance of different models

To comprehensively evaluate the superiority of the Yolo-CAD model in strawberry disease and pest detection tasks, seven representative object detection models were selected as comparative baselines. These models encompass various developmental stages and technical paradigms (including one-stage and two-stage detectors, as well as different YOLO iterations), specifically comprising SSD [15], Retinanet [16], Yolov3, Yolov5n, Yolov8n, Yolov9t [10] and Yolov10n [17], alongside the proposed Yolo-CAD model.

All models were trained and evaluated using the identical strawberry disease and pest dataset—partitioned into training, validation, and testing sets with a ratio of 74%, 23%, and 3%, respectively, and an input resolution of 640×640 pixels. Furthermore, consistent experimental platforms and hyperparameter settings were maintained throughout the process.

Key performance indicators focused on the mean Average Precision (mAP50), which comprehensively reflects both localization and classification capabilities; Recall, which serves as a direct measure of the detection rate; and Model Size, which determines the feasibility of practical deployment. Detailed comparative results are presented in Table 4.

Table 4. Comparison of model results

mould	image size	Average precision/%	Recall rate/%	Model size/MB
SSD	640×640	64.75	61.08	95.7
Retinanet	640×640	70.11	68.21	139
Yolov3	640×640	69.11	67.61	198
Yolov5n	640×640	70.08	67.20	5.05
Yolov8n	640×640	71.25	70.46	5.98
Yolov9t	640×640	70.89	69.46	6.03
Yolov10n	640×640	70.12	65.74	5.52
Yolo-CAD	640×640	76.33	71.20	3.37

In comparison with classical one-stage models such as SSD and RetinaNet, the lightweight YOLOv8n maintains excellent detection performance—leading by 6.50 and 1.14 percentage points in mAP50, respectively—while achieving a significant reduction in model size by 89.72 MB (96.37%) and 130.41 MB (95.61%). This validates the significant advantages of modern lightweight YOLO architectures in terms of efficiency, aligning with the deployment requirements of mobile and edge computing platforms. When compared with the earlier YOLOv3 and contemporary lightweight models (YOLOv5n, YOLOv9t, and YOLOv10n), YOLOv8n also exhibits a lead of 2.14, 1.07, 0.36, and 1.13 percentage points in mAP50, as well as 4.99, 4.30, 0.00, and 3.72 percentage points in Recall, respectively. These results indicate that selecting YOLOv8n as the baseline for YOLO-CAD represents a choice characterized by a high performance starting point and an optimal balance of efficiency. While Shi Jie et al. [6] introduced GhostNet and EIOU to lightweight YOLOv5s to address mobile deployment challenges, the unmodified YOLOv8n baseline in this study demonstrates comparable performance with a smaller model size.

Building upon the lightweight and efficient characteristics of YOLOv8n, the proposed YOLO-CAD achieves significant performance improvements. Specifically, it attained an mAP50 of 76.33% and a Recall of 71.20%. Relative to YOLOv8n, precision increased by 5.08 percentage points, and Recall improved by 1.74 percentage points. Simultaneously, the model size was further reduced by 43.65% to merely 3.37 MB, representing a decrease of 2.62 MB. Consequently, YOLO-CAD realizes high precision while simultaneously enhancing model compression effects.

Compared with YOLOv3, YOLOv5n, YOLOv9t, and YOLOv10n, the mean Average Precision of YOLOv8n improved by 2.14, 1.17, 0.36, and 1.13 percentage points, respectively, while Recall increased by 2.85, 3.26, 1.00, and 4.72 percentage points. This demonstrates that within the YOLO series, the YOLOv8 model exhibits superior performance, making improvements based on YOLOv8n more suitable for strawberry disease and pest classification. The YOLO-CAD model achieved an average precision of 72.12 and a Recall of 69.23, with a model size of 3.38 MB. This performance is superior to the lychee disease model proposed by Wang Weixing [5] (20.1 MB). In summary, compared to YOLOv8n, the YOLO-CAD model size is reduced by 43.65%, with an increase of 5.08% in average precision and 0.74% in Recall, indicating optimized model performance. Figure 5 illustrates the training accuracy, while Figure 6 depicts the training loss.

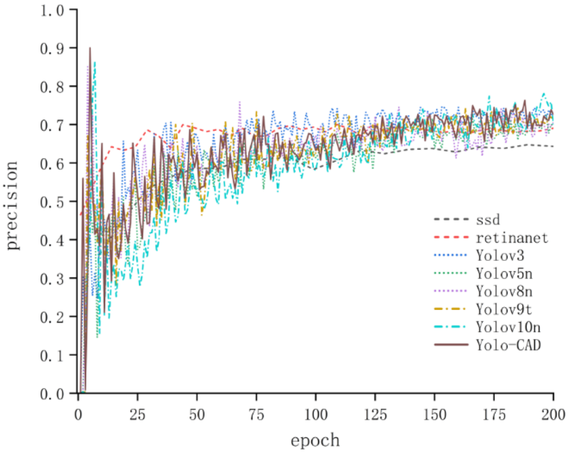


Fig. 5 Model training accuracy

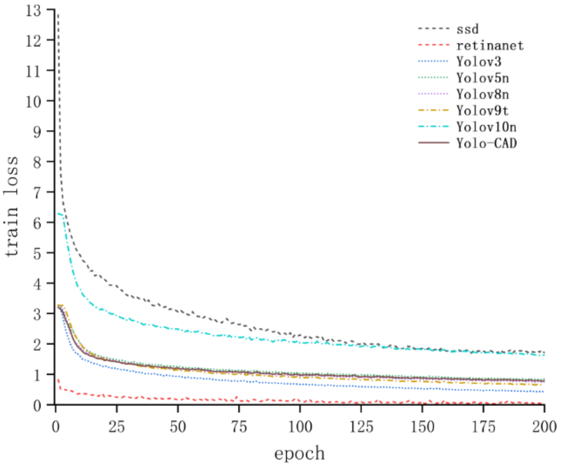


Fig. 6 Model training loss

4.2. Ablation Studies

To verify the effectiveness of the proposed Yolo-CAD model, ablation experiments were designed to analyze key metrics, including mean Average Precision (mAP50), Recall, and Model Size. These experiments involved the YOLOv8n baseline, YOLOv8n+CCFM, YOLOv8n+ADown, and the fully integrated YOLOv8n+CCFM+ADown configurations. As presented in the table, the results indicate that all three modification strategies contribute to a reduction in model size.

When solely introducing the CCFM to replace the original Neck (PAN-FPN) of YOLOv8n, the mAP50 increased by 0.14 percentage points to 71.39%, Precision rose by 0.74 percentage points to 72.27%, and Recall improved by 0.66 percentage points to 70.12%. Simultaneously, the model size experienced a slight decrease of 0.10 MB (approximately 1.67%). Through its embedded RepBlocks and cross-scale fusion mechanism, the CCFM enhances feature representation and model discriminability (as evidenced by the rise in Precision), thereby leading to a marginal improvement in detection precision and recall. The slight reduction in model size is attributed to more efficient feature processing, which eliminates certain redundant connections or operations. This effect is similar to the strategy of incorporating the CBAM module to reinforce feature extraction proposed by Yang Kun et al. [8]; however, the CCFM represents a more comprehensive optimization of the multi-scale fusion architecture.

By exclusively incorporating ADown to replace the traditional convolutional down-sampling modules (Conv) in the Backbone and Neck of YOLOv8n, the mAP50 increased by 1.76 percentage points to 73.01%, Precision rose by 3.71 percentage points to 75.24%, and Recall improved by 1.19 percentage points to 70.65%. Concurrently, the model size was compressed by 42.40% (a reduction of 2.54 MB) to 3.45 MB. These results validate the core advantage of ADown: its dual-path information preservation strategy—comprising an average pooling path to retain global context and a max pooling path to capture key feature points—significantly outperforms traditional convolutional down-sampling. It achieves model lightweighting in a highly structured manner while maximizing the retention of critical information, particularly for small-scale target features. This aligns with the objective of parameter reduction using the GhostModule proposed by Wang Weixing et al. [5]; however, the targeted design of ADown in the down-sampling layers achieves a superior balance between parameter count and precision。

The integration of both CCFM and ADown yielded a significant synergistic enhancement effect. The mAP50 increased to 76.33%, surpassing the simple summation of improvements from individual modules. Precision increased by 4.66 percentage points to 76.19%, and Recall improved by 1.74 percentage points to 71.20%, while the model size was compressed to 3.37 MB (a reduction of 2.62 MB, or 43.65%). These results demonstrate the following mechanisms:Feature Preservation: ADown provides a richer and higher-quality source of multi-scale feature information for subsequent fusion, thereby allowing the fusion potential of CCFM to be fully realized. Without the effective retention of critical information by ADown in the feature extraction layers—particularly details of small targets that are easily blurred by traditional down-sampling—the performance improvement of CCFM would be constrained. Effective Utilization: Relying solely on ADown to preserve features, without the cross-scale contextual fusion mechanism of CCFM, fails to effectively integrate this information into strongly discriminative feature representations for the detection head. CCFM ensures that the information meticulously preserved by ADown is utilized effectively.

Table 5. Performance comparison of different loss functions

mould	image size	Average precision/%	Accuracy/%	Recall rate/%	Model size/MB
Yolov8n	640×640	71.25	74.01	70.46	5.98
CCFM	640×640	70.59	74.15	67.34	4.00
ADown	640×640	73.01	77.60	73.71	5.20
CCFM+ADown	640×640	72.12	78.67	69.23	3.38

4.3. Ablation Studies

To evaluate the efficacy of the proposed improvements, a comparative analysis was conducted on the CCFM+ADown framework using various loss functions, including CIoU, GIoU, SIoU, EIoU, ShapeIoU, FocalerIoU, and DIoU.

In contrast to CIoU, which solely accounts for the overlap area and aspect ratio, DIoU directly guides the model to minimize the normalized distance between the center points of the predicted and ground truth boxes. This mechanism imposes a more intuitive and potent geometric constraint, thereby facilitating faster convergence and higher precision. This is particularly advantageous in the detection of strawberry diseases and pests, where targets are frequently subject to occlusion (e.g., leaves covering lesions or overlapping lesions) and characterized by blurred boundaries. In such scenarios, the center-point distance constraint of DIoU enables the model to localize the predicted box to the center of the ground truth box more rapidly, significantly enhancing robustness against localization ambiguity. Consequently, the training convergence speed was improved by 38%, a finding that is consistent with the acceleration conclusions reported in Reference [12].

While Shi Jie et al. [6] selected EIoU to enhance precision in their improved YOLOv5s model, the DIoU employed in the Yolo-CAD framework of this study yielded a superior improvement margin compared to EIoU (an increase of 3.79 percentage points). This discrepancy indicates that the performance of loss functions may vary depending on the specific task and model architecture. However, within the specific context of strawberry disease and pest detection—supported by the effective features extracted via ADown and CCFM—DIoU exhibited the most outstanding adaptability and optimization effects.

As presented in the table, compared with CIoU, GIoU, SIoU, EIoU, ShapeIoU, and FocalerIoU, the DIoU loss function achieved improvements in mean Average Precision (mAP) of 5.7, 3.66, 6.55, 3.79, 4.37, and 4.65 percentage points, respectively. These results demonstrate superior comprehensive performance, confirming that the DIoU loss function significantly enhances model performance in this specific detection task.

Table 6. Performance comparison of different loss functions

mould	image size	Average precision/%	Accuracy/%	Recall rate/%
CIoU	640×640	70.63	74.04	68.15
GIoU	640×640	72.67	78.56	69.39
SIoU	640×640	69.78	75.59	68.06
EIoU	640×640	72.54	76.99	68.28
ShapeIoU	640×640	71.96	73.19	71.29
FocalerIoU	640×640	71.68	75.42	70.34
DIoU	640×640	76.33	76.19	71.20

4.4. Comparative Analysis of Heatmap Performance

Heatmap visualization serves as a critical technique in object detection. The process initiates with feature extraction and prediction performed via Convolutional Neural Networks (CNNs). Subsequently, a set of anchor boxes is assigned to each unit of the partitioned input image to predict the probability of the category corresponding to the units location. Upon obtaining the prediction results, a heatmap is generated based on the confidence scores and positional information of the targets. The color intensity within the heatmap reflects the degree of confidence in object detection: warm tones (e.g., red and orange) indicate higher confidence levels, whereas cool tones (e.g., blue and green) signify lower confidence.

In this experiment, Gradient-weighted Class Activation Mapping (Grad-CAM) was utilized to generate detection heatmaps for the layers of interest, based on the trained model weight files (.pt). As indicated in Table 7, compared with other neural network models, the high-confidence regions (depicted in red) in the detection heatmaps generated by the Yolo-CAD model align more closely with the actual areas of disease and pest infestation.

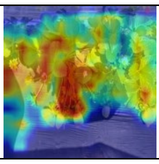
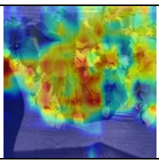
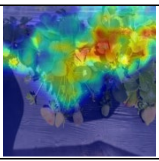
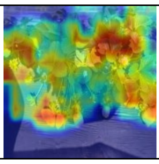
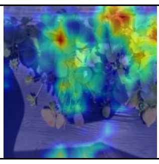
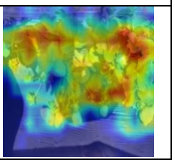
For the generation of heatmaps in the Yolo-CAD model, Layer 10 (corresponding to the RT-DETR equivalent convolutional layer), Layers 12 and 17 (the input layers for CCFM parameters), and Layers 22 and 25 (associated with the PAN structure) were selected. The heatmaps are arranged in ascending order based on layer indices.

As illustrated in Table 8, Layer 10 (situated near the RT-DETR equivalent convolutional layer) possesses a relatively large receptive field. Consequently, the heatmaps demonstrate a predominant focus on the main body of the strawberry fruit, larger lesion patches, and the holistic relationship between the target and its environment.

Layers 12 and 17 (the key parameter input layers for CCFM) are situated within the intermediate stage of the network and possess moderate receptive fields. The heatmaps reveal significant activation in regions corresponding to smaller, morphologically complex lesion points, such as early-stage anthracnose spots and the powdery zones of powdery mildew. This provides direct evidence that the CCFM module, by effectively incorporating and fusing spatial detail information inherent in intermediate features, significantly enhances the models cap ability to capture minute, early-stage, and morphologically irregular targets that are prevalent in strawberry diseases and pests.

Regarding Layers 22 and 25, following the final cross-scale fusion and decoding processes via the PAN pathway, the features exhibit enhanced robustness and generalization capabilities.

Table 7. Yolo series model heat map

Label the original image	Yolov3	Yolov5	Yolov8	Yolov9	Yolov10	Yolo-CAD
						

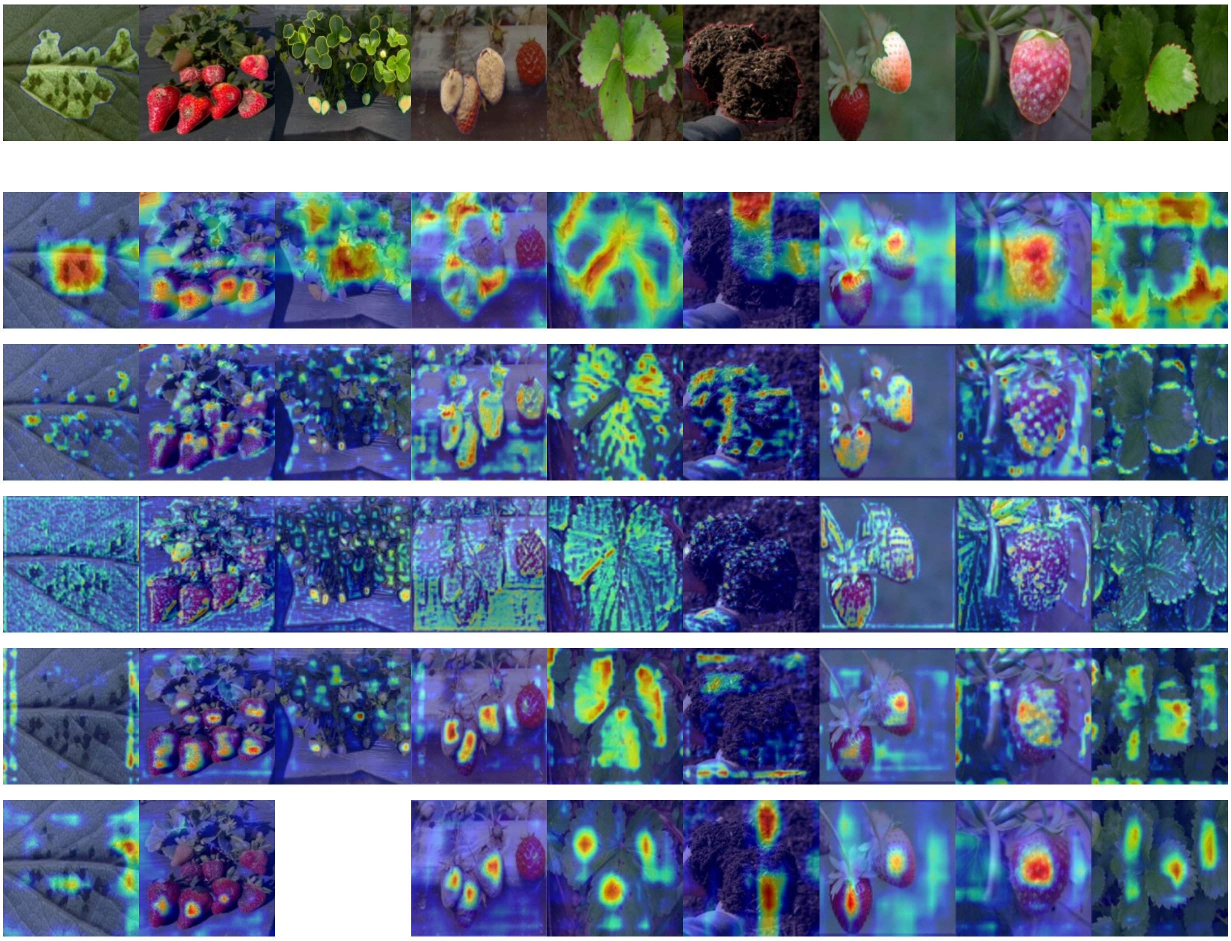


Fig. 7 Yolo-CAD model heat map

5. Conclusions

(1) Addressing the challenges of multi-label classification and inaccurate bounding box localization in strawberry disease and pest detection, a lightweight detection method named Yolo-CAD has been designed based on the YOLOv8n framework. Experimental results on the validation set demonstrate that the proposed model achieves a mean Average Precision (mAP50) of 76.33%, a Precision of 76.19%, and a Recall of 71.20%.

(2) To validate the efficacy of the improvements, four groups of ablation experiments were designed, comprising the

baseline YOLOv8n, YOLOv8n+CCFM, YOLOv8n+ADown, and the fully integrated YOLOv8n+CCFM+ADown. The model performance on the validation set was evaluated using four key metrics: mAP50, Precision, Recall, and Model Size. The findings indicate that the Yolo-CAD model exhibits superior performance compared to other configurations in the detection of strawberry diseases and pests.

(3) A comparative performance analysis was conducted between the improved Yolo-CAD model, various iterations of the YOLO series, and other one-stage algorithms. The results reveal that compared to SSD, RetinaNet, YOLOv3, YOLOv5n, YOLOv8n, YOLOv9t, and YOLOv10n, the mAP50 has been improved by 11.58, 6.22, 7.22, 6.25, 5.08, 5.44, and 6.21 percentage points, respectively. Furthermore, the model size has been reduced to 3.37 MB, representing a 43.65% reduction compared to YOLOv8n, thereby rendering it highly conducive to deployment on mobile devices.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 52374371) and the Open Fund of the Anhui Provincial Key Laboratory of Intelligent Manufacturing of Construction Machinery (Grant No. IMCM2023KF04).

References

[1] Song H, Shang Y, He D. Research progress of fruit object recognition technology based on deep learning[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(1): 1-19.

[2] Lin J, Wu X, Chai Y, et al. Structure optimization of convolutional neural networks: A survey[J]. Acta Automatica Sinica, 2020, 46(1): 24-37.

[3] Xiang X, Wang K, Ding Y, et al. Research on strawberry diseases and pests detection algorithm based on AM-YOLOX[J]. Research and Exploration in Laboratory, 2023, 42(10): 35-42, 60.

[4] Wang X, Zheng H, Wang F. Research and application of GCD-EF-CV model for Baihe strawberry disease and pest identification[J]. Journal of Shanxi Agricultural University (Natural Science Edition), 2023, 43(1): 65-74.

[5] Wang W, Liu Z, Gao P, et al. Detection model of litchi diseases and pests based on improved YOLO v4[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(5): 227-235.

[6] Shi J, Lin S, Zhang W, et al. Detection method of maize diseases and insect pests based on lightweight improved YOLOv5s[J]. Jiangsu Journal of Agricultural Sciences, 2024, 40(3): 427-437.

[7] Yan Y, Hao S, Gao Y, et al. Pest and disease detection method in kiwifruit orchard based on air-ground multi-source information[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(S2): 294-300.

[8] Yang K, Fan X, Bo W, et al. Plant disease and insect pest detection based on visual enhanced attention model[J]. Journal of Nanjing Forestry University (Natural Sciences Edition), 2023, 47(3): 11-18.

[9] Xing W, Li J, Zhong L, et al. Research on crop pest detection model based on improved RetinaNet [J]. Journal of Sichuan Agricultural University, 2023, 41(1): 153-157, 184.

[10] Wang C Y, Yeh I H, Mark Liao H Y. Yolov9: Learning what you want to learn using programmable gradient information[C]//European Conference on Computer Vision. Springer, Cham, 2025: 1-21.

[11] Xin C, Hartel A, Kasneci E. Dart: An automated end-to-end object detection pipeline with data diversification, open-vocabulary bounding box annotation, pseudo-label review, and model training[J]. Expert Systems with Applications, 2024, 258: 125124.

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

[13] Redmon J. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[14] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.

[15] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14,2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[16] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE International Conference on computer vision. 2017: 2980-2988.

[17] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection[J]. arXiv preprint arXiv:2405.14458, 2024.