Smart Agricultural Technology 12 (2025) 101513

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

CMD-YOLO: A lightweight model for cherry maturity detection targeting small object

Meng Li [a,b], Xue Ding [c,d,e,*], Jinliang Wang [c,d,e]

[a] School of Information Science and Technology, Yunnan Normal University, Kunming, 650500
[b] Southwest United Graduate School, Kunming, 650500
[c] Faculty of Geography, Yunnan Normal University, Kunming, 650500
[d] Key Laboratory of Resources and Environmental Remote Sensing for Universities in Yunnan Kunming, Kunming, 650500, China
[e] Center for Geospatial Information Engineering and Technology of Yunnan Province, Kunming, 650500

ARTICLE INFO

Keywords:
CMD-YOLO
YOLOv12n
Cherry ripeness
Small object detection
Lightweight

ABSTRACT

Cherry ripeness detection is a critical step in achieving intelligent harvesting. There are three core challenges associated with cherry ripeness detection: (1) complex environmental interference makes it difficult to extract surface features of cherries; (2) the detection of dense small targets results in high rates of missed detections and false positives; (3) the computational load of the model is too high to be deployed on edge agricultural devices. Based on these challenges, this study proposes a lightweight real-time detection model, CMD-YOLO (Cherry Maturity Detection-YOLO), based on an improved YOLOv12 architecture to optimize cherry ripeness detection performance. First, we propose the P23 structure-adaptive detection head with optimized number and scale, selectively removing redundant modules to enhance feature perception for dense small objects. Second, we innovatively propose the CDCHead module, which employs depthwise convolutions and SE attention mechanisms for cascaded channel fusion. This enables refined extraction of multi-scale spatial features while suppressing complex background interference. Finally, we introduce the Shape-IoU loss function. By precisely modeling variations in bounding box shape and scale, it enhances detection accuracy while maintaining low computational complexity. Experiments show that on the Multi-Scenario Cherry Ripeness Dataset V1, compared to the baseline model YOLOv12, the detection accuracy of this model is improved to 70.7 % (an increase of 5.3 %), the recall rate reaches 70.0 % (a rise of 10.3 %), the average precision mAP50 is 74.3 % (a rise of 11.8 %), and mAP50:95 of 54.9 % (an increase of 10.4 %), while reducing the number of model parameters to 0.7 M (a decrease of 73.1 %). The CMD-YOLO detection framework proposed in this study performs exceptionally well in detecting dense, small cherry targets in complex environments, with overall performance superior to that of existing mainstream models.

Introduction

Cherries are an essential economic fruit tree species worldwide [1] and one of the core categories in the high-end fresh fruit market [2]. They are a high-quality raw material for a range of products in the fruit processing industry, including cherry juice [3], fruit wine [4], and freeze-dried products [5]. During the process of ripening and harvesting, the maturity of cherries is primarily determined by the color and hardness of the fruit. When unripe, the fruit is light green or yellow-green. As it ripens, it gradually turns orange-red and eventually becomes a vibrant bright red. This color change exhibits characteristics similar to a continuous spectrum [6]. Fruit firmness is reflected in the fact that cherries with lower maturity levels are harder, and as the fruit matures, the flesh gradually softens [7].

In actual conditions, the synergistic effects of environmental factors (such as uneven light distribution [8], periodic rainwater erosion [9], diurnal temperature fluctuations [10], and ecological humidity changes [11]), as well as biological factors (such as bacterial infection [12] and insect feeding damage [13]), the maturity status of fruits on different branches of the same plant varies. Even within the same cluster of fruits, there are differences in maturity. This directly increases harvesting and sorting costs, especially in complex orchard environments where manual identification becomes time-consuming and costly, resulting in compromised accuracy. Precise detection of cherry ripeness has become

* Corresponding author.
E-mail addresses: 2424420015@ynnu.edu.cn (M. Li), 4228@ynnu.edu.cn (X. Ding).

https://doi.org/10.1016/j.atech.2025.101513
Received 17 August 2025; Received in revised form 3 October 2025; Accepted 5 October 2025
Available online 6 October 2025

a core technological bottleneck constraining industry development, manifesting in three aspects: (1) If precise prediction of ripening time is not possible [14], scientific forecasting of harvesting and market release timings becomes impossible, thereby impacting market share; (2) When maturity differences are significant, mixed processing reduces the flavor stability of the final product, necessitating additional sorting and processing steps, thereby increasing processing costs [15]; (3) Inaccurate maturity identification leads to inefficient use and waste of resources in harvesting, transportation [16], and storage. These three challenges hinder the high-quality development of the cherry industry. Addressing the industry pain point of cherry ripeness detection, the development of high-precision, intelligent detection technologies holds significant importance. By accurately identifying cherry ripeness at an early stage and scientifically scheduling harvesting, sorting, and processing stages, it is possible to promote the high-quality development of the cherry industry, ensure the stability of the fruit supply chain, and provide technical support for the implementation of precision agriculture.

Traditional cherry ripeness detection methods [17] primarily rely on manual visual inspection [18] and semi-automated methods [19] based on image processing detection technology [20]. Manual visual inspection methods rely on farmers' long-term experience to judge cherry ripeness. This method is highly subjective, with varying standards among different individuals, and is constrained by external factors such as weather, leading to low efficiency, high costs, and an inability to meet the demands of large-scale industrial production [21]. Traditional image processing methods, such as edge detection [22] and threshold segmentation [23], first capture cherry images, then apply image preprocessing techniques to remove noise and extract features, including color, texture, and shape, before segmenting the cherry region. In complex backgrounds [24] and low-contrast scenes [25], this method struggles with lighting [26] and shadow changes [27], leading to the loss of geometric features [28] and the failure of multi-scale feature fusion [29]. Additionally, there is a mismatch between the computational requirements of the model and the computational power of edge devices [30].

In recent years, with the rapid development of deep learning technology in computer vision [31], object detection methods based on convolutional neural networks (CNN) [32] and the YOLO (You Only Look Once) series [33] have been widely applied in the agricultural field, opening up new avenues for the automated detection of cherry ripeness. However, CNN models have high computational complexity [34], making their deployment challenging on edge devices with limited computing and storage resources [35]. On the other hand, when processing high-resolution cherry images, CNN models exhibit poor real-time performance, failing to meet the real-time monitoring requirements of actual cherry ripeness detection scenarios [36]. Compared to traditional convolutional neural network (CNN) detection methods, the YOLO series of models has gradually become the focus of research and application in the field of object detection due to their more efficient detection rates and superior recognition accuracy, attracting widespread attention from both academia and industry [37]. In 2016, Redmon et al. first proposed YOLOv1 [38], laying the foundation for the series of models. In 2017, YOLOv3 [39] maintained its detection speed advantage while significantly improving detection accuracy. In 2020, Bochkovskiy et al. introduced YOLOv5 [40], which further optimized detection efficiency by introducing the Focus module, CSP structure, and an improved loss function. In 2022, Li et al. proposed YOLOv6 [41], which abandoned the anchor box mechanism, adopted a simpler detection head design, and combined the SIoU loss function to improve detection accuracy. In 2023, the Ultralytics team open-sourced YOLOv8 [42], which supports multi-task processing and offers advantages such as a lightweight design and ease of deployment. In February 2024, YOLOv9 [43] addressed information loss issues through programmable gradient information and an efficient layer aggregation network. In May of the same year, YOLOv10 [44] introduced an NMS-free training method, which significantly reduced computational costs while maintaining performance. The subsequently released YOLOv11 [45] optimized the backbone and neck architecture, improving detection accuracy and speed. Currently, YOLOv12 [46], as the latest version, introduces an innovative architecture centered on attention mechanisms, achieving significant breakthroughs in detection accuracy and inference speed.

However, when the YOLO series detection model is applied to the specific detection scenario of cherry ripeness in orchards, the existing architecture faces three key challenges due to insufficient scene adaptability: First, from the perspective of cherry morphological characteristics, the differences in ripeness exhibit a continuous spectral-like gradual transition, with subtle changes in fruit skin texture from rough to smooth. Additionally, in open-air orchards, dynamic lighting conditions such as direct sunlight and cloudy shadows, as well as environmental disturbances like branches and leaves obstructing the fruit, occur frequently. The existing feature extraction mechanisms cannot capture the subtle features critical for distinguishing ripeness. Second, in terms of scale adaptability, cherries vary in size, and fruits are densely distributed and overlap. There is significant heterogeneity in size across different maturity stages. Traditional fixed feature fusion models cannot adaptively match the contextual information of multi-scale fruits, making it challenging to balance detection accuracy with scale generalization capabilities. Finally, from the perspective of actual deployment requirements in orchards, commonly used equipment such as portable detection terminals and drone-embedded modules has limited computing and storage resources, resulting in prolonged model loading and inference times. This not only fails to meet the real-time requirements of field-based mobile detection but also increases hardware costs, making it difficult to adapt to low-cost deployment scenarios in large-scale orchards. In response to the complex challenges faced in cherry ripeness detection, as analyzed above, this study proposes an innovative CMD-YOLO detection model, which achieves an effective balance between detection accuracy and computational efficiency through an optimized model architecture. The core innovations of this method are as follows: First, a novel network architecture, P23 is proposed. By employing adaptive detection heads and pruning strategies, it addresses the challenge of detecting minute objects in cherry images. The original model's P5 and P4 detection heads are removed, while retaining P3 and introducing a new P2 detection head. Redundant modules in the trunk and neck regions are pruned, enhancing detection accuracy while reducing model parameters to ensure computational efficiency. Second, the innovative lightweight detection head module CDCHead employs an inverted residual structure to reduce parameters and computational load. Integrating depthwise convolutions and SE attention mechanisms, it dynamically enhances key features through global information compression and channel weight learning. This module refines the regression branch of the detection head while optimizing the classification branch with depthwise convolutions. Channel-dimension fusion across both branches lowers computational costs, enabling efficient feature extraction. Finally, the Shape-IoU loss function is introduced to address the limitation of existing bounding box regression methods that neglect the intrinsic shape and scale properties of the box itself. This proposed regression method focuses on the shape and scale features of the bounding box, thereby optimizing the accuracy of bounding box regression.

Experimental results demonstrate that the CMD-YOLO detection model can accurately detect cherry ripeness in complex environments, quickly locate and assess the distribution of fruit ripeness, and provide precise monitoring results to assist in implementing targeted measures, thereby reducing losses caused by misjudgments of ripeness. The model's lightweight characteristics enable it to be embedded in harvesting and sorting equipment, allowing for real-time maturity grading at the production site. This promotes the mechanization and intelligent development of the cherry industry, providing key technological support for establishing a digital quality control system for the industry.

Test data

Image acquisition and data annotation

The dataset collected by the research institute originates from Malang Community, Qidian Subdistrict, Yangzonghai, Yiliang County, Kunming City, Yunnan Province. Collection dates were April 12, 2025, and April 23, 2025, with the cherry variety being Dahongpao [47]. Da Hong Pao belongs to the Rosaceae family as a deciduous shrub. Young branches grow relatively upright, spreading out after fruiting. Its kidney shaped fruit belongs to a variety that is fleshy, soft, and early maturing. When ripe, the skin turns bright red, and the flesh is thick and juicy. Its appealing sweet-tart flavor, vibrant color, abundant juice, thin peel that peels easily without cracking, and rich content of various vitamins along with high levels of potassium, calcium, and iron make it highly popular in the market.

During the cherry ripening period, images of cherry fruits were captured using a smartphone. Data collection selected mainstream mid-range smartphone models—the iPhone 15 Pro and Xiaomi 14. These models are widely adopted among orchard growers, eliminating the need for additional specialized imaging equipment and lowering the barrier to data acquisition. All shooting parameters were uniformly set to 3072 × 4096 pixels resolution, 72dpi horizontal and vertical resolution, 24-bit color depth, and sRGB color representation mode. To ensure a clear depiction of fruit details such as skin texture and color gradients, shutter speed automatically adapts to ambient light, flash is disabled, and white balance is set to auto mode. This prevents image blur caused by handshakes or slight fruit movement. The shooting location was an open orchard plot. During capture, the smartphone camera was positioned at the same height as the cherry clusters, maintaining a distance of 0.5–1.0 m from the fruit. This simulated the perspective of orchard growers during routine inspections. Images were taken from the northeast, southeast, northwest, and southwest directions of each tree, encompassing conditions such as front lighting, backlighting, single targets, multiple targets, targets partially obscured by foliage, and targets fully exposed. After shooting, blurred, duplicated, and invalid images were filtered out, yielding 964 raw cherry images with varying shooting angles, obstruction levels, and shake intensities. An annotation tool was then used to label the dataset. After annotation, the 964 images comprised a total of 10,824 detection boxes. These boxes were categorized into three states based on cherry ripeness: ripe, semi-ripe, and unripe. The labels were distributed as follows: 5461 ripe, 3389 semi-ripe, and 1974 unripe. The annotated dataset is illustrated in Fig. 1.

Data augmentation

Due to the relatively small sample size of the collected dataset, we utilized Roboflow's built-in preprocessing workflow to standardize the data and perform data augmentation, thereby enhancing the model's generalization ability, improving its robustness, and preventing overfitting. This approach generated three variants for each training sample.

Fig. 1. Dataset annotation diagram.

This process employs a random combination augmentation strategy. For each original image, the system randomly selects at least two augmentation operations from a predefined library of augmentation methods and applies them in combination. As shown in Fig. 2. The data augmentation methods used include:

1) Geometric transformations: Flipping as shown in (Fig. 2b), and (Fig. 2c). 90° Rotation as shown in (Fig. 2d), (Fig. 2e), and (Fig. 2f). Cropping as shown in (Fig. 2*g*). Rotation as shown in (Fig. 2h), and (Fig. 2i). Shear transformation as shown in (Fig. 2j), (Fig. 2k), (Fig. 2l), and (Fig. 2m).
2) Photometric adjustments: Blur as shown in (Fig. 2n). Saturation as shown in (Fig. 2o), and (Fig. 2p). Brightness as shown in (Fig. 2q), and (Fig. 2r). Exposure as shown in (Fig. 2s), and (Fig. 2t). Adding noise as shown in (Fig. 2u).

To further enhance the model's generalization ability, parameter randomization is applied to each sample variant during data processing. Specifically, the parameters of each data augmentation method are randomly sampled within a specified range, ensuring that the perturbation intensity of the same data augmentation operation varies across different variants, thereby simulating the complex variations in real-world scenarios. Through the combined perturbation of geometric transformations and photometric adjustments [48], we ensure that each original sample generates three mutually exclusive enhanced versions, thereby maximizing the coverage of the data distribution and improving the model's generalization performance in complex scenarios. This data augmentation strategy effectively balances data diversity and computational efficiency, avoiding the monotony of patterns caused by a single augmentation method while simulating the complex imaging conditions of real-world scenarios through combined perturbation. Ultimately, the training set in the experimental dataset used in this study was expanded from the original 964 images to 2892 images. The enhanced images were divided into training, validation, and test sets in a 7:2:1 ratio, named the Multi-Scenario Cherry Ripeness Dataset V1.

Network model improvement and training

CMD-YOLO model

In this cherry ripeness detection task, YOLOv12n was ultimately selected as the baseline model. The core reason lies in its adaptability to complex environments with small object detection tasks and its lightweight characteristics, which align highly with the requirements of the cherry detection scenario. Regarding detection accuracy, YOLOv12n incorporates an Area Attention module. By dividing feature maps into multiple regions along vertical or horizontal axes, this module enhances the model's spatial localization capabilities. It dynamically suppresses interference from leaf occlusions and lighting fluctuations while amplifying fruit feature responses. Compared to mainstream models like YOLOv8n and YOLOv10n, it more efficiently captures subtle color gradients and textural features of cherry fruits. For lightweight deployment, YOLOv12n reduces parameter complexity by simplifying the backbone architecture and optimizing neck feature fusion paths, making it well-suited for limited-computing devices like portable orchard terminals.

This study proposes the CMD-YOLO detection model based on YOLOv12, with its innovation stemming from the synergistic optimization of three key components: First, to address the small target scale distribution characteristics of cherries, the detection head architecture was restructured. The 20 × 20 (P5) and 40 × 40 (P4) detection heads from the baseline model were removed to eliminate redundant computations of macro features. The 80 × 80 (P3) detection head was retained, and a new 160 × 160 (P2) detection head was added to enhance sensitivity to small targets. To further improve detection efficiency, the model was modified using pruning strategies, removing some A2C2f modules from the trunk and neck regions, significantly reducing the

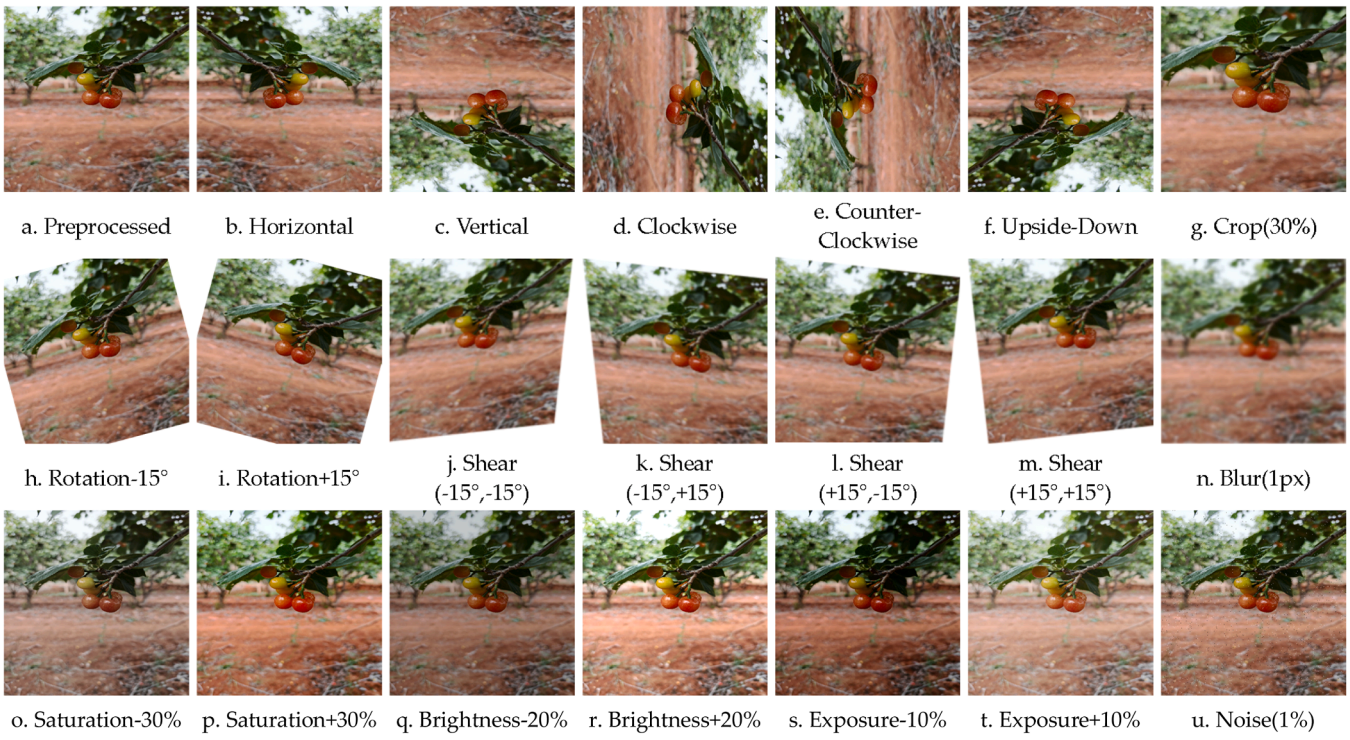


Fig. 2. Schematic diagram of data augmentation methods.

computational load of the trunk network and drastically compressing the number of parameters. Subsequently, a lightweight CDCBlock was proposed. The CDCBlock expands the receptive field through a cascaded channel fusion structure and integrates the depthwise convolutional SE (Squeeze-and-Excitation) attention module to achieve lightweight and efficient feature extraction by compressing global information, dynamically enhancing key features, and suppressing redundant features. By improving the regression branch in the detection head using CDCBlock while retaining the classification branch, the accuracy of cherry ripeness detection is enhanced, and the computational complexity of the model is effectively reduced. Finally, existing bounding box regression methods typically consider the geometric relationship between the ground truth (GT) box and the predicted box, calculating loss based on the relative position and shape of the bounding boxes. Still, they overlook the inherent attributes of the bounding boxes themselves, such as shape and scale, which influence bounding box regression. We introduce the

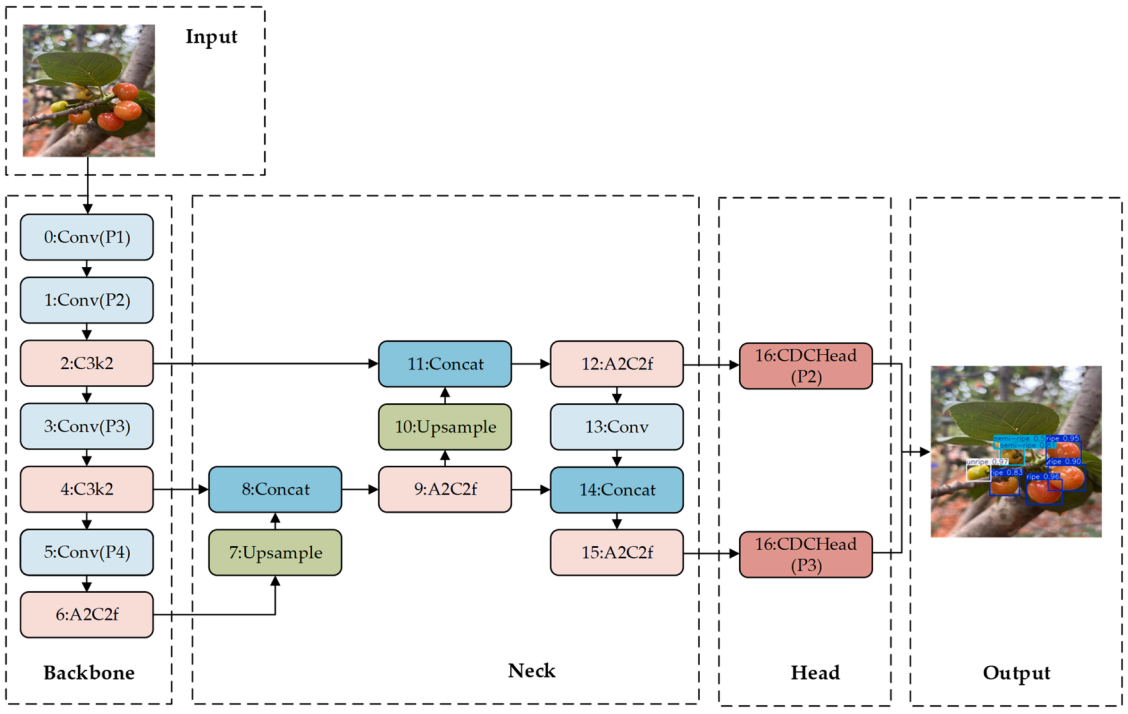


Fig. 3. CMD-YOLO network architecture diagram.

Shape-IoU loss function, a bounding box regression method that focuses on the shape and scale of the bounding box itself, significantly reducing localization errors for small objects and outperforming existing mainstream methods such as IoU and CIoU. The modified model structure is shown in Fig. 3.

Cherry image data undergoes feature extraction in the backbone network, followed by feature enhancement in the neck network, and localization regression in the head network, ultimately outputting detection results. This model demonstrates excellent efficiency and robustness in detecting cherry ripeness.

P23 detection head structural adjustment and pruning strategy

In current lightweight object detection, methods for model compression and architecture optimization primarily fall into two categories: channel pruning and detection head redesign. While both achieve lightweight models, they exhibit significant limitations in scenarios involving small objects in complex environments, like cherry ripeness detection. Existing channel pruning methods, such as regularized pruning and batch normalization (BN) layer scale factor pruning, fundamentally operate by statistically evaluating the importance of channels across model layers. They eliminate redundant channels with low contribution, thereby reducing both the number of parameters and computational complexity.

However, when applied to cherry detection, conventional channel pruning fails to account for the fact that small object features rely on shallow-layer channels, while color features indicating ripeness depend on specific channels. This may lead to the erroneous pruning of shallow-layer channels critical for extracting subtle cherry features. Furthermore, existing channel pruning methods predominantly focus on the backbone network, neglecting the channel compatibility between the detection head and the backbone. If shallow-layer channels in the backbone are excessively pruned, the detection head loses small object features, resulting in degraded detection performance. Existing detector head redesigns typically employ convolutional variants to optimize standard convolutions, retaining the YOLO series' classic P3, P4, and P5 scale detector heads for medium, large, and extra-large objects, respectively. Their scale adaptation remains singular, failing to optimize scale branches for cherries' predominantly small-object characteristics.

To address these limitations in cherry detection, we propose a novel P23 architecture featuring detection head scaling adjustments, quantity optimization, and targeted feature extraction network pruning. The detection head scaling prioritizes small object detection by removing P4 and P5 heads—which contribute minimally to small object detection in the original structure—while retaining the P3 head. And adding a new P2 detection head for shallow-layer features of small objects. The pruning strategy for the P23 architecture avoids global redundant removal across all layers, instead implementing targeted pruning based on cherry detection requirements. In the backbone network, shallow-layer channels primarily provide fine-grained features like cherry color and texture, processed by the C3k2 modules in layers 2 and 4. While deep-layer channels supply generalized background features like foliage and soil, processed by the A2C2f modules in layers 6 and 8. Consequently, pruning removes only redundant channels from deep-layer A2C2f modules while preserving all critical shallow-layer channels.

The P23 architecture achieves feature prioritization under limited computational resources through targeted pruning and a detection head optimized for small object detection, thereby enhancing the model's small object detection capabilities.

Cascaded depthwise channel-fusion detection head

The detection head of YOLOv12 is constrained by the limitations of standard convolution operations and fixed receptive fields when performing object detection, making it challenging to capture the detailed features of objects. This results in small objects being easily missed. When the visual differences between objects and their background are not significant, the model is prone to background interference, resulting in a decrease in detection accuracy. Additionally, YOLOv12 innovatively introduces a decoupled detection head architecture to enhance detection performance. However, compared to the shared head structure, its computational load doubles, imposing extremely high demands on device computing power and storage resources.

To reduce the computational load of the model while maintaining its detection accuracy, this study proposes a lightweight cascaded deep channel fusion detection head, consisting of depthwise convolutions [49], SE attention mechanisms [50], and CDCBlock, as illustrated in Fig. 4. Depthwise convolution, as a lightweight and efficient feature extraction unit, differs from traditional standard convolution in that it performs spatial convolution operations channel-by-channel in the channel dimension.

When a feature $X \in R^{C_{in} \times H_{in} \times W_{in}}$ with input channel count C, height H, and width W undergoes a depthwise convolution operation with a convolution kernel $K \in R^{C_{in} \times K \times K}$ (which has the same input channel count C_{in} as the input feature map, with a kernel size of $K \times K$), the output feature map $Y \in R^{C_{in} \times H_{out} \times W_{out}}$ is produced. The height H_{out} and width W_{out} of the output feature map are:

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times padding - k}{stride} + 1 \right\rfloor \tag{1}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times padding - k}{stride} + 1 \right\rfloor \tag{2}$$

Here, *padding* denotes the number of pixels filled around the input feature map, while *stride* represents the step size when the convolution kernel slides. This is abbreviated as s below. For each channel i of the input feature map, $Y_{i,m,n}$ denotes the value at position (m, n) in the corresponding channel i of the output feature map. This is obtained by performing a convolution operation using the corresponding kernel K_i. u and v represent the offsets of the kernel within the $K \times K$ range. The result for channel i in the output feature map is calculated as follows:

$$Y_{i,m,n} = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} X_{i,s \times m+u, s \times n+v} \times K_{i,u,v} \tag{3}$$

Compared to standard convolution, depthwise convolution performs convolution operations independently on each input channel, efficiently capturing local spatial features such as edges and textures within each channel without requiring repetitive calculations across channels, as is necessary with standard convolution. This characteristic enables it to maintain spatial feature extraction capabilities while significantly reducing computational complexity and parameter size, while preserving the specificity between feature channels. This provides rich and differentiated spatial feature materials for subsequent feature fusion and modulation, making it a key component in constructing lightweight and efficient feature extraction pathways.

For the CDCBlock module, as the core feature extraction unit of CDCHead, the collaborative design of multi-stage feature transformation and dynamic weight control achieves deep refinement of input features. When the input feature map $X \in R^{C_{in} \times H_{in} \times W_{in}}$ is input, the CDCBlock first performs channel compression on the input features to achieve preliminary cross−channel feature aggregation. Let the convolution kernel be $W \in R^{C_{mid} \times C_{in} \times 1 \times 1}$, where C_{mid} is the number of intermediate channels ($C_{mid} < C_{in}$). The purpose is to reduce the computational load of subsequent depth convolution and SE modules. The output feature is X_1:

$$X_1 = X * W_1 + b_1 \tag{4}$$

Where $*$ denotes a convolution operation, b_1 is the bias term, and after channel projection, spatial dimension fine−grained feature extraction is performed using a depthwise convolution with a kernel $K \in R^{C_{mid} \times K \times K}$, outputting feature $X_2 \in R^{C_{in} \times H_{in} \times W_{in}}$. The calculation process follows the formula:

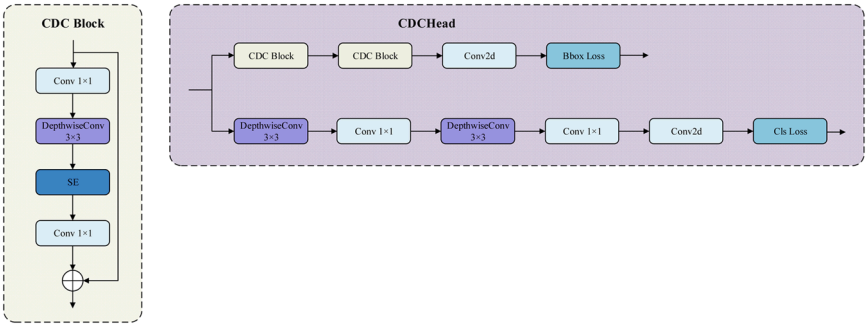


Fig. 4. CDCHead module structure diagram.

$$X_{2(i,m,n)} = \sum_{u=0}^{k-1}\sum_{v=0}^{k-1} X_{1(i,s\times m+u,s\times n+v)} \times K_{(i,u,v)} \tag{5}$$

In this process, depthwise convolutions capture the target contours and edge textures within each channel at extremely low computational cost, providing a rich spatial feature foundation for subsequent SE modules. The SE module performs key feature enhancement and redundancy suppression on the channel dimension of the depthwise convolutional output X_2, including Squeeze and Excitation. The Squeeze operation aggregates spatial information into channel statistics through global average pooling, outputting feature vectors $Z \in R^{C_{mid}}$:

$$Z_i = \frac{1}{H \times W}\sum_{m=1}^{H}\sum_{n=1}^{W} X_{(i,m,n)} \tag{6}$$

Here, Z_i denotes the global average value of the i-th channel after undergoing the Squeeze operation. The excitation operation generates channel attention weights $A \in R^{C_{mid}} (0 \le A_i \le 1)$ through two layers of fully connected networks with Sigmoid activation:

$$A = \sigma(W_2 \cdot \delta(W_1 \cdot Z)) \tag{7}$$

δ denotes the ReLU activation function, σ denotes the Sigmoid activation function, $W_1 \in R^{\frac{C_{mid}}{r}\times C_{mid}}$ and $W_2 \in R^{C_{mid}\times\frac{C_{mid}}{r}}$ are learnable parameters representing the weight matrices of two fully connected layers, where r is the dimension reduction ratio used to reduce model parameters and computational complexity. Feature modulation is then performed by multiplying the attention weights with X_2 channel-wise, yielding the output $X_3 \in R^{C_{mid}\times H_{in}\times W_{in}}$:

$$X_{3(i,m,n)} = X_{2(i,u,v)} \cdot A_i \tag{8}$$

By learning the dependencies between channels, the SE module assigns higher weights A_i to key feature channels and suppresses irrelevant channels such as background noise, allowing feature expressions to focus more on task-related information. Then, through a 1×1 convolution, the number of channels is restored from C_{mid} to C_{in}, and the output $X_4 \in R^{C_{in}\times H_{in}\times W_{in}}$ is obtained:

$$X_4 = X_3 \times W_3 + b_3 \tag{9}$$

Here, $W_3 \in R^{C_{in}\times C_{mid}}$ denotes the weight matrix of the 1×1 convolution layer, and $b_3 \in R^{C_{in}}$ represents the bias term of the 1×1 convolution layer. Finally, the original input X and the transformed feature $X_4$are added together in the channel dimension through residual connections, avoiding feature degradation in deep networks, preserving the original input information to ensure the integrity of feature transmission, and alleviating the vanishing gradient problem to improve the training stability of the module. The output of CDCBlock is obtained:

$$Y = X + X_4 \tag{10}$$

The output feature map $Y \in R^{C_{in}\times H_{in}\times W_{in}}$ can be represented as follows:

$$Y = X + F(X) \tag{11}$$

In the CDCHead module, CDCBlock is applied to the bounding box regression branch to enhance spatial feature extraction. In the bounding box regression branch, consecutive CDCBlock operations can continuously mine and refine features related to the target's spatial position. For cherry ripeness detection, depthwise convolution can capture key spatial information about the target edges, laying the foundation for more accurate prediction of the bounding box position. The CDCBlock employs a cascaded design that first extracts features through dimensionality reduction, then performs dimensionality expansion, and finally connects via residual connections. This design achieves a precise balance between computational efficiency and feature representation capability, providing the core support for enhancing detection performance in the CDCHead module.

Shape-IoU

In the original YOLOv12 model, IoU is measured solely by the ratio of the intersection and union areas of the predicted box and the ground truth box, without considering the shape of the bounding box or the impact of scale differences on the overlap relationship. As shown in Fig. 5. This illustrates the IoU calculation method in the original YOLOv12 model. The yellow region B_{gt} represents the annotated bounding box of the target's actual position, the blue region B_{pre} represents the bounding box predicted by the model, the gray overlapping region $B_{gt} \cap B_{pre}$ represents the overlapping area between the actual box and the target box, the background region C represents the boundary of

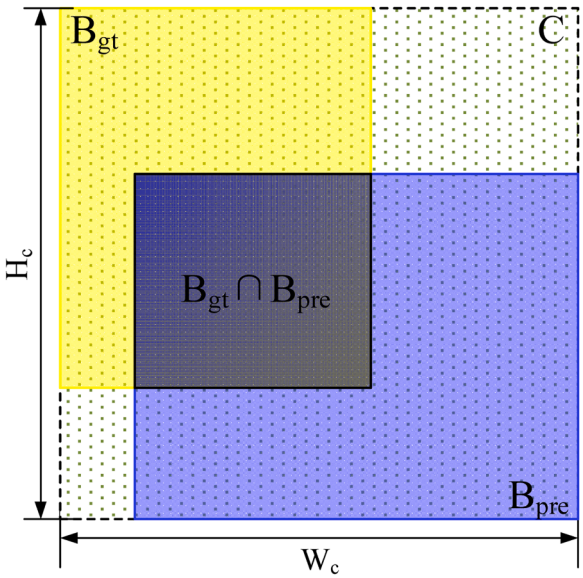


Fig. 5. IoU calculation diagram.

the area occupied by $B_{gt} \cup B_{pre}$, and W_c and H_c represent the width and height of the bounding box, respectively. In this case, the IoU loss is calculated as follows:

$$IoU = \frac{B_{gt} \cap B_{pre}}{B_{gt} \cup B_{pre}} \tag{12}$$

$$L_{IoU} = 1 - IoU + \frac{\rho^2(B_{gt}, B_{pre})}{(W_c)^2 + (H_c)^2} + \alpha v \tag{13}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{14}$$

$$v = \frac{4}{\pi^2}\left(\tan^{-1}\left(\frac{W_{gt}}{H_{gt}}\right) - \tan^{-1}\left(\frac{W_{pre}}{H_{pre}}\right)\right) \tag{15}$$

Where $\rho(B_{gt}, B_{pre})$ denotes the distance between the center points of the ground truth box and the predicted box, and W_{gt},H_{gt},W_{pre}, and H_{pre} denote the width and height of the ground truth box and the predicted box, respectively. α and v are two parameters in the IoU loss function that balance the influence of different terms.

The IoU loss in YOLOv12 defines the aspect ratio as a relative value, so it cannot reflect the proper relationship between the width and height of the predicted box and the ground truth box. As shown in Fig. 6. All GT boxes in the bounding box regression samples have the same shape deviation (all 0), the same position deviation (10 units to the right), and the GT box size of sample A is the same as that of sample B, while the GT box size of sample C is the same as that of sample D; the GT box shapes of A and C are the same, and the GT box shapes of B and D are the same; however, the bounding box scales of A and B are smaller than those of C and D, resulting in the following issues:

1) (1): Under the same conditions of shape deviation and position deviation, the shapes of the GT boxes are different, resulting in different IoU values.
2) (2): In the figure, A and B have the same position deviation values and the exact bounding box sizes, but the shapes of the GT boxes for A and B are different. The deviation direction for A is along the long side of the rectangular box, while the deviation direction for B is along the short side of the rectangular box. This results in B having an IoU value 0.11 lower than A, indicating that deviations along the long side have a minor impact on the IoU value. In contrast, deviations along the short side have a larger impact.
3) (3): In the figure, the deviation values of positions C and D are the same, and the bounding box sizes are the same, but the shapes of the GT boxes for C and D are also different. The deviation direction for C is along the long side of the rectangular box, while the deviation direction for D is along the short side of the rectangular box. The IoU value for D is 0.7 lower than that for C, which is smaller than the difference between A and B. This indicates that when the bounding box scale is larger, the shape has a weaker impact on the IoU value under the same deviation.

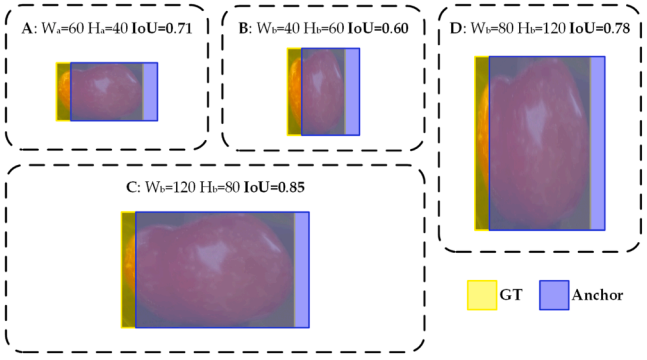


Fig. 6. Sample of bounding box regression analysis.

Shape-IoU extends the basic IoU metric by incorporating a measure of bounding box shape [51]. As shown in Fig. 7. Shape-IoU comprehensively accounts for overlap, shape distance deviation, and width-to-height ratio penalties. It achieves this by calculating weight coefficients ww and hh for the true box width and height:

$$ww = \frac{2 \times (w^{gt})^{scale}}{(w^{gt})^{scale} + (h^{gt})^{scale}} \tag{16}$$

$$hh = \frac{2 \times (h^{gt})^{scale}}{(w^{gt})^{scale} + (h^{gt})^{scale}} \tag{17}$$

To determine the weights associated with the actual bounding box dimensions, reflecting the influence of different shapes in different directions, where *scale* is the scaling factor related to the target's scale. Next, utilize the distance deviation between the center points of the scale-normalized bounding boxes $distance^{scale}$:

$$distance^{scale} = hh \times \left(\frac{(x_c - x_c^{gt})}{c}\right)^2 + ww \times \left(\frac{(y_c - y_c^{gt})}{c}\right)^2 \tag{18}$$

Calculate the distance deviation between the center of the predicted bounding box and the center of the ground truth bounding box after accounting for shape weighting, where c is the parameter used for scale normalization. Subsequently, quantify the differences in width and height between the predicted and ground truth bounding boxes using the Ω^{shape} metric:

$$\Omega^{shape} = \sum_{t=w,h}(1 - e^{-\omega_t})^4 \tag{19}$$

$$\begin{cases} \omega_w = hh \times \dfrac{|w - w^{gt}|}{\max(w, w^{gt})} \\ \omega_h = ww \times \dfrac{|h - h^{gt}|}{\max(h, wh^{gt})} \end{cases} \tag{20}$$

ω_w and ω_h are weighting coefficients for width-height differences, used to adjust the influence of these differences in shape evaluation. The core function of Ω^{shape} is to incorporate shape deviations of the bounding box into the loss calculation rather than purely positional deviations,

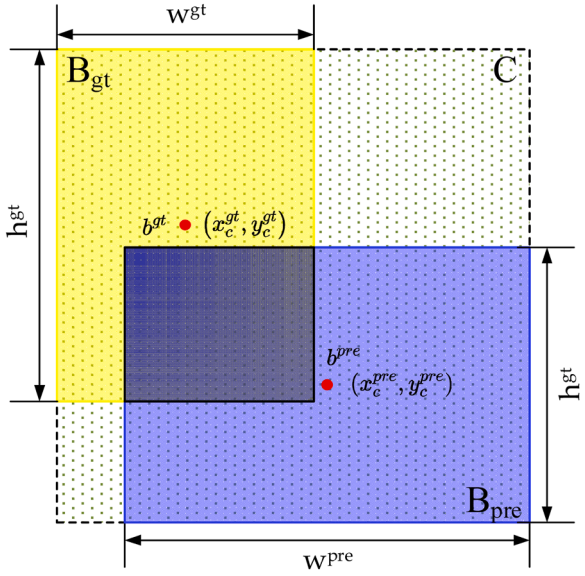


Fig. 7. Schematic diagram of Shape-IoU calculation method.

helping the model learn the intrinsic shape features of different objects more accurately. Ultimately, the bounding box regression loss for shape-IoU can be expressed by the following formula:

$$L_{shape-IoU} = 1 - IoU + distance^{scale} + 0.5 \times \Omega^{shape} \quad (21)$$

Test environment and evaluation indicators

Test environment and parameter settings

To validate the significant improvement in detection performance of the CMD-YOLO network, experiments were conducted on a sample dataset. Table 1 provides a detailed overview of the hardware and software configuration of the experimental environment, based on the Linux operating system. The hardware configuration includes an Intel (R) Xeon(R) Platinum 8474C CPU and an RTX 4090D (24GB) GPU. The software configuration utilizes the Anaconda+PyCharm compiler, with Python 3.8 as the programming language, and a deep learning framework based on PyTorch 2.0.0, corresponding to CUDA version 11.8.

The hyperparameter configuration used for model training is shown in Table 2. During training, mosaic data augmentation technology was employed to enhance the model's generalization capability, the specific parameters are: hue adjustment range of 0.015, saturation adjustment range of 0.7, luminance adjustment range of 0.4, translation range of 0.1, scaling range of 0.5. The optimizer selected was Adam, with an initial learning rate of 0.01, a momentum factor of 0.937, and a weight decay coefficient of 0.0005. The number of epochs is set to 300, with both workers and Batch size set to 32. The input image resolution is 640 × 640, Cache is set to False, and Amp is set to False.

Evaluation indicators

In this study, we systematically evaluated the CMD-YOLO detection model using multidimensional metrics. The metrics were divided into two categories: performance metrics and complexity metrics [52]. The performance metrics included:

Precision (P): The reliability of model prediction results is defined as the proportion of correctly predicted positive samples among all predicted positive samples, where *TP* denotes true positives, *FP* denotes false positives, and *FN* denotes false negatives:

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

Recall (R): Represents the model's ability to cover positive samples, calculated as the proportion of correctly identified positive samples to the actual positive samples.

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

F1-Score: The harmonic mean of accuracy and recall, which combines the balance between accuracy and recall. The higher the F1-Score, the better the model performs in terms of accuracy and recall. It is expressed as:

$$F1 - Score = \frac{2 \times P \times R}{(P + R)} \quad (24)$$

Mean Average Precision (mAP): Dynamically evaluate the overall performance of models based on the Intersection over Union (IoU) threshold. mAP50 denotes the mean average precision at an IoU threshold of 50 %, while mAP50:95 represents the mean average precision across IoU thresholds incrementally adjusted from 50 % to 95 %, reflecting the network's robustness in target localization and classification. Here, $p(r)$ denotes the precision value at recall rate r, N represents the number of categories in the dataset, AP_i indicates the average precision for category i (measuring both accuracy and completeness of individual category predictions), and IoU_{thresh} is the intersection-over-union threshold. Different threshold values j influence the calculation of average precision.

Table 1
Experimental software and hardware configuration information.

Component	Configuration
Operating System	Linux
CPU	Intel(R) Xeon(R) Platinum 8474C
GPU	RTX 4090D(24GB)
Compiler	Anaconda+PyCharm
Python Version	3.8
Deep Learning Framework	2.0.0
CUDA Version	11.8

Table 2
Training hyperparameter setting information.

Component	Configuration
Optimizer	Adam
Initial learning Rate	0.01
Momentum	0.937
Weight Decay	0.0005
Epochs	300
Workers	32
Batch	32
Input Image Size	640 × 640
Cache	False
Amp	False

$$AP = \int_0^1 p(r)dr \quad (25)$$

$$mAP50 = \frac{1}{N}\sum_{i=1}^{N} AP_i(IoU_{thresh} = 0.5) \quad (26)$$

$$mAP50:95 = \frac{1}{10}\sum_{j}\left(\frac{1}{N}\sum_{i=1}^{N} AP_i(IoU_{thresh} = j)\right) \quad (27)$$

Complexity metrics include:

Parameters: The total number of trainable parameters in a model, which is directly related to storage requirements and space complexity.

Computational efficiency (GFLOPs): The number of billion floating point operations required for a single forward inference, which quantifies the computational complexity of a model. A low GFLOPs value indicates that the network requires less hardware computing power when deployed.

Real-time Metrics: FPS serves as a hard metric for evaluating system real-time performance, directly quantifying the matching threshold between algorithm inference latency and scene dynamic bandwidth. It acts as a key constraint variable for balancing accuracy and efficiency at the deployment end. In the formula, N represents the number of images the model can process per unit time, while T denotes the total time required to process N images.

$$FPS = \frac{N}{T} \quad (28)$$

Test results and analysis

Ablation experiment

To thoroughly investigate the specific impact of each component in the CMD-YOLO network on its performance, this study conducted ablation tests. While keeping other conditions consistent, key elements in the model were removed one by one to observe changes in the model's performance and complexity metrics. The ablation test system evaluated the impact of each improved module and its combinations on model performance.

Detection head scale test

To accommodate the needs of multi-scale object detection, the YOLO series model architecture sets detection heads P3, P4, and P5 to correspond to small, medium, and large-scale objects, respectively. In cherry ripeness detection, statistical analysis of the dataset shows that the bounding box area of most ripe cherries is $\leq 32 \times 32$ pixels, exhibiting a significantly small-scale clustering feature. Based on the scale distribution patterns of cherry fruits, a strategic adjustment strategy is implemented to optimize the number of detection heads and their corresponding scales in the model.

The experimental data are shown in Table 3. When the P5 detection head, designed for large-scale targets, was removed, the model's accuracy, as measured by the P metric, decreased slightly from 65.4 % to 64.5 %, but the recall rate, as measured by the R metric, increased from 59.7 % to 65.1 %. mAP50 rose from 62.5 % to 65.7 %, and the model's parameter count (from 2.6 MB to 1.8 MB) and computational complexity (from 6.3 GB to 5.9 GB) also decreased slightly, thereby improving the model's lightweight nature. This indicates that the impact of this operation on the overall model detection performance is within a controllable range. This is because, in the Multi-Scenario Cherry Ripeness Dataset V1, large-scale target samples appear with low frequency, and their proportion in the total sample pool does not provide sufficient support for detection results. Therefore, the P5 detection head plays a non-dominant role in feature extraction and discrimination processes. The experimental results presented above fully validate the significant effectiveness of this optimization strategy in constructing lightweight models suitable for resource-constrained scenarios.

To further investigate the impact of detection head scale on model performance, we conducted experiments by introducing a 160×160 (P2) detection head after removing the P5 detection head. The results show that after introducing the P2 detection head, the model's precision rate P increased from 64.5 % to 70.0 %, recall R adjusted from 65.1 % to 69.1 %, mAP50 increased by 7 % (reaching 72.7 %), and mAP50:95 improved by 7 % (reaching 55.3 %), resulting in a significant enhancement in detection performance. However, due to the large feature map size and high resolution of the P2 detection head, its introduction also resulted in a significant increase in computational cost, with the model's computational workload (GFLOPs) surging from 5.9 GFLOPs to 14.5 GFLOPs. This indicates that while the P2 detection head enhances the ability to capture small object features, it also increases the demand for computational resources.

To optimize the balance between accuracy and efficiency, the P4 detection head was removed for testing. The results show that the model equipped with only P2 and P3 detection heads maintains an average precision mean mAP50 of 72.7 %, an increase of 0.4 percentage points to 55.7 % for mAP50:95, a rise of 0.4 percentage points to 69.5 % for recall rate R, and a precision P value that is nearly equivalent to the model equipped with all three detection heads (P2, P3, and P4), with a decrease of only 0.2 %. In terms of computational cost, the number of model parameters was reduced from 1.5 million to 0.8 million, which is approximately 30.8 % of the parameter scale of YOLOv12n. GFLOPs decreased from 14.5 G to 12.4 G, with an increase within an acceptable range, achieving a better balance between accuracy and efficiency. Through the above architectural optimization strategies, the model achieves a new balance between parameter scale, computational cost, and detection accuracy, enhancing the extraction of small-scale object features and providing a practical approach for designing object detection models tailored to resource-constrained scenarios.

Ablation test for the CDCHead module

To demonstrate the rationale behind the cascaded design of depthwise convolutions and SE attention in CDCHead, and to quantify the value of depthwise convolutions, the SE attention mechanism, and their cascaded structure, we sequentially replaced the detection head components under identical experimental conditions and compared their performance differences. The results are shown in Table 4. In the table, the DWHead model indicates replacing the regular convolutions in the regression branch of the original YOLOv12 detector head with only depthwise convolutions; the SEHead model incorporates the SE attention mechanism into the regression branch of the YOLOv12 detector head; while CDCHead represents the model employing the novel detector head proposed in this paper.

Based on experimental results, while DWHead significantly reduces computational complexity (GFLOPs decrease from 6.5G to 5.0G compared to YOLOv12n, and parameters decrease from 2.569 million to 2.195 million), accuracy shows a noticeable decline. The core accuracy metric mAP50 drops from 62.5 % to 60.3 %, and mAP50:95 from 44.5 % to 42.0 %. This indicates that models relying solely on depthwise convolutions lose inter-channel correlation information due to their per-channel independent computation, resulting in insufficient feature discrimination capabilities for objects and reduced accuracy. SEHead achieves an mAP50 of 61.4 %, 1.1 percentage points higher than DWHead, demonstrating enhanced key channel feature extraction. However, its lightweight advantage is not pronounced. This stems from the SE attention mechanism's reliance on local channel information for weight calculation. When directly applied to high-dimensional features from standard convolutions, it incurs additional computational overhead, preventing the full realization of its lightweight benefits. CDCHead achieves an mAP50 of 64.9 %, surpassing DWHead by 4.6 percentage points and SEHead by 3.5 percentage points, while exceeding the baseline model by 2.4 percentage points. It delivers the highest accuracy metrics among all models, with parameters only approximately 2.3 % higher than DWHead. Its GFLOPs are comparable to SEHead, significantly lower than the baseline YOLOv12n's 6.5 G.

This experiment demonstrates that CDCHead combines the advantages of depthwise convolutional local feature extraction and lightweight design with the benefits of SE attention's global channel semantic enhancement. It simultaneously addresses the accuracy degradation caused by the loss of depthwise convolutional channel correlations and resolves the computational redundancy resulting from SE attention directly applied to standard convolutions. Ultimately, it achieves an optimal balance across three dimensions: accuracy, lightweight design, and inference efficiency.

Comparative study of CDCHead and mainstream lightweight detection heads

To comprehensively evaluate the efficiency and performance advantages of the lightweight detector head CDCHead, we conducted comparative experiments between CDCHead and current mainstream lightweight detector heads ASFF [53], DyHead [54], Conv2Formers [55], and the two proposed detectors RFAHead and PPAHead in this study. While maintaining identical backbone architectures and training hyperparameters, only the detector modules at the network end were replaced. Performance differences among detectors were systematically compared using accuracy metrics and model lightweighting indicators.

Table 3
Test results of detection head scale.

YOLOv12n	P2	P3	P4	P5	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)
√	×	√	√	√	65.4	59.7	62.5	44.5	62.42	2.569	6.3
√	×	√	√	×	6 4.5	65.1	65.7	48.3	64.80	1.830	**5.9**
√	√	√	√	×	**7 0.0**	69.1	72.7	55.3	69.55	1.475	14.5
√	√	√	×	×	69.8	**69.5**	**72.7**	**55.7**	**69.65**	**0.834**	12.4

Table 4
Ablation test for the CDCHead module.

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)
YOLOv12n	65.4	59.7	62.5	44.5	62.42	2.569	6.5
DWHead	62.4	59.8	60.3	42.0	61.07	**2.195**	**5.0**
SEHead	63.0	60.5	61.4	43.1	61.72	2.242	5.2
CDCHead	**67.5**	**61.9**	**64.9**	**48.6**	**64.58**	2.249	5.2

Detailed experimental data are presented in Table 5.

As shown in Table 5, CDCHead achieves comprehensive performance leadership across P, R, mAP50, and F1-score metrics at 67.5 %, 61.9 %, 64.9 %, and 64.58 % respectively, while minimizing parameter count. Its performance on GFLOPs ranks second only to DyHead, with a difference of merely 0.2 and a relative increment of just 4 %. The results demonstrate that CDCHead achieves significant performance gains by capturing more detailed features of objects through its cascaded design combining depthwise convolutions and SE attention. This approach enhances the model's lightweight nature, and the improvement is clearly evident. The parameters of ASFFHead are about 4.26 M and GFLOPs are 9.9G , which is almost twice that of CDCHead. This may be because the core of ASFF is the weighted fusion of three scale features, which requires an additional 3×3 convolution to generate the weight map, which brings additional parameters and calculation. However, when ASFF is applied to models such as yolov12n whose network capacity is compressed to the limit, the model can only use limited computing power to fit these weights, rather than really distinguish between the target and the background. The final output fusion features are mixed with a large number of invalid disturbances, which leads to a reduction in accuracy. DyHead leads in computational efficiency and demonstrates improvements across all accuracy metrics, as it employs lightweight dynamic convolutions to provide dynamic weights for targets. However, its accuracy gains are less pronounced than those of CDCHead. In contrast, CDCHead achieves substantial accuracy improvements at the minimal computational cost of just 0.2 G additional operations. Overall, CDCHead strikes the optimal balance between accuracy and efficiency, making it the preferred solution for lightweight cherry ripeness detection models.

To further validate that the CDCHead module enhances small object feature representation, Grad-CAM heatmaps were used to compare feature activations before and after applying this module [56]. After processing all images, three representative images depicting complex small object detection scenarios were selected for demonstration, as shown in Fig. 8. (Fig. 8a). Illustrates the detection of small cherries obscured by branches and leaves. except for the two cherries closest to the lens, all others suffer varying degrees of branch and leaf occlusion. In this scenario, the YOLOv12 model fails to generate an attention region for the partially occluded target in the upper left corner. Conversely, the model incorporating CDCHead effectively eliminates leaf interference, precisely focusing on the target area while also showing improvement in missed detections. (Fig. 8b). Densely clustered cherries in the foreground mutually obscure each other, leading to missed detections in both models. Foreground cherries in the background exhibit blurring, compression, and minimal pixel coverage. The CDCHead-applied model produces smoother attention regions for foreground targets and achieves higher detection rates for background targets than the CDCHead-unapplied YOLOv12. (Fig. 8c). The background features soil nearly matching the target color along with intertwined branches, leaves, and tree trunks. The CDCHead-applied model detected more objects, with stronger responses in the red regions of the heatmap. These scenarios demonstrate that the CDCHead module enhances small object feature representation. Even in complex scenarios, the CDCHead-applied model remains adept at capturing the features of small cherry targets.

Loss function test

To validate the effectiveness of shape-IoU in cherry ripeness detection, the consistency of the experimental environment was strictly controlled, and multiple loss functions were introduced for the YOLOv12n model for comparison. These included CIoU, DIoU (which accelerates regression convergence by focusing on the distance to the center point), EIoU (which enhances shape matching by refining the width and height dimensions), GIoU (which introduces the minimum bounding box), and SIoU (which combines angle distance with multi-dimensional optimization for regression). The experimental results are shown in Table 6.

The experimental results show that CIoU improves the model's P to 68.4 % and R to 69.0 %, with mAP50 and mAP50:95 at 71.8 % and 54.4 %, respectively, effectively enhancing localization and detection accuracy; The P and R values of the DIoU model are 67.4 % and 69.3 %, respectively. While the mAP metric shows some improvement, it remains slightly weaker than CIoU. Indicating that DIoU focuses solely on optimizing the distance between center points in regression, whereas CIoU considers geometric constraints such as bounding box aspect ratios and shape adaptability more comprehensively; EIoU first increased the model P value to 70.1 %, but its R value was inferior to CIoU and DIoU, indicating that it has a unique role in shape matching optimization but is slightly weaker in terms of overall constraint balance, with mAP50:95 exceeding 53.6 %; The model incorporating GIoU achieves an R value of 70.4 %, the highest among all models, with mAP metrics comparable to CIoU. GIoU addresses the recall shortcomings of traditional IoU through a minimum bounding box penalty mechanism, offering advantages in recall optimization for scenarios prone to missed detections, such as cherry ripeness detection. Although SIoU combines multidimensional optimization of angle and distance, its performance is inferior to models incorporating other loss functions due to insufficient specificity for cherry morphology. In contrast, shape-IoU analyzes boundary box shape and scale differences, making it more sensitive to short-side deviations and better suited to characterizing cherry morphology. mAP50 is 71.8 %, the same as CIoU and GIOU, mAP50:95 reaches 54.5 %, and P and R also remain at relatively high levels of 67.4 % and 69.2 %,

Table 5
Experimental results of CDCHead and mainstream lightweight detection heads.

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)
YOLOv12n	65.4	59.7	62.5	44.5	62.42	2.569	6.5
ASFFHead	59.7	60.2	59.0	41.2	59.95	4.263	9.9
DyHead	66.1	60.2	62.4	45.6	63.01	2.366	**5.0**
Conv2Formers	63.9	60.9	62.3	43.5	62.36	2.381	6.0
RFAHead	63.9	59.5	61.0	44.1	61.62	2.639	6.8
PPAHead	65.5	60.4	63.3	45.9	62.85	3.188	9.5
CDCHead	**67.5**	**61.9**	**64.9**	**48.6**	**64.58**	**2.249**	5.2

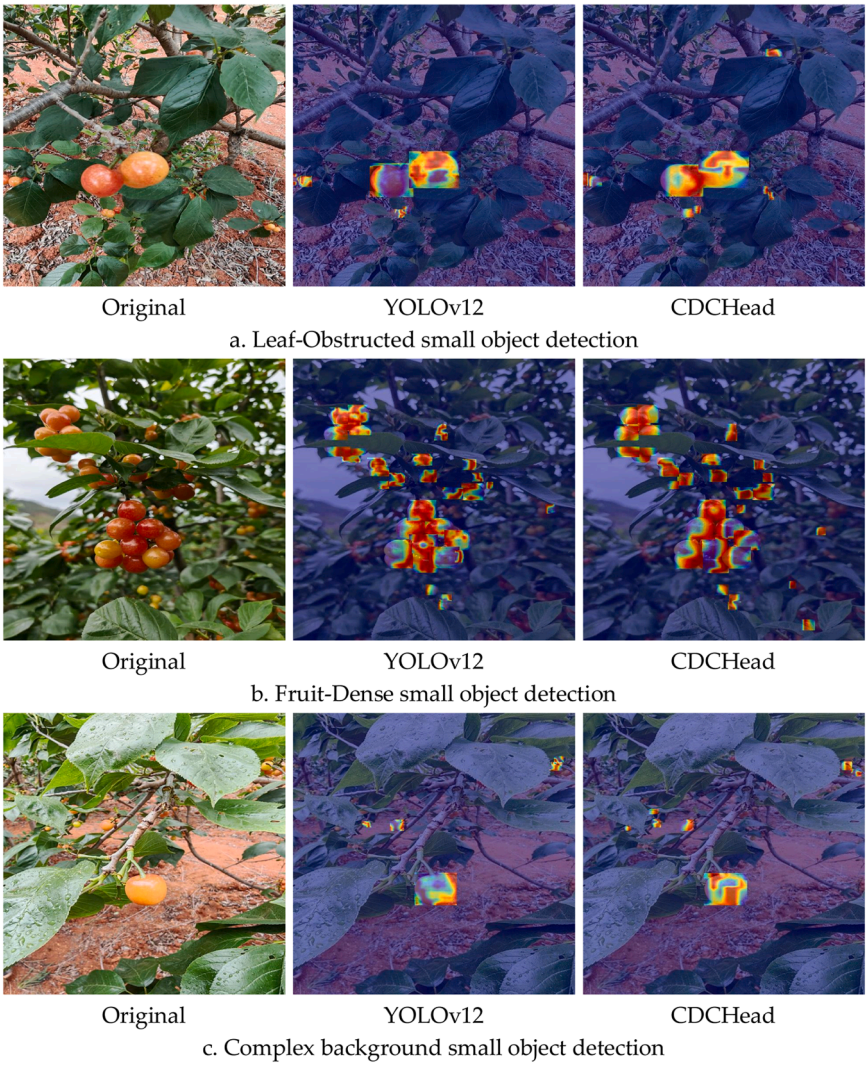


Fig. 8. Comparison of Grad-CAM heatmap activation before and after using CDCHead module features.

Table 6
Experimental results of different loss functions on the baseline model YOLOv12.

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)
YOLOv12n	65.4	59.7	62.5	44.5	62.42	2.569	6.5
+CIoU	68.4	69.0	71.8	54.4	**73.89**	2.569	6.5
+DIoU	67.4	69.3	70.7	53.5	68.34	2.569	6.5
+EIoU	**70.1**	67.2	71.4	53.6	68.62	2.569	6.5
+GIoU	66.5	**70.4**	71.8	54.5	68.39	2.569	6.5
+SIoU	64.2	63.7	65.1	47.8	63.95	2.569	6.5
+ShapeIoU	67.4	69.2	**71.8**	**54.5**	68.29	2.569	6.5

demonstrating shape-IoU's advantages in accurately capturing the impact of shape on regression and balancing accuracy and recall in cherry ripeness detection scenarios.

In summary, loss functions influence model performance through differentiated calculation methods, with CIoU, GIoU, and shape-IoU standing out in cherry ripeness detection. Shape-IoU, leveraging its adaptability to cherry morphology, validates its effectiveness in optimizing bounding box regression and enhancing detection performance while maintaining the mean average precision (mAP) metric, providing practical guidance for selecting and optimizing loss functions in cherry ripeness detection models.

Ablation testing of combination modules

To thoroughly investigate the specific impact of each component on performance, this study conducted ablation tests on the combination modules. While keeping other conditions consistent, key components were removed from the model one by one to observe changes in performance and complexity metrics. The ablation test system evaluated the impact of each improved module and its combination on model performance. The results, in terms of performance and complexity, are presented in Table 7.

The experimental results show that when the backbone network of the model is changed to the P23 structure alone, the accuracy on the test set improves from 65.4 % to 69.8 %, the recall rate improves from 59.7 % to 69.5 %, and mAP50 and mAP50 are significantly improved (by 10.2

Table 7
Ablation test results for combination modules.

YOLOv12n	P23	CDCHead	ShapeIoU	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)	FPS
√	×	×	×	6 5.4	59.7	62.5	44.5	62.42	2.569	6.5	84.4
√	√	×	×	69.8	69.5	72.7	55.7	69.65	0.834	12.4	**87.8**
√	×	√	×	6 7.5	61.9	64.9	48.6	64.58	2.249	5.2	81.0
√	×	×	√	67.4	69.2	71.8	54.5	68.30	2.569	6.5	82.5
√	√	√	×	6 9.4	70.2	73.6	**56.4**	69.80	0.674	7.9	86.4
√	√	×	√	69.3	**71.9**	**74.4**	54.8	**70.60**	0.834	12.4	86.5
√	×	√	√	67.4	70.0	72.4	52.6	68.67	2.249	**5.2**	78.5
√	√	√	√	**70.7**	70.0	74.3	54.9	70.35	**0.674**	7.7	86.5

Note: In the table, A denotes the model using the P23 architecture, B denotes the CDCHead module, and C denotes the use of the Shape-IoU loss function.

% and 11.2 %, respectively). The number of parameters was compressed from 2.6 to 0.8, but the computational complexity increased from 6.5 to 12.4, nearly doubling that of the original model. This phenomenon reveals that removing the P4 and P5 detection heads effectively compresses the model's parameters, and the addition of the P2 detection head for tiny targets enables the model to capture key feature information of small cherry targets more efficiently. However, since the feature extraction network corresponding to the P2 detector head must process larger-scale feature maps, this leads to a significant increase in computational overhead.

The model using the CDCHead module alone achieved 67.5 % accuracy, 61.9 % recall, 64.9 % mAP50, and 48.6 % mAP50:95. While its improvement in accuracy metrics is less pronounced than the P23 structure, in terms of complexity metrics, the model using the CDCHead module alone not only reduces the number of parameters from 2.6 to 2.2, and GFLOPs from 6.5 to 5.2, indicating that CDCHead significantly reduces redundant computations through lightweight computations in depthwise convolutions and information focusing in the SE module. By replacing the brute-force computations of traditional convolutions with lightweight feature selection logic, it achieves a precise balance between accuracy and efficiency while efficiently compressing complexity.

The model incorporating the shape-IoU loss function alone achieves an accuracy of 67.4 %, with recall significantly improved to 69.2 %, mAP50 rising to 71.8 %, and mAP50:95 is 54.5 %. Shape-IoU incorporates key directional deviations in the short side direction of cherry shapes based on differences in object bounding box shapes and scales, ensuring detection accuracy while reducing missed detections caused by shape misjudgments and improving recall.

Module combination experiments further reveal the synergistic mechanisms and performance boundaries between different components. When the P23 structure is combined with the CDCHead module, all model accuracy metrics improve, with better results than when used alone. The mAP50:95 reaches the highest value among all models, the number of parameters is compressed to 0.7 for the first time, and the computational complexity increases only slightly. This indicates that the synergistic effect of the P23 structure and the CDCHead module ensures the model's detection accuracy while also making it more lightweight.

When the P23 structure is combined with the shape-IoU loss function, the model achieves the best performance in terms of accuracy metrics. Still, its computational complexity remains at a high level, indicating that the integration of the P23 structure and the shape-IoU loss function fundamentally reconstructs the model's computational logic. By synergistically optimizing feature extraction and bounding box regression processes, it achieves a significant improvement in accuracy. Still, it fails to optimize lightweight metrics, reflecting that the current combination scheme has not yet achieved an effective balance between accuracy and computational cost control.

When the CDCHead module and shape-IoU loss function work together, the model's complexity metrics decrease, with the number of parameters compressed from 2.6 to 2.2, and GFLOPs from 6.5 to 5.2. This result once again validates the advantages of CDCBlock in lightweight design. The CDCHead module and shape-IoU loss function, through the synergistic effect of lightweight feature regression mechanisms and targeted loss constraints, not only strengthen the model's ability to distinguish features but also effectively reduce the computational load of the model.

In summary, the integrated model with three enhanced modules achieved a detection accuracy of 70.7 % on the test set, the highest among all models, with a recall rate of 70.0 % at a relatively high level. It also improved mAP50 to 74.3 %, a 0.1 % decrease from the highest value of 74.4 %. The F1 score is 70.35, while maintaining the lowest parameter count of 0.7 million and a minimal computational load of 7.7 GFLOPs. This successfully balances performance and complexity, providing a solution for efficient object detection tasks in edge computing scenarios that combines high accuracy with lightweight characteristics.

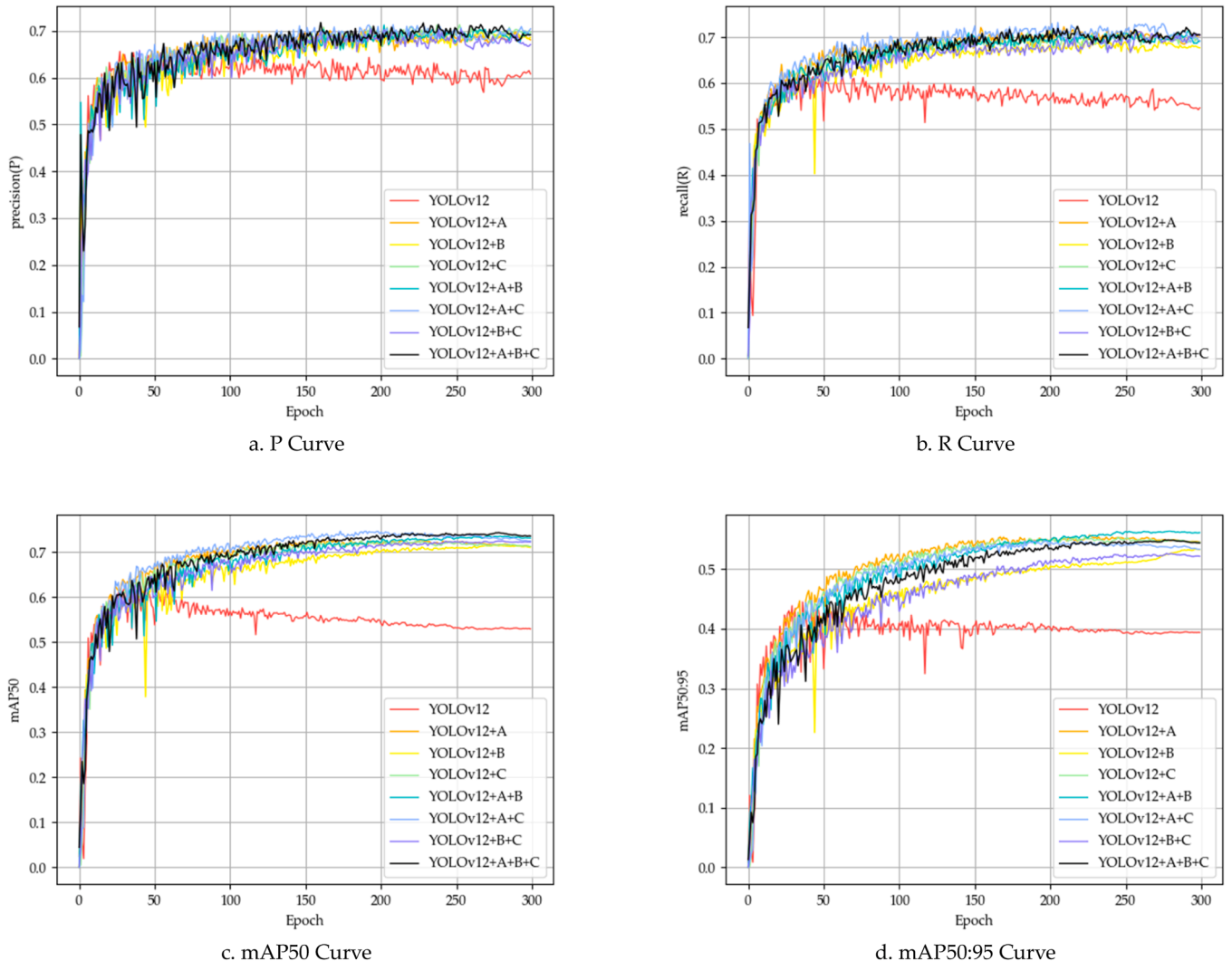
As shown in Fig. 9. (Fig. 9a). Presents the accuracy trends of different module combinations at various training stages, (Fig. 9b). Shows the recall rate evolution curves of varying module combinations over training stages, and (Fig. 9c). And (Fig. 9d). Respectively reflect the dynamic changes of mAP50 and mAP50:95 for different module combinations during training. Based on the four core metrics of accuracy, recall rate, mAP50, and mAP50:95, the comprehensive model CMD-YOLO outperforms other module combinations in all aspects. This result further validates the superior comprehensive performance of the extensive model in object detection tasks from a dynamic training perspective. The deep integration of its feature enhancement mechanism and bounding box regression strategy enables it to demonstrate stronger robustness and adaptability in cherry ripeness detection tasks in complex scenarios.

Note: In the figure, A denotes the model employs the P23 architecture, B represents the CDCHead module, and C indicates the use of the Shape-IoU loss function.

Comparative test

To further validate the performance of the CMD-YOLO network relative to existing object detection models, a systematic comparison and analysis were conducted with currently mainstream lightweight object detection models under identical conditions (including consistent parameter settings and the same dataset). The experiment selected eight representative single-stage YOLO series models: YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n, and YOLOv12n. It also included the classic two-stage object detection model: Faster R-CNN [57]. The experimental results are shown in Table 8.

Table 8. systematically compares the performance of multiple lightweight object detection models through metrics including accuracy, recall, model parameters, and computational efficiency. In high-accuracy scenarios, CMD-YOLO demonstrates superior performance, leading with 70.7 % precision, 74.3 % mAP50, and 54.9 % mAP50:95, showcasing its precise detection capabilities. Its F1-Score of 70.35 further indicates a favorable balance between high precision and high recall. Among all models, CMD-YOLO features the lowest number of parameters (only 0.674 M), relatively low GFLOPs at 7.7G , and a relatively high FPS of 86.5. Although YOLOv5n achieves 4.2 G GFLOPs,



a. P Curve

b. R Curve

c. mAP50 Curve

d. mAP50:95 Curve

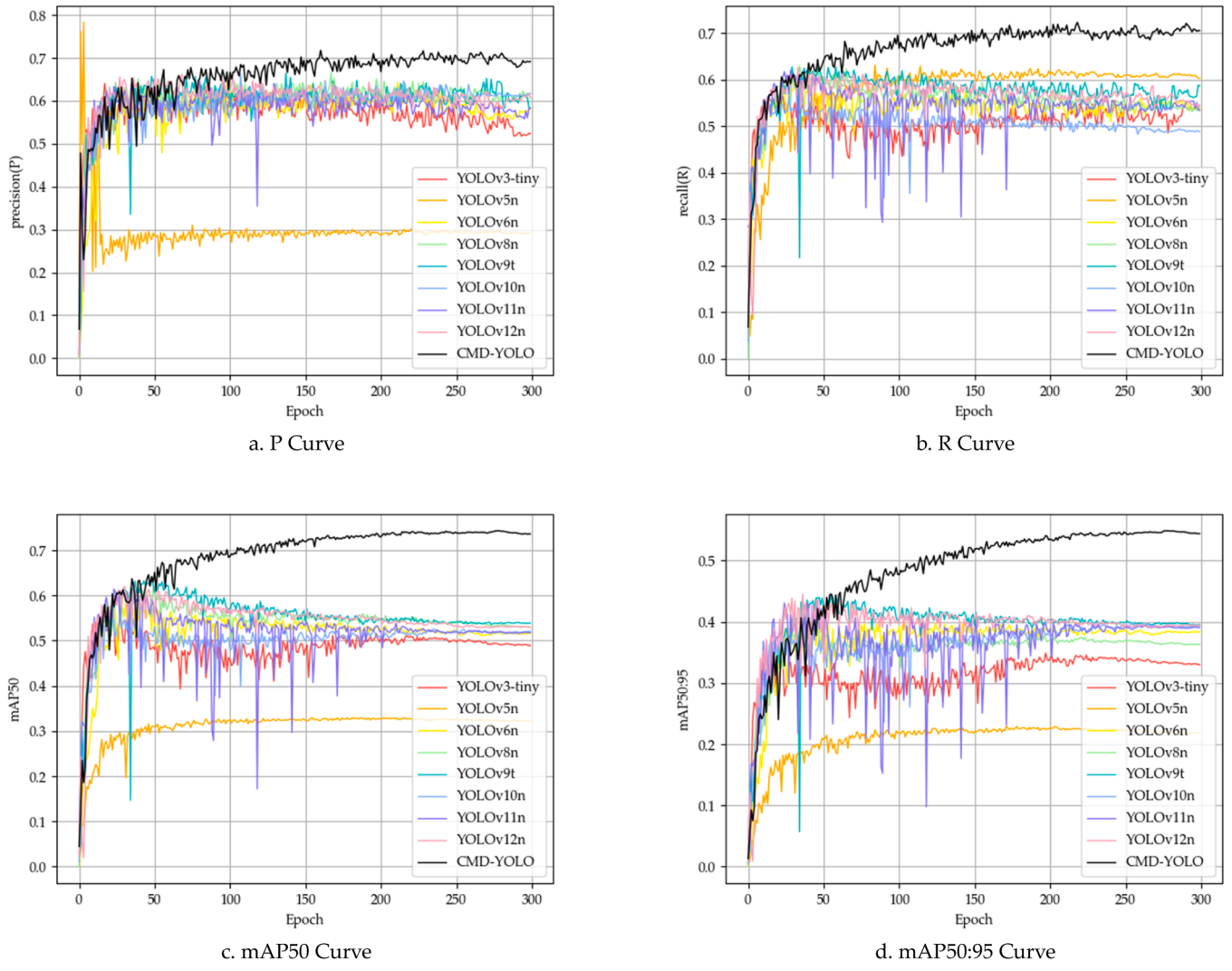
Fig. 9. Performance index change curves of Different Module Combinations in Different Training Periods.

Table 8
Comparison of accuracy and performance indicators of different models.

Model	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)	FPS
Faster R-CNN	65.1	72.1	62.9	33.3	68.42	137.099	370.2	0.37
YOLOv3-tiny	64.0	55.3	57.2	35.5	59.35	12.134	19.0	94.5
YOLOv5n	30.8	61.5	33.3	23.3	41.53	1.767	**4.2**	28.4
YOLOv6n	58.8	59.2	58.0	40.6	59.00	4.238	11.9	93.3
YOLOv8n	64.3	59.6	61.5	39.3	61.83	3.011	8.2	91.8
YOLOv9t	65.2	60.5	63.1	44.4	62.78	2.006	7.9	81.2
YOLOv10n	60.7	49.7	52.5	39.8	54.64	2.709	8.4	**94.9**
YOLOv11n	62.0	60.6	59.9	43.3	61.29	2.590	6.4	88.5
YOLOv12n	65.4	59.7	62.5	44.5	62.42	2.557	6.3	84.4
CMD-YOLO	**70.7**	**70.0**	**74.3**	**54.9**	**70.35**	**0.674**	7.7	86.5

its precision is only 30.8 %, the lowest among all models. To achieve a lighter model design, it sacrifices significant precision, and these precision losses are critical. As a representative two-stage object detection model, Faster R-CNN demonstrates strong recall performance. However, its parameter count is approximately 137 million, roughly 205 times that of CMD-YOLO. Its GFLOPs of 370.2 G is 48 times that of CMD-YOLO, while its FPS is only 0.37, which may limit its performance in real-time applications. Based on the analysis of the aforementioned experiments, CMD-YOLO demonstrates itself as a lightweight model featuring high accuracy and low computational requirements. This highlights its practicality in resource-constrained environments, making it suitable for target scenarios such as cherry ripeness detection in edge agriculture applications.

To visually illustrate the changes in performance metrics during model training and the final training results, the changes and comparisons of performance metrics for each model on the same dataset during training are shown in Fig. 10. (Fig. 10a). Displays the accuracy change curves for different models at different training stages. During the initial training phase, CMD-YOLO exhibits steady improvement, with little difference from other models, and continues to increase steadily to a higher level in the later stages. (Fig. 10b). Shows the recall rate change curves of different models at different training stages. CMD-YOLO's



a. P Curve

b. R Curve

c. mAP50 Curve

d. mAP50:95 Curve

Fig. 10. Performance index change curves of different models in different training periods.

improvement rate is similar to that of some models in the early stage. In the middle stage, most mainstream models stagnate or decline, while CMD-YOLO continues to grow steadily. In the later stage, it leads to a reduction, indicating that it can effectively avoid missed detections caused by crack characteristics. (Fig. 10c). And (Fig. 10d). Show the mAP50 and mAP50:95 curves of different models as training progresses. CMD-YOLO rapidly converges during the early learning phase, continues to optimize in the middle and later stages, and outperforms other models. It maintains good detection performance under different accuracy requirements, demonstrates stronger adaptability to complex backgrounds, and achieves high accuracy standards in both localization and category classification, resulting in overall superior detection performance.

This study employs the Gradient-based Activation Map (Grad-CAM) technique to systematically evaluate the visual interpretability of the improved model for cherry ripeness. By quantitatively analyzing the distribution of feature attention during the model's decision-making process, the study provides empirical evidence for the model's interpretability. As shown in Fig. 11. The activation intensity (red regions) in the heatmap exhibits a strict monotonic relationship with the model's confidence in crack features, providing empirical support for the model's interpretability research.

The Grad-CAM heatmap is shown in Fig. 11. Illustrates: (Fig. 11a). A scenario with leaf obstruction in a complex background, where numerous leaf elements in the background partially obscure the target, resulting in an incomplete effective feature region for the target. The model struggles to accurately distinguish the target from background noise, leading to a high likelihood of missed detections; (Fig. 11b). Shows a scene with densely distributed fruits: severe mutual occlusion exists among the densely distributed fruits, resulting in highly overlapping target features. The boundary features of individual fruits are weakened or even lost, making it difficult for the model to precisely locate the complete region of each fruit, leading to false positives or false negatives; (Fig. 11c). The figure shows a small target detection scenario: small targets occupy an extremely low proportion of pixels in the image, carry limited practical distinguishing features, and are easily affected by image noise and subtle changes in the surrounding environment, making it difficult for the model to extract stable target features from complex backgrounds, resulting in insufficient detection accuracy.

The CMD-YOLO model exhibits significant advantages in suppressing complex background interference, detecting dense scenes, and identifying small targets. Compared to traditional YOLO series models, the heatmap feature distribution of CMD-YOLO exhibits stronger focus. As observed in (Fig. 11a). Of the Grad-CAM heatmap, CMD-YOLO significantly reduces its response to background noise. In the lower right corner of the original image, there are two cherry targets highly occluded by leaves. Among all models, only CMD-YOLO simultaneously detects both targets. Most traditional YOLO series models focus on only one target and do not generate heatmaps for highly obscured mature targets. In contrast, YOLOv12n does not generate heatmaps for either of

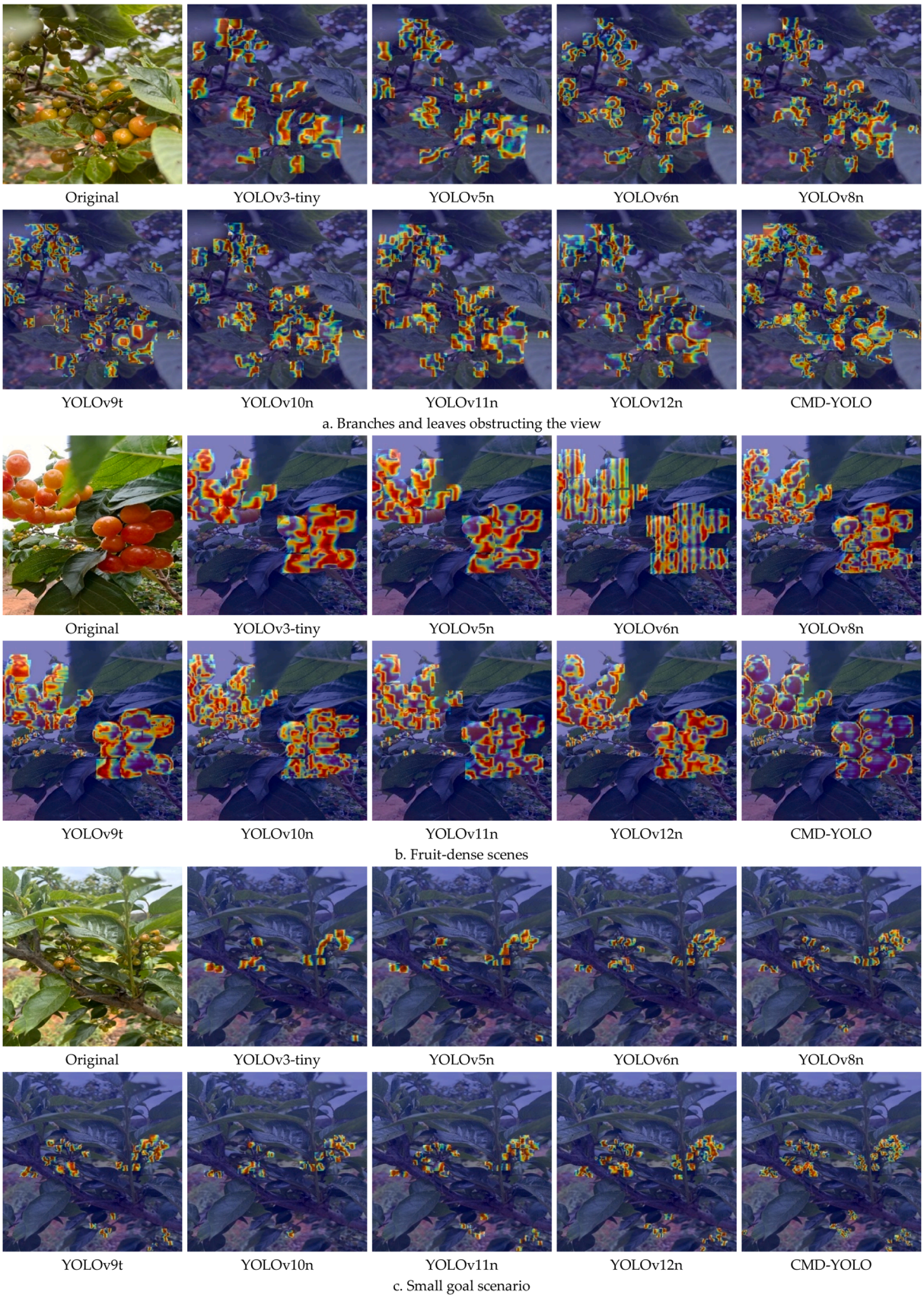


Fig. 11. Visualized Grad-CAM heatmaps of inference experiments for each model.

the two highly obscured targets.

In the fruit-dense scene task shown in (Fig. 11b). The red regions in the Grad-CAM heatmap of the CMD-YOLO model are focused on the edges of the cherry fruits. The intensity distribution is uniform and continuous, indicating that the CMD-YOLO model clearly and accurately defines the edges of the fruits, effectively addressing the issue of blurred boundaries caused by mutual occlusion between fruits in dense scenes. This provides strong support for the accurate detection and segmentation of the fruits.

In small object detection scenarios, due to the small size of cherry fruits (generally no $>2\ cm^3$), when the camera is at a significant distance from the fruits, the cherry targets occupy an extremely low pixel ratio in the image, resulting in limited discriminative features and making it easy for the fruit to be obscured by background information and preventing the model from reliably capturing practical target features. This makes the model prone to missed detections in small object detection scenarios. However, in the small object detection scenario shown in (Fig. 11c). The CMD-YOLO model can precisely cover the small cherry fruits in the red area. Compared to other models, it extracts small object features more accurately. For the unripe cherry fruits in the bottom-right corner of the image, the models YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv8n, and YOLOv10n all fail to detect them. However, the CMD-YOLO model can accurately identify target features even when the fruit pixels account for a low proportion of the image, demonstrating that the CMD-YOLO model has higher accuracy in feature recognition and target localization.

To systematically evaluate the detection performance of the CMD-YOLO model, we visually compared the detection results of each model, particularly in challenging scenarios such as branch and leaf occlusion, dense fruit clusters, small targets, and uneven lighting conditions (sunny vs. cloudy days). This allowed us to intuitively present the detection boundaries, confidence distributions, and false-negative/false-positive rates of each model. The results are shown in Fig. 12. As shown in (Fig. 12a). In the leaf-obscured scenario, the left cherries are partially obscured by branches, and leaves partially obscure the lower-right cherries. Among all models, YOLOv5n, YOLOv5n, YOLOv9t, YOLOv10n, and YOLOv11n misclassify half-ripe cherries as ripe cherries, while among the models that do not misclassify, CMD-YOLO has the highest detection confidence. In the fruit-dense scene shown in (Fig. 12b). Where cherry fruits overlap and obstruct each other, with some overlaps exceeding 50 %, the models YOLOv3-tiny and YOLOv10 exhibit missed detections. At the same time, CMD-YOLO has fewer missed detections and provides comprehensive coverage of dense fruits. In the small object detection scenario shown in (Fig. 12c). The detection rate of the CMD-YOLO model for small cherry fruits in non-central areas of the image is significantly higher than that of other models. In the sunny detection sample shown in (Fig. 12d). When comparing the detection results of various models, YOLOv3-tiny, YOLOv5n, YOLOv6n, and YOLOv11n can mark some cherries, but there are severe missed detections; YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv12n have fewer missed detections, but there are misjudgments. Overall, the CMD-YOLO model has fewer missed detections and provides reasonable confidence annotations for cherries of different maturities. In the overcast scene experiment shown in (Fig. 12e). Due to the combined effects of insufficient lighting conditions and leaf obstruction, the color information of the cherries themselves is distorted, and their features are weakened, significantly increasing the difficulty for the model to extract practical details. In this complex environment, the model is highly prone to missing or misclassifying targets due to its inability to capture target features accurately. Comparing the detection results of various models, only YOLOv3-tiny and CMD-YOLO did not exhibit missed detection issues; however, further examination of confidence scores reveals that YOLOv3-tiny has lower confidence values for detected targets, while CMD-YOLO demonstrates more stable and reasonable confidence performance. This highlights the model's stronger adaptability and reliability in complex scenarios such as overcast conditions and occlusions for cherry target detection.

To further validate the advanced capabilities of CMD-YOLO in the field of small object detection in agriculture, two representative studies published within the past two years were selected for comparison. The analysis focused on the model's ability to balance detection precision, lightweight design, and real-time performance. The results are shown in Table 9.

As shown in Table 9, the GFLOPs of CMD-YOLO is 7.7 G, which is slightly higher than that of TeaBudNet (6.1 G) [58]. However, the mAP50 of CMD-YOLO reaches 74.3 %, representing a 2.6 percentage point improvement compared to TeaBudNet—a model specifically designed for tea bud detection. This achieves a better trade-off between detection accuracy and model lightweighting, demonstrating that CMD-YOLO features more efficient parameter utilization. For the CTB-YOLO model [59], although its FPS is as high as 137.2, its GFLOPs of 16.96 means that its inference process requires substantial computational resource support. This proves costly for small object detection tasks deployed in outdoor scenarios, such as cherry ripeness detection. In terms of small object detection in agricultural application scenarios, CMD-YOLO exhibits excellent performance in balancing accuracy, lightweighting, and real-time performance, making it more suitable for practical agricultural production scenarios.

Discussion

This study conducted systematic innovations based on the YOLOv12n architecture and successfully constructed a lightweight model, CMD-YOLO, for cherry ripeness detection. Experimental results show that compared to the baseline model YOLOv12n, the CMD-YOLO model achieves significant improvements in multiple key metrics: detection accuracy increases from 65.4 % to 70.7 % (+5.3 %), recall rate increases from 59.7 % to 70.0 % (+10.3 %), mAP50 reaches 74.3 % (+11.8 %), mAP50:95 increased to 54.9 % (+10.4 %), and the number of parameters was compressed to 0.7 MB with a compression rate of 26.9 %. Although the computational complexity of the model in terms of GFLOPs increased from 6.3 to 7.7, it remains within a manageable range. CMD-YOLO significantly reduces its demand for computational resources during use, effectively validating the feasibility of lightweight design.

Through quantitative analysis of ablation experiments, it was found that optimizing the main network and detection head scale did indeed improve accuracy metrics, but also introduced additional computational overhead. To address this overhead, this study innovatively designed a new detection head module, CDCHead, which effectively aggregates cross-channel features through depthwise convolution and cascaded channel fusion, effectively suppressing background interference. Additionally, to address the limitations of traditional loss functions, the shape-IoU loss function was introduced, further improving detection accuracy while maintaining model lightweightness. This synergistic optimization design strategy provides a new technical paradigm reference for cherry ripeness target detection tasks in complex agricultural scenarios.

Although the CMD-YOLO model demonstrates high performance and low computational complexity in cherry ripeness detection tasks, it still has certain limitations. Under extremely complex conditions with abundant similar texture interference, the model's ability to accurately distinguish ripeness levels requires further enhancement. For extremely small and blurred cherry targets in the data images, constrained by the current feature extraction mechanism and the characteristics of RGB images, future research should explore fusion methods based on multimodal data. Furthermore, this study has not yet conducted comparative evaluations against mainstream lightweight state-of-the-art (SOTA) models such as PP-PicoDet and NanoDet. To continuously optimize model architecture, further reduce computational complexity and parameter counts, and enhance operational efficiency on edge devices—enabling more stable and efficient performance in resource-

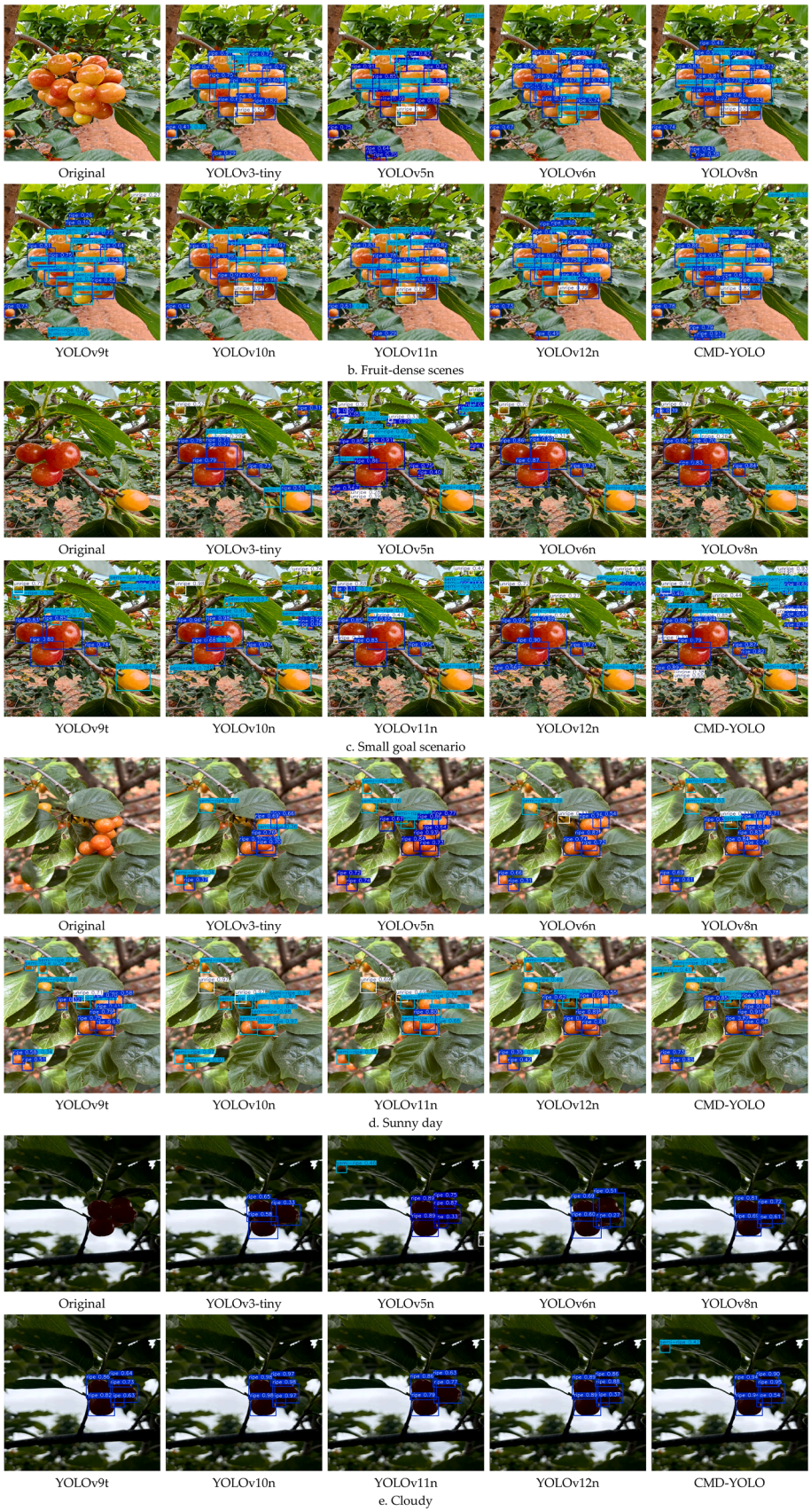


Fig. 12. Visualized detection results of inference experiments for each model.

Table 9
Key performance comparison of the latest research on CMD-YOLO and agricultural small target detection.

Model	Data	P(%)	R(%)	mAP50(%)	mAP50:95(%)	F1-Score	Parameters(M)	GFLOPs(G)	FPS
CTB-YOLO	tip-burn and powdery mildew in coriander	70.8	66.1	73.0	38.2	68.37	3.880	16.96	**137.2**
TeaBudNet	tea bud dataset D1	70.1	71.4	71.7	46.6	70.74	0.950	**6.1**	–
CMD-YOLO	Multi-Scenario Cherry Ripeness Dataset V1.	70.7	70.0	**74.3**	54.9	70.35	0.674	7.7	86.5

constrained environments—integrating evaluations of mainstream lightweight SOTA models into lightweight scenarios represents a key direction for future research.

Conclusion

This study addresses the core challenges in the field of cherry ripeness detection, including complex background interference, difficulties in detecting small objects, and computational resource constraints. It innovatively proposes the CMD-YOLO lightweight detection model. This model achieves breakthroughs in performance and efficiency through multi-dimensional collaborative optimization:

1) In terms of model architecture optimization, the model is deeply modified based on the YOLOv12 architecture. Through structured pruning strategies to remove redundant modules from the main network and adjust the scale of the detection head, the model achieves a detection accuracy of 69.8 % on the Multi-Scenario Cherry Ripeness Dataset V1, significantly enhancing its ability to extract features of small targets in complex field environments. Compared to the YOLOv12 baseline model, the improvement is significant, effectively overcoming the interference caused by the complex orchard environment on the maturity recognition of small cherry targets.
2) A new CDCHead module was proposed, whose cascaded channel fusion structure design ensures the integrity of small target features during hierarchical transmission. It integrates depthwise convolutions and SE attention mechanisms to enhance key channel feature expression while suppressing redundant channels caused by background noise, achieving extreme compression of the model's computational load. Experimental results show that the addition of this module reduces the number of model parameters and computational complexity while maintaining detection accuracy, significantly enhancing the model's advantages for deployment in agricultural edge scenarios.
3) The introduction of shape-IoU dynamically allocates weights based on the aspect ratio of real bounding boxes, strengthening constraints on deviations in the short-side direction. This makes the model's bounding box regression more targeted, ensuring low computational complexity while further improving detection accuracy.

These technological innovations collectively establish a new paradigm for lightweight detection of cherry ripeness in agricultural scenarios, providing important technical references for real-time visual detection in fields such as crop phenotyping and precision agriculture.

Ethics approval and consent to participate

The data of our study came from open access database and do not involvedata of human participants. All methods were performed in accordance withthe relevant quide lines and regulations.

Funding

Science and Technology Major Project of Yunnan Province (Science and Technology Special Project of Southwest United Graduate School - Major Projects of Basic Research and Applied Basic Research): Vegetation change monitoring and ecological restoration models in Jinsha River Basin mining area in Yunnan based on multi-modal remote sensing (Grant No.: 202302AO370003); Yunnan Provincial Basic Research Program General Project: Remote Sensing Estimation of Vegetation Aboveground Carbon Sink in Central Yunnan Urban Agglomeration and Its Response to Climate Change and Human Activities, Project No.: 202401AT070103; Yunnan Provincial Basic Research Program General Project: Research on Forest Aboveground Carbon Storage Estimation in Typical Mountainous Plateau Regions Based on ICESat-2/ATLAS Data, Project No.: 202501AT070008.

Data availability

All data are included in the article. To obtain data and code, please email the first author or corresponding author.

CRediT authorship contribution statement

Meng Li: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xue Ding:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition. **Jinliang Wang:** Writing – review & editing, Validation, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data Availability Statement: All data are included in the article. To obtain data and code, please email the first author or corresponding author.

References

[1] A.R. Soares Mateus, et al., By-products of dates, cherries, plums and artichokes: a source of valuable bioactive compounds, Trends. Food Sci. Technol. 131 (Jan. 2023) 220–243, https://doi.org/10.1016/j.tifs.2022.12.004.
[2] F. Carvalho, R.A. Lahlou, L.R. Silva, Phenolic compounds from cherries and berries for chronic disease management and cardiovascular risk reduction, Nutrients. 16 (11) (May 2024) 1597, https://doi.org/10.3390/nu16111597.
[3] N. Rungraung, N. Muangpracha, D. Trachootham, Twelve-week safety and potential lipid control efficacy of coffee cherry pulp juice concentrate in healthy volunteers, Nutrients. 15 (7) (Mar. 2023) 1602, https://doi.org/10.3390/nu15071602.
[4] S. Liu, B. Simonato, C. Rizzi, G. Zapparoli, F. Bianchi, S. Vincenzi, Sustainable sparkling cherry wine production from early and late varieties: insights into technological properties and volatile compounds, Food Bioprocess Technol 18 (8) (Aug. 2025) 7083–7094, https://doi.org/10.1007/s11947-025-03851-4.
[5] H. Lu, et al., Variable-frequency ultrasound synergistic hot air drying of Cherry: effect on drying characteristics and physicochemical quality, Ultrason. Sonochem. 119 (Aug. 2025) 107387, https://doi.org/10.1016/j.ultsonch.2025.107387.
[6] Z. Liu, et al., Comparative metabolomics profiling highlights unique color variation and bitter taste formation of Chinese cherry fruits, Food Chem. 439 (May 2024) 138072, https://doi.org/10.1016/j.foodchem.2023.138072.
[7] T. Tian, et al., Genome-wide characterization and comparative transcriptomics unravel CpMADS47 as a positive regulator during fruit ripening and softening in Chinese cherry, Postharvest Biol. Technol. 219 (Jan. 2025) 113287, https://doi.org/10.1016/j.postharvbio.2024.113287.

[8] Y. Yin, G. Liu, S. Li, Z. Zheng, Y. Si, Y. Wang, A method for predicting canopy light distribution in cherry trees based on fused point cloud data, Remote Sens. (Basel) 15 (10) (May 2023) 2516, https://doi.org/10.3390/rs15102516.
[9] S. Yang, Z. Huang, F. Yang, Y. Chi, Y. Chi, Mechanisms and management strategies for rain-cracking in greenhouse and open-air cherry, Horticultural Plant Journal (Jul. 2025), https://doi.org/10.1016/j.hpj.2025.04.008. S246801412500127X.
[10] W. Xing, et al., Development of predictive models for shelf-life of sweet cherry under different storage temperatures, LWT 217 (Feb. 2025) 117442, https://doi.org/10.1016/j.lwt.2025.117442.
[11] F. Coye, A. Calderón-Orellana, J.P. Zoffoli, C. Contreras, Influence of preharvest environmental conditions and postharvest relative humidity on the appearance of orange peel disorder in sweet cherry during fruit development and storage, Chil. J. agric. res. 84 (6) (Dec. 2024) 803–816, https://doi.org/10.4067/S0718-58392024000600803.
[12] R. Chen, et al., Limosilactobacillus fermentum MQ10 reduces decay of sweet cherry during storage by altering surface microbiome and enhancing phenylpropane metabolism, Food Control 177 (Nov. 2025) 111448, https://doi.org/10.1016/j.foodcont.2025.111448.
[13] M. Yazdani, D. Bao, J. Zhou, A. Wang, R.D. Van Klinken, Single-wavelength near-infrared imaging and machine learning for detecting Queensland fruit fly damage in cherries, Smart Agricult. Technol. 12 (Dec. 2025) 101090, https://doi.org/10.1016/j.atech.2025.101090.
[14] P.K. Srivastava, N. Sit, Physicochemical characterization of Spanish cherry (Mimusops elengi) fruit at different growth stages and its mass modelling using machine learning algorithms, Food Measure 18 (5) (May 2024) 3906–3922, https://doi.org/10.1007/s11694-024-02464-3.
[15] Q. Qiao, et al., Detecting the physiological and molecular mechanisms by which abscisic acid (ABA) regulates the consistency of sweet cherry fruit maturity, Sci. Rep. 15 (1) (Feb. 2025) 6311, https://doi.org/10.1038/s41598-025-85821-6.
[16] X. Fu, J. Sun, C. Lyu, X. Meng, H. Guo, D. Yang, Evaluation on simulative transportation and storage quality of sweet cherry by different varieties based on principal component analysis, Food Sci. Technol 42 (2022) e30420, https://doi.org/10.1590/fst.30420.
[17] Y. Sun, Z. Sun, W. Chen, The evolution of object detection methods, Eng. Appl. Artif. Intell. 133 (Jul. 2024) 108458, https://doi.org/10.1016/j.engappai.2024.108458.
[18] Y. Djenouri, A.N. Belbachir, T. Michalak, A. Belhadi, G. Srivastava, A knowledge-enhanced object detection for sustainable agriculture, IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing 18 (2025) 728–740, https://doi.org/10.1109/JSTARS.2024.3497576.
[19] P. Dubey, R.K. Mittan, A critical study on suspicious object detection with images and videos using machine learning techniques, SN. Comput. Sci. 5 (5) (Apr. 2024) 505, https://doi.org/10.1007/s42979-024-02869-3.
[20] X. Hu, Z. Du, F. Wang, Research on detection method of photovoltaic cell surface dirt based on image processing technology, Sci. Rep. 14 (1) (Jul. 2024) 16842, https://doi.org/10.1038/s41598-024-68052-z.
[21] N.U.A. Tahir, Z. Zhang, M. Asim, J. Chen, M. ELAffendi, Object detection in autonomous vehicles under adverse weather: a review of traditional and deep learning approaches, Algorithms. 17 (3) (Feb. 2024) 103, https://doi.org/10.3390/a17030103.
[22] W. Yang, X.-D. Chen, H. Wang, X. Mao, Edge detection using multi-scale closest neighbor operator and grid partition, Vis. Comput. 40 (3) (Mar. 2024) 1947–1964, https://doi.org/10.1007/s00371-023-02894-y.
[23] L. Qiao, K. Liu, Y. Xue, W. Tang, T. Salehnia, A multi-level thresholding image segmentation method using hybrid Arithmetic Optimization and Harris Hawks Optimizer algorithms, Expert. Syst. Appl. 241 (May 2024) 122316, https://doi.org/10.1016/j.eswa.2023.122316.
[24] R. Chen, X. Tian, Gesture detection and recognition based on object detection in complex background, Applied Sciences 13 (7) (Mar. 2023) 4480, https://doi.org/10.3390/app13074480.
[25] F. Xu, et al., An object planar grasping pose detection algorithm in low-light scenes, Multimed. Tools. Appl. 84 (9) (Apr. 2024) 5583–5604, https://doi.org/10.1007/s11042-024-19128-5.
[26] Z. Li, J. Xiang, J. Duan, A low illumination target detection method based on a dynamic gradient gain allocation strategy, Sci. Rep. 14 (1) (Nov. 2024) 29058, https://doi.org/10.1038/s41598-024-80265-w.
[27] Z. He, X. Chen, T. Yi, F. He, Z. Dong, Y. Zhang, Moving target shadow analysis and detection for ViSAR imagery, Remote Sens. (Basel) 13 (15) (Jul. 2021) 3012, https://doi.org/10.3390/rs13153012.
[28] S. Agrawal, P. Natu, OBB detector: occluded object detection based on geometric modeling of video frames, Vis. Comput. 41 (2) (Jan. 2025) 921–943, https://doi.org/10.1007/s00371-024-03374-7.
[29] Y. Wang, et al., Multi-scale hierarchical feature fusion for infrared small-target detection, Remote Sens. (Basel) 17 (3) (Jan. 2025) 428, https://doi.org/10.3390/rs17030428.
[30] K. Oksuz, B.C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: a review, IEEe Trans. Pattern. Anal. Mach. Intell. 43 (10) (Oct. 2021) 3388–3415, https://doi.org/10.1109/TPAMI.2020.2981890.
[31] W. Wei, Y. Cheng, J. He, X. Zhu, A review of small object detection based on deep learning, Neural Comput & Applic 36 (12) (Apr. 2024) 6283–6303, https://doi.org/10.1007/s00521-024-09422-6.
[32] W. Shi, X. Lyu, L. Han, An object detection model for power lines with occlusions combining CNN and transformer, IEEe Trans. Instrum. Meas. 74 (2025) 1–12, https://doi.org/10.1109/TIM.2025.3529073.
[33] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of Yolo algorithm developments, Procedia Comput. Sci. 199 (2022) 1066–1073, https://doi.org/10.1016/j.procs.2022.01.135.
[34] M. Yang, et al., LCSED: a low complexity CNN based SED model for IoT devices, Neurocomputing. 485 (May 2022) 155–165, https://doi.org/10.1016/j.neucom.2021.02.104.
[35] X. Guo, Q. Jiang, A.D. Pimentel, T. Stefanov, Model and system robustness in distributed CNN inference at the edge, Integration. 100 (Jan. 2025) 102299, https://doi.org/10.1016/j.vlsi.2024.102299.
[36] P. Ruiz-Barroso, F.M. Castro, N. Guil, Real-time unsupervised video object detection on the edge, Fut. Gener. Comput. Syst. 167 (Jun. 2025) 107737, https://doi.org/10.1016/j.future.2025.107737.
[37] S. Hou, Y. Pang, J. Wang, J. Hou, B. Wang, RS-YOLO: a highly accurate real-time detection model for small-target pest, Smart Agricultural Technology 12 (Dec. 2025) 101212, https://doi.org/10.1016/j.atech.2025.101212.
[38] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.
[39] J. Redmon, A. Farhadi, YOLOv3: an incremental improvement, arXiv preprint (Apr. 08, 2018), https://doi.org/10.48550/arXiv.1804.02767 arXiv:1804.02767.
[40] O.M. Lawal, YOLOv5-LiNet: a lightweight network for fruits instance segmentation, PLoS. One 18 (3) (Mar. 2023) e0282297, https://doi.org/10.1371/journal.pone.0282297.
[41] C. Li, et al., YOLOv6: a single-stage object detection framework for industrial applications, arXiv preprint (Sep. 07, 2022), https://doi.org/10.48550/arXiv.2209.02976 arXiv:2209.02976.
[42] N. Ma, Y. Su, L. Yang, Z. Li, H. Yan, Wheat seed detection and counting method based on improved YOLOv8 model, Sensors 24 (5) (Mar. 2024) 1654, https://doi.org/10.3390/s24051654.
[43] C. Liu, et al., Detection of surface defects in soybean seeds based on improved Yolov9, Sci. Rep. 15 (1) (Apr. 2025) 12631, https://doi.org/10.1038/s41598-025-92429-3.
[44] A. Wang et al., "YOLOv10: real-time end-to-end object detection," Oct. 30, 2024, *arXiv preprint*: arXiv:2405.14458. doi: 10.48550/arXiv.2405.14458.
[45] C. Zou, S. Yu, Y. Yu, H. Gu, X. Xu, Side-scan sonar small objects detection based on improved YOLOv11, JMSE 13 (1) (Jan. 2025) 162, https://doi.org/10.3390/jmse13010162.
[46] Y. Tian, Q. Ye, D. Doermann, YOLOv12: attention-centric real-time object detectors, arXiv preprint (Feb. 18, 2025), https://doi.org/10.48550/arXiv.2502.12524 arXiv:2502.12524.
[47] M. Guo, et al., First report of plum bark necrosis stem pitting-associated virus affecting sweet cherry in Yunnan, China, J. Plant Pathol. 104 (2) (May 2022) 829–830, https://doi.org/10.1007/s42161-022-01037-x.
[48] J. Lin, G. Hu, J. Chen, Mixed data augmentation and osprey search strategy for enhancing YOLO in tomato disease, pest, and weed detection, Expert. Syst. Appl. 264 (Mar. 2025) 125737, https://doi.org/10.1016/j.eswa.2024.125737.
[49] T. Zhang, W. Xu, B. Luo, G. Wang, Depth-wise convolutions in vision transformers for efficient training on small datasets, Neurocomputing. 617 (Feb. 2025) 128998, https://doi.org/10.1016/j.neucom.2024.128998.
[50] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEe Trans. Pattern. Anal. Mach. Intell. 42 (8) (Aug. 2020) 2011–2023, https://doi.org/10.1109/TPAMI.2019.2913372.
[51] Z. Zhao, X. Ma, Y. Shi, X. Yang, Multi-scale defect detection for plaid fabrics using scale sequence feature fusion and triple encoding, Vis. Comput. 41 (7) (May 2025) 5205–5221, https://doi.org/10.1007/s00371-024-03716-5.
[52] I. Park, S. Kim, Performance indicator Survey for object detection, in: *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, Busan, Korea (South), IEEE, Oct. 2020, pp. 284–288, https://doi.org/10.23919/ICCAS50221.2020.9268228.
[53] S. Liu, D. Huang, Y. Wang, Learning spatial fusion for single-shot object detection, arXiv preprint (Nov. 25, 2019), https://doi.org/10.48550/arXiv.1911.09516 arXiv:1911.09516.
[54] K. Han, Y. Wang, J. Guo, E. Wu, ParameterNet: parameters are all you need, arXiv preprint (Jan. 14, 2024), https://doi.org/10.48550/arXiv.2306.14525 arXiv:2306.14525.
[55] Q. Hou, C.-Z. Lu, M.-M. Cheng, J. Feng, Conv2Former: a simple transformer-style ConvNet for visual recognition, arXiv preprint (Nov. 22, 2022), https://doi.org/10.48550/arXiv.2211.11943 arXiv:2211.11943.
[56] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, Lake Tahoe, NV, Mar. 2018, pp. 839–847, https://doi.org/10.1109/WACV.2018.00097.
[57] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEe Trans. Pattern. Anal. Mach. Intell. 39 (6) (Jun. 2017) 1137–1149, https://doi.org/10.1109/TPAMI.2016.2577031.
[58] P. Chutichaimaytar, Z. Zongqi, K. Kaewtrakulpong, T. Ahamed, An improved small object detection CTB-YOLO model for early detection of tip-burn and powdery mildew symptoms in coriander (Coriandrum sativum) for indoor environment using an edge device, Smart Agricultural Technology 12 (Dec. 2025) 101142, https://doi.org/10.1016/j.atech.2025.101142.
[59] Y. Li, et al., TeaBudNet: a lightweight framework for robust small tea bud detection in outdoor environments via weight-FPN and adaptive pruning, Agronomy 15 (8) (Aug. 2025) 1990, https://doi.org/10.3390/agronomy15081990.