

Article

YOLO-SPM: Lightweight Apple Detection Algorithm in Complex Orchard Environments

Jingyue Li ^{1,2,3}, Hongfei Yang ^{1,2,3}, Guangchuan Hou ^{1,2}, Junqi Xu ^{1,2}, Jinyong Zhu ^{1,2}, Zhiyuan Zhang ^{1,2,3}, Jingbin Li ^{1,2,3,*} and Shuanming Li ^{1,2,*}

¹ College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, China; 20242009034@stu.shzu.edu.cn (J.L.); yanghf20@shzu.edu.cn (H.Y.); 20252109007@stu.shzu.edu.cn (G.H.); xujunqi@stu.shzu.edu.cn (J.X.); zhujinyong@stu.shzu.edu.cn (J.Z.); zzy_mac@shzu.edu.cn (Z.Z.)

² Xinjiang Production and Construction Corps Key Laboratory of Modern Agricultural Machinery, Shihezi 832003, China

³ National-Local Joint Engineering Laboratory for Specialty Crop Production Machinery and Equipment, Shihezi 832003, China

* Correspondence: lijingbin@shzu.edu.cn (J.L.); lsm_inf@shzu.edu.cn (S.L.)

Abstract

Under the dwarf-rootstock dense planting method, existing apple detection models for intelligent harvesting suffer from excessive parameter counts that hinder deployment on resource-constrained devices, while lightweight alternatives often sacrifice detection accuracy. To address this dilemma, this paper proposes YOLO-SPM, a lightweight apple detection model based on the YOLOv12n architecture, specifically designed for complex orchard environments. The core innovation lies in a problem-driven, three-stage collaborative optimization strategy: first, PConv is introduced to replace standard convolutions in the A2C2f module, reducing computational redundancy by exploiting channel-wise feature similarity of apple targets; second, the parameter-free SimAM attention mechanism is embedded in the neck network to enhance the model's focus on occluded fruit features without increasing model size, while MBConv is integrated into the detection head to further reduce computational cost; third, WIoU v3 is adopted as the loss function to compensate for the accuracy loss incurred by lightweight design through its dynamic focusing mechanism on difficult samples. This complementary design ensures that each module addresses a distinct bottleneck of the native YOLOv12n in orchard scenarios, achieving a balance between efficiency and accuracy rather than simple module stacking. Experimental results demonstrate that YOLO-SPM achieves a precision of 92.8% and mAP@0.5 of 93.1%, outperforming the baseline by 4.8 and 5.3 percentage points, respectively, while reducing parameter count, FLOPs, and memory footprint by 40.2%, 35.4%, and 41.8%. This study provides a feasible solution for high-precision apple identification in dwarf-rootstock dense planting orchard environments, with the potential for integration into automated harvesting systems upon future on-device validation.

Keywords: apple; unstructured orchard; cluttered environment object detection; YOLO-SPM; lightweight

Academic Editor: Hongbin Pu

Received: 14 May 2026

Revised: 24 June 2026

Accepted: 25 June 2026

Published: 26 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

Apples are an important edible fruit globally [1]. As the world's leading apple producer and consumer, China's annual apple production and planting area over the past

ten years are shown in Figure 1. In 2023, the annual apple production reached 4960.17 million tons, accounting for over 50% of the world's total production. The increase in apple production poses a huge challenge to harvesting efficiency. Since apple harvesting in China is still primarily manual, it is inefficient and costly [2–5]. In 2023, the average total cost of apple production per unit area in China was CNY 5451.3/mu. Therefore, intelligent apple-picking robots have huge development prospects [6]. In the development of intelligent apple-picking systems, how to overcome interference from complex environmental factors and achieve precise and efficient fruit identification is a key technical problem that needs to be solved urgently [7,8]. In natural complex scenes, traditional detection methods are easily affected by lighting changes, leaf occlusion, and background noise interference [9], leading to significant degradation in detection performance and difficulty in meeting practical application requirements. To address labor shortage issues, improve production efficiency, and reduce production costs, it is necessary to study more efficient and reliable apple identification algorithms that can meet lightweight deployment requirements for picking robots.

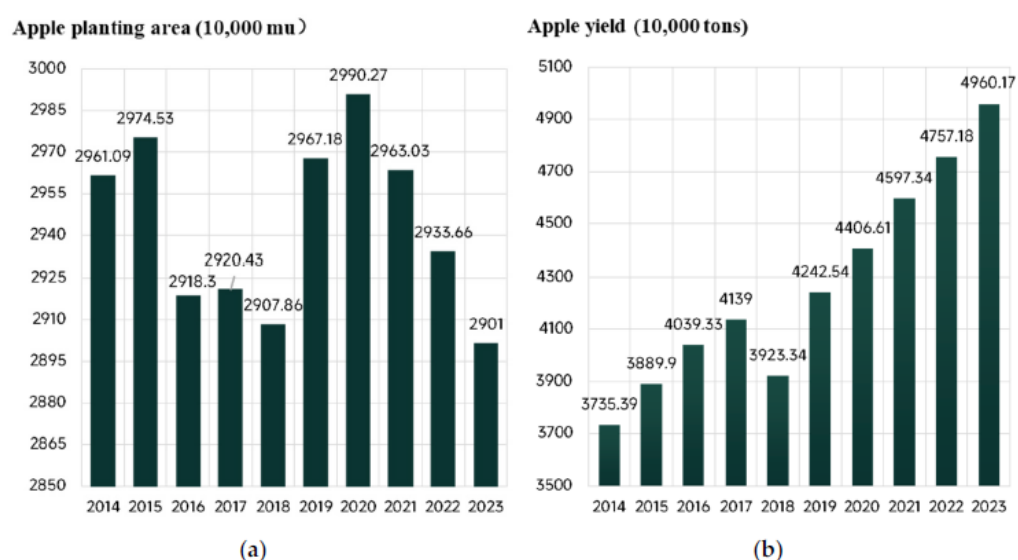


Figure 1. China's apple planting area and production in the past ten years: (a) China's apple planting area in the past ten years; (b) China's apple production in the past ten years.

The rapid development of the artificial intelligence industry has driven new deep learning-based object detection methods, which are mainly divided into two-stage object detection and one-stage object detection. Two-stage object detection first obtains Mask anchor boxes for detection targets, then performs classification and regression, e.g., Mask R-CNN [10] and Faster R-CNN [11]. One-stage object detection algorithms require only one step to achieve end-to-end detection, e.g., SSD [12], RetinaNet [13], and the YOLO [14] series. In particular, the detection methods based on the YOLO series have shown good adaptability in challenging unstructured orchard environments. Many scholars have used the YOLO series for object recognition research. Jiuxin Wang et al. introduced MBConv to improve the standard convolution module in the YOLOv5s backbone, introduced the least squares method to correct misjudged data output results, introduced an object association recognition method, and designed a multi-object picking path, reducing the model size to 6.01 MB and increasing efficiency by 89% [15]. Liusong Yang et al. addressed the detection difficulties of bagged apples in natural environments due to appearance changes, occlusions, and complex backgrounds and proposed a lightweight model AAB-YOLO based on improved YOLOv11. By introducing the ADown module, C3k2_ContextGuided module, Detect_SEAM module, and Inner_EIoU loss function, the model's mAP@50 is

increased to 0.921, its precision is increased to 0.951, its recall is improved by 1.04%, and the parameter count and computational load are reduced by 37.7% and 38.1%, respectively, achieving a balance between high accuracy and lightweight design to meet the real-time detection needs of orchards [16]. Heng Fu et al. proposed an improved apple detection model MSOAR-YOLOv10. A 160×160 scale feature layer is added to the backbone, the C2fCIB convolution module in the backbone embeds a squeeze-excitation attention mechanism, a diversified branch module is introduced to enhance the representation capability of the convolutional neural network, and the normalized Wasserstein distance loss function is adopted, raising the model's precision, recall, and average precision mean to 89.3%, 89.8%, and 92.8%, respectively, which are improvements of 3.1%, 2.2%, and 3.0% over the original model [17]. Zhipeng Wang et al. addressed the difficulty of the real-time detection of the stem/calyx position during the post-harvest commercialization of fruits based on the YOLO-v5 algorithm by performing detection-head search, layer pruning, and channel pruning optimization. The optimized model's parameter count and weight size are reduced by about 71%, the detection speed for the CPU reaches 25.51 frames per second, the mean average precision is 0.880, the F1 score is 0.851, and the detection accuracy for stems/calyxes of different apple varieties reaches 93.89%, providing a technical basis for automated fruit loading and packaging systems [18].

Despite the successes of YOLO-series detectors in various agricultural tasks, the native YOLOv12n model still exhibits notable limitations when applied to dwarf-rootstock dense-planting orchards. First, its standard convolutional layers in the A2C2f module introduce substantial computational redundancy, as the feature maps of apple targets often exhibit high channel-wise similarity; processing all channels with equal density leads to unnecessary FLOPs and memory access, which is particularly detrimental for edge-device deployment. Second, the attention mechanism in YOLOv12n, while effective for generic object detection, tends to produce scattered energy maps under high-noise orchard backgrounds (e.g., dappled light, leaf clutter, and fruit overlap), failing to consistently focus on the discriminative textures of apples such as stems, spots, and partially occluded contours [16]. Third, the detection head using standard convolution lacks efficient feature recalibration for small or heavily occluded fruits, resulting in increased missed detections. Fourth, the CIoU loss function imposes rigid geometric penalties on the aspect ratio, which is suboptimal for near-circular apple targets and sensitive to low-quality annotations caused by occlusion and illumination variation [17]. These deficiencies motivate the design of YOLO-SPM, which addresses each bottleneck with tailored lightweight modules.

Although existing research has achieved certain results in network lightweighting or recognition accuracy, there remain two prominent challenges: the performance of lightweight models is insufficient, especially in terms of robustness and real-time processing capability under complex environmental conditions; on the other hand, high-performance models lack lightweighting, leading to high resource consumption and difficulty in deployment on embedded platforms with limited computational power [15,17]. To address the above issues, this study takes apples in a dwarf-rootstock dense-planting orchard environment as the research object and, based on the YOLOv12n model, proposes a lightweight apple detection algorithm that has the potential for deployment on edge devices: YOLO-SPM. The overall method of this paper is shown in Figure 2.

The main work completed includes the following four points:

- (1) In the baseline model, introduce PConv in the A2C2f module to replace the ordinary convolution structure, improving computational efficiency.
- (2) Introduce the SimAM attention mechanism in the neck network to enhance recognition accuracy without increasing model complexity or computational burden.
- (3) Fuse MBConv in the detection head for optimization, improving recognition efficiency and further reducing model computational load.

- (4) Introduce WIoU v3 to improve the loss function, achieving optimal allocation of limited computational resources and enhancing model generalization capability. This research result provides a feasible solution for effectively handling apple recognition problems in resource-constrained environments, filling many limitations of the existing detection models.

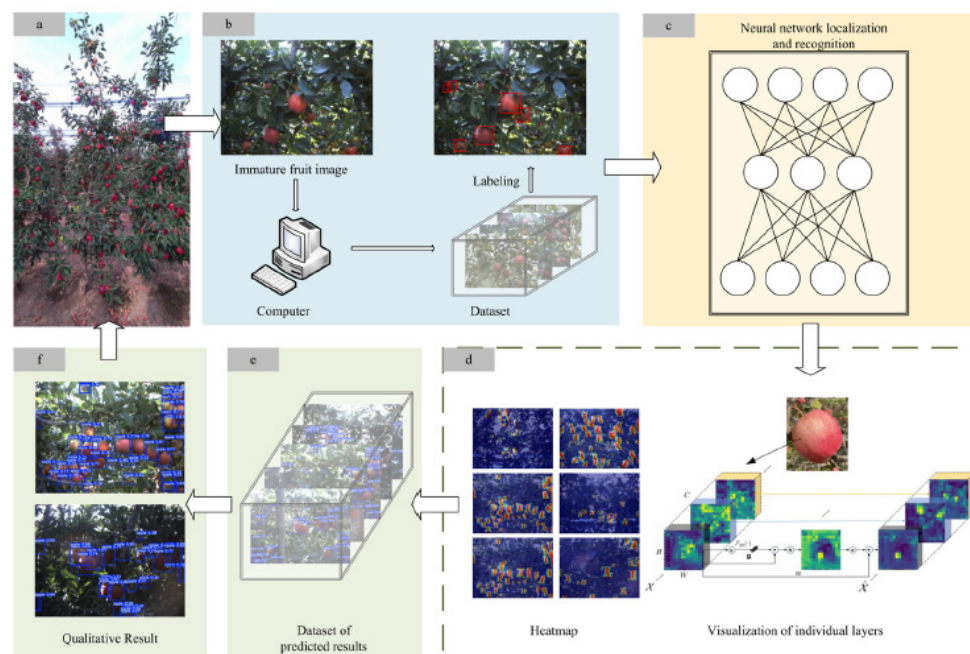


Figure 2. Method overview illustration: (a) image collection; (b) data labeling and dataset preparation, red bounding boxes are used to indicate apple targets; (c) model training; (d) visualization of individual layers and heatmap results; (e) dataset; (f) ground truth result graph.

2. Materials and Methods

2.1. Image Acquisition

The image dataset was captured from late September to early October 2025 at a dwarf-rootstock dense-planting apple orchard in the Fifth Regiment, 13th Company of Aksu District, Xinjiang Uyghur Autonomous Region. At the time of image acquisition, the apples were already mature and about to enter the concentrated harvest period; the fruit skin showed red coloration, and there was severe fruit-to-fruit occlusion. The orchard's apple cultivation methods consist of V-shaped trellis-assisted planting and standard dwarf-rootstock planting.

Multiple image acquisition devices were used, and the images were saved in JPG format. The camera was fixed at a distance of 0.5 m from the tree planting point and moved back and forth along the plant row direction to simulate actual picking viewpoint changes. During acquisition, for five consecutive days, images were captured at different times of day in different planting zones to enrich the dataset. After quality assessment of the collected images, 5325 images were retained for model training and testing. As shown in Figure 3, the dataset includes various situations such as leaf occlusion, fruit overlap, branch occlusion, and sufficient light without occlusion under different lighting conditions.



Figure 3. Dataset examples (partial): (a) sufficient light without occlusion; (b) glare and backlight; (c) uneven lighting; (d) leaf occlusion; (e) fruit overlap; (f) branch/foreign object occlusion.

2.2. Data Set Pre-Processing

The images were manually annotated using the Labelme image annotation software, drawing the minimum bounding rectangle for apple targets in the tree images. To facilitate intelligent harvesting, apples with large-area occlusion or full occlusion were not annotated, and the generated annotation files were saved in XML format.

To enrich the image dataset and improve the model's generalization ability and robustness, data augmentation was applied to the training set images, including horizontal rotation, vertical rotation, central cropping, brightness variation, contrast enhancement, saturation transformation, Gaussian noise, and salt-and-pepper noise, among other augmentation methods.

The original images were captured using multiple consumer-grade smartphone devices, primarily including Huawei Mate 70 Pro, Xiaomi 14, and so on, and saved in JPG format. The original resolution of all images is 1920×1080 pixels. During model training and inference, each image is uniformly resized to 640×640 pixels via letterboxing (preserving the original aspect ratio while padding the remaining areas with gray pixels) to meet the input requirement of the YOLOv12n backbone. Before each training session, the entire dataset of 5325 images was randomly split into training, validation, and test sets at a ratio of 7:2:1, resulting in approximately 3727 images for training, 1065 for validation, and 533 for testing. This random partitioning strategy ensures the model's generalization ability across different orchard scenes and acquisition conditions.

2.3. YOLO-SPM Apple Detection Model

As a representative one-stage object detection algorithm, the YOLO series models integrate localization and classification into an end-to-end regression task; compared with

two-stage object detection algorithms, they offer higher real-time performance and can achieve efficient detection [19,20].

In dwarf-rootstock dense-planting orchards, apples grow in an environment where branches intersect and leaves overlap; dense occlusions and cluttered backgrounds can easily lead to missed detections or false detections, while orchard operations such as intelligent harvesting and automatic inspection impose higher requirements on the detection model's real-time performance and accuracy.

Therefore, YOLOv12n was selected as the baseline for the following reasons:

- (1) It is the smallest and fastest variant in the YOLOv12 series, making it the most suitable starting point for lightweight design.
- (2) Compared to YOLOv8n and YOLOv11n, YOLOv12n incorporates an area-attention mechanism (A2) that provides stronger multi-scale feature representation, which is beneficial for detecting apples of varying sizes in orchard environments.
- (3) At the time of this study, YOLOv12n was the state-of-the-art YOLO variant with the best trade-off between speed and accuracy among nano-scale models.

To ensure a fair comparison, we also included YOLOv8n and YOLOv11n in the comparative experiments, and YOLO-SPM consistently outperforms them, as shown in the ablation experiment section.

The YOLOv12 series includes five specifications: n, s, m, l, and x. Among them, YOLOv12n, the smallest specification in the series, has extremely fast inference speed and relatively high detection accuracy. However, in the dwarf-rootstock dense-planting mode, the common complex environment of fruit overlap and branch-leaf occlusion, as well as the inevitable complex lighting conditions during image acquisition, cause the existing attention mechanism to fail to accurately capture the corresponding target features, easily leading to feature confusion and resulting in higher false detection and missed detection rates.

Studies on lightweight apple detection have primarily focused on pruning or single-module substitution [15,18], not multi-module synergistic integration. Attention mechanisms have been applied to fruit detection [16] but without considering the joint optimization of architecture and loss function. Loss function optimization has been explored [17] but without integration with lightweight architectural design. Therefore, this study proposes a lightweight apple detection algorithm, YOLO-SPM, based on the YOLOv12n framework [21,22]. To overcome the aforementioned limitations, YOLO-SPM introduces four targeted improvements:

- (1) Replacing the standard convolutions in the A2C2f module with Partial Convolution to reduce channel-wise redundancy.
- (2) Embedding a parameter-free SimAM attention module in the neck to enhance salient feature capture without extra parameters.
- (3) Substituting the conventional detection head with an MBConv-based lightweight head that maintains feature discriminability.
- (4) Adopting the WIoU v3 loss function with a dynamic focusing mechanism to handle low-quality samples and near-circular apple shapes.

The subsequent subsections detail each adaptation and explain its specific benefits for apple detection in complex orchards. The improved network architecture diagram and module diagram are shown in Figure 4.

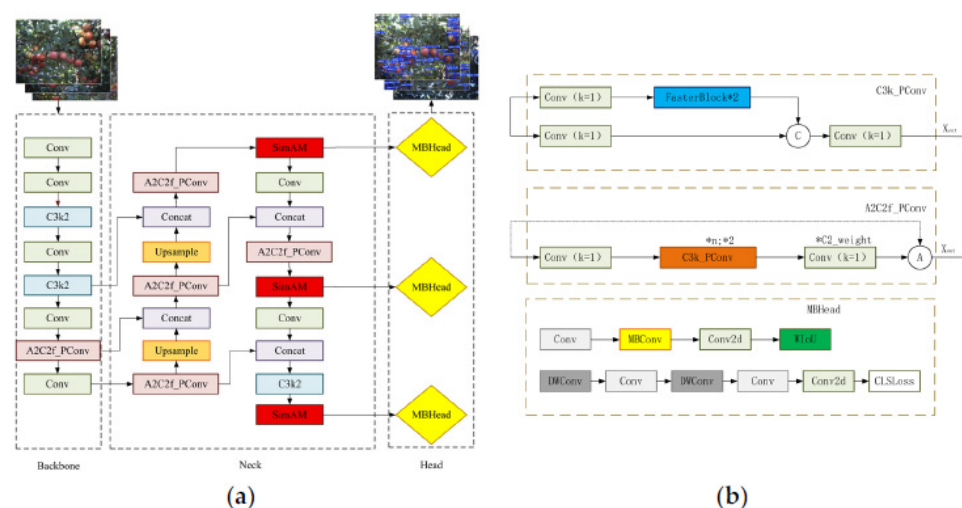


Figure 4. Network architecture diagram: (a) YOLO-SPM structure diagram; (b) structure diagram of the improved module in YOLO-SPM.

In this paper, for the feature extraction module A2C2f in the backbone and neck networks, the C3k module, which connects multiple convolutional layers to implement attention mechanism fusion and dynamic receptive field adjustment, is lightweighted: using FasterBlock to replace the bottleneck to obtain C3k_PConv and replacing the ABlock module in A2C2f to obtain the A2C2f_PConv module achieves lightweight improvement. A SimAM attention module is inserted before the detection head, which significantly improves small-target detection performance with very little computational overhead while preserving important fruit features, enhancing computational efficiency, and further reducing the computational cost. In addition, the standard convolution used for bounding-box regression in the detection head Detect is replaced by MBConv to obtain MBHead, lightweighting the detection head and speeding up prediction inference. The bounding-box regression loss function CIoU in Detect is replaced by WIoU v3; although this causes a slight increase in computational cost, it brings significant accuracy improvement and training stability enhancement, making it suitable for apple recognition scenarios characterized by small-target detection and difficult samples.

2.4. A2C2f_PConv

The apple identification model needs to be deployed on edge devices, so there is a certain requirement for the model's memory footprint. In the original YOLOv12 architecture, each A2C2f module uses two 1×1 standard convolutions, which means that in each computational step, a large number of convolution kernel operations are required, bringing a significant computational burden [23], especially for edge devices, which is a huge computational overhead.

Several lightweight convolution designs have been proposed to reduce this computational burden, as illustrated in Figure 5. However, each has inherent limitations for deployment on resource-constrained devices: Group Convolution [24] reduces FLOPs by dividing channels into groups but may degrade feature representation when cross-group information is required; Ghost Convolution [25] generates features via low-cost operations but introduces additional memory access overhead; depthwise separable convolution [26] decouples spatial filtering and channel combination, yet its independent channel processing ignores inter-channel correlations, weakening feature extraction capability [27]. Therefore, we adopt Partial Convolution (PConv) as a more suitable alternative for apple detection, as detailed below.

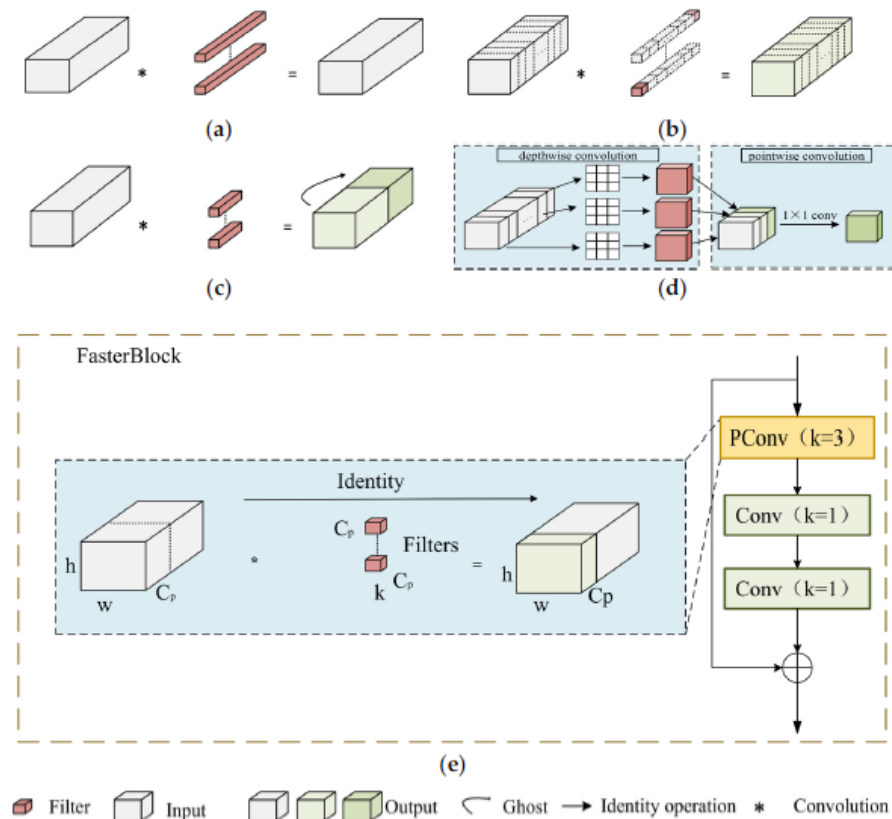


Figure 5. Structure diagram of different lightweight convolution layer methods: (a) standard convolution; (b) Group Convolution; (c) Ghost Convolution; (d) depthwise separable convolution; (e) Partial Convolution.

As shown in Figure 5e, after comprehensive comparison, this paper designs an improved PConv_A2C2f module by borrowing FasterBlock from FasterNet and Partial Convolution [27]. PConv applies filters only on a few input channels, leaving the other channels untouched, leveraging the similarity between different channel features and the redundancy in their feature maps to optimize computational cost and reduce floating-point operations.

Let the input feature map be $X \in R^{C \times H \times W}$ and the output feature map be $Y \in R^{C \times H \times W}$, assuming the same input and output channels for simplicity. A standard convolution with kernel size $K \times K$ has a computational cost (FLOPs) of:

$$FLOPs_{std} = K^2 \times C_{in} \times C_{out} \times H \times W. \quad (1)$$

For PConv, we apply the convolution only on a subset of $C_p = r \times C$ input channels, where $r \in (0, 1]$ is the channel retention ratio (typically $r = \frac{1}{4}$ in practice). The remaining $(1 - r)C$ channels are directly passed through without computation. The FLOPs for PConv become:

$$FLOPs_{PConv} = K^2 \times C_p \times C_{out} \times H \times W = r \times FLOPs_{std}. \quad (2)$$

When $r = \frac{1}{4}$, PConv reduces the computational cost to 25% of the standard convolution. Moreover, because PConv maintains the original channel topology for the untouched channels, no additional memory read/write operations are introduced for those channels, avoiding the “low FLOPs but high memory access” pitfall common in depthwise separable or Ghost Convolutions. The total memory access volume (MAC) for PConv is approximately:

$$MAC_{PConv} = H \times W \times (2C_p + K^2 \times C_p \times C_{out}). \quad (3)$$

which is also reduced proportionally to r . This mathematical advantage directly translates into lower latency and energy consumption on resource-constrained edge devices.

Compared with traditional convolution, it performs high-density spatial convolution only on a subset of input channels, preserving the complete channel information flow topology, optimizing continuous memory access and regular computation patterns, reducing redundant computation and memory access, effectively lowering computational and memory costs, and improving network runtime efficiency, making it more suitable for modern GPU parallel computing architectures and achieving higher computational efficiency in actual hardware deployment.

Apples in dense-planting orchards exhibit strong channel-wise feature similarity due to their uniform color and texture across different spatial locations. Standard convolutions process all channels equally, generating redundant computations that are unnecessary for this redundancy pattern. PConv exploits this similarity by applying dense convolution only on a fraction of input channels while leaving the others untouched, thereby preserving essential information with significantly reduced FLOPs and memory access. This design is particularly beneficial for real-time apple detection on embedded harvesters, where computational budgets are strictly limited.

2.5. SimAM

The attention mechanism extracts the most salient features from the input sequence by computing a weighted combination of input vectors, representing an efficient feature extraction method. The attention module of YOLOv12 employs a complex filter bank that can focus on a single target when multiple different objects appear in the visual field. However, in densely occluded and cluttered orchard scenarios, its global energy computation on the feature map tends to be scattered due to high noise background, dispersing the attention distribution on key fruit regions, making it difficult to capture fruit texture features and affecting the response of occluded apples and edge information. Therefore, to filter out feature combinations that assist recognition, this study integrates the SimAM [28] attention mechanism into the YOLOv12 architecture, which can resolve the feature misalignment problem caused by directly stacking multi-scale components and thus improve overall performance.

As shown in Figure 6, in traditional attention mechanisms, the Channel Attention Mechanism [29] mainly models inter-channel dependencies to optimize features. It treats all neurons within the same channel as identical processing units and uses a two-dimensional weight vector to recalibrate channel-wise features, which cannot compute true three-dimensional weights and requires substantial computational resources. The Spatial Attention Mechanism [30] focuses on establishing a feature selection mechanism in the spatial dimension but remains limited to a single-dimension calibration. The SimAM module is innovatively based on the spatial suppression principle observed in mammalian brains, where the most informative neurons are those that exhibit firing patterns markedly different from the surrounding neurons, and active neurons may suppress the activity of neighboring neurons. To fully exploit the neuron's value, it proposes an improved energy function:

$$e_t(w_t, b_t, y, x_i) = \frac{1}{M-1} \sum_{i=1}^{M-1} \left(-1 - (w_t x_i + b_t) \right)^2 + \left(1 - (w_t t + b_t) \right)^2 + \lambda w_t^2. \quad (4)$$

where t and x_i are input features, $X \in R^{C \times H \times W}$ represents the target neuron and other neurons in a single channel, i is the index in the spatial dimension, $M = H \times W$ is the number of neurons on this channel, and w_t and b_t are the weight and bias of the linear transformation.

$$w_t = -\frac{2(t - \mu_t)}{(t - \mu_t)^2 + 2\sigma_t^2 + 2\lambda} \quad (5)$$

$$b_t = -\frac{1}{2}(t + \mu_t)w_t. \quad (6)$$

where $\mu_t = \frac{1}{M-1} \sum_{i=1}^{M-1} x_i$ and $\sqrt{\sigma_t^2} = \frac{1}{M-1} \sum_{i=1}^{M-1} (x_i - \mu_t)^2$ are the mean and variance of all neurons on this channel, except (t).

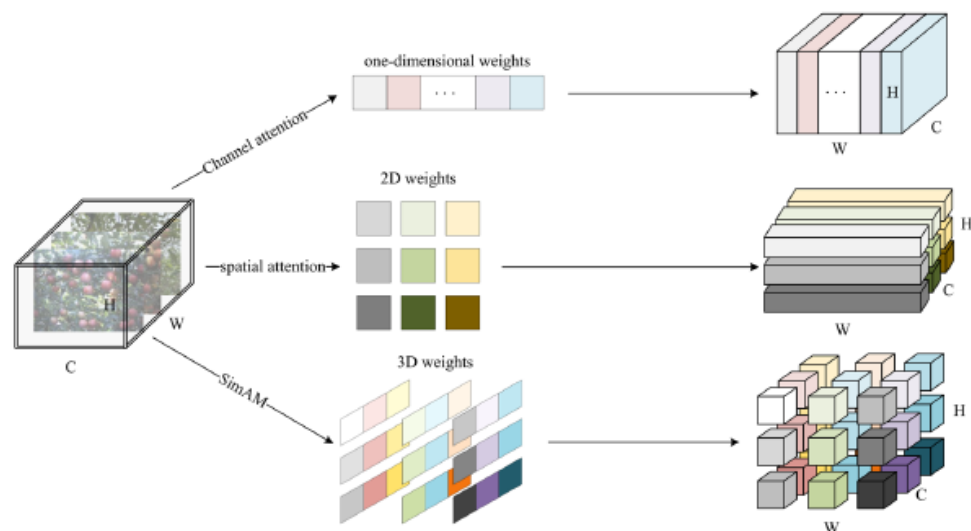


Figure 6. Schematic diagram of different attention mechanisms modules.

Assuming all pixels in a single channel follow the same distribution, we can compute the mean and variance for all neurons once and reuse them on that channel, obtaining the minimum energy computation formula:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda}. \quad (7)$$

where $\hat{\mu} = \frac{1}{M} \sum_{i=1}^M x_i$ and $\hat{\sigma}^2 = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{\mu})^2$ are for the global channel statistics, and $\mu_t = \frac{1}{M-1} \sum_{i \neq t} x_i$ and $\sigma_t^2 = \frac{1}{M-1} \sum_{i \neq t} (x_i - \mu_t)^2$ are for the per-neuron statistics.

The derived closed-form solution enables parameter-free attention weight calculation. It not only enhances the theoretical interpretability of the module but also improves the model's computational efficiency.

The key discriminative features of apples in cluttered orchard scenes—such as fruit stems, calyxes, surface spots, and the boundary contours of partially occluded fruits—are often subtle and easily overwhelmed by background noise. Traditional channel or spatial attention mechanisms require additional parameters and may still mislocalize attention under high interference. SimAM's parameter-free, neuron-level energy function computes a three-dimensional attention weight that inherently suppresses the surrounding irrelevant neurons, making it exceptionally suited for highlighting these fine-grained apple features without increasing model size. The improved attention focus directly translates into higher recall for occluded or small-sized apples. Therefore, the SimAM module is inserted before the detection head to address the key issues of leaf occlusion and cluttered background in dwarf-rootstock dense-planting orchards.

2.6. MBConv_Head: Lightweight Detection Head

In apple detection tasks, target features are often partially occluded and subject to background noise interference. Traditional convolution operations suffer from

insufficient feature expression and high computational redundancy when processing such targets. To enhance the detection head's sensitivity to target features while maintaining model efficiency, this study introduces the MBConv module [31] from EfficientNet [32] to construct the MBConv_Head detection head. Efficient feature extraction and enhancement are achieved through an inverted residual structure and depthwise separable convolution.

As shown in Figure 7, the MBConv module uses a 1×1 convolution to increase dimensions and expand channels, obtaining more dimensions of learnable features; it relies on the depthwise convolution in depthwise separable convolution to extract spatial features, and pointwise convolution fuses channel features, greatly reducing the computational load. Compared with traditional convolution, it can maintain good feature extraction capability while reducing computational cost, making it more suitable for edge devices and other scenarios with limited computational resources. The SE module weights channel features, automatically focusing on important feature channels and suppressing unimportant ones, effectively extracting and utilizing features to improve model accuracy and generalization. Finally, a 1×1 convolution reduces dimensions for feature fusion and compression to obtain the output feature map. Thanks to the shortcut connection, the input features are added to the output features, enabling the network to better learn the mapping between input and output.

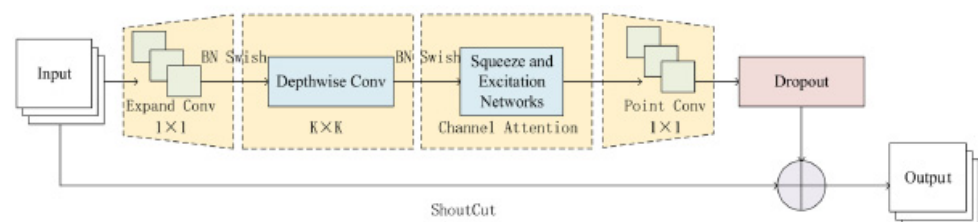


Figure 7. Schematic diagram of the MBConv module.

Through a three-stage cascade transformation architecture, the number of model parameters is reduced, decreasing storage requirements and the risk of over-fitting, making the model more lightweight and easier to deploy. Therefore, introducing MBConv can help the detection head maintain computational efficiency while significantly enhancing the discriminative ability of features in complex backgrounds. MBConv_Head processes three feature maps (layers 14, 17, 20; channels 64, 128, 256), each with an expansion ratio of 4, 3×3 depthwise convolution and SE ratio 0.25. The total model has 2,248,371 parameters and 5.2 GFLOPs (484 layers), reducing the parameter count by 12.2% and FLOPs by 20.0% versus the standard head.

The detection head in lightweight models often becomes a bottleneck: standard convolutions either consume too many resources or lose expressive power. MBConv addresses this through an inverted residual structure and depthwise separable convolution, which reduce computational load while maintaining a large receptive field through the SE module. For apple detection, where targets vary in scale and are often partially occluded, the SE module's channel-wise recalibration helps the head focus on the most informative channels. The shortcut connection also stabilizes training when the gradient flow is weak due to occlusion diversity. Thus, MBConv_Head achieves a favorable trade-off between efficiency and accuracy specifically tailored to orchard environments.

2.7. Loss Function

Compared to the traditional Intersection over Union loss function [33], the bounding-box loss function CIoU [34] chosen for YOLOv12 significantly improves localization accuracy for generic object detection, although it still has some limitations for the apple

detection task in a low-density orchard environment because it imposes strong constraints and penalties on geometric parameters such as center distance and aspect ratio, while apples in images usually appear as near-circular or elliptical and the aspect ratio of their bounding boxes varies within a small dynamic range. The strict, rigid penalty on aspect ratio in the loss function yields limited gain for detecting such relatively uniform-shaped objects and may produce unstable prediction boxes due to the model's generalization, while high-precision geometric measurements greatly increase computational cost. In addition, apple annotation in orchard environments inherently faces difficulties; leaf occlusion, fruit overlap, and illumination variation result in many "low-quality" samples in the dataset with blurry or hard-to-precisely-bound boxes. The strong-geometry-penalty mechanism forces the model to minimize overall loss and sacrifice precise localization of a large number of clear, ordinary samples, preventing optimal resource allocation.

This study selects the dynamic non-monotonic focal loss function WIoU v3 [35] for replacement, introducing a dynamic non-monotonic focusing mechanism to fully exploit the model's potential, using "outlier degree" instead of IoU to evaluate anchor quality and providing a gain-allocation strategy that reduces the competitiveness of high-quality anchors while diminishing harmful gradients from low-quality examples. The formula for WIoU is shown in Equations (8)–(14):

(1) WIoU v1

By constructing a distance-attention mechanism, define:

$$L_{WIoUv1} = R_{WIoU} L_{IoU}, \quad (8)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right). \quad (9)$$

where L_{WIoUv1} is the WIoU v1 function; R_{WIoU} is the penalty term; L_{IoU} is the overlap between predicted and ground-truth boxes; x is the x-coordinate of the predicted point; x_{gt} is the x-coordinate of the ground-truth point; y is the y-coordinate of the predicted point; y_{gt} is the y-coordinate of the ground-truth point; W_g and H_g are, respectively, the width and height of the real anchor box. When the real apple anchor box overlaps well with the target box, $R_{WIoU} \in [1, e]$ will significantly amplify the ordinary-quality anchor boxes L_{IoU} while the high-quality anchor boxes' $L_{WIoU} \in [0, 1)$ will be markedly reduced, with R_{WIoU} making them focus on the distance between center points.

(2) WIoU v2

Learning from the focal loss idea, construct L_{WIoUv1} as the monotonic focusing coefficient $L_{IoU}^{\gamma^*}$ to obtain:

$$L_{WIoUv2} = L_{IoU}^{\gamma^*} \cdot L_{WIoUv1} \quad (\gamma > 0) \quad (10)$$

The gradient changes as follows with the addition of the focusing coefficient:

$$\frac{\sigma L_{WIoUv2}}{\sigma L_{IoU}} = L_{IoU}^{\gamma^*} \frac{\sigma L_{WIoUv1}}{\sigma L_{IoU}}. \quad (11)$$

Introduce L_{IoU} and its mean value as a normalization factor to solve the problem of slower convergence in later stages.

$$L_{WIoUv2} = \left(\frac{L_{IoU}^*}{\overline{L_{IoU}}}\right)^{\gamma} L_{WIoUv1}. \quad (12)$$

where $\overline{L_{IoU}}$ is the exponential moving average with momentum m , dynamically updating the normalization factor so that the gradient gain $r = \left(\frac{L_{IoU}^*}{\overline{L_{IoU}}}\right)^{\gamma}$ is kept overall at a higher level.

(3) WIoU v3

Define the degree of anomaly as:

$$\beta = \frac{L_{IoU}}{\overline{L_{IoU}}} \quad (13)$$

According to β , construct a non-monotonic focusing coefficient:

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \quad (14)$$

where δ is such that when $\beta = \delta r = 1$ is applied to WIoU v1 to obtain WIoU v3, i.e., $L_{WIoUv3} = r L_{IoUv1}$, and when the true apple anchor box outlier satisfies $\beta = C$ (C is a constant), the true apple anchor box will receive the highest gradient gain. Because $\overline{L_{IoU}}$ is dynamic, the anchor box quality-grading standard is also dynamic, enabling WIoU v3 to make the optimal gradient gain allocation strategy at each moment.

Here, α and δ are hyperparameters ($\alpha = 1.9$, $\delta = 3$). This coefficient directly multiplies the gradient during backpropagation. The resulting behavior is:

For high-quality anchors ($\beta < 1$), r is reduced. This prevents the model from over-focusing on already well-localized boxes, allowing for resources to be allocated elsewhere.

For moderately low-quality anchors ($\beta \in [1, 3]$), r is moderately increased, providing a healthy boost to help refine them.

For extremely low-quality anchors ($\beta \gg 3$), r is suppressed. This reduces the harmful impact of very poor or outlier annotations that would otherwise produce noisy gradients and destabilize training.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v. \quad (15)$$

where v measures aspect ratio consistency. While effective for generic objects, this fixed formulation has two drawbacks for apple detection:

- (1) Apple bounding boxes are nearly circular; their aspect ratios vary within a narrow range. The strict penalty term v adds computational overhead without meaningful benefit and can even degrade performance when annotations are slightly imprecise.
- (2) In orchard datasets, many ground-truth boxes are imperfect due to leaf occlusion, fruit overlap, or motion blur. CIoU forces the model to fit these inaccurate boxes with equal strength, causing overfitting to annotation noise and distracting from well-labeled samples.

WIoU v3 overcomes these limitations by decoupling sample quality assessment from geometric heuristics. The outlier degree β is based solely on the IoU loss relative to the batch average, making it robust to aspect ratio variations and annotation noise. For a heavily occluded apple with a poorly defined bounding box, CIoU would apply a large gradient penalty that may be harmful, whereas WIoU v3 automatically reduces the gradient gain for such extreme outliers, allowing the model to focus on cleaner samples. Meanwhile, for a clear, typical apple, β is low, and WIoU v3 slightly reduces the gradient to avoid over-optimization, improving generalization.

As shown in Table 1, replacing CIoU with WIoU v3 increased precision from 88.5% to 91.6% and mAP@0.5 from 88.4% to 95.4% while keeping the parameter count and FLOPs unchanged. This substantial gain, achieved without any additional computational cost during inference, confirms that the dynamic focusing mechanism is particularly well-suited to the irregular sample quality distribution in complex orchard environments.

Table 1. Comparison results of loss functions.

Metrics	YOLOv12n (CIoU)	DIoU	GIoU	WIoU v3
Precision	88.5	86.9	89.8	91.6
mAP@0.5	88.4	90.2	93.8	95.4
Recall	0.854	0.857	0.87	0.873

The CIoU loss used in YOLOv12n imposes strong penalties on center distance and aspect ratio. However, apple bounding boxes are nearly circular, and their aspect ratios vary within a narrow range, making the rigid geometric penalty largely unnecessary and even harmful when annotations are imprecise due to occlusion or motion blur. Moreover, orchard datasets inherently contain “low-quality” samples. WIoU v3 introduces a dynamic non-monotonic focusing mechanism that uses an “outlier degree” to evaluate anchor quality, reducing the harmful gradients from low-quality examples while allowing high-quality anchors to contribute more. This property is particularly valuable for apple detection, where perfect bounding-box annotations are difficult to obtain. The improved loss leads to more stable training and higher final mAP@0.5 without increasing inference cost.

3. Results

3.1. Evaluation Metrics

In this study, all comparative models were trained with identical data augmentation strategies (horizontal rotation, vertical rotation, central cropping, brightness variation, contrast enhancement, saturation transformation, Gaussian noise, and salt-and-pepper noise) to ensure a fair comparison. The number of parameters (params), floating-point operations (FLOPs), and memory consumption are used to evaluate the complexity of the YOLO-SPM model, and the detection performance of the YOLO-SPM model is evaluated using precision (P) and mean average precision at IoU = 0.5 (mAP@0.5). The corresponding formulas are as follows.

$$Params = C_{in} * C_{out} * K * K, \quad (16)$$

$$FLOPs = W * H * K * K * C_{in} * C_{out}, \quad (17)$$

$$P = \frac{TP}{TP + FP}, \quad (18)$$

$$mAP = \frac{\sum_{i=1}^{N_c} AP_i}{N_c}. \quad (19)$$

where C_{in} and C_{out} are, respectively, the number of input and output image channels; K is the filter size; W and H are, respectively, the width and height of the input image; TP is the number of correctly detected apples in the image; FP is the number of incorrectly detected apple fruits in the image; AP is the average precision, i.e., the area under the PR curve; and i is the number of target detection categories.

3.2. Experimental Environment and Parameters

All experiments were conducted on a workstation running Ubuntu 20.04 LTS, with Python 3.10, CUDA 12.1, and PyTorch 2.1.0 as the deep learning framework. The hardware platform comprised an NVIDIA GeForce RTX 4090D GPU (24 GB GDDR6X VRAM) and an Intel(R) Xeon(R) Platinum 8474C CPU (48 physical cores, 2.1 GHz base frequency) with 80 GB of system memory. This workstation was used for model training and offline evaluation rather than as the final deployment target.

The specific training hyperparameters are listed in Table 2.

Table 2. Training parameter settings.

Parameter	Value
Image size/px	640 × 640
Batch size	64

Threads	4
Momentum	0.937
Initial learning rate	0.01
Optimizer	SGD
Epochs	300

3.3. Ablation Experiment

An ablation experiment is a method that systematically evaluates the contribution of each module to the overall system performance by gradually removing or replacing specific components in the model. In this study, YOLOv12n is used as the baseline model, and under unchanged conditions, only one improved module is added each time to analyze the effectiveness of the improvements; the ablation experiment results are shown in Table 3.

Table 3. Ablation experiment.

Metrics	YOLOv12n	+PConv	+SimAM	+MBHead	+WIoU v3
Precision	88.5	86.3	91.2	89.2	92.8
mAP@0.5	88.4	87.4	89.2	89.8	93.1
Parameters	2.56	2.01	2.03	1.53	1.53
FLOPs	6.5	5.4	5.1	4.2	4.2
Memory	5.5	4.3	4.3	3.3	3.2
Usage/MB					

As can be seen from Table 3, YOLO-SPM introduces the improved A2C2f_PConv as the backbone and the main feature extraction module in the neck, reducing the parameter count and computation load to a certain extent, while the accuracy only slightly declines. Subsequently, the SimAM attention mechanism is introduced to reconstruct the neck, further optimizing performance while reducing parameter count, achieving an increase in detection accuracy. After using MBConv_Head as the detection head, the model complexity is further reduced, although the accuracy declines somewhat, but finally using WIoU v3 as the bounding box loss function enables the model to achieve the best performance, compensating for the accuracy loss introduced by the lightweight detection head. Compared with the baseline model YOLOv12n, the precision and mAP@0.5 increase by 4.8 and 5.3 percentage points, respectively, and the parameter count and FLOPs are reduced by 40.2% and 35.4%, respectively, indicating that the YOLO-SPM model achieves model lightweighting while maintaining high detection accuracy.

Although all experiments were conducted on a high-performance workstation, the proposed model's parameter count of 1.53 M and FLOPs of 4.2 G are significantly lower than those of comparable models, indicating strong potential for deployment on edge platforms. Future work will include on-device latency measurements on NVIDIA Jetson Orin and Raspberry Pi platforms.

To justify the selection of SimAM, we compared it with SE, CA, and CBAM under identical settings. SimAM achieves the highest precision of 91.5%, recall of 88.1%, and mAP@0.5 of 90.3%, outperforming all other attention mechanisms. Notably, SE even causes a slight decline in mAP@0.5 compared to the baseline, while CA and CBAM yield only marginal improvements. More importantly, SimAM delivers this superior performance without introducing any additional parameters, whereas the other modules increase model size. This parameter-free property, together with its energy-based three-dimensional attention weights that effectively suppress background noise and highlight

occluded fruit features, makes SimAM particularly suitable for lightweight deployment in orchard environments.

3.4. Comparative Experiment

To comprehensively evaluate the detection performance of the proposed YOLO-SPM model in real orchard scenarios, we conducted a comparative experiment against several representative object detectors, including SSD, Faster-RCNN, DETR, YOLOv5, YOLOv8, and YOLOv10, as well as two recently proposed lightweight apple detection models, AAB-YOLO [16] and MSOAR-YOLOv10 [17]. As presented in Table 4, YOLO-SPM achieves the highest precision of 92.8% and mAP@0.5 of 93.1% among all compared methods, surpassing the baseline YOLOv12n by 4.8 and 5.3 percentage points, respectively. Notably, this superior accuracy is attained with only 1.53×10^6 parameters and 4.2×10^9 FLOPs, which are substantially lower than those of other YOLO-series models and even more efficient than the concurrent lightweight competitors MSOAR-YOLOv10. In terms of real-time inference, YOLO-SPM achieves 268.74 FPS, which is competitive with the fastest model (YOLOv8 at 275.70 FPS) while delivering higher accuracy and consuming less than half the parameters. These results collectively demonstrate that YOLO-SPM strikes an excellent balance between detection accuracy, model compactness, and inference efficiency, making it a highly practical solution for apple detection tasks on resource-constrained harvesting platforms.

Table 4. Ablation study on attention mechanisms.

Metrics	YOLOv12n	+SE	+CA	+CBAM	+SimAM
Precision	88.5	89.0	88.5	89.3	91.5
mAP@0.5	88.4	87.5	89.8	90.1	90.3
Recall	0.854	0.875	0.872	0.879	0.881

It should be noted that the DETR model, as a transformer-based detector, typically requires a significantly longer training schedule to achieve competitive convergence due to its reliance on bipartite matching and the absence of hand-crafted anchor designs. In this study, to ensure a fair comparison across all evaluated models, each model was trained under identical hardware and hyperparameter settings, including fixed 300 epochs without any task-specific tuning for DETR. Under this limited training budget, DETR failed to fully converge, resulting in lower precision and mAP@0.5 than its potential optimum. This outcome is consistent with known observations in the literature that DETR and its variants benefit substantially from extended training or improved training recipes. Therefore, the relatively poor performance of DETR in Table 5 should be interpreted as a consequence of training-epoch limitation rather than an inherent weakness of the transformer-based paradigm. For applications where longer training is feasible, DETR-like architectures may still be considered, but for the current lightweight and rapid-deployment scenario, YOLO-based models remain a more practical choice.

Table 5. Comparative experiment.

Models	SSD	Faster-RCNN	DETR	YOLOv5	YOLOv8	YOLOv10	AAB-YOLO [16]	MSOAR-YOLO [17]	YOLO-SPM
Precision	80.3	86.1	59.9	88.9	89.8	87.2	-	-	92.8
mAP@0.5	74.6	88.2	54.3	90.8	92.6	90.6	-	-	93.1
Parameters	23.75	28.27	36.74	2.51	3.01	2.71	1.61	9.1	1.53
FLOPs	30.43	37.52	36.81	7.2	8.2	8.4	3.9	29.0	4.2
FPS/s	196.96	120.27	80.31	121.63	275.7	163.88	-	-	268.74
Recall	0.711	0.779	0.578	0.824	0.885	0.82	-	-	0.894

F1	0.74	0.88	0.59	0.86	0.87	0.84	-	-	0.88
Memory Usage/MB	95	113.1	147	5.3	6.3	5.8	3.41	19.3	3.2

Despite the above training-related limitation for DETR, the comparison among the remaining models remains fair and clearly validates the effectiveness of the proposed YOLO-SPM. To further illustrate its robustness under challenging conditions, representative detection results and attention heatmaps are presented in Figures 8 and 9 as qualitative visual evidence supporting the quantitative findings.

In the fruit-overlap and uneven-lighting scenes shown in Figure 8 and in the lighting-change, leaf-occlusion, and scale-variation scenes shown in Figure 9, the YOLO-SPM model has fewer missed and false detections than the baseline models, demonstrating good performance in real unstructured orchard scenes.

3.5. Failure Case Analysis

To verify the generalizability and adaptability of the YOLO-SPM algorithm proposed in this paper, the model's recognition performance was tested under extreme environments, as shown in Figure 10, including high exposure, glare, branch and leaf occlusion, and severe fruit stacking. The results indicate that the YOLO-SPM model can still perform the recognition task in extreme scenes. Although the performance is not ideal, it is still better than the original YOLOv12 model.

3.6. Public Dataset Validation

To verify the generalization performance of the YOLO-SPM model proposed in this paper on different datasets, the public dataset used in this study is an open-access dataset without a designated name, comprising 1200 apple images acquired under various orchard conditions and spanning three maturity stages: immature green, mature green, and mature red. To assess the generalization capability of YOLO-SPM, the model was directly evaluated on this dataset in a zero-shot manner (without any fine-tuning), and its performance was compared with that of SSD, DETR, YOLOv5, and YOLOv8 under identical experimental settings. The YOLO-SPM model and other YOLO series models as well as some single-stage models were compared on a public dataset containing immature green apples, mature green apples, and mature red apples. As shown in Table 6, compared with other models, YOLO-SPM maintains the smallest parameter count and FLOPs while still achieving the best precision, recall, and mAP@0.5. The visualization results are shown in Figure 11. In the scenes with varying fruit colors, leaf occlusion, and uneven lighting shown in Figure 11, YOLO-SPM has fewer missed and false detections than the other models, demonstrating good generalization performance in real unstructured orchard scenes.

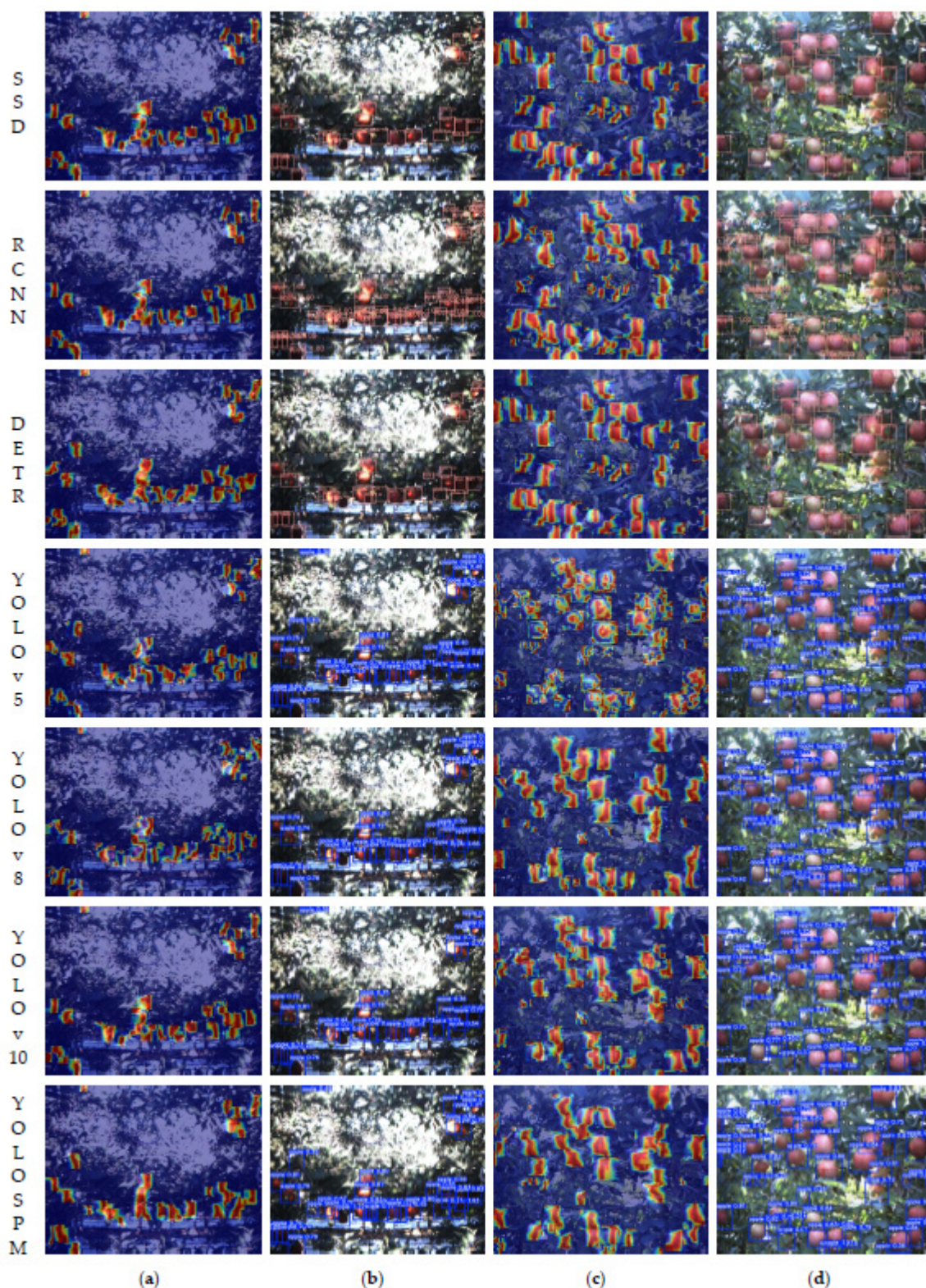


Figure 8. Recognition results under complex environments: (a) heatmap under non-uniform illumination conditions; (b) recognition results under non-uniform illumination conditions; (c) heatmap in overlapping fruit scenes; (d) recognition result image in overlapping fruit scenes. The heatmap colors indicate the attention weight distribution, where warmer colors represent higher attention.

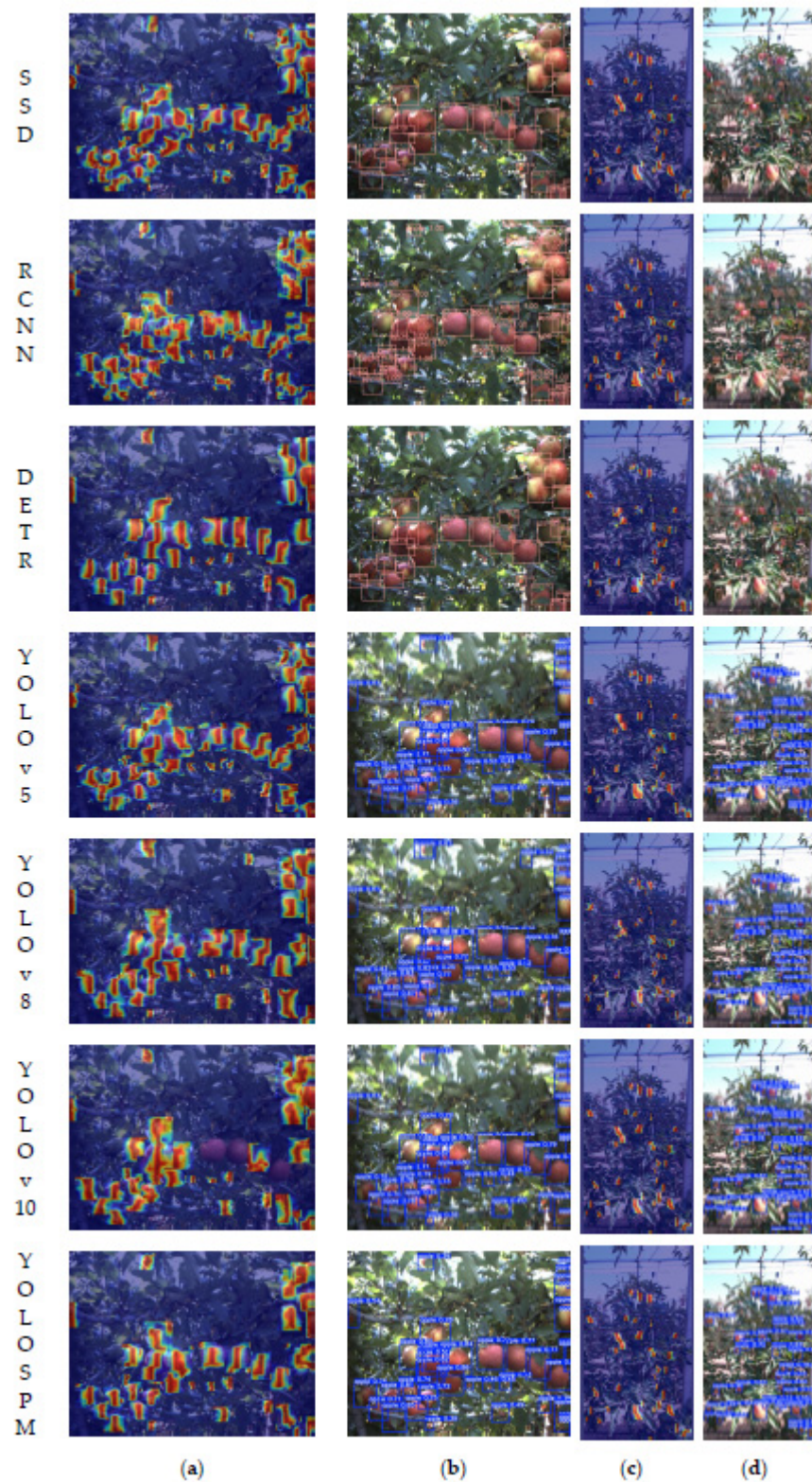


Figure 9. Recognition result image in multi-scale scenes: (a) heatmap under non-uniform fruit color scenes; (b) recognition results in scenes with uneven fruit color; (c) heatmap under branch and leaf occlusion and varying scales; (d) recognition results in scenes with branch and leaf occlusion and varying scales. The heatmap colors indicate the attention weight distribution, where warmer colors represent higher attention.

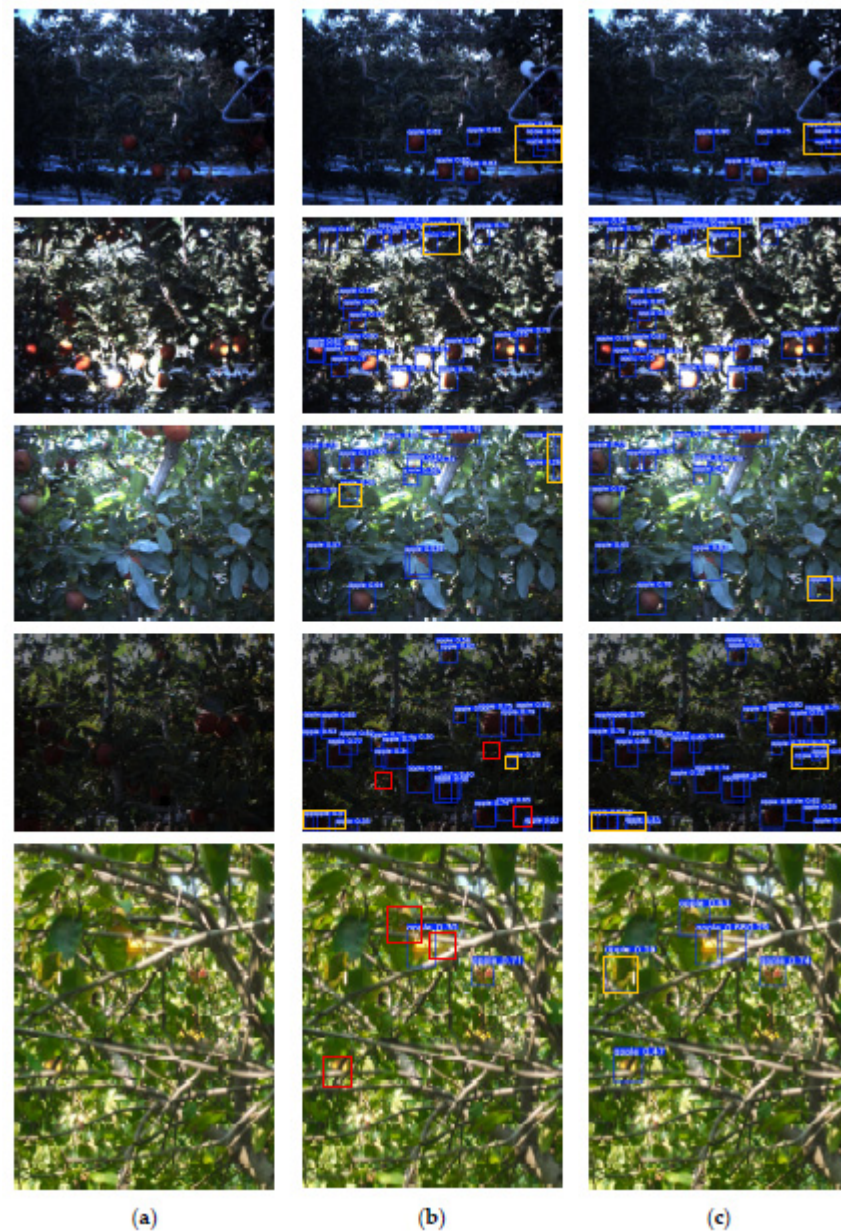


Figure 10. Failure Analysis Result Diagram. Blue boxes indicate algorithm detection results, red boxes indicate missed detections, and yellow boxes indicate false detections. (a) Original image; (b) YOLOv12n model detection result; (c) YOLO-SPM algorithm detection result.

Table 6. Comparison experiment results on public dataset.

Models	SSD	DETR	YOLOv5	YOLOv8	YOLO-SPM
Precision	89.1	53.2	81.4	82.1	91.1
mAP@0.5	85.8	54.0	84.2	83.7	87.2
Parameters	23.75	36.74	2.51	3.01	1.53
FLOPs	30.43	36.81	7.2	8.2	4.2
Memory Usage/MB	95	147	5.3	6.3	3.2

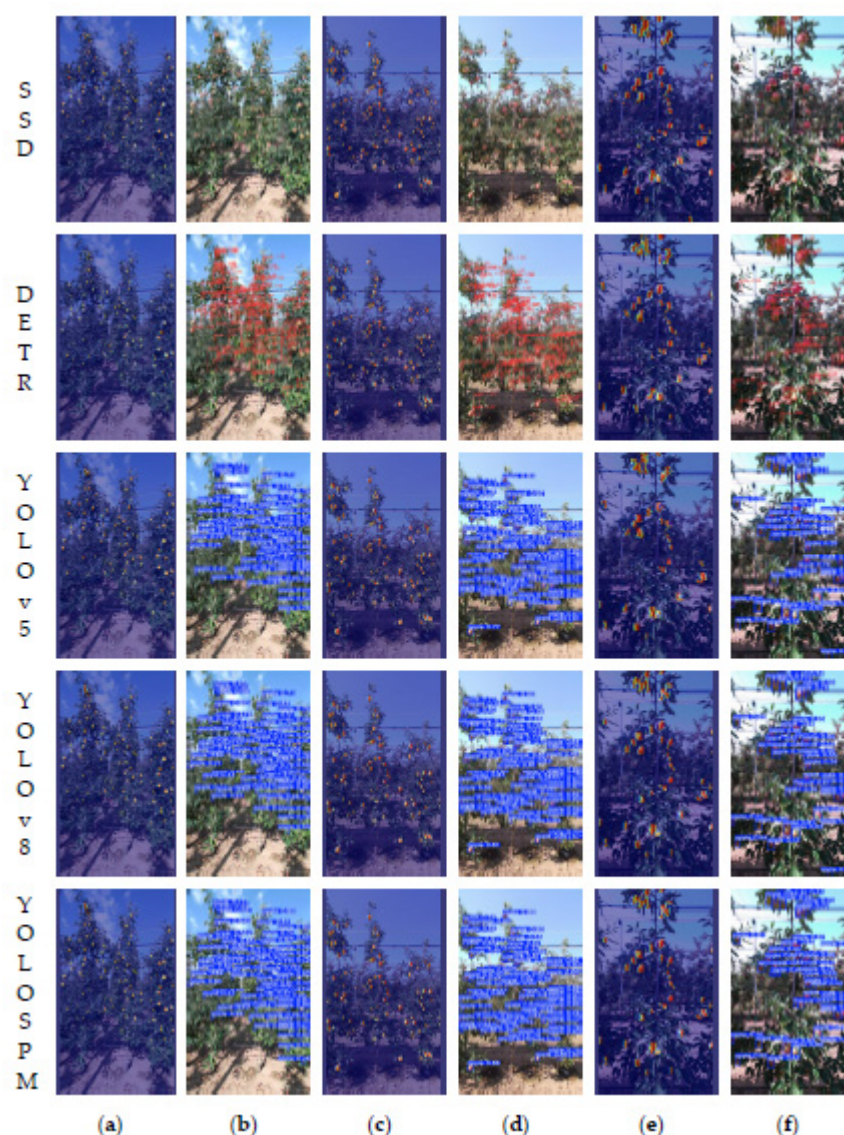


Figure 11. Recognition performance of different models on public datasets. (a,c,e) Heatmaps of the respective models. (b,d,f) Recognition results. The heatmap colors indicate the attention weight distribution, where warmer colors represent higher attention.

4. Discussion

The ablation experiment results clearly reveal the contribution of each improvement module to the overall performance of YOLO-SPM. Replacing the standard convolution in the A2C2f module with PConv reduces FLOPs by 16.9% and the parameters by 21.5%, with a modest performance trade-off that is subsequently compensated by the remaining modules. This indicates that PConv effectively reduces redundant computation and memory access by exploiting the similarity between feature map channels [27], and the slight accuracy loss can be compensated for by the subsequent optimization modules. This phenomenon is consistent with the findings of Chen et al. [23], who reported that standard convolutions introduce significant computational burden in lightweight network design, whereas Partial Convolution can substantially alleviate this issue.

The introduction of the SimAM attention mechanism increased precision and mAP@0.5 to 91.2% and 89.2%, representing improvements of 2.7 and 0.8 percentage points over the baseline model (88.5% and 88.4%, respectively), while the parameter count was further reduced to 2.03×10^6 . This demonstrates that compared with traditional channel attention and spatial attention mechanisms, SimAM can more accurately capture salient

features in complex orchard environments—such as the fruit stems, spots, and edge information of occluded apples—without adding extra parameters. This finding aligns with the core idea proposed by Yang et al., that three-dimensional attention weights derived from an energy function can more effectively suppress background noise interference. In the dwarf-rootstock dense-planting orchard scenario, where branches intersect and leaves overlap, this parameter-free attention design is particularly important for maintaining high detection accuracy.

After adopting MBConv_Head as the detection head, the model parameter count and FLOPs were further reduced to 1.53×10^6 and 4.2×10^9 , achieving a lightweight level, but precision and mAP@0.5 decreased to 89.2% and 89.8%, respectively. This indicates that while the inverted residual structure and depthwise separable convolution in the MBConv module effectively reduce computational cost, some feature expression capability is lost. This trade-off has also been observed in other lightweight detection studies. For instance, Yang et al. reported a similar accuracy fluctuation when introducing the ADown module in AAB-YOLO. However, when WIoU v3 was ultimately introduced as the loss function, the model's precision and mAP@0.5 increased significantly to 92.8% and 93.1%—the optimal values, while maintaining the same compact parameter configuration. This strongly proves that the dynamic non-monotonic focusing mechanism of WIoU v3 can effectively balance the learning of high-quality and low-quality samples. In apple detection tasks, where annotation quality is often degraded by leaf occlusion, fruit overlap, and illumination variation, the “outlier degree”-based gradient gain allocation strategy provides stronger generalization capability than the traditional CIoU loss function, which relies on rigid geometric penalties.

It should be noted that the most significant single-module improvement originates from the WIoU v3 loss function (+7.0 percentage points in mAP@0.5 in the independent test), which is consistent with previous findings that loss function optimization can substantially impact bounding-box regression quality. However, the architectural modules contribute primarily to model compactness and inference efficiency: PConv reduces FLOPs by 16.9%, MBConv_Head reduces parameters by 40.2%, and SimAM maintains accuracy without additional parameters. Therefore, the novelty of YOLO-SPM lies not in any single component but in the synergistic integration of lightweight architecture and dynamic loss function, achieving both high accuracy and efficient deployment—a combination that no existing apple detection model has simultaneously realized.

The comparative experiment results show that YOLO-SPM achieved the best precision, 92.8%, and the best mAP@0.5, 93.1%, among all compared models, with the smallest parameter count of 1.53×10^6 , FLOPs: 4.2×10^9 and memory usage of 3.2 MB. Compared with the two-stage model Faster-RCNN and the anchor-free model DETR, YOLO-SPM demonstrates overwhelming advantages in both accuracy and lightweight design, having potential for deployment on edge devices. Compared with other YOLO series models—such as YOLOv5 with 2.51×10^6 parameters and mAP@0.5 of 90.8% and YOLOv8 with 3.01×10^6 parameters and mAP@0.5 of 92.6%—YOLO-SPM achieves higher accuracy with fewer parameters and computational resources. These results fully validate the effectiveness of the multi-module collaborative optimization strategy proposed in this study. Notably, in fruit overlap and uneven illumination scenarios, YOLO-SPM exhibits significantly fewer missed and false detections than the baseline model. This visual evidence further corroborates the quantitative results—that the combination of the SimAM attention mechanism and WIoU v3 dynamic focusing mechanism enhances the model's robustness to complex backgrounds. The public dataset validation further confirms YOLO-SPM's generalization ability across apple datasets with different maturity levels and color characteristics: immature green, mature green, and mature red apples, suggesting that the model possesses strong cross-dataset adaptability.

Despite YOLO-SPM's excellent performance, the failure case analysis reveals some limitations. Under extreme conditions—such as high exposure, severe glare, dense branch-and-leaf occlusion, and extensive fruit stacking—the model still exhibits missed and false detections, although its performance remains superior to the original YOLOv12n. These extreme scenarios pose challenges common to most vision-based detection systems. On one hand, extreme illumination can cause the loss of fruit texture features; on the other hand, severe occlusion reduces the available pixel area of the targets, making it difficult for even the attention mechanism to extract discriminative features. Future research could explore the following directions:

- (1) Multi-modal or multi-spectral fusion: Integrating RGB images with depth information (e.g., from an RGB-D camera) or near-infrared (NIR) channels could help recover fruit contours under extreme illumination, where texture features are partially lost. This would provide complementary cues for robust detection.
- (2) Small-target and heavily occluded fruit detection head: Inspired by multi-scale detection architectures, a dedicated small-object detection head could be added to the neck network, operating on higher-resolution feature maps to better capture the fine details of occluded or distant apples. Alternatively, a transformer-based feature enhancement module could be explored to model long-range dependencies for occluded targets.
- (3) Edge-computing optimization schemes: Beyond architectural improvements, model compression techniques such as structured pruning, INT8 quantization, and knowledge distillation could be explored to further reduce inference latency on resource-constrained devices like the Jetson Orin or Raspberry Pi. Achieving real-time deployment with end-to-end latency below 30 ms per frame represents a promising direction for enabling practical harvesting robots, though this requires systematic on-device validation in future work.
- (4) Integration with orchard IoT systems: The proposed model can be extended to continuous monitoring scenarios by connecting it to fixed or mobile camera nodes. Real-time fruit detection results could be fused with weather, soil, and growth stage data to enable predictive harvesting scheduling and yield estimation.

5. Conclusions

Aiming at the complex scenario characteristics of densely planted low-stature orchards in unstructured environments, such as branch-leaf occlusion and uneven illumination, as well as the lightweight demand for model deployment, this study proposes a lightweight and efficient apple detection model, using YOLOv12n as the baseline and integrating multiple convolutional neural network architectures (including A2C2f_PConv, SimAM, WIoU v3, and MBConv_Head modules). Among them, a Partial Convolution module is introduced to reconstruct the original A2C2f module in YOLOv12n, forming the A2C2f_PConv structure, which reduces redundant computation and memory access and significantly improves the network's runtime efficiency; the SimAM attention mechanism is integrated into the neck network, greatly enhancing feature extraction capability in complex orchard environments; the detection head combined with the MBConv module achieves higher accuracy while consuming fewer computational resources; the original loss function is replaced with the WIoU v3 loss function, whose dynamic aggregation mechanism adjusts gradient gain according to the magnitude of outliers, balancing the learning outcomes of high-quality and low-quality samples and substantially improving the overall model performance. Data augmentation techniques are employed to increase dataset diversity, thereby optimizing model robustness and generalization. The accuracy improvements are primarily driven by the WIoU v3 loss function, while the architectural modules contribute mainly to model compactness and inference efficiency. The value of

YOLO-SPM lies in the integration of these components to achieve a combined effect—high accuracy from WIoU v3 combined with high efficiency from the lightweight architecture—which no existing apple detection model has simultaneously achieved.

The experimental results verify the YOLO-SPM model's capability to achieve accurate and fast apple detection, providing a feasible visual solution for high-precision fruit identification in complex orchard environments and serving as a potential foundation for future automated harvesting systems.

Author Contributions: J.L. (Jingyue Li): Writing—review and editing, Writing—original draft, Methodology, Investigation, Formal analysis, Conceptualization; H.Y.: Writing—review and editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization; G.H.: Methodology, Formal analysis, Data curation; J.X.: Methodology, Formal analysis, Data curation; J.Z.: Methodology, Formal analysis, Data curation; Z.Z.: Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization; S.L.: Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization, J.L. (Jingbin Li): Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Xinjiang Fruit Industry Technology System (Grant No. XJLGJSTX04-2026); Project of Tianchi Talented Young Doctor (Grant No. CZ002558); Tianchi Talents Introduction Program (BT-2025-TCYC-0074); High-Level Talent Program (RCZK202441); Key Laboratory of Agricultural Equipment in Northwest China; Ministry of Agriculture and Rural Affairs (KFKT202508); the Xinjiang Production and Construction Corps Key Agricultural Core Technology Research Program (NYHXGG2023AA402); Corps-Level Undergraduate Education Reform Project (BTBKXM-2025-Y31) and the Shihezi University Undergraduate Education and Teaching Reform Research Project (JGY-2025-08),

Institutional Review Board Statement: This study did not involve any human participants, animals, genetically modified organisms, or sensitive biological materials. The highest ethical standards were followed in the conduct of the research. All procedures were performed in accordance with relevant ethical guidelines and regulations. The design and implementation of the study primarily focused on ensuring the safety, privacy, and well-being of all participants, as well as the integrity and transparency of the results. No ethical approval was required for this study.

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

Conflicts of 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.

Abbreviations

The following abbreviations are used in this manuscript:

A2C2f	Area-Attention Enhanced Cross-Feature module
CIoU	Complete Intersection over Union
FLOPs	Floating-Point Operations
IoU	Intersection over Union
MBConv	Mobile Inverted Bottleneck Convolution
PConv	Partial Convolution
SimAM	Simple, Parameter-Free Attention Module
SSD	Single-Shot MultiBox Detector
WIoU	Wise Intersection over Union
YOLO	You Only Look Once
mAP@0.5	Mean Average Precision at IoU threshold of 0.5

β	Outlier degree in WIoU v3 loss function
b_t	Bias of linear transformation in SimAM
C_{in}	Number of input channels
C_{out}	Number of output channels
e_t^*	Minimum energy in SimAM
H	Height of input image or feature map
K	Filter size
L_{IoU}	IoU loss
M	Number of neurons on a single channel in SimAM
P	Precision
R_{IoU}	Penalty term in WIoU v1
R	Non-monotonic focusing coefficient in WIoU v3
W	Width of input image or feature map
w_t	Weight of linear transformation in SimAM
μ_t	Mean of neurons on a channel except target neuron
σ^2	Variance of neurons on a channel except target neuron

References

1. Yu, D.; Liu, Y.; Kong, Y.; Yin, J.; Xu, C.; Wang, J.; Wang, G. Algorithm for Recognizing Green Apples Using Image Segmentation and Object Detection. *Agriculture* **2026**, *16*, 247. <https://doi.org/10.3390/agriculture16020247>.
2. Hu, G.; Zhou, J.; Wen, S.; Chen, N.; Chen, C.; Cheng, F.; Chen, Y.; Chen, J. Field Evaluation of a Robotic Apple Harvester with Negative-Pressure Driven End-Effectors on a Simplified 4-DoF Manipulator. *Agriculture* **2026**, *16*, 717. <https://doi.org/10.3390/agriculture16070717>.
3. Bhattacharya, S.; Arunachalam, C.; Zhang, K.; Li, J.; Lu, R.; Li, Z. Detect, Segment, Cluster: Apple Localization for Robotic Harvesting in Complex Orchards. *Smart Agric. Technol.* **2025**, *12*, 101642. <https://doi.org/10.1016/j.atech.2025.101642>.
4. Niu, J.; Yu, Q.; Bi, M.; Zhao, J.; Zhang, T. Deep Reinforcement Learning-Based Cooperative Harvesting Strategy for Dual-Arm Robots in Apple Picking. *Agronomy* **2025**, *15*, 2565. <https://doi.org/10.3390/agronomy15112565>.
5. Wang, J.; Sun, W. Cluster Segmentation and Stereo Vision-Based Apple Localization Algorithm for Robotic Harvesting. *Front. Plant Sci.* **2025**, *16*, 1598414. <https://doi.org/10.3389/fpls.2025.1598414>.
6. Lv, J.; Xu, H.; Han, Y.; Lu, W.; Xu, L.; Rong, H.; Yang, B.; Zou, L.; Ma, Z. A Visual Identification Method for the Apple Growth Forms in the Orchard. *Comput. Electron. Agric.* **2022**, *197*, 106954. <https://doi.org/10.1016/j.compag.2022.106954>.
7. Wang, M.; Li, F. Real-Time Accurate Apple Detection Based on Improved YOLOv8n in Complex Natural Environments. *Plants* **2025**, *14*, 365. <https://doi.org/10.3390/plants14030365>.
8. Xu, Y.; Qiao, X.; Ding, L.; Li, X.; Chen, Z.; Yue, X. Enhanced YOLOv5 with ECA Module for Vision-Based Apple Harvesting Using a 6-DOF Robotic Arm in Occluded Environments. *Agriculture* **2025**, *15*, 1850. <https://doi.org/10.3390/agriculture15171850>.
9. Yang, H.; Liu, Y.; Wang, S.; Qu, H.; Li, N.; Wu, J.; Yan, Y.; Zhang, H.; Wang, J.; Qiu, J. Improved Apple Fruit Target Recognition Method Based on YOLOv7 Model. *Agriculture* **2023**, *13*, 1278. <https://doi.org/10.3390/agriculture13071278>.
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
11. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In *Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
13. Del Prete, R.; Graziano, M.D.; Renga, A. RetinaNet: A Deep Learning Architecture to Achieve a Robust Wake Detector in SAR Images. In Proceedings of the 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), Naples, Italy, 6–9 September 2021; pp. 171–176. <https://doi.org/10.1109/RTSI50628.2021.9597297>.
14. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. <https://doi.org/10.1007/s11042-022-13644-y>.
15. Wang, J.; Su, Y.; Yao, J.; Liu, M.; Du, Y.; Wu, X.; Huang, L.; Zhao, M. Apple Rapid Recognition and Processing Method Based on an Improved Version of YOLOv5. *Ecol. Inform.* **2023**, *77*, 102196. <https://doi.org/10.1016/j.ecoinf.2023.102196>.
16. Yang, L.; Zhang, T.; Zhou, S.; Guo, J. AAB-YOLO: An Improved YOLOv11 Network for Apple Detection in Natural Environments. *Agriculture* **2025**, *15*, 836. <https://doi.org/10.3390/agriculture15080836>.

17. Fu, H.; Guo, Z.; Feng, Q.; Xie, F.; Zuo, Y.; Li, T. MSOAR-YOLOv10: Multi-Scale Occluded Apple Detection for Enhanced Harvest Robotics. *Horticulturae* **2024**, *10*, 1246. <https://doi.org/10.3390/horticulturae10121246>.
18. Wang, Z.; Jin, L.; Wang, S.; Xu, H. Apple Stem/Calyx Real-Time Recognition Using YOLO-v5 Algorithm for Fruit Automatic Loading System. *Postharvest Biol. Technol.* **2022**, *185*, 111808. <https://doi.org/10.1016/j.postharvbio.2021.111808>.
19. Chen, X.; Liu, T.; Fan, X.; Zou, X.; Xu, Y.; Zhou, J. YOLO-EHS: A Lightweight Deep Learning Framework for Xinmei Detection and Multi-Scale Integration in Orchard. *Smart Agric. Technol.* **2026**, *14*, 101946. <https://doi.org/10.1016/j.atech.2026.101946>.
20. Li, A.; Wang, C.; Wang, A.; Sun, J.; Gu, F.; Zhang, T. YOLO-MSRF: A Multimodal Segmentation and Refinement Framework for Tomato Fruit Detection and Segmentation with Count and Size Estimation Under Complex Illumination. *Agriculture* **2026**, *16*, 277. <https://doi.org/10.3390/agriculture16020277>.
21. Sapkota, R.; Meng, Z.; Churuvija, M.; Du, X.; Ma, Z.; Karkee, M. Comprehensive Performance Evaluation of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments. *Agric. Commun.* **2026**, *4*, 100125. <https://doi.org/10.1016/j.agrcom.2026.100125>.
22. Jegham, N.; Koh, C.Y.; Abdelatti, M.; Hendawi, A. YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions. *arXiv* **2024**, arXiv:2411.00201. <https://doi.org/10.48550/arXiv.2411.00201>.
23. Alif, M.A.R.; Hussain, M. YOLOv12: A Breakdown of the Key Architectural Features. *arXiv* **2025**, arXiv:2502.14740. <https://doi.org/10.48550/arXiv.2502.14740>.
24. Zhang, T.; Qi, G.-J.; Xiao, B.; Wang, J. Interleaved Group Convolutions. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4373–4382.
25. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
26. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
27. Ma, X.; Guo, F.-M.; Niu, W.; Lin, X.; Tang, J.; Ma, K.; Ren, B.; Wang, Y. PConv: The Missing but Desirable Sparsity in DNN Weight Pruning for Real-Time Execution on Mobile Devices. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 5117–5124. <https://doi.org/10.1609/aaai.v34i04.5954>.
28. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; PMLR, 2021; pp. 11863–11874.
29. Bastidas, A.A.; Tang, H. Channel Attention Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019.
30. Zhu, X.; Cheng, D.; Zhang, Z.; Lin, S.; Dai, J. An Empirical Study of Spatial Attention Mechanisms in Deep Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6688–6697.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
32. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR, 2019; pp. 6105–6114.
33. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving Object-Centric Image Segmentation Evaluation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15334–15342.
34. Wang, X.; Song, J. ICIoU: Improved Loss Based on Complete Intersection over Union for Bounding Box Regression. *IEEE Access* **2021**, *9*, 105686–105695. <https://doi.org/10.1109/ACCESS.2021.3100414>.
35. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051. <https://doi.org/10.48550/arXiv.2301.10051>.

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