RESEARCH ON A LIGHTWEIGHT TOMATO RIPENESS DETECTION METHOD BASED ON SFH-YOLOv11
/
基于 SFH-YOLOv11 的轻量化西红柿成熟度检测方法研究

Ruijie GONG[1], Lijun CHENG*[1], Yubo ZHANG[1], Zhixiang FENG[1]
[1] College of Software, Shanxi Agricultural University, Taigu, Shanxi / China
correspondent author: Lijun CHENG; Tel: +86-13835441585; E-mail: cljzyb@sxau.edu.cn

DOI: https://doi.org/10.35633/inmateh-77-118

Keywords: *Tomato ripeness, YOLOv11n, attention mechanism, lightweight model*

ABSTRACT

Automated detection of tomato ripeness is crucial for achieving precise harvesting and enhancing agricultural productivity. However, detecting tomatoes in natural scenes poses challenges such as missed detections and false positives due to significant variations in target scale, frequent occlusions, and complex backgrounds. Additionally, existing detection models face limitations when deployed on mobile devices. To address these issues, this paper proposes SFH-YOLOv11, a lightweight detection model based on an improved YOLOv11n. Building upon YOLOv11n, this model achieves lightweight performance while maintaining high accuracy through three key enhancements: introducing an attention mechanism in the backbone network to strengthen feature selection capabilities, designing lightweight convolutional modules to reduce model complexity, and reconstructing the feature pyramid network in the neck to enhance multi-scale feature fusion. Experimental results demonstrate that SFH-YOLOv11 outperforms other algorithms, achieving mAP50 and mAP50-95 scores of 91.8% and 78.2%, respectively—representing improvements of 1.7% and 1.0% over the original model. While enhancing performance, SFH-YOLOv11 reduces the number of parameters, computational complexity, and model size by 37.2%, 15.9%, and 34.5%, respectively, compared to the original model. This research provides effective technical support for lightweight maturity detection tasks in complex agricultural scenarios.

摘要

西红柿成熟度的自动化检测对于实现精准采摘和提升农业生产效率具有重要意义。然而，自然场景下的西红柿图像检测存在目标尺度变化大、遮挡频繁和背景复杂引发的漏检与误检问题，以及现有检测模型在移动端部署的局限性。为此，本文提出一种基于改进YOLOv11n 的轻量化检测模型SFH-YOLOv11。该模型在YOLOv11n的基础上，通过在主干网络中引入注意力机制以强化特征选择能力、设计轻量化卷积模块以降低模型复杂度、在颈部网络中重构特征金字塔网络以增强多尺度特征融合能力这 3 个方面进行改进，使得模型在保持高性能的同时实现轻量化。实验结果表明，SFH-YOLOv11 的性能优于其他算法，mAP50 和mAP50-95 分别达到91.8%和 78.2%，相较于原模型分别提升了 1.7%和 1.0%。在性能提升的同时，SFH-YOLOv11 的参数量、计算量和模型大小相较原模型分别下降了 37.2%、15.9%和 34.5%。本研究为复杂农业场景下的轻量化成熟度检测任务提供了有效的技术支持。

INTRODUCTION

As a globally cultivated fruit and vegetable with exceptionally high consumption rates (*Wang et al., 2025*), tomatoes hold critical economic significance in agricultural production and food supply chains (*Yan et al., 2023*). Their ripeness determines optimal harvest timing, post-harvest quality, transport losses, and market value. Traditional ripeness assessment relies primarily on manual visual inspection (*Badeka et al., 2023*), where fruit maturity is judged through observation of color, size, and shape. This method is not only highly dependent on individual experience but also suffers from significant subjectivity, inconsistent evaluation criteria, and low efficiency, making it difficult to meet the demands of large-scale, precision agriculture. Therefore, against the backdrop of rapid smart agriculture development, achieving automated, precise, and non-destructive monitoring of fruit and vegetable growth conditions has become an essential requirement for enhancing agricultural production efficiency and core competitiveness (*Zhao et al., 2024; Lu et al., 2021; Jia et al., 2022*).

In recent years, with the rapid advancement of artificial intelligence technologies, particularly the widespread application of convolutional neural networks (CNNs) in object detection (*Qian et al., 2023*), significant potential has been demonstrated in agricultural vision tasks. Within smart agriculture's object detection domain, deep convolutional neural network-based detection algorithms have become the core technology for achieving automated, intelligent analysis (*El Sakka et al., 2025*). Based on the algorithmic workflow, CNNs are categorized into two-stage detectors and single-stage detectors. Two-stage detection models, exemplified by the R-CNN series (Fast R-CNN, Faster R-CNN, Mask R-CNN), typically excel in detection accuracy—particularly in precise object localization—due to their secondary refinement of candidate regions and specialized feature extraction. These models are widely applied in agriculture (*Zhang et al., 2025*). Wang et al. proposed an improved Faster R-CNN model for tomato ripeness detection. Experimental results show an average precision of 96.14% in complex scenarios, outperforming common object detection models (*Wang et al., 2022*). Tang et al. enhanced the Mask R-CNN model by incorporating self-calibrating convolutions for precise strawberry ripeness identification. Results indicate improved model performance, achieving an average precision of 0.937 (*Tang et al., 2023*). Zhang et al. proposed an enhanced algorithm named MRS Faster R-CNN for strawberry recognition and ripeness classification. Experimental validation demonstrated that the optimized model achieved average precision improvements of 0.26% and 5.34%, along with precision gains of 0.81% and 6.34%, respectively, compared to the original model in detecting ripe and unripe strawberries (*Zhang et al., 2023*). Zhang et al. designed an improved Faster R-CNN rice spike detection model based on enhanced fast regions. The modified model achieved an average precision of 92.47%, representing a significant improvement over the original Faster R-CNN model (average precision: 40.96%) (*Zhang et al., 2022*). Although R-CNN series models offer high accuracy advantages in agricultural maturity detection scenarios, their high computational load and large model size pose major obstacles for deployment on resource-constrained edge devices (*Liu et al., 2020*). In contrast, single-stage detectors reformulate object detection as an end-to-end regression problem. Such models, exemplified by the YOLO series and SSD, are well-suited for real-time detection and lightweight scenarios. Among these, the YOLO series is widely adopted for agricultural object detection due to its balanced trade-off between speed and accuracy (*Wang et al., 2024*). Zhu et al. constructed the YOLO-LM detection model by integrating CAA, ASFF, and GSConv modules, achieving an mAP50 of 93.18% that outperformed baseline models (*Zhu et al., 2024*). Wang et al. proposed an enhanced YOLO-ALW detection model based on YOLOv8n for chili pepper ripeness detection. Compared to the baseline model, the improved model achieved 3.4%, 5.1%, and 9.0% increases in mean average precision, precision, and recall, respectively (*Wang et al., 2025*). Zhao et al. proposed YOLO-DGS, a lightweight and efficient ripeness detection algorithm. Based on YOLOv10, its improvements significantly enhanced model performance with a 2% increase in mean average precision. Concurrently, inference speed improved by 12.5% and parameters were reduced by 26.3%, making it suitable for lightweight deployment (*Zhao et al., 2025*). Chen et al. integrated the ACmix attention mechanism, FreqFusion-BiFPN architecture, and Inner-Focaler-IoU loss function into the YOLOv11 model to develop AFBF-YOLO for detecting cherry tomato ripeness. Experimental results showed an mAP50 of 85.6%, outperforming multiple mainstream YOLO models (*Chen et al., 2025*).

Despite the optimizations applied to the aforementioned models, they still exhibit significant limitations in scenarios involving severe occlusion and overlap, variable target scales, complex environmental interference, and lightweight requirements. Therefore, this study proposes an improved YOLOv11n model algorithm. By introducing the SimAM attention mechanism, the model's focus on key features is enhanced; it designs the C3k2_FDP module to reduce computational complexity while increasing feature extraction flexibility; and incorporates the HSFPN architecture to lessen computational burden while improving multi-scale feature fusion capabilities. This enhances the model's accuracy and efficiency in detecting tomato ripeness within complex field environments.

MATERIALS AND METHODS

Data Sources

The dataset used in this experiment was sourced from the Tomatoes public dataset on the Roboflow platform. This tomato dataset encompasses key factors such as different growth stages of the fruit, occlusion levels, variations in lighting, and complex environments. It comprises 819 real-world tomato images, each with a resolution of 640×640 pixels and saved in JPG format.

Data Preprocessing

This experiment employed LabelImg software to manually annotate the image dataset in YOLO format, yielding the raw dataset. Following the national standard GH/T 1193—2021, fruit maturity was categorized based on morphological characteristics and color into the following stages: unripe, green-ripe, color-changing, early red-ripe, mid-red-ripe, and late red-ripe. This experiment integrates maturity into three categories: unripe and green ripe fruits with green skin are classified as immature (marked as unripe). Fruits with a coloring area of less than 10% from the color changing period to the mid-red ripening period are classified as semi-ripe (marked as semi-ripe). Fruits with a coloring area exceeding 70% in the late stage of red ripening are classified as ripe (marked as ripe). After annotation completion, generate a text annotation file with the same name as the image file. The file contains category labels and annotation box coordinates. Randomly split the dataset into a training set (573 images) and a validation set (246 images) at a 7:3 ratio. To prevent overfitting and poor generalization caused by insufficient training data, the dataset was augmented using a combination of techniques, including Gaussian blurring, noise addition, brightness adjustment, rotation, cropping, translation, and mirroring. It should be specifically noted that all data augmentation operations are applied only to the training set during the model training phase. The validation set evaluation uses the original images without any augmentation, thereby ensuring the fairness and reliability of the model performance assessment. The augmented training set comprises 2,292 images, as illustrated in Fig. 1.

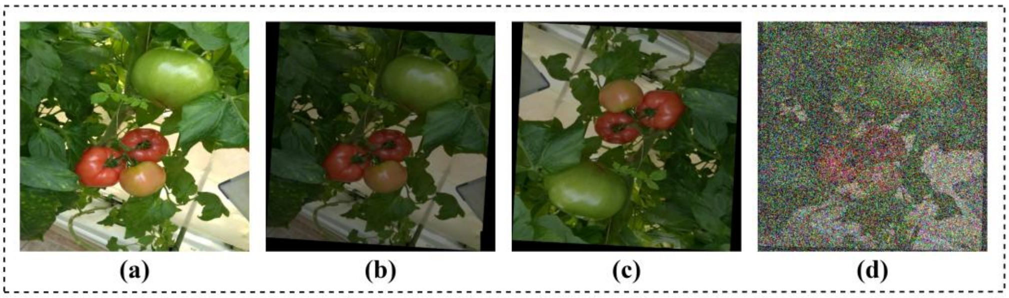


Fig. 1 - Image data augmentation

(a) Original image; (b) Reduce brightness+rotation; (c) Mirror+reduce brightness; (d) Gaussian blur+rotation

YOLOv11n Model Architecture

YOLOv11n is a lightweight variant within the YOLO series, featuring a core structure comprising a backbone network, neck network, and detection head, as illustrated in Fig. 2. The backbone network primarily consists of standard convolutional modules (Conv), C3k2 modules, SPPF modules, and C2PSA modules, tasked with extracting multi-scale feature information from input images. The neck network fuses the multi-scale features extracted by the backbone to generate a feature pyramid rich in semantic and localization information. The detection head employs a decoupled head structure, separately predicting categories and bounding boxes to enhance detection accuracy. YOLOv11n strikes a balance between speed and accuracy, making it suitable for complex multi-class detection tasks and an ideal baseline model for this study.

Improvement of the YOLOv11n Model

The task of accurately and efficiently detecting tomato ripeness in natural field environments presents several challenges: variable target scales and morphologies, overlapping and occlusion among fruits, background interference caused by lighting variations and dense foliage, and the requirement for lightweight models in practical applications. To address these issues, a lightweight SFH-YOLOv11 model is proposed based on YOLOv11n. The optimized SFH-YOLOv11 model architecture is illustrated in Fig. 2.

First, to enhance the model's ability to focus on key maturity features, SimAM is introduced in the deep layers of the backbone network (after the last two C3k2 modules). This module dynamically recalibrates feature maps via an energy function without adding extra parameters, significantly improving the model's feature discrimination and expression capabilities. Second, to simultaneously address high model complexity and weak feature adaptability, a Convolutional Module (FDPConv) that integrates partial convolutions with a frequency-domain dynamic mechanism was designed. This module replaces standard convolutions within the Bottleneck structure and is embedded into the C3k2 module. It effectively reduces computational complexity and parameter count while enhancing feature extraction flexibility. Finally, to optimize the efficiency and accuracy of multi-scale feature fusion, an HSFPN is introduced into the neck network. This architecture achieves more efficient and discriminative multi-scale feature fusion, thereby improving detection performance for objects of varying sizes and occlusion levels. These three improved modules sequentially operate on the critical stages of feature enhancement, feature extraction, and feature fusion, collectively forming a lightweight object detection model that balances high accuracy and computational efficiency.

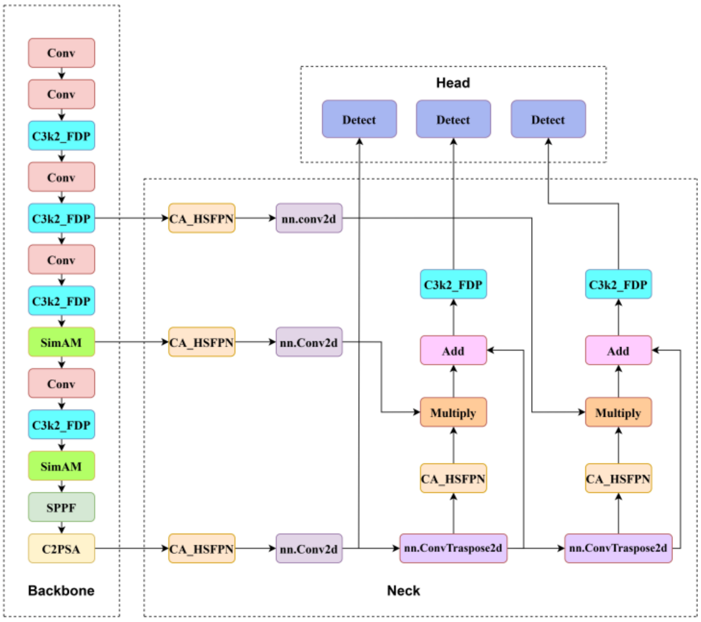


Fig. 2 - SFH-YOLOv11 Network Architecture

Simple Attention Module（SimAM）

To address the issue of insufficient feature selectivity in the model and difficulty in focusing on key regions of fruit maturity from complex backgrounds, and considering the need for lightweight design, this study introduced parameter-free SimAM attention modules at the deep feature positions of the backbone network (after the last two C3k2 modules), as shown in Fig. 3.

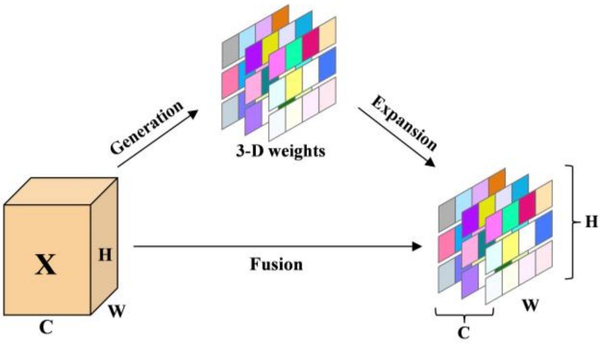


Fig. 3 - SimAM attention module structure

SimAM is a conceptually simple yet highly efficient parameter-free attention module inspired by the spatial inhibition theory in neuroscience (*Webb et al., 2005*). Yang et al. proposed SimAM in 2021 (*Yang et al., 2021*). Its core idea is to define an energy function for each neuron in the feature map. By minimizing this energy function, the distinctiveness of a neuron relative to all other neurons within the same channel can be quantified. The energy function of SimAM is shown in Equation (1).

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

where: t represents the target neuron, x_i represents other neurons, w_t denotes the weight, b_t denotes the bias, and M denotes the total number of neurons.

By minimizing the above energy function, analytical solutions for w_t and b_t can be obtained by taking the derivative and setting it to zero, as shown in Equations (2) and (3).

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

$$b_t = -\frac{1}{2}(t+\mu_t)w_t \tag{3}$$

Substituting the solution into the energy function yields the minimum energy for target neuron t, as shown in Equation (4).

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

Deep features in the network contain rich semantic information associated with the target's category and state. Introducing the SimAM attention module at the aforementioned position enables the model to automatically enhance its response to fruit regions—particularly their surface color and texture features—both spatially and across channels, while simultaneously suppressing interference from irrelevant background elements.

C3k2FDP module structure

The C3k2 module, as the core unit for feature extraction and fusion in YOLOv11, relies on standard convolution for its BottleNeck structure, which has the following problems: firstly, standard convolution performs intensive calculations on all input channels. In pursuit of lightweight goals, there is still parameter and computational redundancy in its internal convolution operations, and there is still room for optimization in computational efficiency. Secondly, fixed-weight convolution kernels cannot dynamically adjust based on the content of the input image, resulting in insufficient flexibility in feature extraction. In order to cope with the interference of target scale changes and complex backgrounds in feature extraction, and to reduce computational costs and achieve lightweight requirements, this study introduces the C3k2FDP module, as shown in Fig. 4(a). Its core is to fuse Frequency Dynamic Convolution (FDConv) and Partial Convolution (PConv) to design the FDPConv module, and based on this, systematically reconstruct the key components Bottleneck, C3k, and C3k2 of the model.

The FDPConv module deeply integrates the strengths of FDConv (*Chen et al., 2025*) and PConv (*Chen et al., 2023*), as shown in Fig. 4(b). This module employs an efficient channel processing strategy: for the input feature map, only a portion of the channels (e.g., 1/4 of the total channels) are fed into the FDConv module for processing, while the remaining channels are directly retained. The FDConv module performs dynamic convolution and modulation of features in the frequency domain, enabling adaptive enhancement or suppression of different frequency components to capture target feature information with greater precision. The remaining channels bypass complex computations and flow directly to the output. Finally, the processed feature channels are concatenated with the retained original channels in the channel dimension. This design achieves more powerful feature extraction capabilities at a low computational cost.

Building upon this foundation, the original Bottleneck architecture was restructured to propose the Bottleneck_FDPConv, as shown in Fig. 4(c). Specifically, both standard convolutional layers (cv1 and cv2) within it were replaced with the aforementioned FDPConv module. This modification enables features to undergo two rounds of adaptive frequency-domain modulation within the Bottleneck architecture with minimal additional parameters and computational overhead, significantly enriching feature hierarchy and expressiveness. Finally, replacing the Bottleneck within the C3k2 module with Bottleneck_FDPConv forms the C3k2_FDP module. The module's initialization process flexibly selects the specific Bottleneck type through conditional checks (c3k), as shown in Fig. 4(d), preserving the architecture's extensibility.

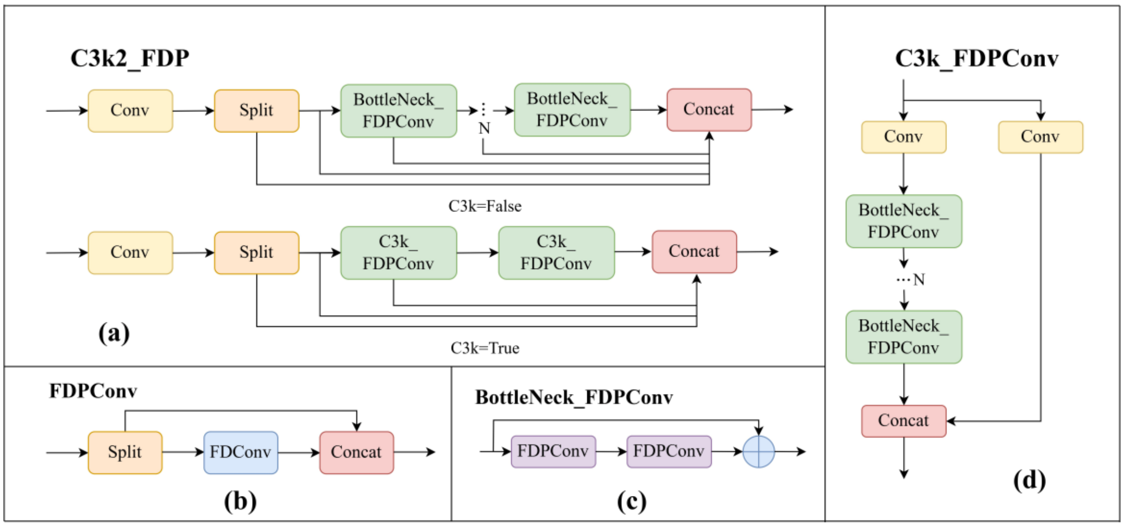


Fig. 4 - Module structure
(a) C3k2FDP module structure; (b) FDPConv module structure; (c) BottleNeck_SDPConv module structure; (d) C3k_FDPConv module structure

High-level Screening-feature Fusion Pyramid Network (HSFPN)

The neck network is responsible for fusing multi-scale features, and its structural design is crucial for detecting fruits of varying sizes. While the PAN-FPN architecture in YOLOv11 enables multi-scale feature fusion, it exhibits limitations when confronting tomato detection tasks characterized by significant target size variations, severe occlusions, and cluttered backgrounds. To address this issue, this paper introduces the HSFPN architecture (*Chen et al., 2024*). This structure takes multi-scale features extracted by the backbone network as input, sequentially processes them through feature selection and feature fusion modules, and ultimately outputs multi-level feature maps rich in semantic information with precise spatial details. These feature maps provide high-quality representations for subsequent detection heads. The structural diagram of HSFPN is shown in Fig. 5(a).

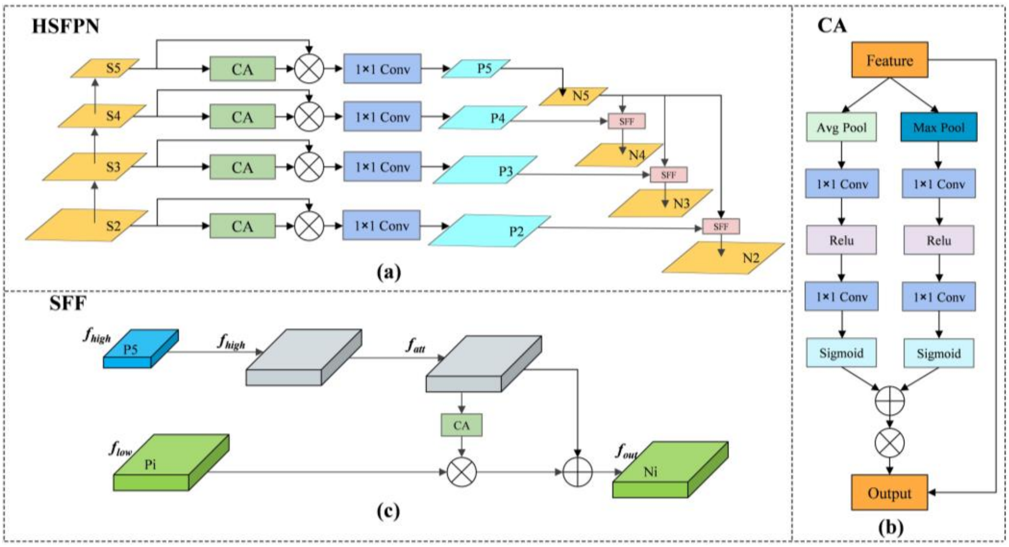


Fig. 5 - High-level Screening-feature Fusion Pyramid Network
(a) The structure of HSFPN; (b) The structure of CA; (c) Structure of SFF

For the feature selection module, the Channel Attention (CA) module serves as the core component for feature filtering in HSFPN. First, the CA module performs global average pooling and global max pooling operations, calculating the mean and maximum responses for each channel, respectively. These statistical values are input into a channel attention module to generate weight vectors matching the number of high-level feature channels, which are then normalized to the [0,1] range via a Sigmoid function. Finally, the original feature map is multiplied channel-wise by the weight vector to achieve feature selection and enhancement. Its structure is illustrated in Fig. 5(b).

For the feature fusion module, HSFPN employs a Selective Feature Fusion (SFF) mechanism to synergistically integrate high-level semantic information with low-level spatial information. Its structure is illustrated in Fig. 5(c). First, high-level features undergo upsampling via transposed convolutions. Subsequently, the upsampled high-level features serve as weight maps for element-wise multiplication with low-level features. Finally, the filtered low-level features undergo residual addition with the high-level features, generating fused features that combine precise localization capabilities with rich semantic information.

Experimental Platform Configuration and Training Strategy

This study was conducted in a standardized experimental environment. The operating system used was Windows 11 Professional Workstation Edition. The hardware configuration included: Intel® Xeon® Platinum 8270 CPU @2.70GHz (2 processors), NVIDIA GeForce RTX 4090 D GPU, and 128GB RAM. The programming language used was Python 3.9.24, integrated with CUDA 11.8 for accelerated model training, and the deep learning framework was built using PyTorch 2.0.1.

All models in this experiment were trained and evaluated using the same experimental configuration environment and training strategy. The input image resolution (image_size) for all models was set to 640×640 pixels, with a batch size (batch_size) of 32 and 300 training epochs (epoch), employing an early stopping mechanism. Stochastic Gradient Descent (SGD) was selected as the optimizer. The initial learning rate (learning_rate) was set to 0.01, the momentum coefficient (momentum) to 0.937, and the weight decay coefficient (weight_decay) to 0.0005.

Evaluation Metrics

To scientifically and objectively quantify and compare the comprehensive performance of the proposed lightweight tomato ripeness detection model, an evaluation framework combining accuracy metrics with efficiency metrics is adopted. Accuracy metrics include: Precision (P), Recall (R), Mean Average Precision (mAP50 and mAP50-95). Efficiency metrics encompass: Parameters (Params), Floating Point Operations (FLOPs), and Model Size (Size). AP is defined as the area under the Precision-Recall curve. For the multi-class detection task in this study, mAP provides a more comprehensive reflection of the model's overall performance (*Meng et al., 2025*). The calculation formulas for P, R, and mAP are shown in (5) to (8).

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \int_0^1 P(R)dr \tag{7}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{8}$$

where: TP denotes the number of positive samples correctly detected, FP denotes the number of samples falsely classified as negative, and FN denotes the number of true positive samples that were not detected.

RESULTS

Ablation Studies

To scientifically and systematically validate the effectiveness and necessity of the three proposed improvement modules, and to deeply analyze their independent contributions and synergistic effects on model performance (accuracy and efficiency), a series of rigorous ablation experiments was designed. All experiments were conducted on the same preprocessed dataset, maintaining identical hyperparameter settings and training strategies to ensure comparability of results. Using the YOLOv11n model as the baseline, a comparative analysis was conducted by progressively introducing each enhancement module. "✓" indicates the model incorporates the corresponding module. Detailed results are summarized in Table 1. Experiment 1 validates the baseline performance of YOLOv11n in detection tasks. Experiments 2, 3, and 4 demonstrate the effects of integrating SimAM, C3k2_FDP, and HSFPN into YOLOv11n, respectively. Experiments 5 and 6 showcase the performance gains achieved through the sequential integration of these enhancement modules.

Table 1

The ablation experiment results of each improved module

Number	Base Line	SimAM	C3k2_ FDP	HSFPN	P	R	mAP50	mAP5 0-95	Params	FLOPs	Size
					[%]	[%]	[%]	[%]	[M]	[G]	[MB]
1	✓				85.4	84.6	90.1	77.2	2.58	6.3	5.5
2	✓	✓			83.9	85.9	91.8	77.8	2.58	6.3	5.5
3	✓		✓		85.3	82.7	90.9	76.9	2.19	5.4	4.7
4	✓			✓	81.2	87.4	90.9	77.7	1.82	5.1	3.9
5	✓	✓	✓		84.8	86.8	91.9	78.2	2.19	5.4	4.7
6	✓	✓	✓	✓	82.0	88.2	91.8	78.2	1.62	5.3	3.6

First, through experiments independently introducing each improvement module, the core functions and contribution directions of each module can be clearly identified. The results of Experiment 2 demonstrate that introducing the SimAM attention mechanism deep within the backbone network (after the last two C3k2 modules) significantly improves the model's accuracy metrics. The mAP50 increased from the baseline 90.1% to 91.8%, representing a 1.7% improvement, while mAP50-95 also rose by 0.6%. This improvement was achieved without altering the model's efficiency metrics (parameter count, computational complexity, model size), demonstrating that the SimAM module effectively enhances classification and localization accuracy by boosting the model's ability to focus on discriminative features, while adhering to lightweight design principles.

Experiment 3 demonstrates that replacing all C3k2 modules in the model with C3k2_FDP yields an mAP50 of 90.9%, representing a 0.8% improvement over the baseline. Concurrently, the number of parameters decreased by 15.1%, computational complexity dropped by 14.2%, and model size shrank by 14.5%. This indicates that the model achieves synergistic optimization of efficiency and accuracy while maintaining high precision. Experiment 4 highlights the module's exceptional capabilities in feature fusion and model compression. It achieves a 0.8% increase in mAP50 and a 0.5% increase in mAP50-95, while significantly reducing parameters by 29.5%, model size by 29.1%, and computational complexity by 19.1%. This confirms that HSFPN can aggregate multi-scale features to enhance detection performance while streamlining the parameters and computational overhead of the neck network.

Second, through modular combination experiments, in-depth analysis of the interactions and synergistic effects among the various improvement modules can be conducted. Further analysis of Experiments 2, 3, and 5 indicates that progressively introducing the SimAM and C3k2_FDP modules onto the baseline model, respectively, improves mAP50 and mAP50-95 by 1.7% and 1%, while reducing the number of parameters by 15.1%, computational complexity by 14.2%, and model size by 14.5%. These results demonstrate positive synergistic effects, reflecting the complementary enhancement between feature selection and dynamic feature extraction. Experiment 6 integrates all three improvements, achieving 1.7% and 1% gains in mAP50 and mAP50-95, respectively, compared to the baseline model. Concurrently, the model achieves extreme lightweighting with a substantial 37.2% reduction in parameters, 15.9% decrease in computational cost, and 34.5% reduction in model size. By comparing the heatmaps of the YOLOv11n, SFH-YOLOv11, and YOLOv12n models across different scenarios. As shown in Fig. 6(a) and (c), it is observed that the detection performance of the YOLOv11n and YOLOv12n models is unstable detection performance in complex environments and small object detection tasks. Particularly in scenarios with cluttered backgrounds, varying object scales, or occlusions, the heatmap response intensity distribution becomes uneven. This leads to incomplete coverage or misalignment of critical targets, consistent with the missed detections and false positives observed in practical detection scenarios. In contrast, the SFH-YOLOv11 model consistently maintains high and stable detection performance under identical conditions, demonstrating superior attention to tomatoes amidst multi-scale target variations and complex scenes compared to the baseline model. As shown in Fig. 6(b), the improved model generates heatmaps that show significantly more concentrated and stable high-response regions under identical conditions, enabling consistent and accurate coverage of tomato targets. This indicates that the proposed enhancement module mitigates the impact of complex backgrounds and multi-scale variations, enabling more effective capture of key features related to ripeness to meet the demands of maturity detection.

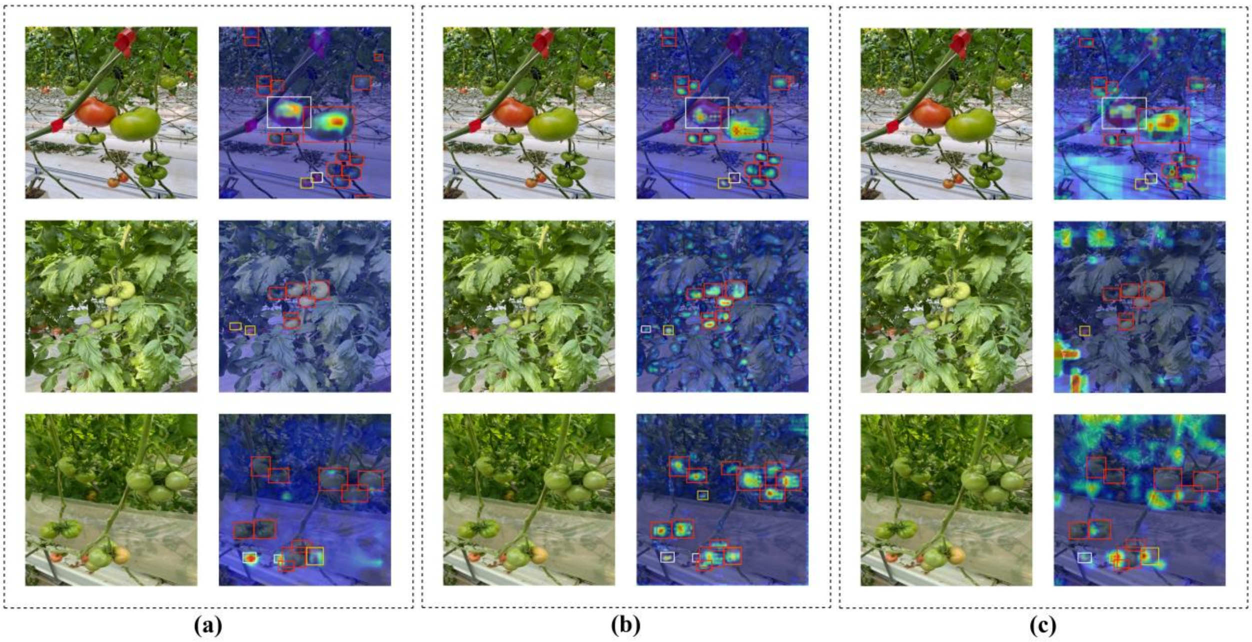


Fig. 6 - Comparison of visualized heatmaps of partial models
(a) YOLOv11n; (b) SFH-YOLOv11; (c) YOLOv12n

In summary, the ablation experiments rigorously demonstrated that each improvement module effectively addresses specific limitations of the original model through systematic comparison, while also quantifying their respective contributions.

Detection Results of SFH-YOLOv11

To visually evaluate the training dynamics and final performance of the improved model, this section conducts an in-depth analysis of key metric changes during training and the model's overall performance on the validation set.

Changes in key metrics during training quantify the model's performance improvement process, as shown in Fig. 7(a). Both precision and recall exhibit a steady upward trend throughout training, stabilizing in the later stages at 82.0% and 88.2%, respectively. During training, the model's key metrics rose synchronously and ultimately converged, remaining stable in the later stages without showing any decline.

These phenomena collectively indicate that the data augmentation strategy and model refinement methods employed in this study are effective. While learning features from the training set, the model did not overfit to the training data and demonstrated strong generalization capabilities. Additionally, the Precision-Recall Curve evaluates the model's detection performance across categories, as illustrated in Fig. 7(b). The three curves correspond to different maturity categories, with the area under each curve representing the average precision for that category. Results indicate that the “immature” category achieved the highest AP value at 95.3%, followed by the ‘mature’ category at 92.7%, while the “semi-mature” category had a relatively lower AP value of 87.4%. The PR curves for all categories cluster predominantly in the upper-right quadrant of the coordinate plot, indicating the model maintains high precision across various recall levels. The model's comprehensive performance metric, mAP50, reached 91.8%. This demonstrates that the improved YOLOv11n model possesses high-precision, robust tomato ripeness detection capabilities overall.

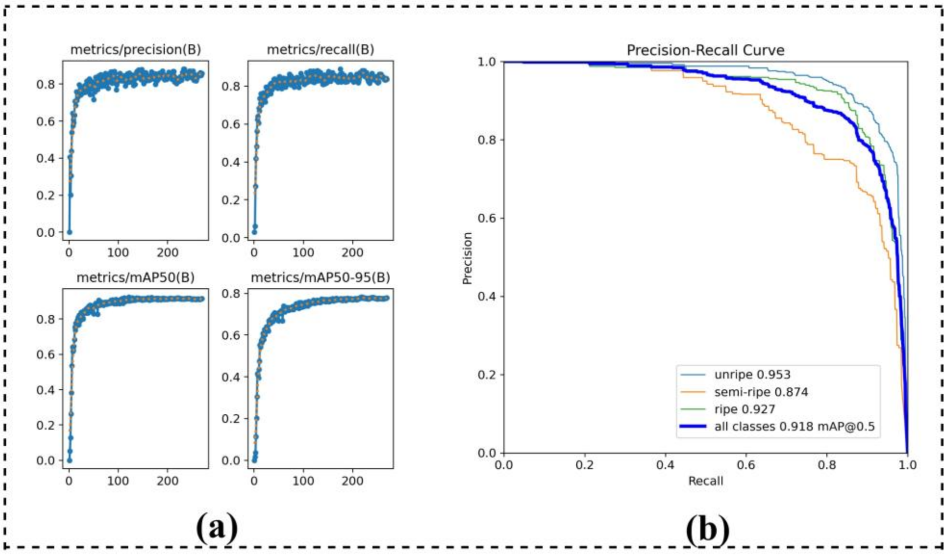


Fig. 7 - Curve changes during training
(a) Changes in key indicators; (b) PR curve

Comparative Experiment

To objectively evaluate the performance of the proposed improved model among mainstream lightweight detection models, a comprehensive comparative experiment was conducted using the same tomato ripeness detection dataset. The model was compared against multiple representative lightweight versions from the YOLO series, including YOLOv5n, YOLOv8n, YOLOv10n, YOLOv11n, and YOLOv12n. The results of the comparative experiment are shown in Table 2.

Table 2

Performance comparison of different models

Number	Model	P	R	mAP50	mAP50-95	Params	FLOPs	Size
		[%]	[%]	[%]	[%]	[M]	[G]	[MB]
1	YOLOv5n	85.5	81.0	89.8	76.6	2.50	7.1	5.3
2	YOLOv8n	81.6	85.2	89.8	76.0	3.00	8.1	6.3
3	YOLOv10n	86.2	78.8	87.5	75.2	2.26	6.5	5.8
4	YOLOv11n	85.4	84.6	90.1	77.2	2.58	6.3	5.5
5	YOLOv12n	80.5	87.0	89.3	76.7	2.55	6.3	5.5
6	SFH-YOLOv11	82.0	88.2	91.8	78.2	1.62	5.3	3.6

First, analyzing both accuracy and efficiency, the improved model achieved the best overall performance among all comparison models in detection accuracy. Its mAP50 reached 91.8%, representing a 1.7% improvement over the original baseline YOLOv11n and significantly outperforming all other comparison models. For the mAP50-95 metric, the improved model achieved 78.2%, surpassing YOLOv11n's 77.2% and other comparison models. This indicates not only high classification accuracy but also precise bounding box localization. Second, the improved model demonstrates even more pronounced advantages in computational efficiency. Its parameter count and model size are only 1.62 million parameters and 3.6 MB, respectively—the lowest among all comparison models. Compared to YOLOv10n, which has the closest parameter count, the improved model reduces parameters by 28.3% and shrinks model size by 37.9%. Furthermore, its computational load is only 5.3 FLOPs, also the lowest among the comparison models.

This demonstrates that the improved model achieves the highest detection accuracy while possessing the smallest complexity and storage overhead, making it suitable for deployment on resource-constrained devices. As shown in Table 2, the YOLOv11n and YOLOv12n models exhibit high similarity in terms of FLOPs and Size. This is primarily because YOLOv12n, as a subsequent iteration of YOLOv11n, focuses its lightweight design optimizations on training strategies, loss functions, or architectural fine-tuning. Consequently, both models share comparable theoretical computational complexity.

To visually demonstrate the performance differences between models in real-world scenarios, Fig. 8 presents a visual comparison of detection results in simple versus complex scenes. In simple scenes (as shown in the upper half of Fig. 8, where fruits are clearly visible and unobstructed), all compared models achieve relatively accurate detection with negligible performance differences. This confirms that under ideal conditions, mainstream lightweight models possess fundamental object detection capabilities. However, in complex scenes (as depicted in the lower half of Fig. 8, featuring dense foliage occlusions, overlapping fruits, and uneven lighting), significant performance disparities emerge among the models. As shown in Fig. 8(b), YOLOv5n exhibited notable false negatives and false positives, particularly failing to recognize some obscured or overlapping shaded fruits while misclassifying leaves as fruits. As shown in Fig. 8(c), (d), and (e), although the YOLOv8n, YOLOv11n, and YOLOv12n models did not exhibit false detections, they missed overlapping fruits. In contrast, the improved model developed in this study demonstrated exceptional stability in complex scenarios, achieving more complete detection of occluded targets while effectively suppressing background interference, as shown in Fig. 8(a).

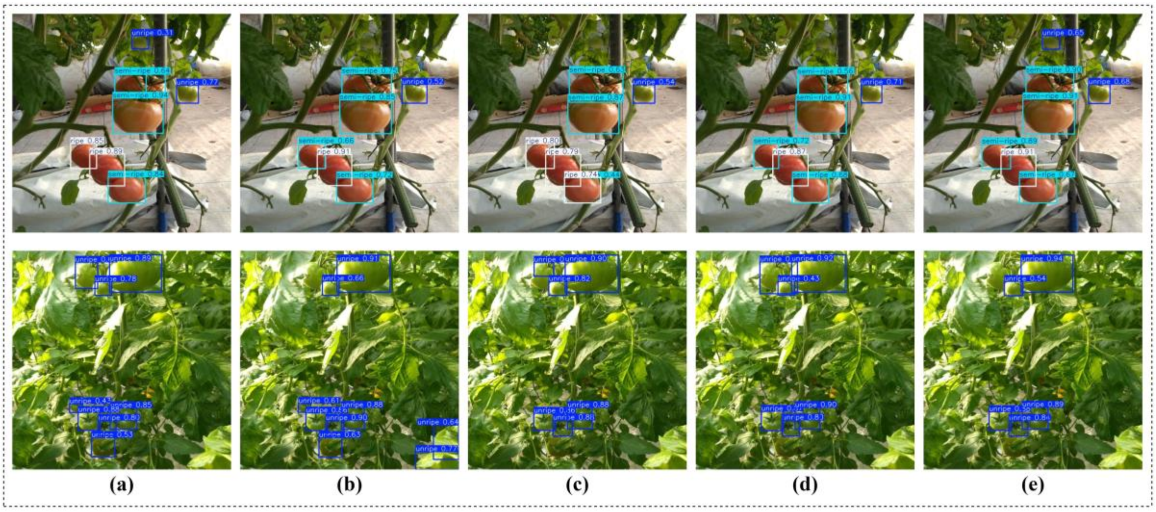


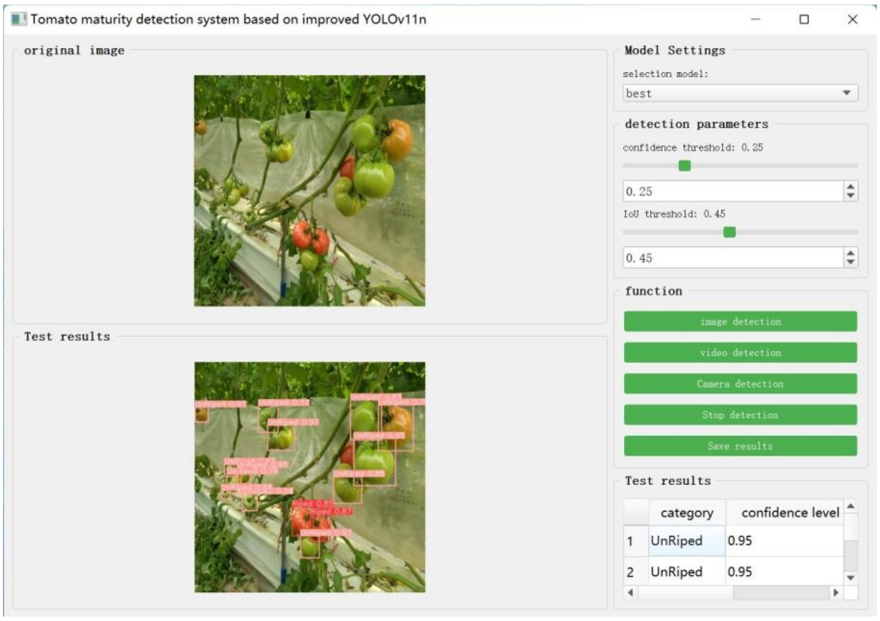
Fig. 8 - Partial model detection results

(a) SFH-YOLOv11n; (b) YOLOv5n; (c) YOLOv8n; (d) YOLOv11n; (e) YOLOv12n

Tomato Ripeness Detection

To visually validate and demonstrate the practical application potential of the improved model developed in this study, a tomato ripeness detection system was designed and implemented. This system provides a user-friendly interface for the model, serving as a reference for subsequent deployment on edge or embedded devices.

As shown in Fig. 9, users select locally stored tomato images or video files via the graphical interface and input the data into the trained model. The system then automatically invokes the algorithm to identify fruits within the input footage. Real-time visualization highlights each fruit with colored bounding boxes corresponding to its maturity category, while also enabling users to save the annotated results locally.



Tomato maturity detection system based on improved YOLOv11n

Model Settings

selection model: best

detection parameters

confidence threshold: 0. 25

0. 25

IoU threshold: 0. 45

0. 45

Test results

	category	confidence level
1	UnRiped	0.95
2	UnRiped	0.95

Fig. 9 - Detection system

CONCLUSIONS

This study proposes a lightweight deployment-oriented tomato ripeness detection model that demonstrates significant performance improvements in complex field environments. Building upon YOLOv11n, the integration of the SimAM attention mechanism module, C3k2_FDP module, and HSFPN feature fusion architecture achieves enhanced detection accuracy in complex scenarios while substantially reducing model complexity.

Ablation experiments demonstrate that each improved module contributes distinct performance gains and efficiency optimizations, validating the feasibility of synergistic optimization across deep feature selection, dynamic feature extraction, and multi-scale feature fusion. Comparative experiments further demonstrate that the proposed improved model outperforms current mainstream lightweight models in both accuracy and efficiency. The improved model demonstrates significant advantages in both detection accuracy (mAP50 reaching 91.8%) and model complexity (parameter count reduced by 37.2%). Additionally, its lightweight benefits—including low memory consumption, reduced computational load, and ease of integration—provide technical support for addressing edge device deployment challenges. This provides a reference for achieving real-time, accurate maturity analysis on computationally constrained field mobile devices or embedded systems.

Based on the above work, future research directions focus on the following aspects: transferring the proposed improvement strategy to other agricultural vision tasks to validate its generalization capability; deploying the model on actual embedded devices for field performance testing and optimization, thereby advancing the development of intelligent agricultural detection.

ACKNOWLEDGEMENT

This research was funded by the Shanxi Provincial Higher Education Institution Science and Technology Innovation Program (2024L058).

REFERENCES

[1] Badeka, E., Karapatzak, E., Karampatea, A., Bouloumpasi, E., Kalathas, I., Lytridis, C., Tziolas, E., Tsakalidou, V. N., & Kaburlasos, V. G. (2023). A deep learning approach for precision viticulture, assessing grape maturity via YOLOv7. *Sensors*, Vol. 23, no. 19, 8126.

[2] Chen, B. J., Bu, J. Y., Xia, J. L., Li, M. X., & Su, W. H. (2025). AFBF-YOLO: An Improved YOLO11n Algorithm for Detecting Bunch and Maturity of Cherry Tomatoes in Greenhouse Environments. *Plants*, Vol. 14, no. 16, 2587.

[3] Chen, L., Gu, L., Li, L., Yan, C., & Fu, Y. (2025). Frequency Dynamic Convolution for Dense Image Prediction. *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 30178-30188.

[4] Chen, J., Kao, S. H., He, H., Zhuo, W., Wen, S., Lee, C. H., & Chan, S. H. G. (2023). Run, don't walk: chasing higher FLOPS for faster neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12021-12031.

[5] Chen, Y., Zhang, C., Chen, B., Huang, Y., Sun, Y., Wang, C., Fu, X., Dai, Y., Qin, F., Peng, Y., & Gao, Y. (2024). Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases. *Computers in biology and medicine*, Vol. 170, 107917.

[6] El Sakka, M., Ivanovici, M., Chaari, L., & Mothe, J. (2025). A review of CNN applications in smart agriculture using multimodal data. *Sensors*, Vol. 25, no. 2, 472.

[7] Jia, W., Liu, J., Lu, Y., Liu, Q., Zhang, T., & Dong, X. (2022). Polar-Net: Green fruit instance segmentation in complex orchard environment. *Frontiers in Plant Science*, Vol. 13, 1054007.

[8] Lu, T., Han, B., Chen, L., Yu, F., & Xue, C. (2021). A generic intelligent tomato classification system for practical applications using DenseNet-201 with transfer learning. *Scientific Reports*, Vol. 11, no. 1, 15824.

[9] Liu, X., Li, Y., Shuang, F., Gao, F., Zhou, X., & Chen, X. (2020). ISSD: Improved SSD for insulator and spacer online detection based on UAV system. *Sensors*, Vol. 20, no. 23, 6961.

[10] Meng, B., & Shi, W. (2025). Small traffic sign recognition method based on improved YOLOv7. *Scientific Reports*, Vol. 15, no. 1, 5482.

[11] Qian, Y., & Wang, B. (2023). A new method for safety helmet detection based on convolutional neural network. *PLoS one*, Vol. 18, no. 10, e0292970.

[12] Tang, C., Chen, D., Wang, X., Ni, X., Liu, Y., Liu, Y., Mao, X., & Wang, S. (2023). A fine recognition method of strawberry ripeness combining Mask R-CNN and region segmentation. *Frontiers in Plant Science*, Vol. 14, 1211830.

[13] Wang, R., & de Maagd, R. A. (2025). Transcriptional control of tomato fruit development and ripening. *Journal of Experimental Botany*, Vol. 76, no. 21, pp. 6311-6326.

[14] Wang, Z., Ling, Y., Wang, X., Meng, D., Nie, L., An, G., & Wang, X. (2022). An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios. *Ecological Informatics*, Vol. 72, 101886.

[15] Wang, Y., Zhang, P., & Tian, S. (2024). Tomato leaf disease detection based on attention mechanism and multi-scale feature fusion. *Frontiers in Plant Science*, Vol. 15, 1382802.

[16] Wang, Y., Ouyang, C., Peng, H., Deng, J., Yang, L., Chen, H., Luo, Y., & Jiang, P. (2025). YOLO-ALW: An Enhanced High-Precision Model for Chili Maturity Detection. *Sensors (Basel, Switzerland)*, Vol. 25, no. 5, 1405.

[17] Webb, B. S., Dhruv, N. T., Solomon, S. G., Tailby, C., & Lennie, P. (2005). Early and late mechanisms of surround suppression in striate cortex of macaque. *Journal of Neuroscience*, Vol. 25, no. 50, 11666-11675.

[18] Yan, C., Yang, T., Wang, B., Yang, H., Wang, J., & Yu, Q. (2023). Genome-wide identification of the WD40 gene family in tomato (Solanum lycopersicum L.). *Genes*, Vol. 14, no. 6, 1273.

[19] Yang, L., Zhang, R. Y., Li, L., & Xie, X. (2021, July). Simam: A simple, parameter-free attention module for convolutional neural networks. *International conference on machine learning*. pp. 11863-11874. PMLR.

[20] Zhao, P., Zhou, W., & Na, L. (2024). High-precision object detection network for automate pear picking. *Scientific Reports*, Vol. 14, no. 1, 14965.

[21] Zhang, Y., & Liu, D. (2025). Rethinking feature representation and attention mechanisms in intelligent recognition of leaf pests and diseases in wheat. *Scientific Reports*, Vol. 15, no. 1, 15624.

[22] Zhang, Y., Zhang, L., Yu, H., Guo, Z., Zhang, R., & Zhou, X. (2023). Research on the strawberry recognition algorithm based on deep learning. *Applied sciences*, Vol. 13, no. 20, 11298.

[23] Zhang, Y., Xiao, D., Liu, Y., & Wu, H. (2022). An algorithm for automatic identification of multiple developmental stages of rice spikes based on improved Faster R-CNN. *The Crop Journal*, Vol. 10, no. 5, pp. 1323-1333.

[24] Zhu, X., Chen, F., Zheng, Y., Chen, C., & Peng, X. (2024). Detection of Camellia oleifera fruit maturity in orchards based on modified lightweight YOLO. *Computers and Electronics in Agriculture*, Vol. 226, 109471.

[25] Zhao, M., Cui, B., Yu, Y., Zhang, X., Xu, J., Shi, F., & Zhao, L. (2025). Intelligent Detection of Tomato Ripening in Natural Environments Using YOLO-DGS. *Sensors*, Vol. 25, no. 9, 2664.