

Article

Detection of Eggplant Fruits and Stems in Complex Greenhouse Environments Using an Improved YOLOv8n

Long Bai ¹, Jianfei Zhu ^{1,*}, Caishan Liu ², Keke Zhang ¹, Sibao Yang ¹ and Yushuo Chen ¹

¹ School of Mechanical and Electrical Engineering, Beijing Information Science and Technology University, Beijing 100192, China; bailong0316jn@126.com (L.B.); 2025020107@bistu.edu.cn (K.Z.); 2025020147@bistu.edu.cn (S.Y.); 2025020160@bistu.edu.cn (Y.C.)

² School of Engineering, Peking University, Beijing 100871, China; liucs@pku.edu.cn

* Correspondence: 2023020080@bistu.edu.cn

Abstract

Accurate perception of eggplant fruits and stems remains challenging for greenhouse harvesting robots because illumination changes, foliage occlusion, fruit overlap, and background branches can degrade target visibility, particularly for small and curved stems. To improve joint fruit-and-stem detection under these conditions, this study develops an enhanced YOLOv8n model using a greenhouse dataset collected across different illumination levels, viewpoints, occlusion degrees, and fruit-overlap situations. The baseline network was modified in three aspects. Selected conventional convolutions in the backbone and neck were replaced by Omni-Dimensional Dynamic Convolution (ODConv) to improve feature adaptation to targets with different scales and shapes. Efficient Multi-Scale Attention (EMA) was placed after the SPPF module to emphasize informative responses from fruit and stem regions while reducing background interference. In addition, C2f_MSBlock was incorporated into the neck to strengthen multi-scale feature representation and fusion. The resulting model achieved 96.4% precision, 97.2% recall, 99.0% mAP@0.5, and 86.0% mAP@0.5:0.95, with 3.74 M parameters, 6.5 GFLOPs, and a model size of 7.9 MB. Relative to the original YOLOv8n, these four detection metrics increased by 2.2, 0.3, 0.5, and 2.9 percentage points, respectively, while GFLOPs decreased by 16.7%. These results indicate that the modified model improves detection robustness in complex greenhouse scenes while maintaining moderate computational requirements, providing a feasible visual perception approach for eggplant fruit recognition and stem localization in robotic harvesting.

Keywords: YOLOv8n; eggplant; object detection; ODConv



Academic Editor: Gniewko Niedbala

Received: 22 July 2026

Revised: 7 September 2026

Accepted: 8 September 2026

Published: 9 September 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Eggplant is an economically important vegetable widely cultivated worldwide and plays a significant role in protected vegetable production because of its high nutritional and commercial value [1,2]. With the rapid development of protected agriculture, greenhouse cultivation has become an important approach for increasing eggplant yield, improving fruit quality, and enabling year-round production [3]. However, greenhouse-grown eggplant plants are characterized by dense foliage and continuous flowering and fruiting, and the identification and harvesting of mature fruits still rely largely on manual labor. Manual harvesting is labor-intensive and inefficient and is strongly affected by workers' experience and labor shortages, thereby restricting the large-scale and intelligent production of greenhouse eggplants. Therefore, the development of eggplant harvesting robots is of

great significance for improving harvesting efficiency, reducing labor costs, and promoting intelligent protected vegetable production [4].

In greenhouse eggplant harvesting robots, visual perception is a key component for target recognition, picking-point localization, and path planning. Fruit detection provides essential information for identifying harvesting targets, whereas peduncle detection is directly related to picking-point determination and end-effector pose adjustment. Compared with fruit-only detection, the joint detection of fruits and peduncles provides more complete target information for robotic harvesting and is therefore important for improving operational accuracy and harvesting success rates [5,6].

Early agricultural target recognition mainly relied on conventional image-processing techniques, such as color thresholding, texture-feature extraction, and morphological processing [7–9]. Moreira et al. [10] proposed an HSV-based model for tomato ripeness classification. Hue histograms were extracted from fruit regions, and a Gaussian mixture model was used to reduce the influence of background pixels. The Gaussian mean of each Hue histogram was then used in a quadratic statistical classifier to categorize tomatoes into four ripening stages. Malik et al. [11] used an improved HSV transformation to remove the background and detect ripe red tomatoes, followed by morphological processing and an improved watershed algorithm to separate connected fruits. The method achieved an overall detection accuracy of 81.6% under uneven illumination and complex backgrounds. Benavides et al. [12] combined image enhancement, edge detection, segmentation, morphological analysis, and geometric descriptors to detect ripe tomatoes and locate their peduncles. The system identified 80.8% of beefsteak tomatoes and 87.5% of cluster tomatoes with visible peduncles as harvestable, with an average processing time of less than 30 ms per image. Singh and Misra [13] proposed a genetic-algorithm-based image-segmentation method for the automatic detection and classification of plant leaf diseases. Indriani et al. [14] combined HSV color features with gray-level co-occurrence matrix texture features and used a K-nearest-neighbor classifier to distinguish five tomato ripeness classes. Although these conventional methods are relatively simple and computationally inexpensive, their performance depends heavily on handcrafted features and is sensitive to illumination variations, occlusion, and changes in target appearance, limiting their robustness in unstructured greenhouse environments.

In recent years, deep learning has been widely applied to agricultural fruit and vegetable recognition, fruit-ripeness assessment, and robotic harvesting because of its strong feature-extraction capability [15–17]. Sun et al. [18] proposed FBoT-Net for small green apple detection by integrating global contextual and local fine-grained features. Yang et al. developed LS-YOLOv8s by incorporating a lightweight Swin Transformer into YOLOv8s for strawberry ripeness detection under complex illumination, clustering, and occlusion [19]. Huang et al. proposed a lightweight RT-DETR-based Transformer model for pear detection, balancing small-target detection performance and model complexity [20]. Mengesha and Mengistie applied transfer learning with CNN architectures and explainable artificial intelligence to tomato leaf disease detection, further demonstrating the applicability of deep learning to agricultural visual recognition tasks [21]. Two-stage object detectors, such as Faster R-CNN, generally achieve high detection accuracy; however, their complex architectures and relatively slow inference speeds limit their applicability to real-time robotic harvesting and deployment on resource-constrained edge devices [22]. By contrast, one-stage detectors, such as SSD and YOLO, integrate object classification and bounding-box regression into an end-to-end framework, thereby offering clear advantages in detection speed and deployment efficiency. In particular, the YOLO series achieves a favorable balance between detection accuracy and inference speed and has become one of the most widely used approaches for agricultural object detection [23–25].

YOLO-based agricultural object detection has made substantial progress in crops such as tomato, tea, citrus, chili pepper, and mango [26–30]. To address missed detections caused by foliage occlusion and illumination variations in complex greenhouse environments, Zheng et al. [31] proposed YOLOv8-Tomato. The model incorporated Large Separable Kernel Attention (LSKA), the DySample dynamic upsampling module, and the Inner-IoU loss function to enhance local feature extraction, improve feature upsampling, and accelerate bounding-box regression. It achieved an mAP@0.5 of 99.4% and a recall of 99.0%. Yang et al. [32] replaced selected standard convolutions in YOLOv8s with depthwise separable convolutions and introduced a Dual-Path Attention Gate (DPAG) and a Feature Enhancement Module (FEM). Their model achieved an mAP of 93.4%, while its size was reduced from 22.5 MB to 16.1 MB.

For fruit detection, Wang et al. [33] developed YOLO-BLBE based on YOLOv5s for blueberry maturity recognition. The model employed GhostNet integrated with coordinate attention as the backbone and introduced BiFPN and the Alpha-EIoU loss function into the neck and prediction components, respectively. It identified mature, semi-mature, and immature blueberries with an mAP of 98.14%. Zhou et al. [34] proposed Kiwi-YOLO by replacing the YOLOv8 backbone with MobileViTv1 and incorporating BiFPN, multi-dimensional collaborative attention (MCA), and the MPDIoU loss function, thereby improving kiwifruit detection under complex illumination, severe occlusion, and dense-fruit conditions. Ji et al. [35] reconstructed the YOLOX-Tiny backbone using ShuffleNetV2 integrated with CBAM and introduced an adaptive spatial feature fusion module and a two-feature-layer structure. The model achieved AP values of 97.29%, 95.53%, and 95.86% for unbagged, bagged, and nighttime apples, respectively, with an overall AP of 96.76%. Xu et al. [36] proposed YOLO-MSNet based on YOLOv11n for pomegranate detection. By incorporating C3k2_UIB, SSAM, DA-BiFPN, AKConv, VFLoss, and ShapeIoU, the model enhanced multi-scale feature extraction and bounding-box localization. Compared with the baseline model, its mAP@0.5 increased by 1.7 percentage points, while the parameter count and model size decreased by 21.5% and 21.8%, respectively, demonstrating favorable lightweight performance and potential for edge deployment.

For the joint detection of fruits and fruiting stems, Gu et al. [37] introduced a Transformer-integrated BRA sparse attention module into the YOLOv8n backbone, adopted a dynamic detection head incorporating deformable convolution, and reconstructed the feature-fusion network using a GSConv-based Slim-Neck. The model achieved detection precisions of 97.63% and 94.50% for mango fruits and fruiting stems, respectively. Wu et al. [38] developed YOLOv5-B by introducing an InvolutionBottleneck module and modifying the loss function to improve the detection of small targets, including banana rachises and flower buds. The model achieved an overall mAP of 93.2% with an average processing time of 9 ms per image. Edge detection was subsequently used to extract the rachis contour, and stereo vision was employed to localize the cutoff point in three-dimensional space. For nighttime litchi detection, Liang et al. [39] first detected fruits using YOLOv3, determined the regions of interest for the fruiting stems from the fruit bounding boxes, and then segmented individual fruiting stems using U-Net. Zhao et al. [40] improved the CSP structure of YOLOv5s using Ghost modules and replaced selected standard convolutions with depthwise separable convolutions, enabling the simultaneous recognition of grape fruits and stems under different illumination conditions. The model achieved an mAP of 98.6% and a model size of 5.8 MB, while increasing the GPU frame rate by 43.5% compared with the original YOLOv5s.

Existing studies on eggplant vision have mainly focused on fruit detection and tracking-based counting, whereas the simultaneous detection of fruits and stems for robotic harvesting has received comparatively limited attention. Zhu et al. [41] developed an improved

YOLOv5s-DeepSORT method for greenhouse eggplant detection and counting by incorporating a lightweight backbone, attention mechanisms, and multi-scale feature fusion. However, uneven illumination, foliage occlusion, fruit overlap, and interference from greenhouse structures may still result in missed or false detections, particularly for small and curved stems that closely resemble branches. Moreover, fruit localization alone cannot fully meet the operational requirements of a robotic harvesting end effector. Therefore, improving the accuracy of joint fruit-and-stem detection while maintaining a low computational cost is essential for the development of eggplant harvesting robots.

To address these challenges, an improved YOLOv8n-based method is proposed for the simultaneous detection of eggplant fruits and stems in complex greenhouse environments. Comparative and ablation experiments were conducted to evaluate the effectiveness of the proposed method. The main contributions are summarized as follows:

- (1) A joint eggplant fruit-and-stem detection dataset was constructed for the visual perception system of greenhouse harvesting robots. The dataset contains images collected under various viewing angles, illumination conditions, foliage occlusion, fruit overlap, and interference from greenhouse structures, providing a reliable data foundation for model training and evaluation.
- (2) ODConv was used to replace selected standard convolutional layers in the backbone and neck to enhance dynamic feature modeling for targets with different scales, shapes, and complex backgrounds. Meanwhile, an EMA module was introduced after the SPPF module to strengthen effective feature responses in fruit and small-stem regions while suppressing interference from foliage, shadows, and complex backgrounds.
- (3) The original C2f modules in the neck were replaced with C2f_MSBlock modules to enhance the interaction between shallow detailed features and deep semantic features through multi-scale feature extraction and fusion. This modification improved the detection of small stems, occluded fruits, and overlapping targets. Model comparison and ablation experiments confirmed the effectiveness of the proposed modules, providing an effective visual detection method for fruit recognition and stem-region localization in greenhouse eggplant harvesting robots.

2. Materials and Methods

2.1. Dataset Construction

Tianjin Kuaiyuan eggplant was selected as the research material. Images were collected from December 2025 to January 2026 at a greenhouse vegetable production base in Jijialishuang Village, Heguan Town, Qingzhou, Weifang, Shandong Province, China, located at 36.844753° N and 118.620348° E. The greenhouse production environment is shown in Figure 1. Images were captured using an iPhone 12 camera (Apple Inc., Cupertino, CA, USA) at 08:00, 10:00, 12:00, 15:00, and 17:00 to obtain eggplant images under varying natural illumination conditions.

To enhance dataset diversity and model robustness, factors including fruit size, shooting distance, viewing angle, and degree of occlusion were considered during image acquisition. Images were captured from top, frontal, side, and upward-facing views under various conditions, including strong illumination, low illumination, local shadows, foliage occlusion, fruit overlap, and interference from greenhouse structures. Considering that eggplant stems are small, curved, and easily occluded, additional images were collected to cover different stem orientations, visibility levels, and spatial relationships between fruits and stems. These images provided a reliable data foundation for training and evaluating the joint eggplant fruit-and-stem detection model.



Figure 1. Greenhouse base environment: (a) interior view; (b) exterior view.

2.2. Image Data Preprocessing

The initial dataset consisted of 1038 greenhouse eggplant images. To increase sample variability, several augmentation operations were implemented in OpenCV, including random rotation and scaling, brightness and contrast modification, hue adjustment, Gaussian-noise injection, and vignetting. These transformations were randomly combined to reproduce changes in viewpoint, illumination, image noise, and local shadow conditions that may occur in greenhouse scenes. Following augmentation, the dataset was expanded to 5188 images. Of these, 4150 images were assigned to the training set, while 519 images were used for validation and another 519 for testing, corresponding approximately to an 8:1:1 split. The augmentation settings are summarized in Table 1, and representative examples are presented in Figure 2.

Table 1. Data augmentation methods and parameter settings.

Data Augmentation Method	Parameter	Setting
Random rotation	Rotation angle, θ	Uniform(-15° , 15°)
Scaling	Scale factor, s	Uniform(0.8, 1.2)
Brightness adjustment	Brightness factor, α	Uniform(0.8, 1.2)
Contrast adjustment	Contrast factor, β	Uniform(0.8, 1.2)
Hue adjustment	Hue offset	Uniform(-10 , 10)
Gaussian noise	Mean, μ ; standard deviation, σ	$\mu = 0$; $\sigma \in \text{Uniform}(5, 15)$
Vignetting simulation	Vignetting intensity factor, λ	Uniform(0.3, 0.6)

Before training, all images were resized from their original resolution of 1280×1706 pixels to 640×640 pixels. Two object categories, “eggplant” and “stem”, were annotated manually in LabelMe by drawing bounding boxes around each visible target. The annotation files generated in JSON format were then converted into TXT files compatible with the YOLO training pipeline.

The distribution of annotated objects was further examined for the training, validation, and test subsets. The training set contained 6263 eggplant instances and 4591 stem instances, while the validation set included 741 eggplants and 565 stems. In the test set, 654 eggplant instances and 481 stem instances were annotated. Overall, the dataset comprised 7658 eggplant objects and 5637 stem objects, yielding 13,295 labeled instances. These statis-

tics, as summarized in Table 2, describe the class composition of the dataset and provide the basis for subsequent model evaluation.

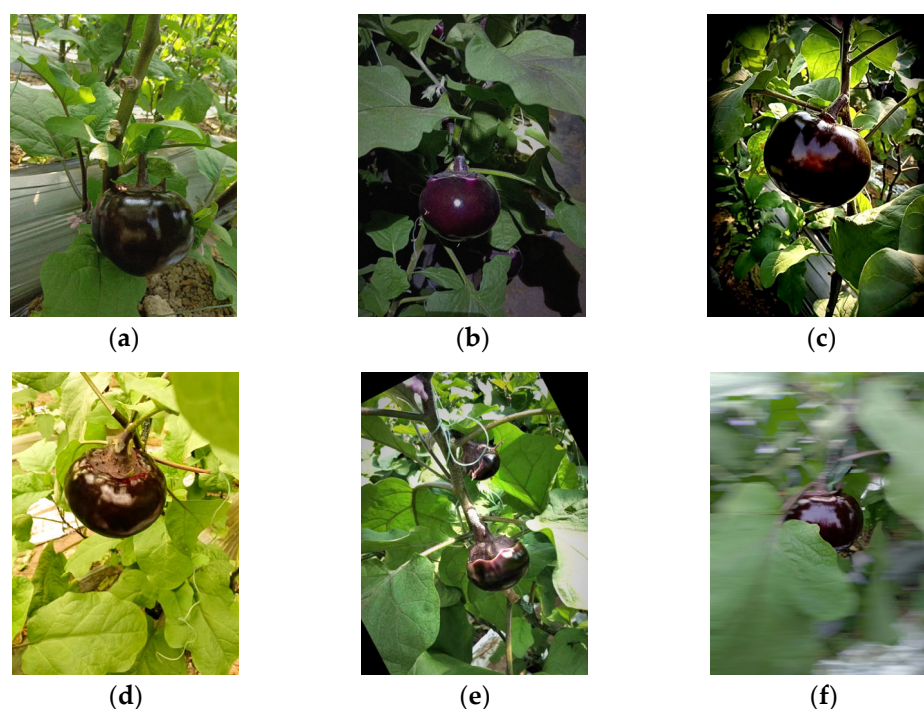


Figure 2. Representative samples from the dataset. (a) normal image; (b) nighttime image; (c) vignette-processed image; (d) brightness-adjusted image; (e) randomly rotated image; (f) motion-blurred image.

Table 2. Dataset split and annotation statistics.

Dataset	Number of Images	Eggplant	Stem	Total Instances
Training	4150	6263	4591	10,854
Validation	519	741	565	1306
Test	519	654	481	1135
Total	5188	7658	5637	13,295

2.3. Model Construction

YOLOv8 follows a single-stage object detection framework. Its architecture is organized into a backbone for hierarchical feature extraction, a neck for aggregating information across different scales, and a prediction head for target classification and localization. In the backbone, feature representations are progressively refined using modules including Conv, C2f, and SPPF. The neck combines features from different resolution levels through upsampling, downsampling, and cross-layer concatenation. At the output stage, classification and bounding-box regression are handled by separate branches within an anchor-free detection head. The targets investigated in this study include relatively large eggplant fruits and small, slender stems. These two target categories differ considerably in scale, appearance, and boundary characteristics, placing high demands on the model's multi-scale feature representation and localization accuracy. Considering detection accuracy, inference efficiency, model complexity, and the deployment requirements of edge devices for harvesting robots, YOLOv8n was selected as the baseline detection model because of its relatively small number of parameters, low computational complexity, and fast inference speed. Targeted improvements were then made to enhance the model's feature extraction capability for

eggplant fruits and stems while maintaining satisfactory detection performance under controlled computational cost.

As shown in Figure 3, the YOLOv8n network architecture was modified in this study. In the backbone, three standard convolutional layers were replaced with ODConv, enabling the network to dynamically adjust the convolution kernel weights according to the input features and enhancing its adaptability to morphological variations in eggplant fruits and stems. Because shallow features contain more edge and texture information, whereas deep features contain richer semantic information, several standard convolutional layers were retained to maintain the stability of network feature extraction. An EMA mechanism was introduced after the SPPF module to adaptively weight channel and spatial features, enhance effective responses in eggplant fruit and stem regions, and suppress interference from foliage, shadows, and greenhouse backgrounds. In the neck, the original C2f modules were replaced with C2f_MSBlock modules, and ODConv was used for downsampling. Through multi-scale feature extraction and cross-layer information fusion, the model's representation capability for small stems, occluded fruits, and overlapping targets was improved. After upsampling, feature concatenation, and multi-scale fusion, the feature maps were fed into three detection heads to classify and localize eggplant fruits and stems at different scales.

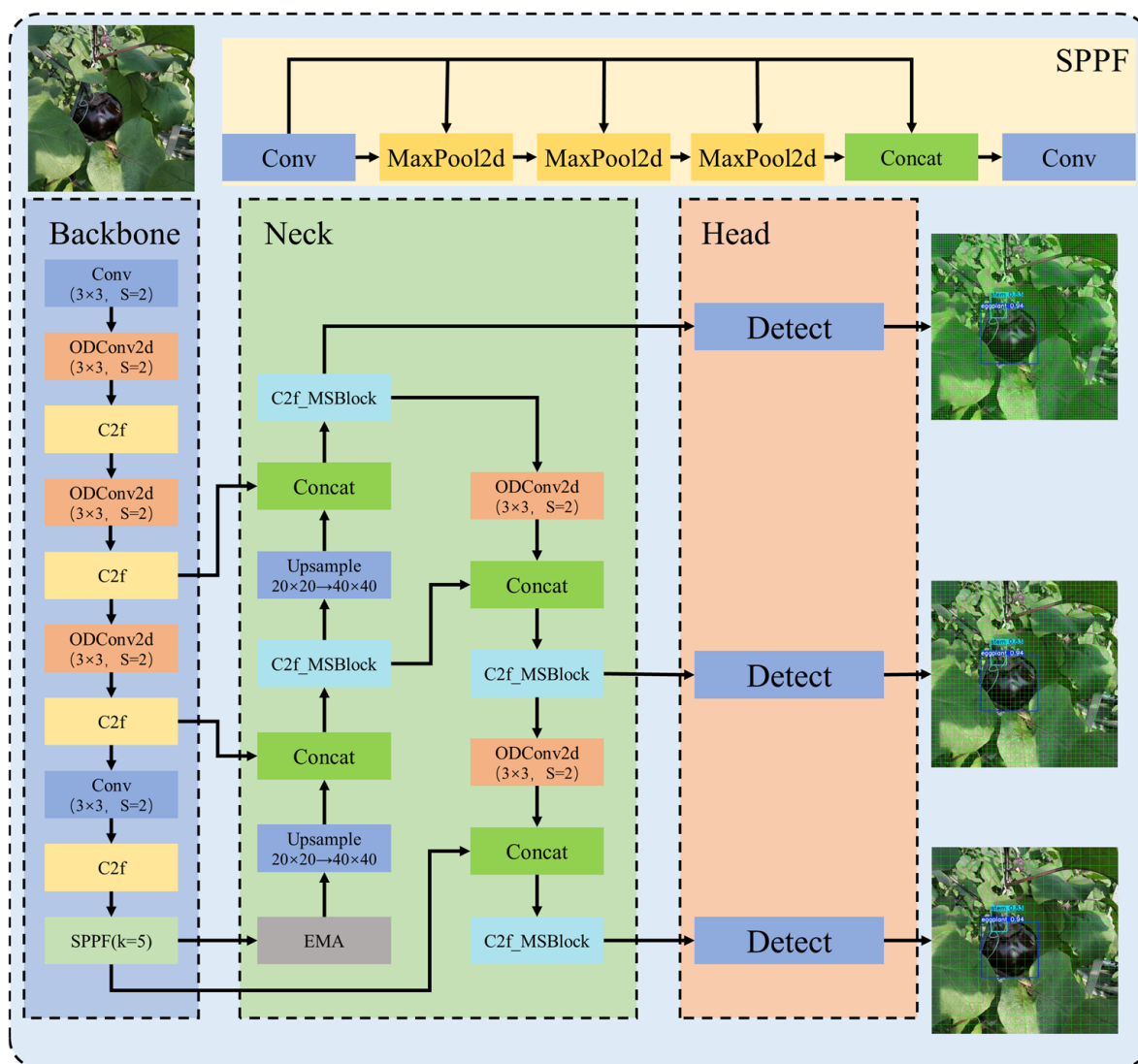


Figure 3. Network architecture of the improved YOLOv8n model.

2.3.1. ODConv Module

In greenhouse environments, eggplant fruits and stems are easily affected by illumination changes, occlusion, and complex backgrounds, while the two targets differ significantly in scale and shape. Fruits are usually large with relatively complete contours, whereas stems are small, slender, and have indistinct boundaries. Conventional convolution uses fixed kernels and therefore cannot effectively handle both target types simultaneously. To address this issue, Omni-Dimensional Dynamic Convolution (ODConv) was introduced into YOLOv8n to enhance the network's adaptive feature extraction capability. The structure of ODConv is shown in Figure 4. This module dynamically generates attention weights from the input features and adaptively modulates and combines candidate convolution kernels. It jointly models the kernels along the spatial, input-channel, output-channel, and kernel-number dimensions, thereby improving the representation ability of the model for eggplant fruits and stems in complex scenes [42].

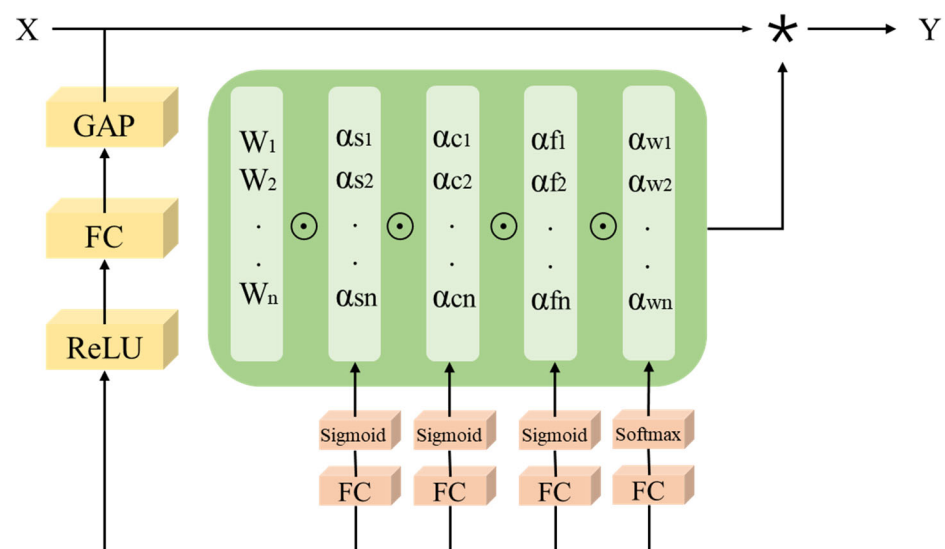


Figure 4. Structure of ODConv.

For an input feature map X , ODConv constructs the effective convolution kernel from n candidate kernels as follows:

$$W_{OD}(X) = \sum_{i=1}^n \alpha_{wi}(X) [\alpha_{fi}(X) \otimes \alpha_{ci}(X) \otimes \alpha_{si}(X)] \odot W_i \quad (1)$$

The output feature Y of ODConv is expressed as:

$$Y = W_{OD}(X) * X \quad (2)$$

where W_i denotes the i -th candidate convolution kernel, and n is the number of candidate kernels; α_{si} , α_{ci} , α_{fi} , and α_{wi} represent spatial, input-channel, output-filter, and kernel attention weights, respectively; $\otimes$ denotes the broadcast-based combination of multidimensional attention weights, $\odot$ denotes element-wise multiplication, and $*$ represents convolution. Because these attention weights are adaptively generated from the input feature X , the effective convolution kernel varies with the input content.

ODConv modulates convolution kernels along four dimensions. Spatial attention α_{si} assigns different weights to spatial positions within each kernel to enhance contour, edge, and local texture features. Input-channel attention α_{ci} adjusts the importance of individual input channels to emphasize informative features and suppress redundant information. Output-filter attention α_{fi} dynamically regulates the responses of different output channels,

while kernel attention α_{wi} adaptively combines candidate kernels according to the input features. Together, these attention mechanisms enable the convolution response to vary dynamically with the input content.

For eggplant fruit and stem detection, ODConv adaptively adjusts the multidimensional weights according to variations in target scale, pose, occlusion, and background, thereby enhancing target representation and discrimination from complex backgrounds.

2.3.2. EMA Mechanism

After multiple convolution and downsampling operations, YOLOv8n extracts deep features containing rich semantic information. However, foliage occlusion, uneven illumination, fruit overlap, and interference from greenhouse structures may cause target and background features to become mixed. In particular, stems are small, slender, and similar in color to foliage, making their effective features susceptible to attenuation during deep feature extraction. Therefore, Efficient Multi-Scale Attention (EMA) was introduced after the SPPF module to enhance the network's focus on key regions of eggplant fruits and stems [43].

Figure 5 illustrates the EMA architecture. At the beginning of the module, the input tensor is partitioned into multiple channel groups, which are subsequently reorganized along the batch dimension so that attention can be estimated independently for each group. Compared with direct channel compression, this grouping strategy preserves more channel information while limiting additional computation.

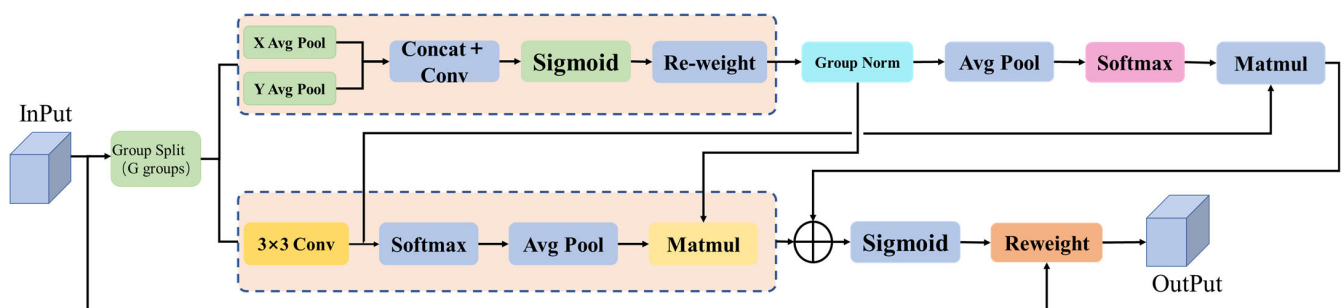


Figure 5. Structure of EMA.

The EMA input is represented by:

$$X \in \mathbb{R}^{B \times C \times H \times W} \quad (3)$$

where B is the batch size, C is the number of channels, H is the height of the feature map, and W is its width.

For grouped feature interaction, the input tensor X is separated into G channel groups as follows:

$$X_g \in \mathbb{R}^{B \times \frac{C}{G} \times H \times W} \quad (4)$$

where X_g denotes the feature tensor of the g -th group, and C/G denotes the number of channels in each group.

Each feature group is processed by two parallel paths. In the first path, directional average pooling is performed separately over the horizontal and vertical axes to capture spatial context across each direction. The operation that averages over the width while retaining the height dimension can be written as:

$$Z_g^h(c, h) = \frac{1}{W} \sum_{w=1}^W X_g(c, h, w) \quad (5)$$

Average pooling along the height dimension while preserving the width information is expressed as:

$$Z_g^w(c, w) = \frac{1}{H} \sum_{h=1}^H X_g(c, h, w) \quad (6)$$

where X_g denotes the input feature of the g -th group; c denotes the channel index in the g -th group; h denotes the height position of the feature map; and w denotes the width position of the feature map.

The second path employs a 3×3 convolution to capture fine-grained local cues, including fruit boundaries, surface patterns, and slender stems. After both paths have produced their responses, the features are averaged over the spatial dimensions and then normalized by Softmax to obtain weighting coefficients. Cross-branch interaction is subsequently performed through matrix operations, enabling the compact descriptor from one path to recalibrate spatial responses in the other path. The two interaction results are added and passed through a Sigmoid function to generate the final spatial attention map. This map is multiplied element-wise by the original grouped feature to obtain the enhanced output. This cross-spatial learning strategy captures pixel-level spatial relationships and promotes the integration of local details with global directional information.

2.3.3. C2f_MSBlock Module

To reduce the number of model parameters and computational cost while enhancing feature extraction for targets at different scales, the original C2f modules in the neck of YOLOv8n were replaced with C2f_MSBlock modules. As shown in Figure 6a, the input features first pass through a convolutional layer for channel adjustment and are then divided into multiple branches through a Split operation. Some features are directly passed to the Concat module through a cross-layer connection, while the remaining features are sequentially processed by multiple MSBlocks for feature extraction. The outputs from different branches and stages are concatenated along the channel dimension and then fused through a convolutional layer [44].

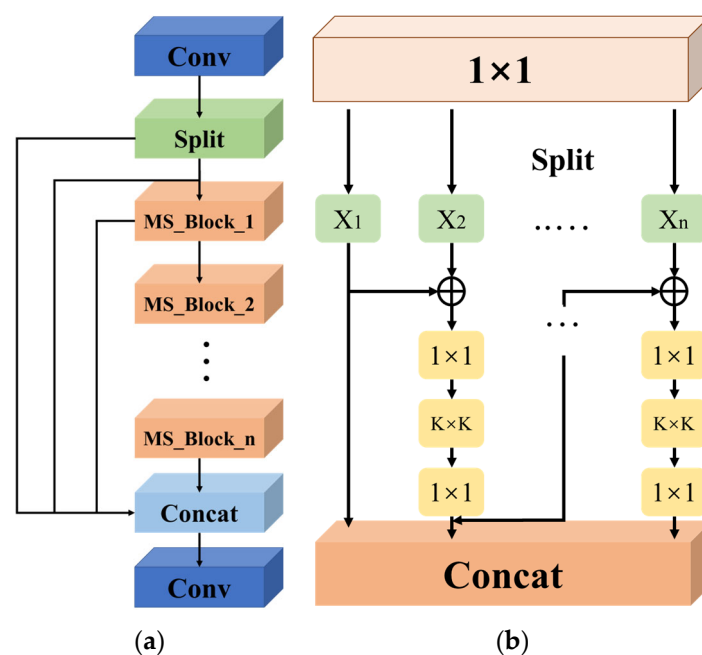


Figure 6. Structures of C2f_MSBlock and MSBlock: (a) C2f_MSBlock. (b) MSBlock.

The structure of MSBlock is shown in Figure 6b. The input features first pass through a 1×1 convolution for channel adjustment and are then divided into multiple sub-feature

branches along the channel dimension. Except for the first branch, each subsequent branch adds its current sub-feature to the output of the preceding branch before passing the fused feature through MSBlock basic units for spatial feature extraction. Each MSBlock basic unit consists of a 1×1 channel-expansion convolution, a $k \times k$ depthwise convolution, and a 1×1 channel-reduction convolution. The outputs of all branches are concatenated along the channel dimension and fused through a 1×1 convolution, enabling multi-scale feature extraction.

3. Results

3.1. Experimental Environment and Parameter Settings

The dataset described in Section 2.2 was used for model training. For YOLOv8n and its improved variants, the input image size was set to 640×640 pixels, the batch size was set to 16, and the number of training epochs was set to 300. Adam was used as the optimizer, with an initial learning rate of 0.001, a momentum parameter of 0.937, and a weight decay coefficient of 0.0005. The remaining parameters followed the default settings of YOLOv8. Table 3 summarizes the experimental setup used in this study.

Table 3. Experimental environment configuration.

Experimental Environment	Configuration
Operating system	Windows 11
GPU	NVIDIA GeForce RTX 4060 Ti
Programming language	Python 3.9
CUDA version	11.3
PyTorch version	1.11.0
Memory	32 GB
Storage	1 TB solid-state drive

3.2. Evaluation Metrics

Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95 were used to evaluate the detection performance of the models. The number of parameters, model size, and floating-point operations (FLOPs) were also used to evaluate model complexity. All models were evaluated using a consistent evaluation protocol. Precision, Recall, and mAP were obtained using the standard evaluation procedure, with mAP calculated from the Precision–Recall curves rather than at a single fixed confidence threshold. Fixed confidence thresholds were used only for confusion matrix analysis and qualitative visualization and were not used to derive the tabulated Precision, Recall, or mAP metrics.

3.3. Comparative Experiments on Backbone Networks

To identify a suitable backbone for joint eggplant fruit and stem detection, YOLOv8n was used as the baseline, and ShuffleNetV1, ShuffleNetV2, MobileNetV3, and ODConv-based variants were compared under the same dataset split, input size, and training conditions for 300 epochs.

According to the results summarized in Table 4, ShuffleNetV1 achieved the lowest computational cost, while ShuffleNetV2 reduced the parameter count but increased GFLOPs. MobileNetV3 achieved the highest P, mAP@0.5, and mAP@0.5:0.95, but with higher model complexity. Introducing ODConv increased P, R, and mAP@0.5:0.95 from 94.2%, 96.9%, and 83.1% to 95.3%, 97.1%, and 83.9%, respectively, while reducing GFLOPs

from 7.8 to 7.1. Therefore, ODConv was retained for further evaluation in the subsequent ablation experiments.

Table 4. Quantitative comparison of the evaluated backbone networks.

Backbone Network	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%	Parameters	GFLOPs
YOLOv8n	94.2	96.9	98.5	83.1	3,055,782	7.8
ShuffleNetV1	95	96.4	98	83.8	3,523,942	6.2
ShuffleNetV2	95.7	98.3	98	85	2,838,162	9
MobileNetV3	97.9	97.1	99.3	90.2	5,703,452	9.1
YOLOv8n-ODConv	95.3	97.1	97.8	83.9	3,353,901	7.1

3.4. Ablation Study

To evaluate the effects of ODConv, EMA, and C2f_MSBlock, as well as their combinations, on the detection performance of YOLOv8n, ablation experiments were conducted using YOLOv8n as the baseline model. The proposed modules were introduced individually or in combination. All experiments used the same dataset and were trained for 300 epochs under identical training settings to ensure a fair comparison.

As shown in Table 5, different modules and their combinations exhibited different characteristics in terms of detection performance and model complexity. With ODConv alone, Precision increased from 94.2% to 94.7%, mAP@0.5:0.95 increased from 83.1% to 83.8%, and GFLOPs decreased from 7.8 to 6.9, while the parameter count increased from 3.06 M to 3.92 M. Combined with its dynamic convolution mechanism, ODConv mainly contributes to enhancing dynamic feature modeling for targets with different scales and shapes. With C2f_MSBlock alone, the parameter count decreased from 3.06 M to 2.85 M and GFLOPs decreased from 7.8 to 7.5, indicating that C2f_MSBlock contributes more directly to reducing structural redundancy and model complexity.

Table 5. Ablation results of different module combinations.

ODConv	EMA	C2f_MSBlock	P/%	R/%	mAP@50/%	mAP@0.5:0.95/%	Parameters	GFLOPs
×	×	×	94.2	96.9	98.5	83.1	3,055,782	7.8
✓	×	×	94.7	96.9	97.7	83.8	3,916,583	6.9
×	✓	×	94.4	96.8	98.1	82	3,056,454	7.8
×	×	✓	94.7	94.6	98	81.1	2,849,862	7.5
✓	✓	×	96.5	97.9	99.2	86	3,917,255	6.9
✓	×	✓	96.9	91.5	95.1	80.5	3,147,981	6.8
×	✓	✓	97.1	97	98.8	86.4	2,850,534	7.5
✓	✓	✓	96.4	97.2	99	86	3,744,103	6.5

The ODConv–EMA configuration yielded the highest recall of 97.9% and the highest mAP@0.5 of 99.2% among the combined variants. The combination of EMA and C2f_MSBlock achieved the highest Precision and mAP@0.5:0.95, reaching 97.1% and 86.4%, respectively. Although the combination of ODConv and C2f_MSBlock further reduced GFLOPs to 6.8, Recall and mAP@0.5:0.95 decreased to 91.5% and 80.5%, respectively.

When ODConv, EMA, and C2f_MSBlock were introduced simultaneously, the complete model achieved a Precision of 96.4%, Recall of 97.2%, mAP@0.5 of 99.0%, and mAP@0.5:0.95 of 86.0%, while GFLOPs decreased to 6.5, the lowest among all configurations. Compared with the original YOLOv8n, mAP@0.5:0.95 increased by 2.9 percentage points while computational cost was reduced. It should be noted that the complete model did not achieve

the best result for every individual metric, indicating trade-offs among detection accuracy, recall capability, and model complexity. Overall, ODConv mainly enhances dynamic feature extraction, EMA strengthens target feature responses and suppresses background interference through the attention mechanism, with more pronounced effects when combined with other modules, while C2f_MSBlock contributes more directly to reducing model complexity. The combination of the three modules enables the complete model to achieve a favorable balance between detection performance and computational cost.

3.5. Comparative Experiments with Different Models

To evaluate the overall performance of the proposed model for joint eggplant fruit and stem detection in greenhouse environments, YOLOv5s, YOLOv7, YOLOv8n, YOLOv10n, YOLOv12n, and YOLO26s were selected for comparison. To balance experimental consistency with the optimization characteristics of different models, all models were trained for 300 epochs using the recommended training configurations of their corresponding implementations. The quantitative comparison is summarized in Table 6.

Table 6. Comparative results of the evaluated detection models.

Model	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%	Parameters	GFLOPs	Model Size/MB
YOLOv5s	95.9	98.4	98.8	85.6	7,066,239	16.5	14.5
YOLOv7	97	99.4	99.4	89	37,200,095	105.1	74.8
YOLOv10n	96.1	98	99	84	2,707,820	8.4	5.8
YOLOv12n	91	95.8	95.6	76.6	2,568,438	6.5	5.5
YOLO26s	94	90.2	88.2	73.3	9,949,412	22.5	20.3
YOLOv8n	94.2	96.9	98.5	83.1	3,055,782	7.8	6.3
Ours	96.4	97.2	99	86	3,744,103	6.5	7.9

As shown in Table 6, the compared YOLO models exhibited clear differences in detection performance and model complexity. YOLOv7 achieved the highest overall detection performance, with Precision, Recall, mAP@0.5, and mAP@0.5:0.95 of 97.0%, 99.4%, 99.4%, and 89.0%, respectively, but required 37.20 M parameters and 105.1 GFLOPs, resulting in a relatively high computational cost. YOLOv5s also achieved good detection performance, with a Recall of 98.4% and an mAP@0.5:0.95 of 85.6%, while requiring 7.07 M parameters and 16.5 GFLOPs. YOLOv10n contained only 2.71 M parameters and achieved an mAP@0.5 of 99.0%, but its mAP@0.5:0.95 was 84.0%. YOLOv12n required 2.57 M parameters and 6.5 GFLOPs, but achieved an mAP@0.5:0.95 of only 76.6%. YOLO26s required 9.95 M parameters and 22.5 GFLOPs, with mAP@0.5 and mAP@0.5:0.95 values of 88.2% and 73.3%, respectively. The original YOLOv8n achieved an mAP@0.5 of 98.5% and an mAP@0.5:0.95 of 83.1% with relatively low model complexity, providing a suitable baseline for subsequent improvement.

After improving YOLOv8n, the proposed model achieved a Precision of 96.4%, Recall of 97.2%, mAP@0.5 of 99.0%, and mAP@0.5:0.95 of 86.0%, with 3.74 M parameters, 6.5 GFLOPs, and a model size of 7.9 MB. Compared with YOLOv5s, the proposed model reduced the parameter count, computational cost, and model size by approximately 47.0%, 60.6%, and 45.5%, respectively, while improving mAP@0.5 and mAP@0.5:0.95 by 0.2 and 0.4 percentage points. Compared with the original YOLOv8n, Precision, Recall, mAP@0.5, and mAP@0.5:0.95 increased by 2.2, 0.3, 0.5, and 2.9 percentage points, respectively, while GFLOPs decreased from 7.8 to 6.5, corresponding to a reduction of approximately 16.7%. Although the parameter count increased by approximately 22.5%, the overall computational cost was reduced. Compared with YOLOv7, which achieved the highest detection

performance, the proposed model required only approximately 10.1% of its parameters and 6.2% of its GFLOPs. Therefore, the proposed model achieves a favorable balance between detection performance and computational efficiency and shows potential for deployment in resource-constrained greenhouse vision systems.

To further analyze the training and convergence characteristics of the compared models, Figure 7 presents the changes in Precision, Recall, mAP@0.5, and mAP@0.5:0.95 over 300 training epochs. As shown in Figure 7, the evaluation metrics increased rapidly during the early stage of training and gradually stabilized thereafter, while mAP@0.5:0.95 showed a relatively slower convergence process. Overall, all models exhibited stable convergence trends.

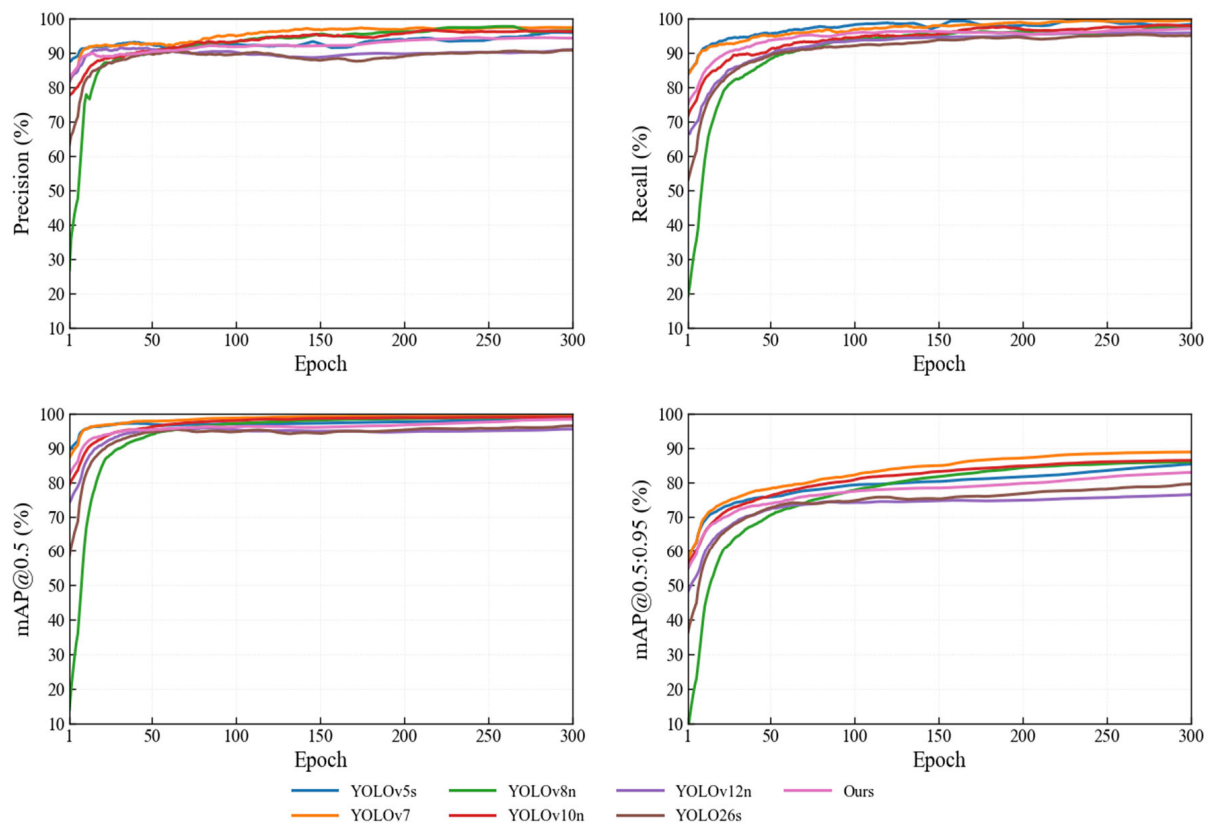


Figure 7. Evolution of detection metrics during training for the compared models.

To further evaluate the training stability before and after model improvement, Figure 8 presents the training and validation loss curves of YOLOv8n and the proposed model over 300 epochs. The training and validation losses dropped sharply in the initial epochs and then approached stable values as training progressed, with no evident abnormal fluctuations.

3.6. Confusion Matrix Analysis

To further analyze class confusion, false detections, and missed detections of eggplant fruits and stems, the complete confusion matrix is shown in Figure 9. The confusion matrix was generated using a confidence threshold of 0.25. Among the 654 ground-truth eggplant instances, 646 were correctly detected, corresponding to a correct-detection rate of 98.78%, while 8 were classified as Background, resulting in a missed-detection rate of 1.22%. Among the 481 ground-truth stem instances, 467 were correctly detected, corresponding to a correct-detection rate of 97.09%; 1 was misclassified as eggplant, corresponding to a class-misclassification rate of 0.21%, and 13 were classified as Background, resulting in a missed-detection rate of 2.70%. In addition, 40 and 35 false-positive detections were

produced for eggplant and stem on background regions, respectively. Overall, direct confusion between eggplant and stem was limited, and the main errors arose from false detections and missed detections involving the Background class.

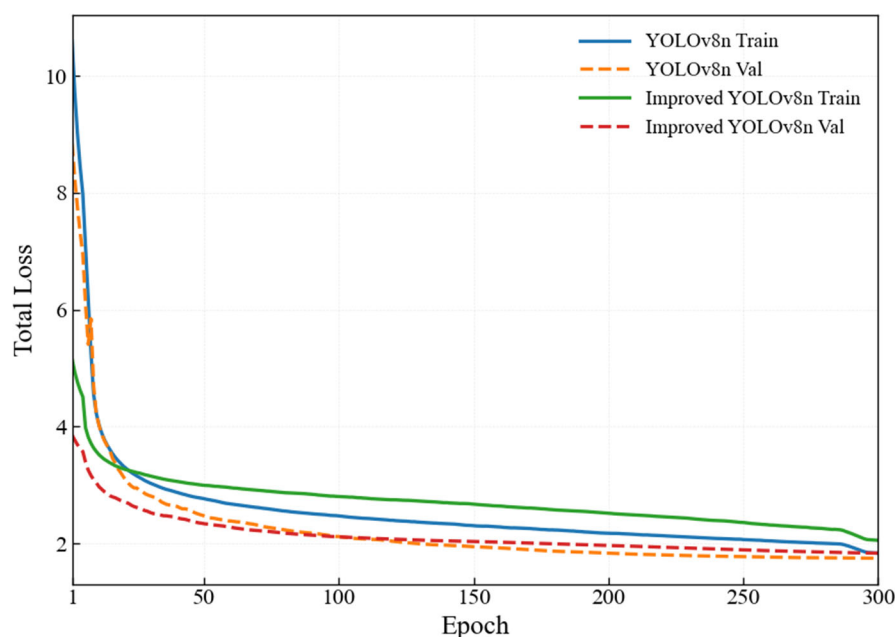


Figure 8. Loss convergence curves of YOLOv8n and the proposed model.

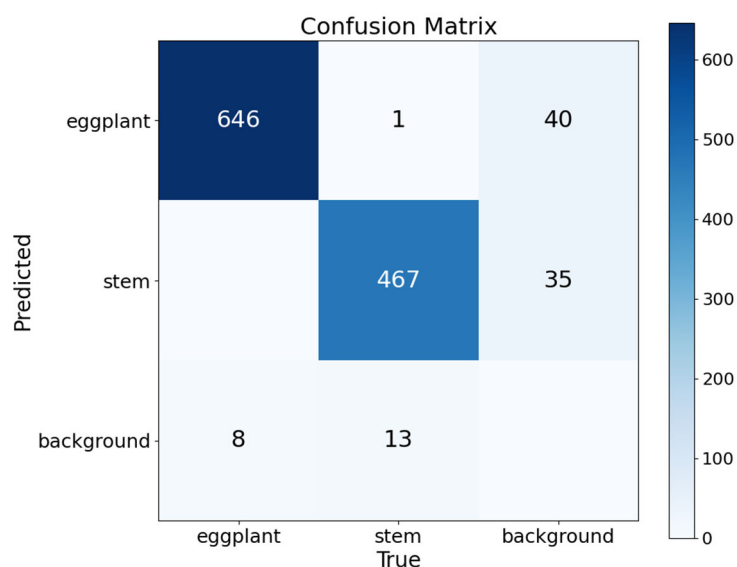


Figure 9. Confusion matrix results for the proposed model.

3.7. Qualitative Analysis

To further evaluate the detection performance of different models under complex greenhouse conditions, YOLOv5s, YOLOv7, YOLOv8n, YOLOv10n, YOLOv12n, YOLO26s, and the proposed model were qualitatively compared under five representative scenarios: nighttime low-light conditions, sufficient illumination, foliage occlusion, branch occlusion, and fruit overlap. These scenarios were used to evaluate the adaptability of different models to illumination variations, target occlusion, small targets, and complex background interference. To ensure a consistent visual comparison, the same confidence threshold of 0.25 was applied to all models for prediction filtering and visualization. This threshold was

used only for qualitative visualization and was not involved in the calculation of quantitative metrics such as Precision, Recall, and mAP.

As shown in Figure 10, under nighttime low-light conditions, local shadows and insufficient illumination reduce the feature contrast between fruits, slender stems, and the background. YOLOv10n and YOLOv12n exhibit different degrees of missed detections, whereas the proposed model retains more complete detections of the main fruits and stems, indicating better feature responses to low-contrast targets.

Under sufficient illumination, most models successfully detect the main targets, although YOLOv10n and YOLOv12n still miss some objects. In contrast, the proposed model maintains relatively complete detections of fruits and stems at different scales and produces relatively high confidence scores for several key targets, demonstrating stable target representation under favorable illumination.

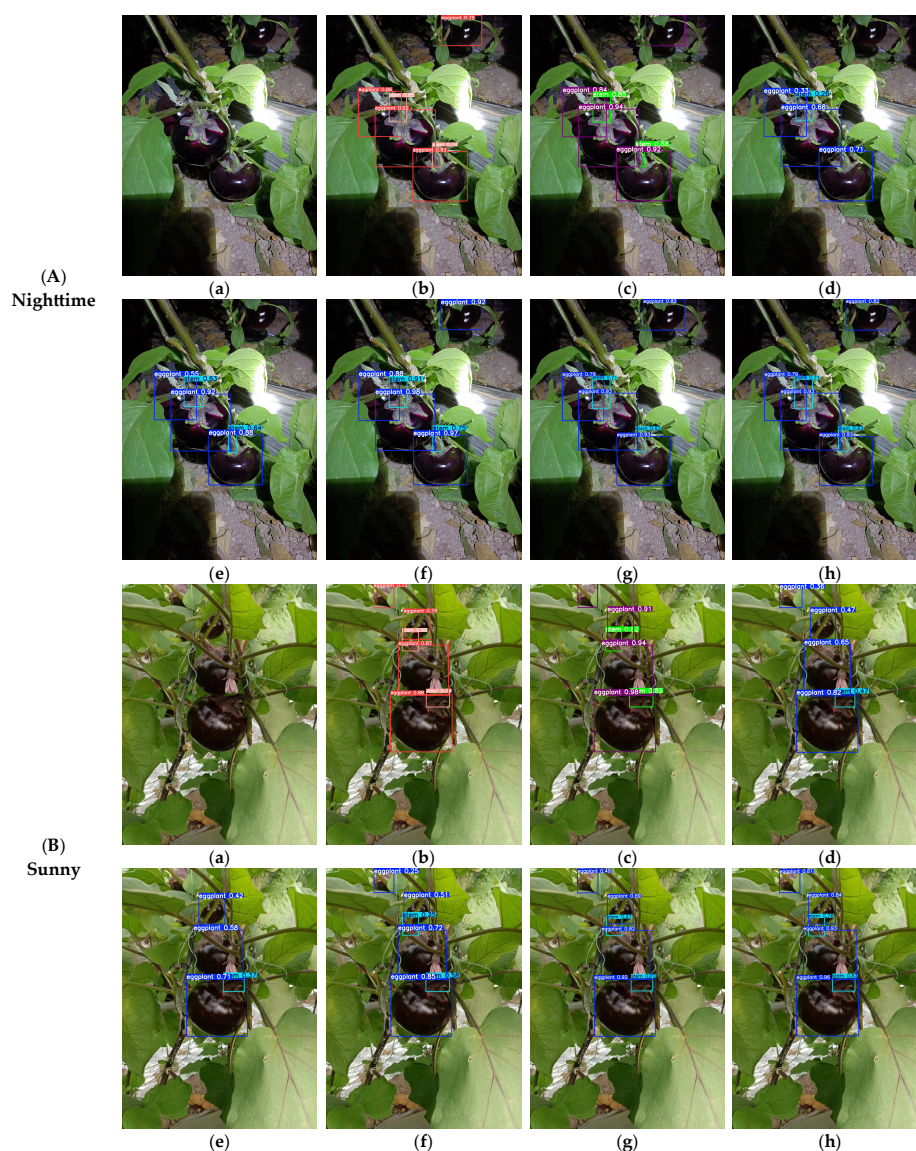


Figure 10. Cont.



Figure 10. Detection results of different models (A–E) under representative greenhouse scenes: (a) input image; (b) YOLOv5s; (c) YOLOv7; (d) YOLOv10n; (e) YOLOv12n; (f) YOLO26s; (g) YOLOv8n; (h) proposed model.

Under foliage occlusion, partial fruit contours and stem structures are obscured by leaves, increasing the difficulty of target recognition. Some comparison models miss slender stems or partially visible targets, whereas the proposed model shows no obvious missed detections in this example and preserves more complete detections of partially occluded objects, indicating better robustness to occlusion.

Branch occlusion and viewpoint variation alter the apparent shape, position, and visible area of targets, making accurate feature representation and localization more difficult. Most comparison models exhibit different degrees of missed detection, accompanied by relatively low confidence scores for some targets. The proposed model, however, detects the main fruits and stems more completely while maintaining relatively stable confidence outputs, showing better adaptability to viewpoint variation and weak-feature targets.

In fruit-overlap scenes, mutual occlusion between adjacent fruits, together with interference from leaves and branches, increases the difficulty of target separation and slender-stem detection. Some models exhibit missed detections, duplicate detections, or insufficient separation of adjacent fruits. In contrast, the proposed model more effectively distinguishes overlapping fruits while maintaining valid stem detections, showing better target separation and background discrimination in this representative scene.

Overall, under the same confidence threshold, the proposed model exhibits relatively stable detection performance under low illumination, occlusion, viewpoint variation, and fruit-overlap conditions, with fewer missed and duplicate detections. It also maintains effective responses to slender stems and partially occluded targets. These qualitative observations are generally consistent with the preceding quantitative results, further demonstrating the adaptability and detection stability of the proposed method in complex greenhouse environments.

4. Discussion

To address the challenges of large-scale differences between eggplant fruits and stems, weak features of slender stems, foliage occlusion, and complex background interference in greenhouse environments, YOLOv8n was improved in this study. Experimental results showed that the improved model achieved a Precision of 96.4%, Recall of 97.2%, mAP@0.5 of 99.0%, and mAP@0.5:0.95 of 86.0%, representing increases of 2.2, 0.3, 0.5, and 2.9 percentage points, respectively, compared with the original YOLOv8n. These results indicate that the improved model reduced false detections while enhancing target recall and localization performance under stricter IoU criteria. Meanwhile, the computational cost decreased from 7.8 to 6.5 GFLOPs, although the parameter count increased from 3.06 M to 3.74 M. This suggests that the proposed method does not simply pursue model-size reduction, but instead improves detection performance and reduces theoretical computational cost at the expense of a moderate increase in parameters.

The ablation experiments further revealed trade-offs among different module combinations across the evaluation metrics. The combination of ODConv and EMA achieved the highest Recall and mAP@0.5, reaching 97.9% and 99.2%, respectively, whereas the combination of EMA and C2f_MSBlock achieved the highest Precision and mAP@0.5:0.95, reaching 97.1% and 86.4%, respectively. The complete model achieved corresponding values of 96.4%, 97.2%, 99.0%, and 86.0%. Therefore, the complete model did not achieve the best result for every individual metric; rather, it maintained high detection performance while reducing GFLOPs to 6.5, reflecting an overall trade-off among detection accuracy, recall capability, and computational complexity. A similar trend was observed in the comparison with other detection models. Although YOLOv7 achieved a higher mAP@0.5:0.95 of 89.0%, together with a Precision of 97.0% and Recall of 99.4%, its parameter count and computational cost reached 37.20 M and 105.1 GFLOPs, respectively. In contrast, the proposed model required

only 3.74 M parameters and 6.5 GFLOPs. Thus, the main advantage of the proposed model lies in its favorable balance between detection performance and computational efficiency rather than achieving the absolute best value for every evaluation metric.

In addition, this study has several limitations. The dataset was mainly collected from a single greenhouse, with a limited number of eggplant varieties and during a specific period. Therefore, the generalization capability of the model across different varieties, seasons, greenhouse structures, and imaging devices still requires further validation. Under severe occlusion, low-light conditions, and dense fruit overlap, slender stems may still be missed or localized unstably. Moreover, two-dimensional bounding boxes cannot directly provide the precise harvesting points and spatial poses required for robotic cutting. In addition, the current results are based on a fixed dataset split and a single training run, without validation using multiple random seeds or statistical intervals; therefore, the stability of some relatively small performance improvements remains to be further assessed.

At present, model complexity is mainly evaluated in terms of parameter count, GFLOPs, and model size, while inference speed, power consumption, and actual harvesting performance on embedded platforms have not yet been tested. Therefore, the current results primarily demonstrate the potential of the proposed model for resource-constrained greenhouse vision systems. Future work will expand the dataset to more diverse scenarios, conduct repeated training with multiple random seeds, and incorporate keypoint detection, instance segmentation, and depth information to further evaluate and improve the model's generalization ability, statistical robustness, and practical deployment performance.

5. Conclusions

To address the difficulties in detecting eggplant fruits and stems caused by illumination variation, foliage occlusion, fruit overlap, and complex background interference in greenhouse environments, this study proposed an improved YOLOv8n-based detection model. Eggplant images were collected in a real greenhouse, and a joint fruit–stem detection dataset covering different illumination conditions, viewing angles, occlusion levels, and fruit-overlap scenarios was constructed.

ODConv, EMA, and C2f_MSBlock were introduced into YOLOv8n. ODConv enhanced dynamic feature extraction for targets with different scales, poses, and occlusion levels. EMA strengthened effective responses in fruit and stem regions while suppressing interference from foliage, shadows, and greenhouse facilities. C2f_MSBlock improved the representation of small stems and overlapping fruits through multi-scale feature fusion while reducing computational cost.

On the test set, the final model yielded a precision of 96.4% and a recall of 97.2%, while its mAP reached 99.0% at IoU = 0.5 and 86.0% over the IoU range of 0.5–0.95. The network contains 3.74 M parameters, requires 6.5 GFLOPs, and has a storage size of 7.9 MB. Compared with the original YOLOv8n, these four detection metrics increased by 2.2, 0.3, 0.5, and 2.9 percentage points, respectively, while computational cost decreased by approximately 16.7%, although the parameter count increased by approximately 22.5%. Compared with other detection models such as YOLOv7, the proposed model did not achieve the highest value for every detection metric, but required substantially lower computational complexity. Overall, the proposed model achieved a favorable balance between detection performance and computational cost.

Overall, the proposed model showed relatively stable detection performance under nighttime low-light conditions, foliage occlusion, branch occlusion, and fruit-overlap scenarios, demonstrating its potential for fruit recognition and stem localization in greenhouse harvesting robots. Practical deployment performance will be further evaluated through embedded-platform and robotic harvesting experiments.

Author Contributions: Conceptualization, L.B., J.Z., C.L., K.Z., S.Y. and Y.C.; methodology, J.Z. and L.B.; software, J.Z. and L.B.; validation, J.Z. and L.B.; formal analysis, J.Z., L.B. and K.Z.; investigation, J.Z., L.B. and K.Z.; resources, L.B.; data curation, J.Z. and L.B.; writing—original draft preparation, J.Z. and L.B.; writing—review and editing, L.B.; visualization, K.Z. and S.Y.; supervision, L.B.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Projects & Technology Demonstration Projects) (No. 2022CXGC020701—Research and Development of Motion Planning and Intelligent Drive Technology for Agricultural Manipulators).

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

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Oladosu, Y.; Rafii, M.Y.; Arolu, F.; Chukwu, S.C.; Salisu, M.A.; Olaniyan, B.A.; Fagbohun, I.K.; Muftaudeen, T.K. Genetic Diversity and Utilization of Cultivated Eggplant Germplasm in Varietal Improvement. *Plants* **2021**, *10*, 1714. [[CrossRef](#)] [[PubMed](#)]
2. Kadoglidou, K.I.; Krommydas, K.; Ralli, P.; Mellidou, I.; Kalyvas, A.; Irakli, M. Assessing Physicochemical Parameters, Bioactive Profile and Antioxidant Status of Different Fruit Parts of Greek Eggplant Germplasm. *Horticulturae* **2022**, *8*, 1113. [[CrossRef](#)]
3. Achour, Y.; Ouammi, A.; Zejli, D. Technological progresses in modern sustainable greenhouses cultivation as the path towards precision agriculture. *Renew. Sustain. Energy Rev.* **2021**, *147*, 111251. [[CrossRef](#)]
4. Yi, Z.; Liu, Z.; Li, X.; Kong, F.; Li, H.; Wen, K. Dual end-effectors for grasping and cutting: An adaptive and damage-free approach to robotic pitaya picking. *Comput. Electron. Agric.* **2026**, *251*, 112024. [[CrossRef](#)]
5. Montoya-Cavero, L.-E.; Torres, R.D.d.L.; Gómez-Espinosa, A.; Cabello, J.A.E. Vision systems for harvesting robots: Produce detection and localization. *Comput. Electron. Agric.* **2022**, *192*, 106562. [[CrossRef](#)]
6. Rong, J.; Dai, G.; Wang, P. A peduncle detection method of tomato for autonomous harvesting. *Complex Intell. Syst.* **2021**, *8*, 2955–2969. [[CrossRef](#)]
7. Camargo, A.; Smith, J. An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosyst. Eng.* **2009**, *102*, 9–21. [[CrossRef](#)]
8. Arivazhagan, S.; Shebiah, R.N.; Ananthi, S.; Varthini, S.V. Detection of Unhealthy Region of Plant Leaves and Classification of Plant Leaf Diseases using Texture Based Clustering Features. *CIGR J.* **2013**, *15*, 211–217.
9. Patil, J.K.; Kumar, R. Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features. *Eng. Agric. Environ. Food* **2017**, *10*, 69–78. [[CrossRef](#)]
10. Moreira, G.; Magalhães, S.A.; Pinho, T.; dos Santos, F.N.; Cunha, M. Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato. *Agronomy* **2022**, *12*, 356. [[CrossRef](#)]
11. Malik, M.H.; Zhang, T.; Li, H.; Zhang, M.; Shabbir, S.; Saeed, A. Mature Tomato Fruit Detection Algorithm Based on improved HSV and Watershed Algorithm. *IFAC-PapersOnLine* **2018**, *51*, 431–436. [[CrossRef](#)]
12. Benavides, M.; Cantón-Garbín, M.; Sánchez-Molina, J.A.; Rodríguez, F. Automatic Tomato and Peduncle Location System Based on Computer Vision for Use in Robotized Harvesting. *Appl. Sci.* **2020**, *10*, 5887. [[CrossRef](#)]
13. Singh, V.; Misra, A. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Inf. Process. Agric.* **2017**, *4*, 41–49. [[CrossRef](#)]
14. Indriani, O.R.; Kusuma, E.J.; Sari, C.A.; Rachmawanto, E.H.; Setiadi, D.R.I.M. Tomatoes classification using K-NN based on GLCM and HSV color space. In Proceedings of the 2017 International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 2–4 November 2017; pp. 1–6. [[CrossRef](#)]
15. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [[CrossRef](#)]
16. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning—Method overview and review of use for fruit detection and yield estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234. [[CrossRef](#)]
17. Masood, M.; Nawaz, M.; Nazir, T.; Javed, A.; Alkanhel, R.; Elmannai, H.; Dhahbi, S.; Bourouis, S. MaizeNet: A deep learning approach for effective recognition of maize plant leaf diseases. *IEEE Access* **2023**, *11*, 52862–52876. [[CrossRef](#)]

18. Sun, M.; Zhao, R.; Yin, X.; Xu, L.; Ruan, C.; Jia, W. FBoT-Net: Focal bottleneck transformer network for small green apple detection. *Comput. Electron. Agric.* **2023**, *205*, 107609. [[CrossRef](#)]
19. Yang, S.; Wang, W.; Gao, S.; Deng, Z. Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. *Comput. Electron. Agric.* **2023**, *215*, 108360. [[CrossRef](#)]
20. Huang, Z.; Zhang, X.; Wang, H.; Wei, H.; Zhang, Y.; Zhou, G. Pear Fruit Detection Model in Natural Environment Based on Lightweight Transformer Architecture. *Agriculture* **2024**, *15*, 24. [[CrossRef](#)]
21. Mengesha, A.T.; Mengistie, M.A. Applying transfer learning in CNN model architectures for detecting tomato leaf disease with explainable artificial intelligence. *Smart Agric. Technol.* **2025**, *11*, 101034. [[CrossRef](#)]
22. Sharma, A.; Kumar, V.; Longchamps, L. Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agric. Technol.* **2024**, *9*, 100648. [[CrossRef](#)]
23. Spancerski, J.S.; Filho, P.L.d.P.; Kugler, M.; Schneider, F.K. A comprehensive benchmark of multi-generation YOLO architectures for forest species identification from macroscopic wood images. *Sci. Rep.* **2026**, *16*, 25802. [[CrossRef](#)] [[PubMed](#)]
24. Wen, F.; Wu, H.; Zhang, X.; Shuai, Y.; Huang, J.; Li, X.; Huang, J. Accurate recognition and segmentation of northern corn leaf blight in drone RGB Images: A CycleGAN-augmented YOLOv5-Mobile-Seg lightweight network approach. *Comput. Electron. Agric.* **2025**, *236*, 110433. [[CrossRef](#)]
25. Sun, J.; Zhou, J.; He, Y.; Jia, H.; Rottok, L.T. Detection of rice panicle density for unmanned harvesters via RP-YOLO. *Comput. Electron. Agric.* **2024**, *226*, 109371. [[CrossRef](#)]
26. Liu, G.; Nouaze, J.C.; Mbouembe, P.L.T.; Kim, J.H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. *Sensors* **2020**, *20*, 2145. [[CrossRef](#)] [[PubMed](#)]
27. Xie, S.; Sun, H. Tea-YOLOv8s: A Tea Bud Detection Model Based on Deep Learning and Computer Vision. *Sensors* **2023**, *23*, 6576. [[CrossRef](#)] [[PubMed](#)]
28. Lin, Y.; Huang, Z.; Liang, Y.; Liu, Y.; Jiang, W. AG-YOLO: A Rapid Citrus Fruit Detection Algorithm with Global Context Fusion. *Agriculture* **2024**, *14*, 114. [[CrossRef](#)]
29. Paul, A.; Machavaram, R.; Ambuj, Kumar, D.; Nagar, H. Smart solutions for capsicum Harvesting: Unleashing the power of YOLO for Detection, Segmentation, growth stage Classification, Counting, and real-time mobile identification. *Comput. Electron. Agric.* **2024**, *219*, 108832. [[CrossRef](#)]
30. Zhong, Z.; Yun, L.; Cheng, F.; Chen, Z.; Zhang, C. Light-YOLO: A lightweight and efficient YOLO-based deep learning model for mango detection. *Agriculture* **2024**, *14*, 140. [[CrossRef](#)]
31. Zheng, S.; Jia, X.; He, M.; Zheng, Z.; Lin, T.; Weng, W. Tomato Recognition Method Based on the YOLOv8-Tomato Model in Complex Greenhouse Environments. *Agronomy* **2024**, *14*, 1764. [[CrossRef](#)]
32. Yang, G.; Wang, J.; Nie, Z.; Yang, H.; Yu, S. A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy* **2023**, *13*, 1824. [[CrossRef](#)]
33. Wang, C.; Han, Q.; Li, J.; Li, C.; Zou, X. YOLO-BLBE: A Novel Model for Identifying Blueberry Fruits with Different Maturities Using the I-MSRCR Method. *Agronomy* **2024**, *14*, 658. [[CrossRef](#)]
34. Zhou, J.; Sun, F.; Wu, H.; Lv, Q.; Feng, F.; Zhao, B.; Li, X. Kiwi-YOLO: A Kiwifruit Object Detection Algorithm for Complex Orchard Environments. *Agronomy* **2025**, *15*, 2424. [[CrossRef](#)]
35. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [[CrossRef](#)]
36. Xu, L.; Li, B.; Fu, X.; Lu, Z.; Li, Z.; Jiang, B.; Jia, S. YOLO-MSNet: Real-Time Detection Algorithm for Pomegranate Fruit Improved by YOLOv11n. *Agriculture* **2025**, *15*, 1028. [[CrossRef](#)]
37. Gu, Z.; He, D.; Huang, J.; Chen, J.; Wu, X.; Huang, B.; Dong, T.; Yang, Q.; Li, H. Simultaneous detection of fruits and fruiting stems in mango using improved YOLOv8 model deployed by edge device. *Comput. Electron. Agric.* **2024**, *227*, 109512. [[CrossRef](#)]
38. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [[CrossRef](#)]
39. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A visual detection method for nighttime litchi fruits and fruiting stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [[CrossRef](#)]
40. Zhao, J.; Yao, X.; Wang, Y.; Yi, Z.; Xie, Y.; Zhou, X. Lightweight-Improved YOLOv5s Model for Grape Fruit and Stem Recognition. *Agriculture* **2024**, *14*, 774. [[CrossRef](#)]
41. Zhu, J.; Bai, L.; Liu, C.; Nian, C.; Zhang, K.; Yang, S. Research on Greenhouse Eggplant Fruit Detection and Tracking-Based Counting Using an Improved YOLOv5s-DeepSORT. *Agriculture* **2026**, *16*, 253. [[CrossRef](#)]
42. Li, C.; Zhou, A.; Yao, A. Omni-dimensional dynamic convolution. In Proceedings of the Tenth International Conference on Learning Representations (ICLR 2022), Virtual Event, 25–29 April 2022.

43. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 4–10 June 2023; pp. 1–5. [[CrossRef](#)]
44. Chen, Y.; Yuan, X.; Wang, J.; Wu, R.; Li, X.; Hou, Q.; Cheng, M.-M. YOLO-MS: Rethinking multi-scale representation learning for real-time object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 4240–4252. [[CrossRef](#)] [[PubMed](#)]

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.