



Article

EDI-YOLO: An Instance Segmentation Network for Tomato Main Stems and Lateral Branches in Greenhouse Environments

Peng Ji ^{1,2}, Nengwei Yang ^{1,3}, Sen Lin ^{3,*} and Ya Xiong ³

¹ School of Machinery and Equipment Engineering, Hebei University of Engineering, Handan 056038, China

² Xiongan Institute of Green Water Network and Life Health, Xiongan New Area 071799, China

³ Research Center of Intelligent Equipment, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China

* Correspondence: linseng@nercita.org.cn

Abstract

Agricultural robots operating in greenhouse environments face substantial challenges in detecting tomato stems, including fluctuating lighting, cluttered backgrounds, and the stems' inherently slender morphology. This study introduces EfficientV1-C2fDWR-IRMB-YOLO (EDI-YOLO), an enhanced model built on YOLOv8n-seg. First, the original backbone is replaced with EfficientNetV1, yielding a 2.3% increase in mAP50 and a 2.6 G reduction in FLOPs. Second, we design a C2f-DWR module that integrates multi-branch dilations with residual connections, enlarging the receptive field and strengthening long-range dependencies; this improves slender-object segmentation by 1.4%. Third, an Inverted Residual Mobile Block (iRMB) is inserted into the neck to apply spatial attention and dual residual paths, boosting key-feature extraction by 1.5% with only +0.7GFLOPs. On a custom tomato-stem dataset, EDI-YOLO achieves 79.3% mAP50 and 33.9% mAP50-95, outperforming the baseline YOLOv8n-seg (75.1%, 31.4%) by 4.2% and 2.6%, and YOLOv5s-seg (66.7%), YOLOv7tiny-seg (75.4%), and YOLOv12s-seg (75.4%) by 12.6%, 3.9%, and 3.9% in mAP50, respectively. Significant improvement is achieved in lateral branch segmentation (60.4% → 65.2%). Running at 86.2 FPS with only 10.4GFLOPs and 8.0 M parameters, EDI-YOLO demonstrates an optimal trade-off between accuracy and efficiency.

Keywords: YOLOv8; instance segmentation; main stem and lateral branches; tomato; greenhouse environment



Academic Editor: Araceli Peña

Received: 17 August 2025

Revised: 13 October 2025

Accepted: 15 October 2025

Published: 18 October 2025

Citation: Ji, P.; Yang, N.; Lin, S.; Xiong, Y. EDI-YOLO: An Instance Segmentation Network for Tomato Main Stems and Lateral Branches in Greenhouse Environments. *Horticulturae* **2025**, *11*, 1260. <https://doi.org/10.3390/horticulturae11101260>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Tomatoes, a major economic crop with global annual production of 186.2 million tons [1], involve multiple precision-intensive operations—including seeding, transplanting, pruning, pollination, and harvesting—that make them an ideal testbed for agricultural robotics [2]. Although greenhouse cultivation offers relatively controlled operating conditions [3], the dynamic nature of plant morphology, complex in-greenhouse lighting, and mutual occlusion among branches and leaves continue to impede robotic performance [4]. These issues are particularly acute for tomato stem segmentation: (i) complex lighting—varying illumination and shadows produce unstable textures and edges with strong reflections and backlighting; (ii) similar elongated structures—stems appear nearly identical to support strings and lateral branches, confounding local features; and (iii) dense occlusion—tightly packed plants with overlapping stems, branches, and fruits lead to fragmented or incorrectly connected segmentations.

These challenges expose limitations in prevailing instance-segmentation approaches. Two-stage methods such as Mask R-CNN [5,6] offer high accuracy but are too computationally demanding for real-time mobile deployment. Single-stage methods such as CondInst [7,8] improve speed yet suffer from localization and boundary errors in complex greenhouse scenes. As a mainstream single-stage approach, YOLOv8n-seg [9] is attractive for efficiency and small-target detection; however, our preliminary tests indicate frequent missed detections and spurious connections between separate stems under challenging lighting and occlusion, as well as limited capacity to represent elongated structures. Consequently, current methods cannot reliably segment elongated, occluded tomato stems while maintaining real-time performance.

To address the difficulties posed by uneven lighting and foliage occlusion in greenhouse environments for tomato main stem and lateral branch segmentation, this study develops the EDI-YOLO algorithm, based on YOLOv8n-seg, to enhance detection and segmentation of elongated plant organs. The primary contributions are as follows:

- We substituted the original backbone with EfficientNetV1 [10], achieving a balance between feature representation and computational efficiency while improving segmentation of morphologically complex tomato stems.
- We developed the C2f-DWR module [11], which incorporates multi-scale dilated convolution to strengthen feature perception across spatial scales, directly addressing the challenge of similar elongated structures.
- We incorporated the iRMB into the neck network [12] to enhance spatial-structure perception of tomato stems, specifically targeting dense occlusion in greenhouse environments.
- We constructed a specialized tomato stem segmentation dataset collected in greenhouses and applied five data-augmentation strategies to improve generalization across diverse lighting and occlusion scenarios.

2. Related Works

Early explorations in tomato recognition and segmentation relied primarily on traditional computer vision algorithms. Zhao et al. [13] achieved tomato fruit recognition through multi-feature image fusion, reaching 93% detection accuracy under uniform lighting conditions, though performance deteriorated significantly when facing alternating light intensities or complex backgrounds. Malik et al. [14] developed an enhanced HSV and watershed algorithm for mature fruit detection (mAP $\approx$ 81.6%), but its effectiveness remained heavily dependent on fruit surface color saturation, showing insufficient adaptability to greenhouse lighting variations and foliage occlusion. Tian et al. [15] employed adaptive k-means clustering with automatic optimal cluster determination for tomato leaf segmentation, improving leaf extraction precision, although this approach focused specifically on leaf segmentation optimization. Zhang et al. [16] combined adaptive morphological segmentation with multi-feature principal component analysis to detect grasping points on randomly positioned fruit clusters, achieving 94.50% average accuracy, but primarily targeted specific grape cluster scenarios. Liu et al. [17] introduced a cucumber recognition approach employing color-based segmentation and shape characterization, attaining a 86% recognition rate, yet it remained prone to false detections under complex backgrounds and varying illumination conditions. Wang et al. [18] performed tomato stem-leaf point cloud segmentation using skeleton extraction combined with supervoxel clustering, achieving 88% AP, though stem-leaf adhesion and noise points led to accuracy degradation. Ma et al. [19] introduced a phenotypic computation scheme for greenhouse canopy branch skeleton extraction, but this method suffered from data loss due to internal leaf and branch cross-occlusion in dense plant structures. These conventional approaches predominantly relied on color or geometric heuristic rules, lacking robustness against lighting variations and

foliage occlusion, making stable extraction of elongated stem targets difficult in practical greenhouse environments.

With the rapid advancement of deep learning technologies, tremendous potential has emerged in crop recognition and segmentation tasks. Gao et al. [20] enhanced YOLOv5 with strengthened feature fusion to improve tomato fruit detection performance in complex greenhouse scenarios, ultimately achieving excellent results with 97.3% mAP and 90.5% recall under occlusion and lighting variation conditions. Appe et al. [21] designed the CAM-YOLO model incorporating channel and spatial attention mechanisms, along with improved NMS and DIoU, to enhance detection capabilities for small and occluded tomato fruits, reaching 88.1% average precision. Solimani et al. [22] optimized YOLOv8 for detecting tomato flowers, fruits, and nodes, improving small target recognition accuracy (mAP50 of 86.13%). However, these studies primarily concentrated on fruit detection, with relatively limited research on instance segmentation of plant stem structures.

Regarding stem structure detection and segmentation, existing research has achieved breakthroughs but still faces numerous challenges. Zhang et al. [23] achieved seedling stem-leaf point cloud segmentation employing a refined version of the red-billed blue magpie algorithm, reaching 96.5% accuracy. However, this approach depends on high-precision point clouds and complex processing workflows, resulting in poor real-time performance and unverified applicability in mature plant greenhouse environments. Xiang et al. [24] proposed a Hybrid Joint Neural Network method based on edge dual relationships and deep network fusion, specifically designed for nighttime tomato main-lateral stem recognition, demonstrating multimodal potential for low-light scenarios. Feng et al. [25] constructed a main stem visual tracking and centerline extraction system based on Mask R-CNN and transfer learning techniques. Although this method achieved favorable accuracy performance, the high computational complexity of Mask R-CNN resulted in insufficient real-time capabilities. Li et al. [26] achieved multi-part segmentation and pruning point estimation under ideal conditions, but showed high dependency on image quality and limitations when handling structurally complex or highly variable samples. Liang et al. [27] relied on specific hardware to acquire seedling point cloud data under darkroom conditions, possessing certain segmentation capabilities but difficult to extend to mature plants or complex greenhouse environments, with limited applicability. Li et al. [28] constructed the MTA-YOLACT network, incorporating fruit cluster detection and main stem, fruit stalk segmentation within a unified framework, achieving 95.4% F1-score for fruit clusters and 51.9% mAP for main stem segmentation. However, main stem segmentation served only as an auxiliary task, failing to distinguish between main stems and lateral branches, inadequate for supporting precision management requirements like pruning. Zhang et al. [29] proposed YOLOv8n-DDA-SAM for cherry tomato cutting point estimation, utilizing segmentation branches to extract main stem masks. Despite high accuracy (mAP50 of 86.13%), it failed to differentiate main and lateral branches, overlooking the impact of structural variations on cutting point localization.

3. Materials and Methods

3.1. Dataset

3.1.1. Data Acquisition

Data were collected in tomato greenhouses at Beijing Cuihu Technology Co., Ltd. (Beijing, China) Two imaging devices were used: an Intel RealSense D435F depth camera (Intel Corp., Santa Clara, CA, USA) (1280 × 720 pixels; 607 images) and a Huawei smartphone camera (Huawei Technologies Co., Ltd., Shenzhen, China) (3456 × 3456 pixels; 512 images). Both cameras were mounted on a common support affixed to a mobile platform to ensure

stable image acquisition and to emulate potential deployment on agricultural robots. The overall setup and greenhouse environment are shown in Figure 1.

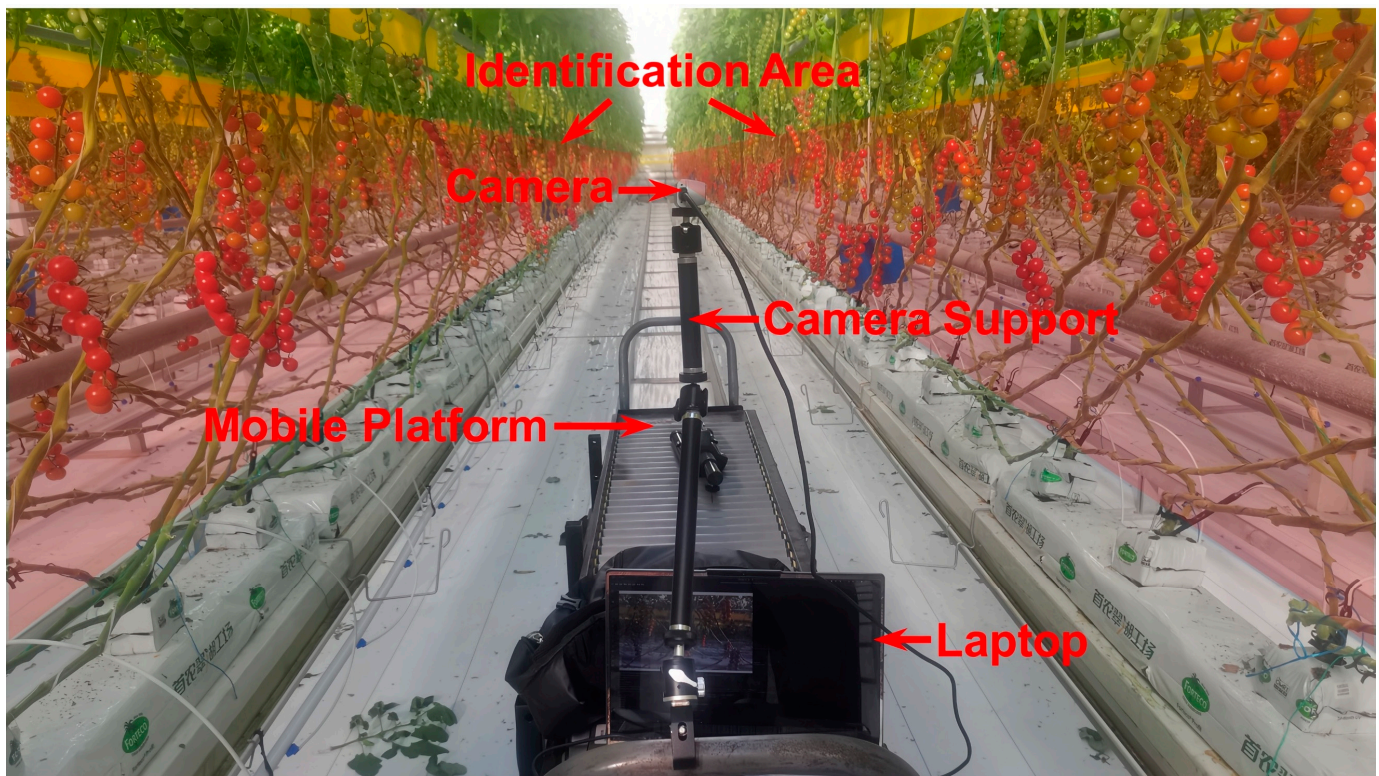


Figure 1. Greenhouse environment and data collection platform for tomato imaging. The red transparent area indicates the model identification area.

To increase variability and capture representative real-world scenarios, the dataset encompassed diverse imaging conditions (Figure 2), including sunny and cloudy weather; front and back lighting; horizontal and overhead perspectives; and occluded and non-occluded scenes. During image acquisition, the shooting distance was maintained at 50–70 cm.

All images were annotated using the Labelme tool (version 5.8.1). Two categories were defined—main stem and lateral branch—each delineated with polygonal masks. Annotators received standardized training and followed consistent guidelines to ensure label accuracy and reliability. An example is shown in Figure 3, where the main stem is marked in red and the lateral branch in green.

3.1.2. Data Augmentation

After quality screening, all images were standardized to a resolution of 640×640 pixels and randomly partitioned at a 7:2 ratio into training (869 images) and validation (250 images) sets. To enhance model robustness, the training set was expanded to 3074 images via random combinations of five augmentation techniques (Figure 4). These augmentations improved dataset diversity across lighting conditions, viewing angles, plant morphology, and image quality, thereby mitigating acquisition constraints in agricultural environments and providing a solid foundation for precise segmentation of main stems and lateral branches.

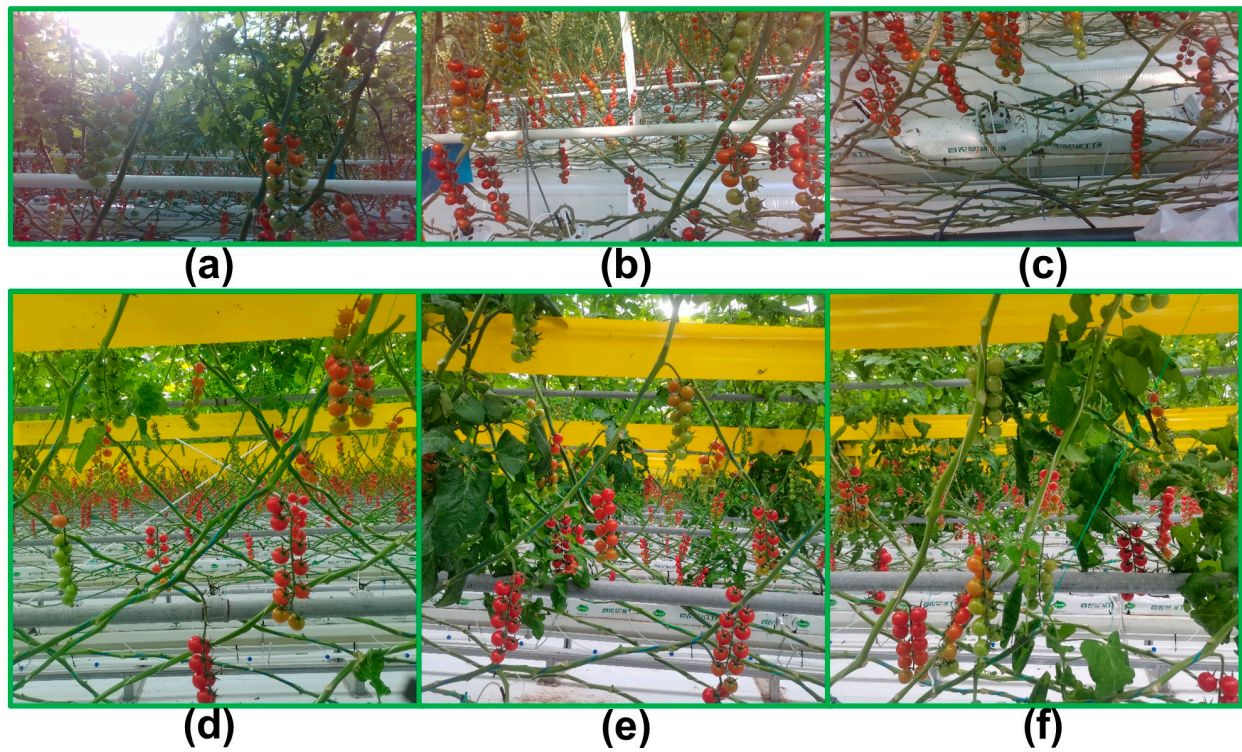


Figure 2. Representative tomato images under diverse acquisition conditions (a–c) sunny; (d–f) cloudy; (c) overhead perspective; (e,f) occluded scenes.

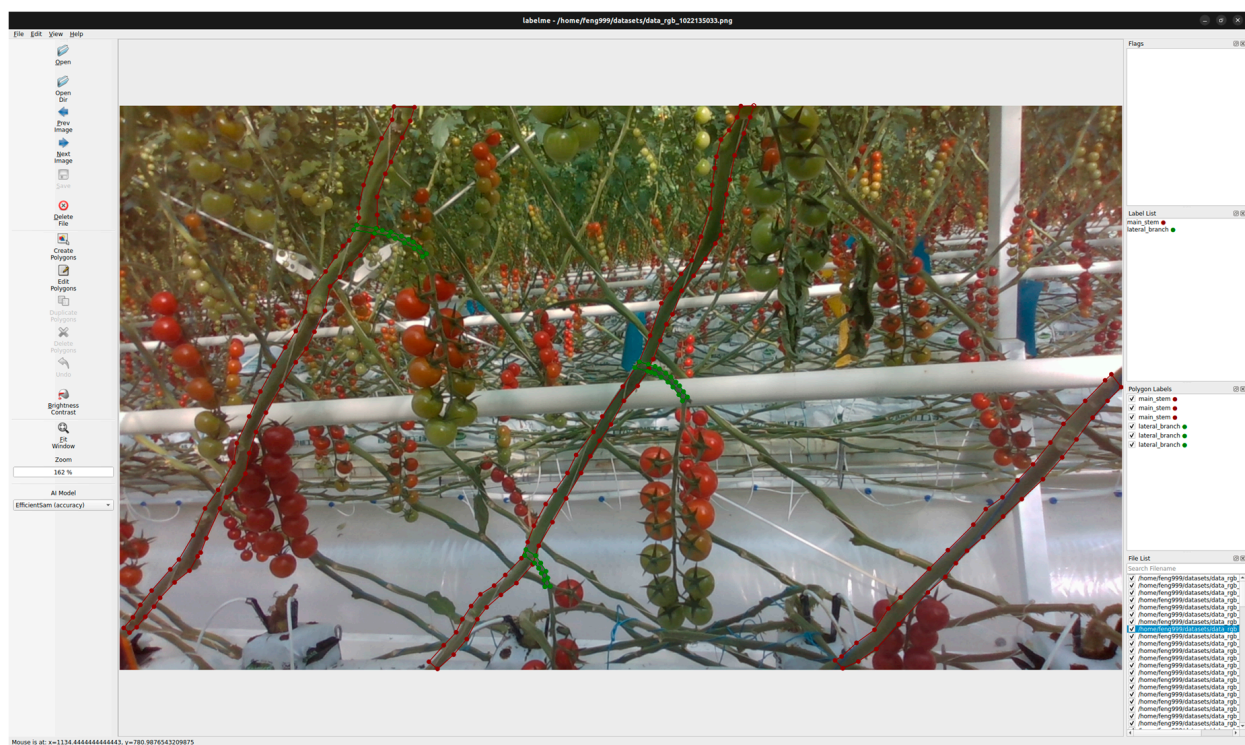


Figure 3. Example of tomato stem and branch annotation using Labelme.

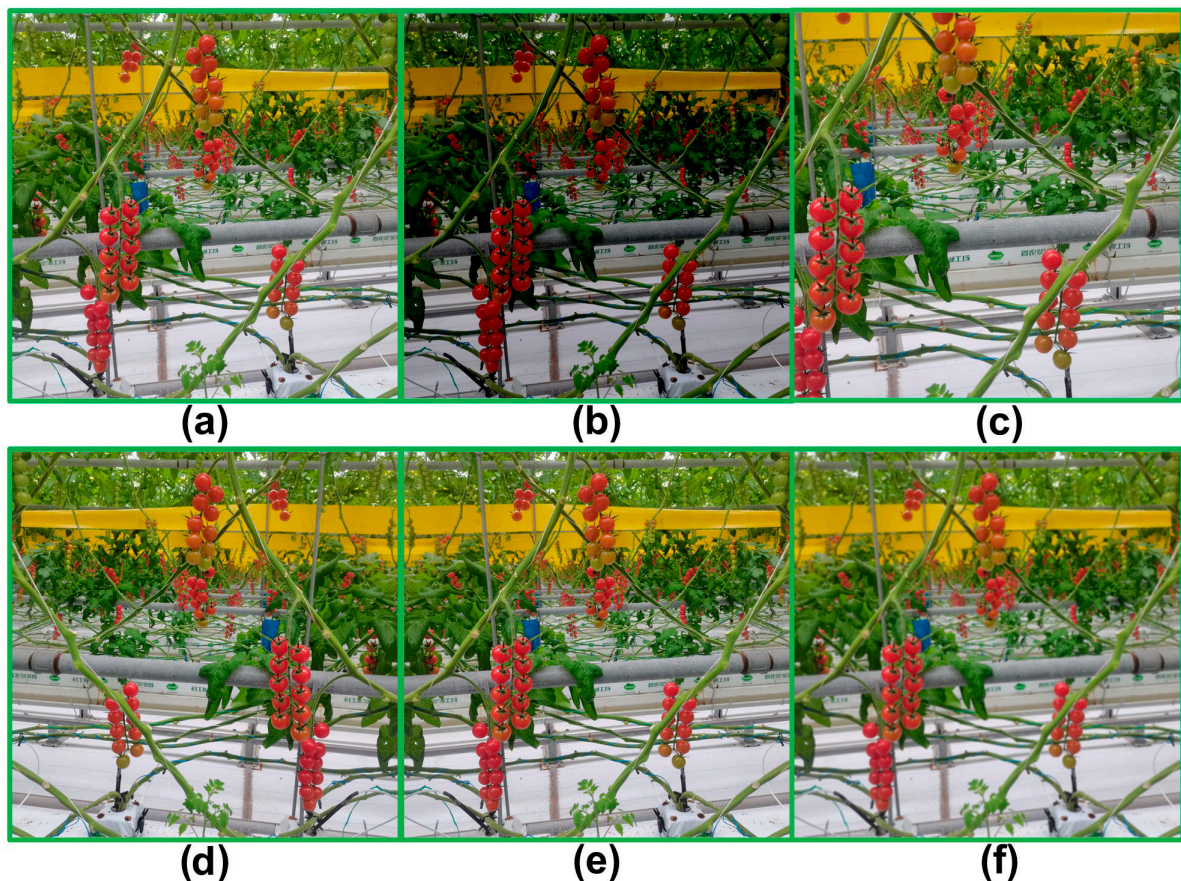


Figure 4. Examples of data augmentation techniques for tomato Images. (a) original image; (b) illumination adjustment; (c) cropping and scaling; (d) horizontal flipping; (e) Gaussian noise; (f) blur effect.

3.2. The Structure of EDI-YOLO

YOLO (You Only Look Once), a representative algorithm in the field of real-time object detection, has received extensive attention in computer vision applications owing to its strong balance between detection speed and accuracy [30]. Nonetheless, an inherent trade-off between speed and accuracy persists, driving ongoing refinement of the YOLO architecture. Among these iterations, YOLOv8n [9] represents a mature development that enhances detection precision while maintaining computational efficiency through optimized network structure design and improved feature extraction mechanisms.

To address the specific challenges of tomato stem detection for agricultural robots in greenhouse environments, the EDI-YOLO segmentation model was developed (Figure 5). The model incorporates three key innovations based on the YOLOv8n-seg [9] architecture: (i) replacement of the original backbone network with EfficientNetV1 [10] to enhance feature extraction capabilities; (ii) implementation of a C2f-DWR [11] structure to optimize segmentation performance for edges and elongated structures; and (iii) integration of iRMB [12] in the neck component to improve critical feature capture. In addition, the head component retains the original YOLOv8n-seg design, incorporating multi-scale detection branches (P3, P4 and P5) for classification and localization, and a Proto module that generates prototype masks combined with detection outputs to produce final segmentation results.

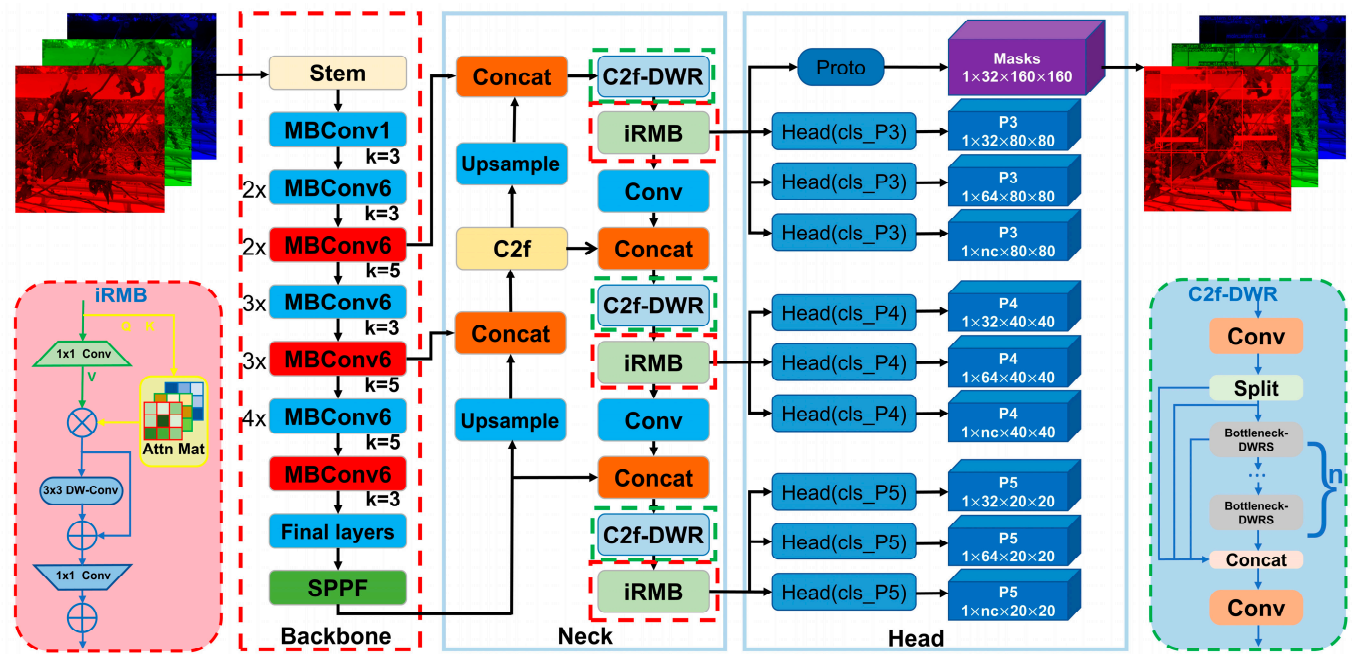


Figure 5. EDI-YOLO network structure diagram.

3.3. EfficientNetV1

To enhance YOLOv8n-seg performance in tomato stem segmentation tasks, this research replaces the original backbone with EfficientNetV1 [10]. The EfficientNetV1 family includes versions from B0 to B7. In this work, we chose EfficientNetV1-B0 because it offers a good balance: the model stays lightweight and efficient while still delivering solid accuracy. Higher versions could improve accuracy a bit more, but they also make the network much heavier, which is not worthwhile for this task. EfficientNetV1 [10] employs a compound scaling strategy that jointly optimizes network depth, width, and input resolution, significantly reducing computational complexity while maintaining high accuracy. As shown in Figure 5, EfficientNetV1-B0 comprises a Stem layer, Final layers, and multiple MBConv (Mobile Inverted Bottleneck Convolution) modules. Various sized MBConv modules are alternately stacked, forming an efficient feature extraction network particularly suited for capturing multi-scale texture features of tomato stems.

MBConv, the core building block of EfficientNetV1, is illustrated in Figure 6a (originally proposed in [10]). It consists of four key components: an expansion convolution (1×1 Conv) that expands input channels by a factor of t ; a $k \times k$ Depthwise Separable Convolution (DW-Conv) that efficiently extracts spatial features; an SE (Squeeze-and-Excitation) attention mechanism that enhances critical feature representation; and a projection convolution (1×1 Conv) that compresses channels back to output dimensions. When input and output dimensions are identical, MBConv introduces a residual connection, mathematically expressed by Equation (1).

As shown in Figure 6b (originally proposed in [10]), the SE module is positioned after the DW-Conv and before the projection convolution. It enhances feature discrimination through three steps: global average pooling to obtain channel descriptors as shown in Equation (2); a two-layer fully connected network generating channel weights as shown in Equation (3), where δ represents Swish activation and σ represents Sigmoid activation; and finally, channel-wise weighting of the original features as shown in Equation (4).

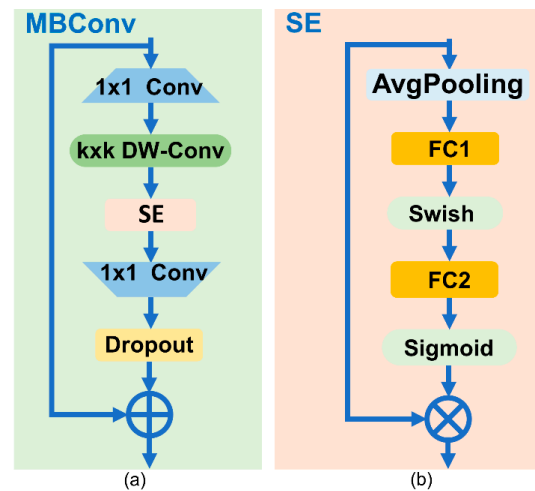


Figure 6. Structure diagrams of MBConv (a) and SE (b) (originally proposed in [10]).

While these architectural components are well-established designs from the original EfficientNetV1 [10] work, our contribution lies in their strategic integration with YOLOv8n-seg for tomato stem segmentation tasks. The selection of EfficientNetV1-B0 represents a carefully considered design choice based on its compound scaling methodology, which systematically balances network depth, width, and input resolution. Although EfficientNetV1-B0 introduces additional parameters compared to the original YOLOv8n backbone, its DW-Conv in MBConv modules decompose standard convolutions into depth-wise and pointwise operations, significantly reducing computational complexity while preserving feature representation capability. Furthermore, the incorporation of the SE attention mechanism provides targeted advantages for agricultural computer vision tasks through channel-wise attention weighting, enabling the network to adaptively emphasize relevant features while suppressing background interference. This is particularly crucial for distinguishing between tomato stems and surrounding background under varying lighting conditions and complex agricultural environments.

$$y = x + F(x) \quad (\text{when conditions are satisfied}) \quad (1)$$

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_c(i, j) \quad (2)$$

$$s_c = \sigma(W_2 \cdot \delta(W_1 \cdot z_c)) \quad (3)$$

$$\tilde{x}_c = s_c \cdot x_c \quad (4)$$

3.4. C2f-DWR

To enhance YOLOv8n-seg's performance in segmenting slender structures like tomato stems, we propose the C2f-DWR module, which maintains the original C2f architecture while replacing standard convolution components with Bottleneck-DWR modules. As shown in Figure 5, C2f-DWR processes input features through an initial convolution and then bifurcates the feature map: one path for direct transmission and another for deep feature extraction via Bottleneck-DWR modules, before merging and fusion. The Bottleneck-DWR module, as shown in Figure 7a, serves as the basic building unit, configuring residual connections based on the input and output dimensions and expanding the receptive field through stacked Conv-DWR layers.

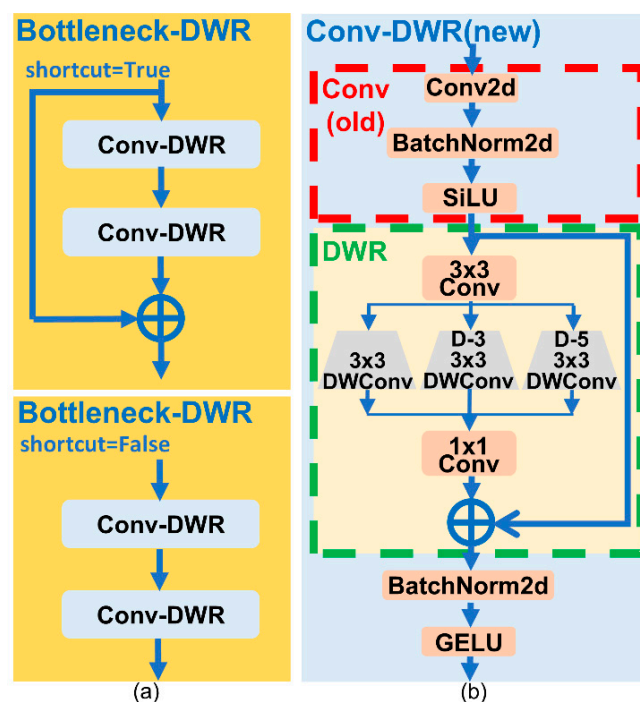


Figure 7. Structure diagrams of Bottleneck-DWR (a) and Conv-DWR (b).

As shown in Figure 7b, The Conv-DWR module represents our core innovation, integrating Dilation-wise Residual (DWR) module into standard convolution operations, enhancing the model's perception of multi-scale spatial information. Unlike previous work [31] that only replaced one convolution layer in each C2f bottleneck, we improve every convolution within the bottleneck using DWR, making the design more thorough. To the best of our knowledge, this is the first attempt to apply such a design to slender-structure segmentation tasks like tomato stems, where maintaining continuity along thin edges is critical. The modification has a clear impact on the baseline model, delivering effective improvements in accuracy and segmentation quality. As illustrated in the green box part of Figure 7b [11], the DWR structure incorporates a standard 3×3 Conv alongside three parallel 3×3 DW-Conv (depth-wise convolutions) with different dilation rates ($d = 1, 2, 3$). These multi-scale features are concatenated along the channel dimension, fused through a 1×1 Conv, added to the input features, and finally processed through normalization and activation functions.

The DWR structure offers several advantages: expanded receptive field without increasing model parameters; establishment of long-range pixel correlations, enhancing segmentation continuity; preservation of spatial resolution, improving edge precision; and simultaneous capture of local details and global structures.

3.5. iRMB

To enhance the model's capability in modeling complex spatial structures of tomato main stems and lateral branches in a complex greenhouse environment, this study integrates the iRMB [12] into the neck network of the YOLOv8n-seg model. The iRMB combines spatial attention with inverted residual design, significantly improving long-range dependency modeling while maintaining computational efficiency, as illustrated in Figure 5.

In the iRMB, the input is divided into two branches. The **V** (value) is generated through a 1×1 Conv from one branch, while the **Q** (query) and **K** (key) are directly

acquired from the input. The $\mathbf{Q}$ and $\mathbf{K}$ are used to compute the attention matrix (Attn Mat), as shown in Equation (5):

$$\mathbf{A}_{i,j} = \frac{\exp(\mathbf{Q}_i \cdot \mathbf{K}_j^\top / \sqrt{d})}{\sum_{j=1}^N \exp(\mathbf{Q}_i \cdot \mathbf{K}_j^\top / \sqrt{d})} \quad (5)$$

where $\mathbf{A}_{i,j}$ represents the attention weight from position i to position j , d denotes the dimension of each head, and N indicates the number of spatial positions within the window. Subsequently, the $\mathbf{V}$ is weighted and aggregated with the Attn Mat $\mathbf{A}$ in the spatial dimension to obtain enhanced features $\mathbf{Y}_i$ at each position, as shown in Equation (6):

$$\mathbf{Y}_i = \sum_{j=1}^N \mathbf{A}_{i,j} \mathbf{V}_j \quad (6)$$

The rearranged $\mathbf{Y}_i$ across all spatial positions constitutes the complete output feature map $\mathbf{Y}$, as shown in Equation (7):

$$\mathbf{Y} = \text{Reshape}(\{\mathbf{Y}_i\}_{i=1}^N) \quad (7)$$

The aggregated features are processed through 3×3 DW-Conv to extract local information and complete the first residual connection with the original path, as shown in Equation (8):

$$\mathbf{Z}_1 = \mathbf{Y} + \text{DWConv}_{3 \times 3}(\mathbf{Y}) \quad (8)$$

Finally, the features are integrated across channels through 1×1 Conv and establish a second residual connection with the input features, yielding the final output $\mathbf{Z}_2$, as shown in Equation (9):

$$\mathbf{Z}_2 = \mathbf{Y} + \text{DWConv}_{3 \times 3}(\mathbf{Z}_1) \quad (9)$$

This dual residual architecture both preserves original information and enhances feature representation, adaptively strengthening key structural features of main stems and lateral branches while effectively suppressing background noise.

Consequently, we implement iRMB as independent attention units serially inserted into the multi-scale feature fusion pathway of the YOLOv8n-seg neck structure, specifically deployed after the 2nd, 3rd, and 4th C2f modules, forming “C2f + iRMB” combinations. In the downsampling path, iRMB enhances the global modeling capability of high-level semantic features P5 and P4; in the upsampling path, iRMB improves the discriminative ability of fusion outputs for fine-grained structures. This multi-scale attention mechanisms achieves effective complementarity between semantic and spatial information, improving the model’s capacity to interpret complex topological structures of tomato plants.

4. Experimental Results and Discussion

4.1. Experiment Environment

Model training and inference evaluation were conducted using an NVIDIA GeForce RTX 3090 Ti (NVIDIA Corporation, Santa Clara, CA, USA) graphics processing unit (GPU) running Ubuntu 22.04. The development environment utilized VSCode 1.100 integrated development environment with software dependencies including Python 3.9.20, PyTorch 2.1.2 framework, and CUDA 12.1 acceleration libraries. Key hyperparameters employed during model training are presented in Table 1.

Table 1. Hyperparameter setting.

Parameter Category	Parameter Setting
Initial learning rate	0.01
Optimizer weight decay	0.0005
Optimizer	SGD
IoU	0.7
Training cycle	200
Image size	640 × 640
Batch	16
Workers	8

4.2. Evaluation Metrics

This study targets accurate and real-time segmentation of tomato stems for greenhouse robotics. To this end, multiple metrics are employed to comprehensively evaluate both detection precision and computational efficiency. Detection performance is measured by *Precision* (P), *Recall* (R), and mean Average Precision (mAP), which reflect the model's capability to accurately identify and segment slender stem structures. Efficiency is assessed using Parameters, GFLOPs, and *FPS*, evaluating the model's lightweight design and suitability for real-time deployment. These metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (11)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (12)$$

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (13)$$

$$mAP50 = \frac{1}{N} \sum_{i=1}^N AP_i \times 100\%, IoU \geq 0.5 \quad (14)$$

$$mAP50 - 95 = \frac{1}{10} \sum_{j=0}^9 mAP(IoU = 0.5 + 0.05j) \times 100\% \quad (15)$$

$$FPS = \frac{1000}{t_{preprocess} + t_{inference} + t_{NMS}} \quad (16)$$

Here, True Positives (*TP*) represents correctly segmented objects, False Positives (*FP*) denotes incorrectly segmented objects, and False Negatives (*FN*) indicates missed objects that should have been segmented. *IoU* (Intersection over Union) metric quantifies the overlap between predicted and ground truth boundaries. Model scale is characterized by Parameters, while GFLOPs reflects the model's complexity. *FPS*, an indicator of real-time processing capability, encompasses the time required for preprocessing, inference, and non-maximum suppression.

4.3. Experimental Analysis

4.3.1. Data Augmentation Comparative Experiment

In this study, YOLOv8n-seg was used as the baseline model, and comparative evaluations were conducted on the original and augmented datasets. As shown in Table 2, the model trained on the augmented dataset achieved clear improvements over the model trained on the original dataset: P increased by 1.8%, R by 4.9%, mAP50 by 4.8%, and

mAP50–90 by 5.1%. These results indicate that effective data augmentation substantially enhances multiple key performance metrics for object detection and segmentation, thereby strengthening robustness and generalization in complex environments.

Table 2. Comparison of model performance using original data and augmented data.

Data	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)
Origin data	77.9	68.4	70.3	26.3
Enhanced data	79.7	73.3	75.1	31.4

4.3.2. Backbone Network Comparison Experiment

In this study, YOLOv8n-seg was employed as the baseline model, and comparative experiments were conducted by substituting the original backbone network with ResNet50 [32], EfficientViT2 [33], and EfficientNetV1 [10] networks. As illustrated in Table 3, YOLOv8n-seg integrated with the EfficientNetV1 backbone demonstrated superior performance, with P, R, mAP50, and mAP50-95 surpassing all other network configurations by 3.2%, 0.7%, 2.3%, and 2.4%, respectively, compared to the baseline. This network combination enhanced detection and segmentation performance while reducing model complexity, with GFLOPs decreasing by 2.6 relative to the baseline model. Compared with other backbones, ResNet50 showed lower accuracy, and EfficientViT2 exhibited relatively high computational cost with lower precision. EfficientNetV1 achieved the optimal balance between accuracy and efficiency. By contrast, EfficientNetV1 achieved the best trade-off between accuracy and efficiency, making it the most suitable choice for our task.

Table 3. Performance comparison of different backbone networks.

Model	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	GFLOPs
Origin-Backbone	79.7	73.3	75.1	31.4	12.0
ResNet50	79	68.1	70.5	26.7	10.3
EfficientViT2	76.3	66.6	68.7	26.4	13.3
EfficientNetV1	82.9	74	77.4	33.8	9.4

4.3.3. Comparative Experiment of Feature Extraction Module

In this comparative experiment, three enhancement strategies for the C2f module in YOLOv8's neck network were evaluated. Using standard C2f-based YOLOv8 as the baseline model, the C2f module was systematically replaced with three variants: C2f-AKConv [34], C2f-DualConv [35], and C2f-DWR [11].

As shown in Table 4, the YOLOv8 model incorporating the C2f-DWR structure demonstrated superior overall performance. Despite a marginal decrease of 0.1% in P, it achieved an mAP50 of 76.5%, representing a 1.4% improvement over the baseline model. Simultaneously, R increased from 73.3% to 73.8%, while model complexity increased slightly from 12 to 12.3 GFLOPs. These results confirm that the C2f-DWR enhancement strategy improves detection capabilities without substantially increasing computational burden, attributed to its multi-scale dilated convolution architecture effectively capturing feature information across various spatial dimensions.

Table 4. Performance comparison of different feature extraction module.

Model	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	GFLOPs
C2f	79.7	73.3	75.1	31.4	12.0
C2f-AKConv	79.8	72.1	73.9	30.3	11.8
C2f-DualConv	81.5	70.2	74.4	30.8	11.3
C2f-DWR	79.6	73.8	76.5	30.9	12.3

4.3.4. Comparative Experiment on Attention Mechanisms

This study employed YOLOv8n-seg as the foundational architecture to systematically evaluate the performance of three attention mechanisms: TripletAtt [36], MultiSEAM [37], and iRMB [12]. As shown in Table 5, experimental results demonstrate that the YOLOv8 variant incorporating iRMB achieved optimal performance across multiple evaluation metrics. Although its P was 0.4% lower than the MultiSEAM [37] configuration, it significantly outperformed in R and mAP50 by 2.6% and 2.2%, respectively. Compared to the original architecture, the YOLOv8-iRMB model achieved improvements of 1.2%, 0.6%, and 1.5% in P, R, and mAP50, respectively, with computational overhead increasing by only 0.7 GFLOPs.

Table 5. Performance comparison of different attention mechanisms.

Model	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)	GFLOPs
None	79.7	73.3	75.1	31.4	12
TripletAtt	80.6	71.7	75.6	31.5	12
MultiSEAM	81.3	71.3	74.4	31.1	12.4
iRMB	80.9	73.9	76.6	31.4	12.7

Heatmap visualization analysis further validated that the model equipped with iRMB enhanced tomato stem feature extraction capabilities. As shown in Figure 8a,b, YOLOv8-iRMB demonstrates superior feature localization capabilities, precisely delineating the contours of main stems and lateral branches while effectively mitigating background interference. These results illustrate the advantages of combining spatial attention with inverted residual blocks in iRMB, maintaining computational efficiency while capturing long-distance spatial dependencies.

4.3.5. Ablation Experiments of Different Improved Modules

To systematically evaluate the independent contribution and synergistic effects of the three proposed model improvement components on performance, a series of ablation experiments were conducted using YOLOv8n-seg as the baseline model. The experimental data presented in Table 6 demonstrate that each component yields positive effects when independently integrated. Specifically, the introduction of EfficientNetV1 alone increased mAP50 to 77.4% (a 2.3% improvement), while C2f-DWR and iRMB raised this metric to 76.5% and 76.6%, respectively. EfficientNetV1 enhanced not only segmentation accuracy but also improved mAP50-90 from 31.4% to 33.8%, demonstrating superior cross-scale feature extraction capabilities. When all three components were applied simultaneously, the model achieved mAP50 and mAP50-90 of 79.3% and 33.9%, respectively, representing substantial improvements of 4.2% and 2.5% over the baseline model. For the identification of highly complex lateral branch structures, mAP50 increased from 60.4% to 65.2%, corresponding to a 4.8% improvement.

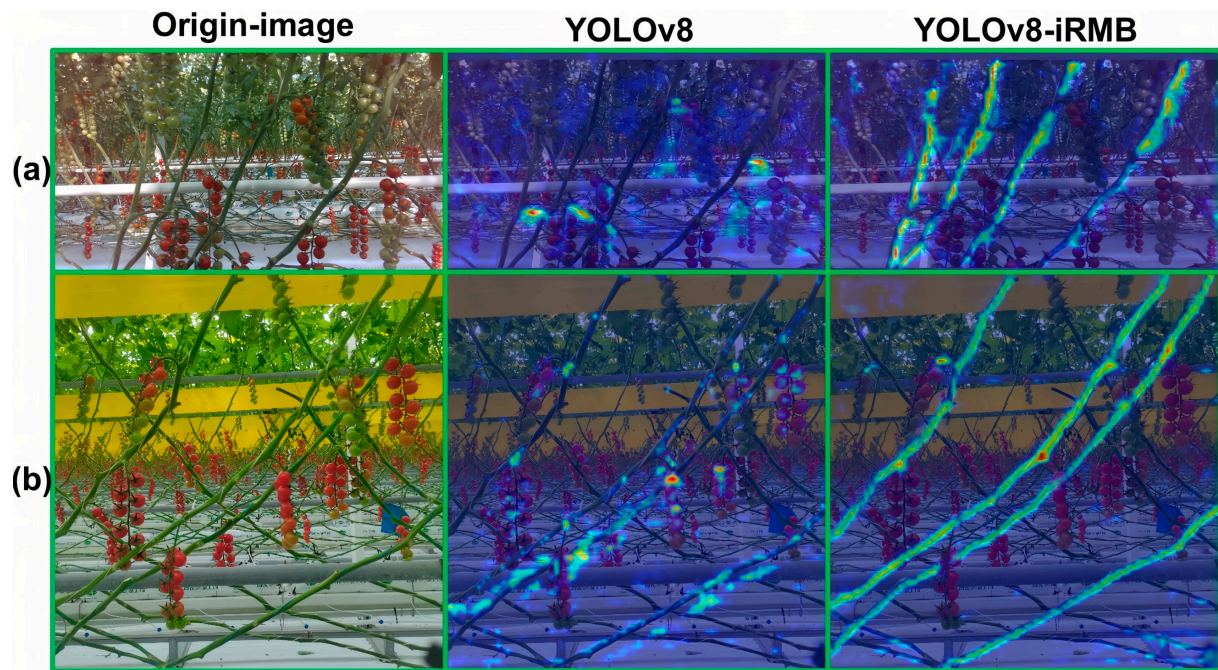


Figure 8. Heatmap visualization of feature extraction by YOLOv8 and YOLOv8-iRMB. (a) Visualization example 1; (b) Visualization example 2. The yellow-green regions indicate the areas where the model focuses its attention.

Table 6. The results of ablation experiments.

Model	EfficientNetV1	C2f-DWR	iRMB	P (%)	R (%)	F1 (%)	mAP50 (%)	mAP50-95 (%)
YOLOv8n-seg				79.7	73.3	76.4	75.1	31.4
E-YOLO	✓			82.9	74.0	78.2	77.4	33.8
C-YOLO		✓		79.6	73.8	76.6	76.5	30.9
I-YOLO			✓	80.9	73.9	77.2	76.6	31.4
ED-YOLO	✓	✓		83.1	74.6	78.4	78.3	33.4
EI-YOLO	✓		✓	81.5	75.4	78.2	78.6	34.5
DI-YOLO		✓	✓	82.9	72.6	77.2	76.7	31.4
EDI-YOLO	✓	✓	✓	83.5	76.0	79.4	79.3	33.9

4.3.6. Comparative Experiment Between EDI-YOLO and YOLOv8n-seg

Based on the ablation experiment, we further conducted a comprehensive comparative analysis of EDI-YOLO and YOLOv8n-seg to verify the practical effect of the improved scheme we proposed in the greenhouse environment.

Table 7 presents the performance comparison between EDI-YOLO and YOLOv8n-seg for segmentation tasks involving main stems and lateral branches of tomato plants. The experimental results demonstrate that EDI-YOLO achieves substantial improvements across all evaluation metrics. For main stem segmentation, EDI-YOLO attains 93.4% mAP50, representing a 3.6% increase over the baseline model, while achieving 46.6% mAP50-95, exceeding the baseline by 4.0%. For lateral branch segmentation, EDI-YOLO achieves 65.2% mAP50, outperforming the baseline by 4.8%, and 21.1% mAP50-95, showing a 0.9% improvement over the baseline model. These results indicate superior performance for small target detection.

Table 7. Performance comparison of different categories.

Model	Class	mAP50 (%)	mAP50-95 (%)
YOLOv8n-seg	main_stem	89.8	42.6
	lateral_branch	60.4	20.2
EDI-YOLO	main_stem	93.4	46.6
	lateral_branch	65.2	21.1

Figure 9 illustrates the comparative tomato stem segmentation results between EDI-YOLO and YOLOv8n-seg across four representative scenarios (a–d). Visual analysis reveals significant differences in segmentation quality between the two models. YOLOv8n-seg exhibits several limitations: noticeable segmentation discontinuities in the upper-left regions of Figure 9a,b,d; missed detection of a lateral branch in Figure 9b; and erroneous double detection frames for the same lateral branch in Figure 9d. In contrast, EDI-YOLO demonstrates distinct advantages, achieving continuous and complete stem segmentation, accurately identifying fine lateral branches overlooked by the baseline model, and effectively eliminating duplicate detection occurrences. These results demonstrate the model’s capability to accurately perform segmentation of tomato main stems and lateral branches in complex greenhouse scenarios characterized by varying lighting conditions and intertwined branch structures.

4.3.7. Comparative Experiments of Different Models

EDI-YOLO is designed to balance detection accuracy with computational efficiency, prioritizing accuracy while satisfying real-time constraints (≥ 30 FPS). To comprehensively evaluate performance, we compare EDI-YOLO with the traditional two-stage architecture Mask R-CNN [5], the single-stage architecture CondInst [7], and several evolutionary YOLO variants (YOLOv5s-seg [38], YOLOv7tiny-seg [39], YOLOv9c [40], YOLOv11s-seg [41], and YOLOv12s-seg [42]).

As shown in Table 8, EDI-YOLO consistently improves upon the YOLOv8n-seg baseline across all accuracy metrics: mAP50 increases from 75.1% to 79.3% (+4.2%), F1-score from 76.4% to 79.4% (+2.9%), and P from 79.7% to 83.5% (+3.8%). Although the inference speed decreases from 288.4 FPS to 86.2 FPS, it remains sufficient for real-time applications, while the computational cost is reduced from 12.0 to 10.4 GFLOPs.

Table 8. The results of comparative experiments with different models.

Model	P (%)	R (%)	mAP50 (%)	F1 (%)	GFLOPs	Parameter (M)	FPS (f/s)
Mask-RCNN	47.4	58.8	48.2	52.4	18.6	44.0	33.2
CondInst	72.1	67	63.4	69.4	20.9	34.0	67.3
YOLOv5s-seg	72.7	67	66.7	69.7	25.7	7.4	291.9
YOLOv7tiny-seg	80	75.5	75.4	77.7	47.7	7.0	240.2
YOLOv8n-seg	79.7	73.3	75.1	76.4	12.0	3.3	288.4
YOLOv9c	82.7	76.8	79.4	79.6	157.6	27.6	81.5
YOLOv11s-seg	79.4	74.4	76.1	79.8	35.3	10.1	201.4
YOLOv12s-seg	81.9	72	75.4	76.6	35.2	9.9	142.4
EDI-YOLO	83.5	76	79.3	79.4	10.4	8.0	86.2

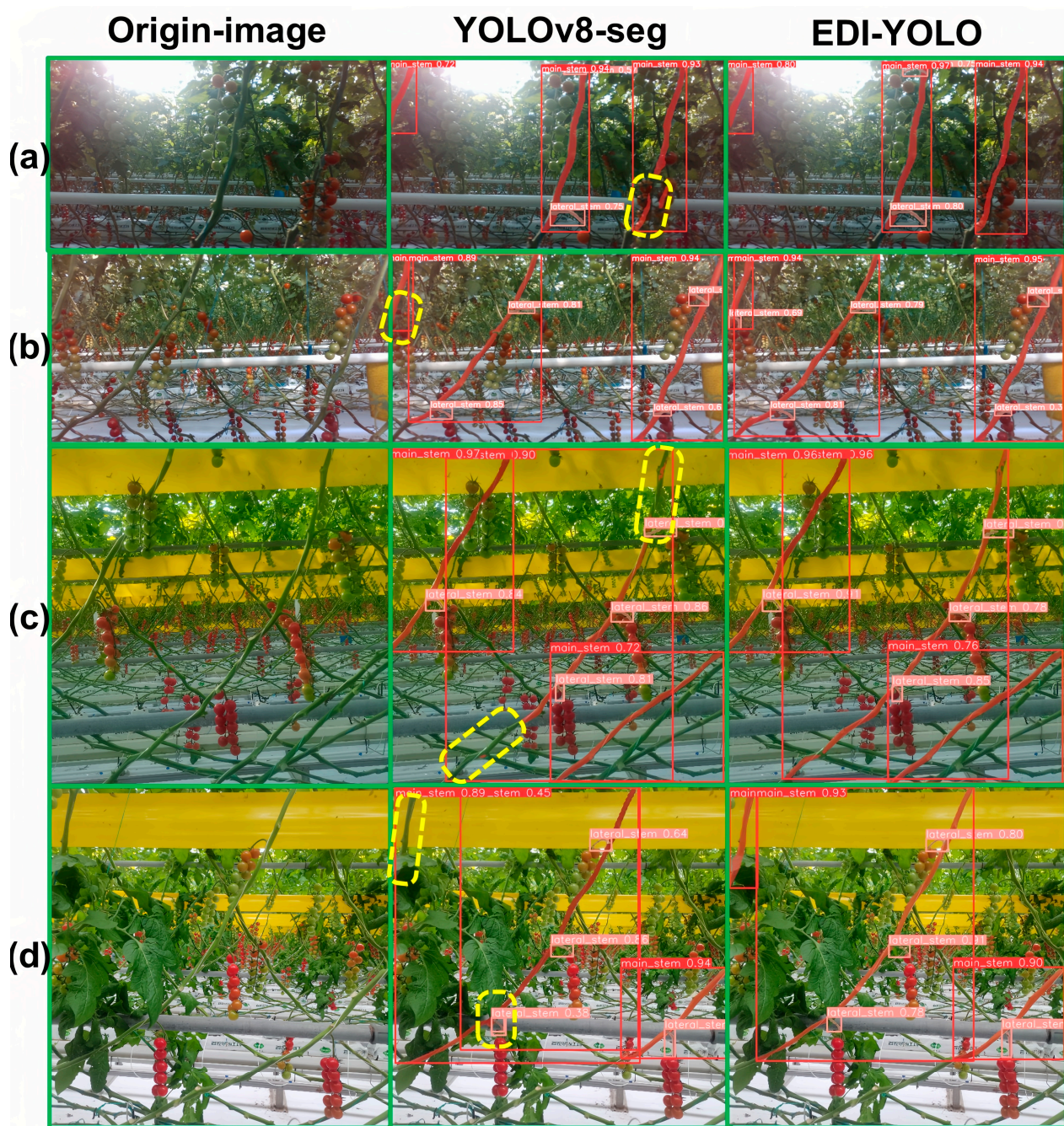


Figure 9. Comparison of tomato stem segmentation between YOLOv8-seg and EDI-YOLO. (a) backlight; (b) frontlight; (c) without obvious occlusion; (d) with occlusion.

Compared with traditional detectors, Mask R-CNN [5]—representing a typical two-stage model—achieves only 48.2% mAP50 with a speed of about 33.2 FPS and a parameter size of 44.0 M, rendering real-time deployment on resource-constrained hardware impractical. CondInst [7] improves inference speed (≈ 67.3 FPS), but its accuracy remains limited (63.4% mAP50), and both its parameter size (34.0 M) and complexity (20.9 GFLOPs) are substantially higher than those of EDI-YOLO.

Within the YOLO family, EDI-YOLO offers a more balanced trade-off. In terms of parameter size, smaller-scale models such as YOLOv5s-seg [38], YOLOv7tiny-seg [39], and YOLOv8n-seg achieve very high speeds (≈ 240 – 291 FPS) but at the expense of accuracy (mAP50 of 66.7%, 75.4%, and 75.1%, respectively). Among mid-scale models, YOLOv11s-seg [41] attains 76.1% mAP50—lower than EDI-YOLO’s 79.3%—while requiring more

parameters (10.1 M vs. 8.0 M) and a greater computational load (35.3 vs. 10.4 GFLOPs). YOLOv12s-seg [42] reaches a P of 81.9%, below EDI-YOLO's 83.5%, and its mAP50 (75.4%) and F1-score (76.6%) are also lower, with higher resource demands. For larger-scale models such as YOLOv9c [40], although the mAP50 is slightly higher (79.4%), it requires more than three times the parameters (27.6 M vs. 8.0 M) and a much larger computational budget (157.6 vs. 10.4 GFLOPs).

Notably, EDI-YOLO achieves near-large-model accuracy with a compact footprint, which confers clear practical value. With only 8.0 M parameters, it operates efficiently on resource-constrained edge devices, and its inference speed of 86.2 FPS is sufficient to ensure real-time performance without increasing computational burden. Overall, EDI-YOLO consistently exhibits a clear advantage across comparisons: it delivers competitive accuracy at a reasonable computational cost, making it especially suitable for scenarios where both accuracy and responsiveness are critical, such as precision agriculture.

5. Conclusions

The segmentation of tomato stems in greenhouse environments is hindered by challenges such as variable lighting, cluttered backgrounds, and the difficulty of detecting slender plant structures. To address these, we developed the EDI-YOLO algorithm, integrating an EfficientNetV1 backbone, three C2f_DWR feature extraction modules, and the iRMB attention mechanism. This approach facilitates reliable identification of slender plant structures under complex environmental conditions, substantially improving the segmentation accuracy of both main stems and lateral branches and making a meaningful contribution to precision greenhouse agriculture.

Our experimental evaluation on a custom tomato stem dataset reveals substantial improvements, with the proposed method achieving an mAP50 of 79.3%, a 4.2% increase over the baseline model. Notably, lateral branch segmentation accuracy improved by 4.8 percentage points. Additionally, the system maintains computational efficiency with an inference speed of 86.2 FPS and requires only 10.4 GFLOPs, balancing precision with processing efficiency. By providing higher accuracy while maintaining reliable real-time performance, the model can improve the efficiency of tomato operations and support the sustainability of protected tomato cultivation through reduced manual intervention.

Despite these advancements, two key limitations remain. First, while computational efficiency has been improved, further reductions in model parameters and computational demands are possible. Second, the current model focuses on two-dimensional analysis, lacking comprehensive 3D plant architecture modeling; thus, it cannot detect lateral branches completely occluded behind main stems.

Future research directions include leveraging the dataset and annotation strategy developed in this study as a foundation for adapting the proposed framework to other trellised or climbing crops through transfer learning. To address the limitation of 2D analysis and enhance perception of complex plant structures, we plan to integrate multi-modal data sources such as RGB-D and multispectral imaging, which will provide depth information for comprehensive 3D plant architecture modeling and can support downstream tasks in robotic operations. Furthermore, we aim to implement advanced model compression and quantization techniques to further reduce deployment costs and enhance detection speed on edge devices. Since FPS on embedded hardware (e.g., NVIDIA Jetson series) is an important indicator of practical value, we will further test and report the model's real-time performance on such platforms in our future work.

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

Funding: This study was supported financially by the Facility Gardening Robotic Operation System (Phase II) Project of Haidian District, Beijing (NO. HDNY20250112-3), the National Key R&D Program Subproject: Development and Integrated Application of Precision Decision-Making Model Integrating Human-Machine-Environment-Agronomy for Open-field Vegetables (No. 2024YFD2000802-4) and the Xiongan R&D Program project: the Key Technology of Intelligent Picking Robot for Facility Solanum Fruit (No. 20250922).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

1. Wikipedia. List of Countries by Tomato Production. Available online: https://en.wikipedia.org/wiki/List_of_countries_by_tomato_production (accessed on 7 April 2025).
2. Kim, J.Y.; Pyo, H.R.; Jang, I.; Kang, J.; Ju, B.K.; Ko, K.E. Tomato harvesting robotic system based on Deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem. *Comput. Electron. Agric.* **2022**, *201*, 107300. [CrossRef]
3. Li, H.; Xu, L. The development and prospect of agricultural robots in China. *Acta Agric. Zhejiangensis* **2015**, *27*, 865–871.
4. Rajendran, V.; Debnath, B.; Mghames, S.; Mandil, W.; Parsa, S.; Parsons, S.; Ghalamzan-E., A. Towards autonomous selective harvesting: A review of robot perception, robot design, motion planning and control. *J. Field Robot.* **2023**, *41*, 2247–2279. [CrossRef]
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2980–2988.
6. Shen, L.; Su, J.; Huang, R.; Quan, W.; Song, Y.; Fang, Y.; Su, B. Fusing attention mechanism with Mask R-CNN for instance segmentation of grape cluster in the field. *Front. Plant Sci.* **2022**, *13*, 934450. [CrossRef]
7. Tian, Z.; Shen, C.; Chen, H. Conditional Convolutions for Instance Segmentation. In *Computer Vision—ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 282–298.
8. Yue, X.; Qi, K.; Na, X.; Zhang, Y.; Liu, Y.; Liu, C. Improved YOLOv8-seg network for instance segmentation of healthy and diseased tomato plants in the growth stage. *Agriculture* **2023**, *13*, 1643. [CrossRef]
9. Ultralytics. Explore Ultralytics YOLOv8. Available online: <https://docs.ultralytics.com/models/yolov8/> (accessed on 7 April 2025).
10. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
11. Wei, H.; Liu, X.; Xu, S.; Dai, Z.; Dai, Y.; Xu, X. DWRSeg: Rethinking efficient acquisition of multi-scale contextual information for real-time semantic segmentation. *arXiv* **2022**, arXiv:2212.01173.
12. Zhang, J.; Li, X.; Li, J.; Liu, L.; Xue, Z.; Zhang, B.; Jiang, Z.; Huang, T.; Wang, Y.; Wang, C. Rethinking Mobile Block for Efficient Attention-Based Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1389–1400.
13. Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. Robust tomato recognition for robotic harvesting using feature images fusion. *Sensors* **2016**, *16*, 173. [CrossRef]
14. Malik, M.H.; Zhang, T.; Li, H.; Zhang, M.; Shabbir, S.; Saeed, A.S.M. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. *IFAC-PapersOnLine* **2018**, *51*, 431–436. [CrossRef]
15. Tian, K.; Li, J.; Zeng, J.; Evans, A.; Zhang, L. Segmentation of tomato leaf images based on adaptive clustering number of k-means algorithm. *Comput. Electron. Agric.* **2019**, *165*, 104962. [CrossRef]
16. Zhang, Q.; Gao, G. Grasping point detection of randomly placed fruit cluster using adaptive morphology segmentation and principal component classification of multiple features. *IEEE Access* **2019**, *7*, 158035–158050. [CrossRef]
17. Liu, W.; Sun, H.; Xia, Y.; Kang, J. Real-time cucumber target recognition in greenhouse environments using color segmentation and shape matching. *Appl. Sci.* **2024**, *14*, 1884. [CrossRef]

18. Wang, Y.; Liu, Q.; Yang, J.; Ren, G.; Wang, W.; Zhang, W.; Li, F. A method for tomato plant stem and leaf segmentation and phenotypic extraction based on skeleton extraction and supervoxel clustering. *Agronomy* **2024**, *14*, 198. [\[CrossRef\]](#)
19. Ma, X.; Jiang, Q.; Guan, H.; Wang, L.; Wu, X. Calculation method of phenotypic traits for tomato canopy in greenhouse based on the extraction of branch skeleton. *Agronomy* **2024**, *14*, 2837. [\[CrossRef\]](#)
20. Gao, G.; Wang, S.; Shuai, C.; Zhang, Z.; Zhang, S.; Feng, Y. Recognition and detection of greenhouse tomatoes in complex environment. *Trait. Signal.* **2022**, *39*, 291–298. [\[CrossRef\]](#)
21. Appe, S.N.; Arulselvi, G.; Balaji, G.N. CAM-YOLO: Tomato detection and classification based on improved YOLOv5 using combining attention mechanism. *PeerJ Comput. Sci.* **2023**, *9*, e1463. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Solimani, F.; Cardellicchio, A.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [\[CrossRef\]](#)
23. Zhang, L.; Huang, Z.; Yang, Z.; Yang, B.; Yu, S.; Zhao, S.; Zhang, X.; Li, X.; Yang, H.; Lin, Y.; et al. Tomato stem and leaf segmentation and phenotype parameter extraction based on improved red billed blue magpie optimization algorithm. *Agriculture* **2025**, *15*, 180. [\[CrossRef\]](#)
24. Xiang, R.; Zhang, M.; Zhang, J. Recognition for stems of tomato plants at night based on a hybrid joint neural network. *Agriculture* **2022**, *12*, 743. [\[CrossRef\]](#)
25. Feng, Q.; Cheng, W.; Zhang, W.; Wang, B. Visual Tracking Method of Tomato Plant Main-Stems for Robotic Harvesting. In Proceedings of the 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Jiaying, China, 27–31 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 886–890.
26. Li, X.; Fang, J.; Zhao, Y. A multi-target identification and positioning system method for tomato plants based on VGG16-UNet model. *Appl. Sci.* **2024**, *14*, 2804. [\[CrossRef\]](#)
27. Liang, X.; Yu, W.; Qin, L.; Wang, J.; Jia, P.; Liu, Q.; Lei, X.; Yang, M. Stem and leaf segmentation and phenotypic parameter extraction of tomato seedlings based on 3D point cloud. *Agronomy* **2025**, *15*, 120. [\[CrossRef\]](#)
28. Li, Y.; Feng, Q.; Liu, C.; Xiong, Z.; Sun, Y.; Xie, F.; Li, T.; Zhao, C. MTA-YOLACT: Multitask-aware network on fruit bunch identification for cherry tomato robotic harvesting. *Eur. J. Agron.* **2023**, *146*, 126812. [\[CrossRef\]](#)
29. Zhang, G.; Cao, H.; Jin, Y.; Zhong, Y.; Zhao, A.; Zou, X.; Wang, H. YOLOv8n-DDA-SAM: Accurate cutting-point estimation for robotic cherry-tomato harvesting. *Agriculture* **2024**, *14*, 1011. [\[CrossRef\]](#)
30. Ghasemi, Y.; Jeong, H.; Choi, S.H.; Park, K.-B.; Lee, J.Y. Deep learning-based object detection in augmented reality: A systematic review. *Comput. Ind.* **2022**, *139*, 103661. [\[CrossRef\]](#)
31. Wang, R.; Liang, F.; Wang, B.; Zhang, G.; Chen, Y.; Mou, X. An Efficient and Accurate Surface Defect Detection Method for Wood Based on Improved YOLOv8. *Forests* **2024**, *15*, 1176. [\[CrossRef\]](#)
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.
33. Cai, H.; Li, J.; Hu, M.; Gan, C.; Han, S. EfficientViT: Lightweight Multi-Scale Attention for High-Resolution Dense Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 17256–17267.
34. Zhang, X.; Song, Y.; Song, T.; Yang, D.; Ye, Y.; Zhou, J.; Zhang, L. LDConv: Linear deformable convolution for improving convolutional neural networks. *Image Vis. Comput.* **2024**, *149*, 105190. [\[CrossRef\]](#)
35. Zhong, J.; Chen, J.; Mian, A. DualConv: Dual convolutional kernels for lightweight deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 9528–9535. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to Attend: Convolutional triplet attention module. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 3138–3147.
37. Wang, Y.; Zhang, J.; Kan, M.; Shan, S.; Chen, X. Self-Supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 12275–12284.
38. Ultralytics. Ultralytics YOLOv5. Available online: <https://docs.ultralytics.com/models/yolov5/> (accessed on 7 April 2025).
39. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475.
40. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision—ECCV 2024, Proceedings of the 18th European Conference, Milan, Italy, 29 September–4 October 2024*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2024; pp. 1–21.

41. Ultralytics. Ultralytics YOLO11. Available online: <https://docs.ultralytics.com/models/yolo11/> (accessed on 7 April 2025).
42. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.

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.