Smart Agricultural Technology 12 (2025) 101465

Contents lists available at ScienceDirect

Smart Agricultural Technology

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

# GCD-YOLO: A deep learning network for accurate tomato fruit stalks identification in unstructured environments

Wuxiong Weng [a,c], Zhenhui Lai [a,c], Zheming Cui [a,c], Zhixiong Chen [a,c], Hongbin Chen [a,c], Tianliang Lin [b], Jufei Wang [a,c], Shuhe Zheng [a,c,*], Guoqing Chen [d,*]

[a] College of Mechanical and Electrical Engineering, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[b] Fujian Key Laboratory of Green Intelligent Drive and Transmission for Mobile Machinery Xiamen 361021, China
[c] Fujian University Engineering Research Center for Modern Agricultural Equipment, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[d] College of Engineering, Northeast Agricultural University, Harbin 150030, China

ARTICLE INFO

*Keywords:*
Machine vision
Deep learning
YOLOv8
Tomato fruit stalks
Edge device deployment

ABSTRACT

Accurate identification of tomato fruit stalks is critical for end-effectors to perform precise cutting operations, which contributes to enhanced harvesting efficiency and reduced reliance on manual labor. However, actual field environments may compromise the accuracy of tomato fruit stalks identification. To address this issue, this study proposes an improved GCD-YOLO model based on YOLOv8n. Specifically, GAM is integrated into the YOLOv8 backbone network, the original Neck is replaced with CCFM, and Dyhead is adopted as the detection head for tomato fruit stalks identification. The results show that the different improvement modules can effectively enhance the feature extraction ability of the model, thus improving the detection accuracy of the GCD-YOLO model for tomato fruit stalks. The GCD-YOLO model achieves a detection precision of 94.4% and an mAP@50 of 91.7% for tomato fruit stalks, significantly outperforming YOLOv8n, SSD, Faster R-CNN, YOLOv5n, RT-DETR, YOLOv9t, YOLOv11n and YOLOv12n. Furthermore, the GCD-YOLO model was deployed on an NVIDIA Jetson Orin Nano for field experiments. Experimental results demonstrate that the GCD-YOLO model delivers outstanding detection performance on edge devices, achieving an inference speed of 26.24FPS in actual field environments. The research findings contribute to the development of smart agriculture and facilitate the implementation of machine vision in tomato harvesting applications.

## 1. Introduction

Tomatoes are one of the most widely cultivated cash crops globally and rank as the world's second-largest agricultural crop [1]. However, the current harvesting of ripe tomato fruits primarily relies on manual labor, where each fruit is individually picked. This traditional harvesting method is labor-intensive and time-consuming. Particularly in large-scale farming scenarios, manual harvesting methods exhibit low efficiency and struggle to meet the high-efficiency production requirements of modern agriculture. By contrast, mechanical harvesting significantly reduces reliance on manual labor, but it still requires human intervention and demands high technical proficiency in machinery operation from workers [2]. With the rapid advancement of artificial intelligence, the realization of automated harvesting represents a pivotal trend in the development of smart agriculture. Within this process, fruit identification and accurate determination of tomato maturity stand as critical technical components in automated harvesting systems. Current research predominantly focuses on tomato harvesting through fruit recognition [3]. While these methods can localize the main body of tomatoes, such techniques fail to directly obtain the coordinates of the stalk cutting points. When robotic end-effectors attempt to grasp fruits based on positional estimates from such methods, squeezing-induced fruit damage frequently occurs [4]. It is noteworthy that stalk identification provides critical support for determining cutting points. By directly shearing the stalk, contact between the end-effector and tomato fruits can be minimized, thereby significantly reducing mechanical damage to the fruits.

Hyperspectral imaging [5], LiDAR (light detection and ranging) [6], and machine vision technologies [7] have been widely utilized in agricultural applications, such as weed identification [8], crop classification [9], and fruit harvesting [10], among others. Hyperspectral imaging captures continuous spectral information, enabling simultaneous acquisition of characteristic spectral features from both the fruit

* Corresponding authors.
*E-mail addresses:* zsh@fafu.edu.cn (S. Zheng), gqchen0937@163.com (G. Chen).

https://doi.org/10.1016/j.atech.2025.101465
Received 23 July 2025; Received in revised form 15 September 2025; Accepted 20 September 2025
Available online 22 September 2025

**Nomenclature**

| | |
|---|---|
| Avg pool | Average pooling |
| AP | Average precision |
| CBAM | Convolutional block attention module |
| CCFM | Cross-scale feature fusion module |
| CIoU | Complete intersection over union |
| Conv $1 \times 1$ | $1 \times 1$ convolutional layer |
| Conv $3 \times 3$ | $3 \times 3$ convolutional layer |
| Conv $7 \times 7$ | $7 \times 7$ convolutional layer |
| CPU | Central processing unit |
| CUDA | Computer unified device architecture |
| cuDNN | NVIDIA CUDA deep neural network library |
| C2f | Cross stage partial fusion with 2 convolutions |
| DFL | Distribution focal loss |
| Dyhead | Dynamic multi-scale detection head |
| Faster R-CNN | Faster region-based convolutional neural network |
| Fc | Fully connected layer |
| FPN | Feature pyramid network |
| FPS | Frames per second |
| GAM | Global attention mechanism |
| GCD-YOLO | YOLOv8n architecture + GAM + CCFM + Dyhead |
| GFLOPs | Giga floating point operations |
| HSV | The color channels of hue, saturation, and value |
| IoU | Intersection over union |
| JPG | Joint photographic expert group |
| LiDAR | Light detection and ranging |
| m | Meter |
| mAP | Mean average precision |
| MLP | Multilayer perceptron |
| PAN | Path aggregation network |
| RGB | The color channels of red, green, and blue |
| SPPF | Spatial pyramid pooling fast |
| SSD | Single shot multi-Box detector |
| VFL | Varifocal loss |
| YOLO | You only look once |

epidermis and stalk tissues, thereby achieving high-precision stalk detection [11]. However, this technology exhibits significant sensitivity to illumination conditions, where uneven lighting can induce spectral feature distortions. In comparison, LiDAR leverages its active measurement capability to reliably construct 3D point cloud models of fruits and stalks, effectively overcoming lighting interference issues. It also provides centimeter-level spatial positioning data to guide robotic arm harvesting. However, this technology still has limitations. On one hand, LiDAR cannot assess biological characteristics such as fruit maturity through geometric features. On the other hand, a single LiDAR scan generates millions of data points, making it challenging to deploy on edge devices with constrained computing resources. In this context, machine vision technology demonstrates unique application advantages through its real-time processing capabilities. It enables stalk identification and fruit ripeness determination with low computational overhead, providing a robust technical foundation for automated harvesting systems.

Early researchers predominantly utilized traditional image detection methods for both fruit and fruit stalk identification. Zhuang et al. [12] integrated the RGB color space, HSV color space, and Otsu's method to detect litchi fruits and their stalks. Xiong et al. [13] utilized RGB information and morphological processing to achieve grape cluster detection in nighttime images, combined with the Hough line detection method to model grape stalks. However, in practical orchard environments, these methods are susceptible to occlusion and illumination variations, leaving their robustness and accuracy requiring further improvement. With the advancement of convolutional neural networks (CNN), deep learning has been progressively applied to robotic fruit harvesting. In recent years, the YOLO series algorithms have undergone continuous iteration and demonstrated excellent performance in tomato phenotypic analysis and maturity discrimination. For instance, Chen et al. [14] developed a multi-task convolutional neural network detection model based on YOLOv7, which can simultaneously identify tomato fruit clusters and determine fruit maturity. Angelo Cardellicchio et al. [15] proposed an enhanced learning method based on YOLOv11 to achieve comprehensive and continuous analysis of phenotypic traits in tomato plants. Another study utilized YOLOv11 to achieve accurate recognition and counting of tomato fruits [16]. However, their research is limited to the phenotypic analysis of tomato fruits and has not yet applied the model to the recognition of tomato fruit stalks. Notably, when confronted with complex orchard environments, CNN-based deep learning algorithms demonstrate robust learning capabilities and exceptional feature representation in stalk detection tasks. For instance, Liang et al. [17] implemented nighttime detection of litchi fruits and their peduncles by leveraging the YOLOv3 and U-Net networks. Meanwhile, Wu et al. [18] enhanced the YOLOv5 model, optimizing its inference efficiency and achieving accurate detection of banana peduncles. Chen et al. [19] innovatively enhanced the YOLOv8-Pose key-point detection algorithm, reducing the model's parameter count and computational complexity while enabling simultaneous detection of grape clusters and their peduncles. It is noteworthy that their studies did not evaluate model performance on edge devices. To address this gap, researchers have dedicated significant efforts to edge deployment. For instance, Zhang et al. [20]; Ji et al. [21]; Huang et al. [22] deployed optimized models on the Jetson Nano platform, while Wang et al. [23] implemented a binocular stereo vision system on the Raspberry Pi 4B Although these works successfully achieved the deployment of object detection algorithms on edge devices, they have predominantly focused on fruit body detection rather than stalk-specific identification. However, large-fruited tomatoes pose significant challenges for existing algorithms due to their short stalks and complex background noise, which lead to insufficient feature representation. Additionally, accurately determining tomato ripeness remains a major hurdle for robotic harvesters, as successfully identifying mature tomatoes is a critical step in the harvesting process.

Building upon this context, this study targets large-fruited tomatoes in actual field environments and proposes a novel object detection model named GCD-YOLO, designed to simultaneously identify tomato fruit stalks and assess fruit maturity. The main research contents of this paper are as follows:

(1) A tomato stalk dataset was compiled by capturing images from multiple angles under varying illumination conditions.
(2) A standalone detection system was developed for identifying tomato fruit stalks, which is capable of simultaneously determining tomato maturity status during stalk recognition.
(3) By deploying the improved GCD-YOLO model on the NVIDIA Jetson Orin Nano, effective detection of tomato fruit stalks has been achieved on edge computing platforms with limited computational power and memory capacity.

Furthermore, the stalk detection system was implemented in operational tomato greenhouses for field experiments to evaluate its detection capabilities in actual field environments.

## 2. Materials and methods

To achieve high-precision detection of tomato fruit stalks in actual

field environments, this study improved the YOLOv8n model and validated the performance of the GCD-YOLO model on both PC and Jetson Orin Nano platforms. The overall research flowchart is illustrated in Fig. 1.

### 2.1. Data acquisition

#### 2.1.1. Image acquisition

This study collected a tomato stalk images dataset at different growth stages from the Yinong Agricultural Base (25° 55′ 5″-25° 55′ 17″ N, 119° 28′ 23″–119° 28′ 30" E) in Changle District, Fuzhou City, Fujian Province. The tomato cultivar was Syngenta Spectrum, with mature plant heights of 150–180 cm and a growth cycle of 70–100 days. Images were captured using a device with a resolution of 3024×4032 pixels and stored in JPG format. To simulate real-world robotic operating conditions, the acquisition device was positioned at heights of 0.5 m, 1.0 m, and 1.5 m above ground level. Multi-angle imaging of tomato fruit stalks was performed to enhance model robustness. Data collection was conducted from March to April 2024, yielding 2052 images encompassing diverse lighting conditions, shooting distances, and occlusion degrees. All images were acquired at distinct temporal and spatial intervals to prevent overlapping regions, with a subset exemplifying the dataset illustrated in Fig. 2.

#### 2.1.2. Data augmentation

To enhance the training efficacy and robustness of the network, this study employed data augmentation techniques to expand the sample size. The data augmentation process was implemented using PyCharm 2024 (JetBrains, Prague, The Czech Republic) and image processing tools, incorporating operations such as mirroring, rotation, random noise injection, scaling, and translation. The data augmentation techniques employed in this study were primarily designed based on practical operational conditions in greenhouse environments. Examples of considered factors include variations in lighting and deviations in camera angles. The number of labels before and after data augmentation is shown in Table 1, while representative examples of data augmentation are presented in Fig. 2. This study expanded the original dataset to a total of 6156 images through data augmentation techniques, and divided the augmented dataset into training set (4925 images), validation set (616 images), and test set (615 images) with an 8:1:1 ratio. The number of labels in each subset is shown in Table 1. This study maintained consistency in the label ratio during the training phase to facilitate a more effective assessment of the mode's performance.

Fig. 3

#### 2.1.3. Dataset curation

In this study, the open-source annotation tool MakeSense (https://www.makesense.ai/) was employed to annotate the dataset. Following the execution of the MakeSense script, tomato fruit stalks in each image were labeled with predefined tags: "ripe-stalk" for stalks of mature tomatoes, "unripe-stalk" for stalks of immature tomatoes, while heavily occluded stalks were excluded from annotation. A corresponding "TXT" file was generated for each image, containing the stalk category and normalized coordinates (center-x, center-y, width, height) of bounding boxes. Image augmentation operations preserved the physical positions of stalks but recalculated bounding box coordinates through

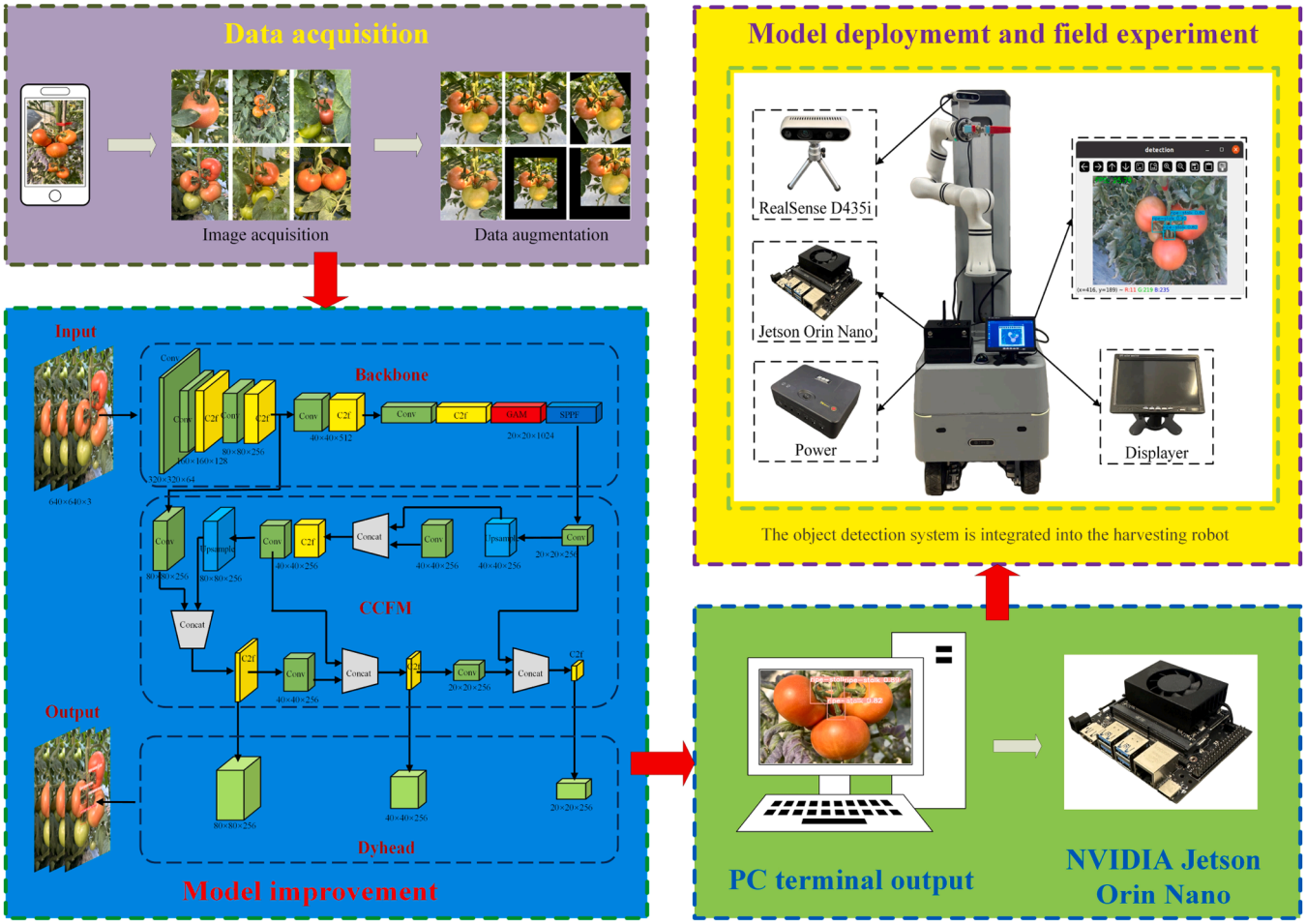


**Fig. 1.** The overall flowchart of the research.


**Fig. 2.** Images of tomato fruit stalks in different environments: (a) single fruit, (b) multiple fruits, (c) occlusion, (d) dark light, (e) normal light, (f) intense light.

**Table 1**
Number of labels in the dataset.

| | Before Augmentation | After Augmentation | Train (80 %) | Val (10 %) | Test (10 %) |
|---|---|---|---|---|---|
| ripe-stalk | 2031 | 6093 | 4850 | 656 | 587 |
| unripe-stalk | 2006 | 6018 | 4788 | 603 | 627 |
| Total | 4037 | 12,111 | 9638 | 1259 | 1214 |

geometric transformations, thereby updating positional data in the "TXT" files for neural network training.

### 2.2. Establishment of the GCD-YOLO model

In the domain of object detection, YOLO-series [24,25] models have been widely adopted due to their single-stage detection framework that balances real-time performance with speed-accuracy trade-offs. Compared to two-stage detectors, YOLO directly outputs target locations and categories through end-to-end inference mechanisms. Leveraging these advantages, this study selected YOLOv8 [26] (https://github.com/ultralytics/ultralytics) as the base architecture, which provides five distinct network variants (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x) with varying network depths and feature map widths. During the harvesting process, real-time detection of tomato pedicels is essential, requiring not only high accuracy but also extremely fast inference speed. Given the hardware constraints of the aforementioned application scenario, our model prioritizes speed and lightweight design. Among the YOLOv8 series, YOLOv8n and YOLOv8s are two models with relatively lower parameter counts and computational complexity.

As shown in **Supplementary Table S1**, YOLOv8n achieves a Precision of 91.1 % at an extremely low computational cost (only 3.001 M Parameters), with an inference speed as high as 140 FPS. In comparison, although larger models can bring improvements of 0.4 % in Recall and 0.6 % in mAP, their computational cost and model size increase by orders of magnitude, accompanied by a significant reduction in inference speed. YOLOv8n strikes the optimal balance between accuracy and efficiency. To enable deployment on edge devices with limited computational resources [27], the parameter-light YOLOv8n [28] was chosen as the baseline model for subsequent accuracy enhancement.

The YOLOv8 architecture comprises four components: Input, Backbone, Neck, and Head. The Backbone extracts multi-scale gradient flow features through Conv and C2f modules, followed by global information fusion via the SPPF [29] module's multi-kernel pooling, which are subsequently transmitted to the Neck. The neck employs a bidirectional fusion strategy integrating FPN (top-down semantic propagation) and PAN (bottom-up localization propagation) to enhance multi-scale contextual awareness. The head adopts a decoupled structure: classification tasks utilize varifocal loss (VFL) to balance small-target weights, while regression tasks combine distribution focal loss (DFL) for rapid localization with complete intersection over union (CIoU) to optimize bounding boxes, ultimately improving detection accuracy and efficiency. The YOLOv8 network model introduces increased parameters and higher computational demands, posing significant challenges for deployment on edge computing devices with limited processing capabilities. To address this challenge, this study proposes three architectural optimizations to YOLOv8: Firstly, global attention mechanism is integrated into the ninth layer of the YOLOv8 backbone network. Secondly, the Neck module is redesigned using cross-scale feature-fusion module. Finally, the original YOLOv8 detection head is replaced with dynamic multi-scale detection head. The optimized model significantly improves the detection accuracy of tomato fruit stalks while maintaining

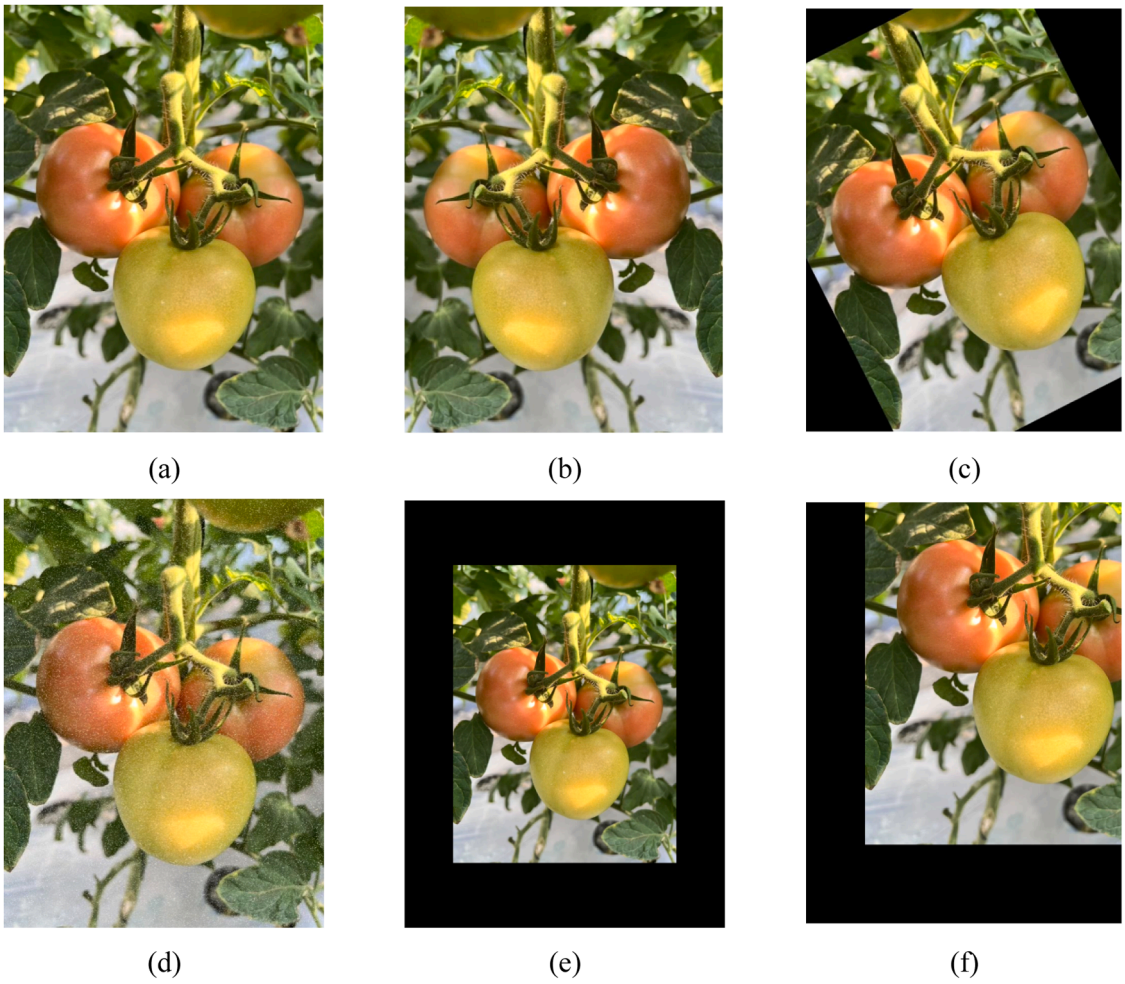


**Fig. 3.** Image after data augmentation: (a) original image, (b) horizontal flip, (c) rotation, (d) Random noise, (e) scaling, (f) image translation.

lightweight characteristics suitable for edge deployment. The architecture of the optimized model is illustrated in Fig. 4.

#### 2.2.1. Global attention mechanism

To enhance feature representation and selection, enabling the model to better focus on critical characteristics of tomato fruit stalks, this study introduces the global attention mechanism module (GAM) [30] into YOLOv8. Building upon the convolutional block attention module (CBAM) [31], GAM redesigns CBAM's submodules using sequential channel-spatial attention mechanisms. GAM primarily consists of a channel attention submodule [32] and a spatial attention submodule [33]. The structural configuration is illustrated in Fig. 5.

In the optimized YOLOv8n model, GAM is positioned at Layer 9 of the Backbone network. Prior to feature fusion, it processes, filters, and enhances targets of varying scales. The operational workflow of GAM is as follows:

$$F_2 = M_C(F_1) \otimes F_2 \tag{1}$$

$$F_3 = M_S(F_2) \otimes F_1 \tag{2}$$

Where $F_1 \in R^{C\times H\times W}$, $C$denotes the number of channels, $H$indicates the image height, $W$represents the image width, $F_1$ corresponds to the input feature map, $F_2$ signifies the intermediate weighted result, $M_C$ and $M_S$denote the channel-wise and spatial output results, and $\otimes$ represents matrix operations.

#### 2.2.2. Cross-scale feature-fusion module

This study optimizes the feature fusion architecture in YOLOv8n using the cross-scale feature-fusion module (CCFM) [34], aiming to reduce model complexity while maintaining high precision, thereby laying the groundwork for subsequent deployment on computationally constrained edge devices. The CCFM structure is depicted in Fig. 3. CCFM incorporates convolutional layers into the fusion path to adjust channel dimensions and integrates bottom-up processing through a Path Aggregation Network (PANet). By extracting multi-scale features, CCFM enables effective complementarity between low-level and high-level features, enhancing the model's capability to recognize tomato fruit stalks in complex agricultural environments.

#### 2.2.3. Dynamic multi-scale detection head

Dynamic multi-scale detection head (Dyhead), proposed by Dai et al. [35], aims to reduce missed detections and false positive rates while enhancing model generalization capability. Its architectural framework is illustrated in Fig. 6. Dyhead integrates scale-aware attention, spatial-aware attention, and task-aware attention, converting the attention mechanism into three sequential modules, each focusing on a specific aspect. For an input feature tensor $F \in R^{L\times S\times C}$, the Dyhead module can be formulated as follows:

$$W(F) = \pi_C(\pi_S(\pi_L(F)\cdot F)\cdot F)\cdot F \tag{3}$$

Where $\pi(x)$ represents the attention function acting on the feature tensor $F$, which has dimensions $L \times S \times C$. Specifically, $\pi_L$represents scale-aware attention, $\pi_S$represents spatial-aware attention, $\pi_C$ represents task-aware attention.

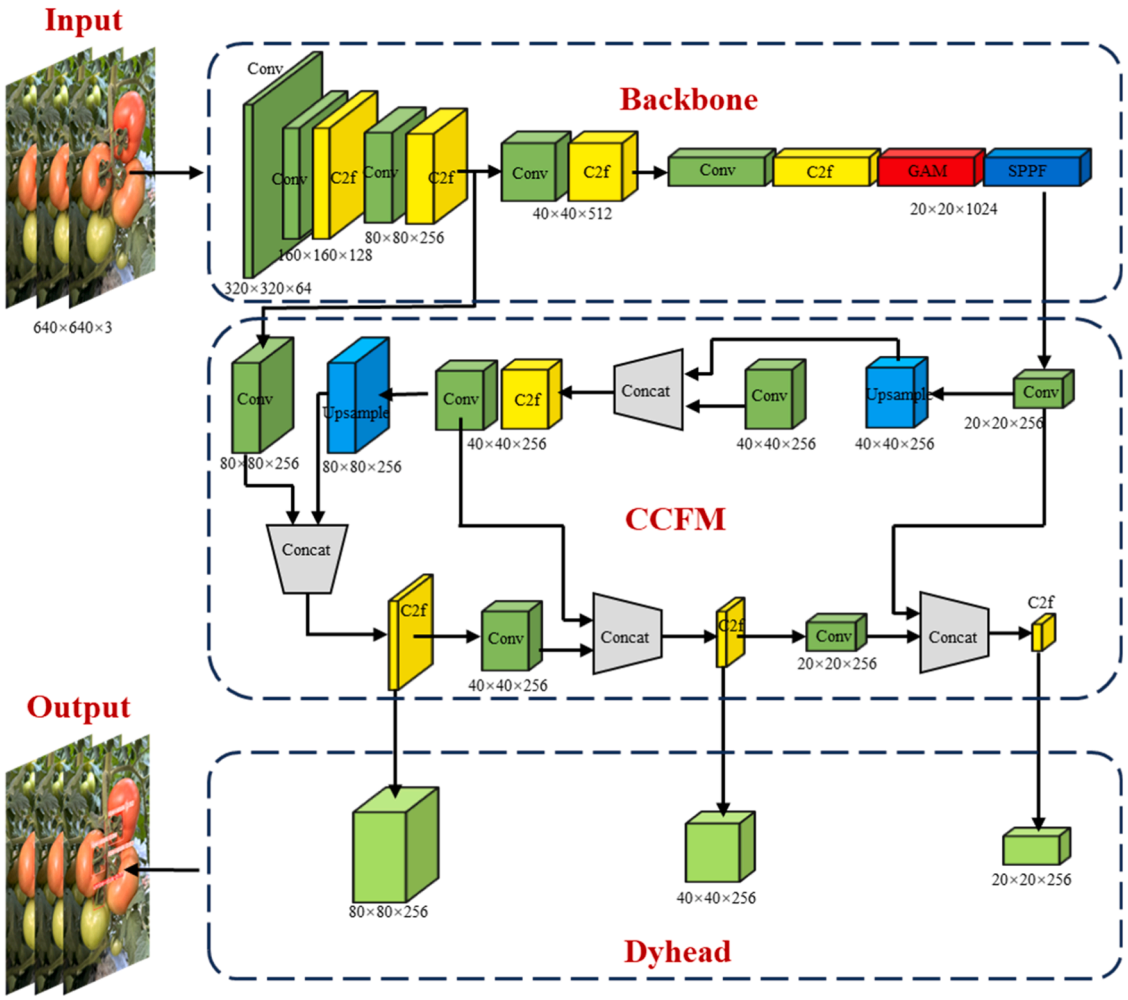


**Fig. 4.** The structure of GCD-YOLO network.
Note: Conv represents the convolution layer, C2f represents cross stage partial fusion with 2 convolutions, GAM represents global attention mechanism, SPPF represents spatial pyramid pooling fast, Upsample represents the upsampling layer, and Concat represents the concatenation layer.

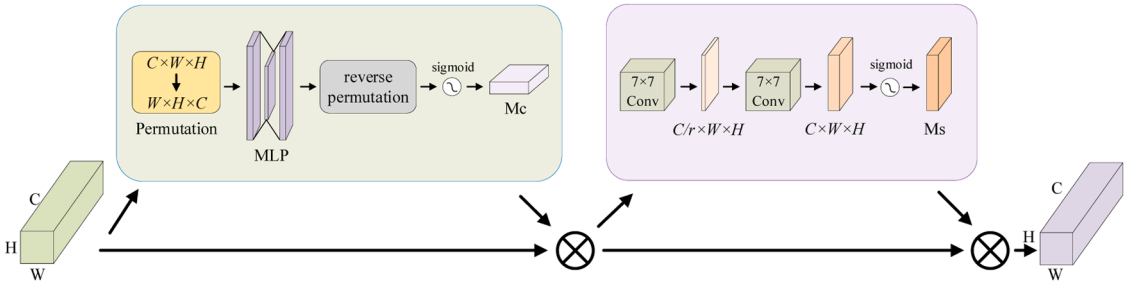


**Fig. 5.** The network structure of GAM.
Note: C, H, and W represent the number of channels, height, and width respectively. Sigmoid represents the sigmoid activation function. Conv 7 × 7 represents the 7 × 7 convolution layer.

### 2.3. Ablation study

This study enhances feature representation and selection by introducing GAM, integrates the CCFM into the backbone network for image feature extraction, and replaces the original YOLOv8 detection head with Dyhead to improve the conventional YOLOv8n. To validate the superiority of these architectural enhancements and evaluate their synergistic effects, ablation studies were conducted to compare and analyze the performance of the detection networks before and after the improvements [36]. Ablation studies, widely adopted in complex neural network research [37], involve systematically removing components to investigate their impact on model performance. The models and corresponding modules evaluated in this ablation study are summarized in Table 2.

### 2.4. Experimental platform and training parameters

All models in this study were trained under identical hardware and software configurations. The hardware setup consisted of an Intel® Core™ i5–12,490 CPU (3.6 GHz), 16 GB DDR4 memory, and an NVIDIA

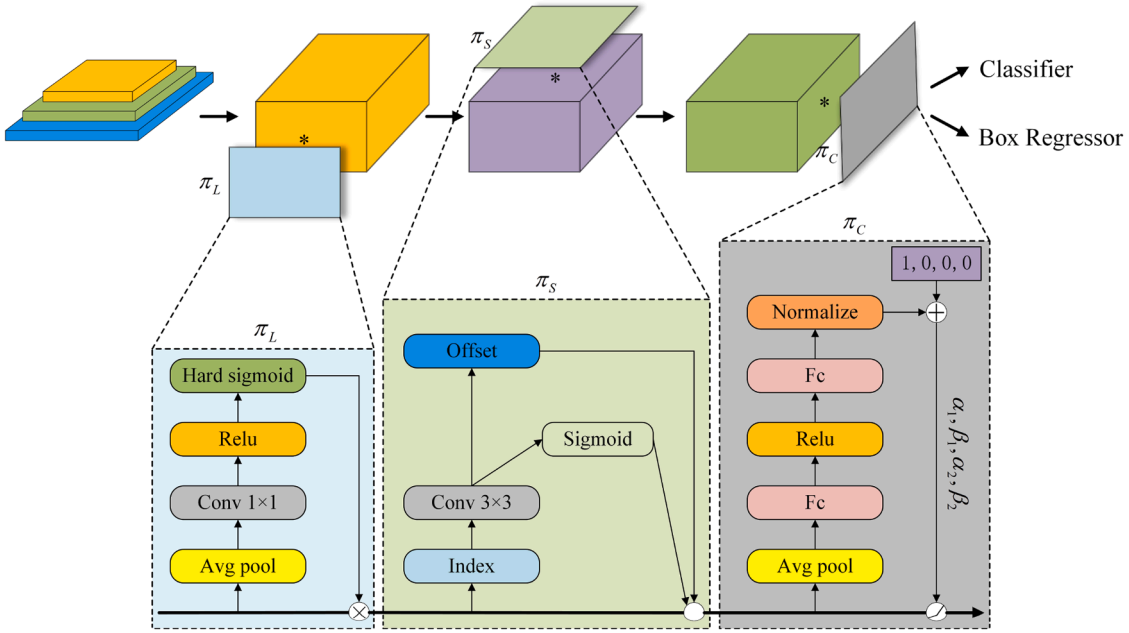


**Fig. 6.** The structure of dynamic multi-scale detection head.
Note: Avg pool represents the average pooling layer, Conv represents the convolution layer, Relu represents the rectified linear unit, Hard sigmoid represents piecewise linear sigmoid approximation, Index represents grid cell coordinate, Offset represents bounding box adjustment value, FC represents the fully connected layer, and Normalize represents the feature standardization.

**Table 2**
Improved models for ablation study.

| Improved models | Module | | | |
|---|---|---|---|---|
| | YOLOv8n | GAM | CCFM | Dyhead |
| Case1 | √ | — | — | — |
| Case2 | √ | √ | — | — |
| Case3 | √ | — | √ | — |
| Case4 | √ | — | — | √ |
| Case5 | √ | √ | √ | — |
| Case6 | √ | √ | — | √ |
| Case7 | √ | — | √ | √ |
| Case8 | √ | √ | √ | √ |

GeForce RTX 4060 graphics card. The deep learning environment was configured with CUDA 11.6, Python 3.8, and PyTorch 1.13.1. The model training parameters were as follows: the initial learning rate of the network was set to 0.01, the weight decay coefficient was 0.0005, and the momentum coefficient was 0.937. The models were trained for 200 epochs with a batch size of 16, selected to enhance detection performance through larger batch processing.

### 2.5. Model deployment

This study established a tomato fruit stalks target detection system based on the NVIDIA Jetson Orin Nano development board. The system consists mainly of the Jetson Orin Nano [38], an Intel RealSense D435i depth camera [39,40], a power module, and a display, as illustrated in Fig. 7. By integrating the stalk detection system into the tomato harvesting robot, this system achieves efficient detection of tomato fruit stalks in actual field environment. The NVIDIA Jetson Orin Nano, a high-performance edge computing device capable of efficient real-time model processing, serves as an ideal embedded platform for edge deployment. TensorRT is a high-performance deep learning inference engine and optimizer developed by NVIDIA that significantly improves the inference speed of models [41]. To validate the effectiveness of the improved GCD-YOLO model on edge devices and advance future tomato harvesting robotics, the optimized model was deployed on the NVIDIA Jetson Orin Nano. Specifically, the trained ".pt" model file was converted to a ".engine" format on the host PC, followed by deployment to the edge device. Detailed deployment environment configurations are provided in Table 3.

### 2.6. Performance evaluation

To evaluate the performance of the model, we selected commonly used evaluation metrics for object detection, including *Precision, Recall, Average Precision*, and *mAP*. The Intersection over Union (IoU) serves as a key criterion for measuring the overlap between predicted bounding boxes and ground truth bounding boxes, and is used to determine the correctness of detection results. Typically, an IoU threshold is set, and a prediction is considered a True Positive (TP) only if the IoU between the predicted box and the ground truth box is not lower than this threshold; otherwise, it is considered a False Positive (FP). *Precision* is defined as the proportion of true positives among all samples predicted as positive, measuring the accuracy of the model's detections. *Recall* represents the proportion of actual positive instances that are correctly detected by the model, reflecting its ability to cover target objects. *Average Precision* is calculated based on the Precision-Recall curve, integrating performance across different confidence thresholds to evaluate the model's overall detection capability for a specific class. The *mAP* is the mean of *AP* values across all classes, providing a comprehensive assessment of the model's performance in multi-class object detection tasks. Additionally, *mAP@50* specifically refers to the *mAP* result calculated at an IoU threshold of 0.5.

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

$$mAP = \frac{1}{k} \sum_{i=1}^{k} \int_{0}^{1} Precision_i d(Recall_i) \tag{9}$$

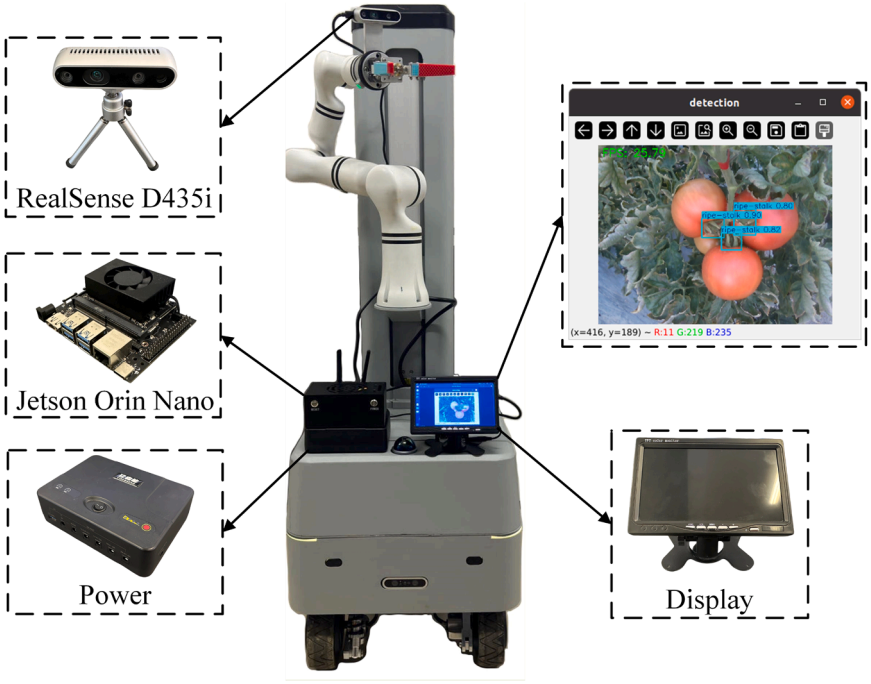


**Fig. 7.** Integration architecture of the stalk detection system within the tomato harvesting robot.

**Table 3**
Deployment environment of NVIDIA Jetson Orin Nano.

| Hardware/Software Environment | Version |
|---|---|
| Development board | NVIDIA Jetson Orin Nano |
| Operating system | Ubuntu 20.04 |
| ultralytics | 8.3.27 |
| python | 3.8.10 |
| torch | 2.0.0 |
| torchvision | 0.15.1 |
| CUDA | 11.4.315 |
| cuDNN | 8.6.0.166 |
| TensorRT | 8.5.2.2 |

To more accurately evaluate the efficiency of the model, metrics such as *Parameters, GFLOPs, Model Size*, and *FPS* were introduced. *Parameters* represent the total number of trainable parameters, indicating model complexity and storage requirements. Giga floating point operations (*GFLOPs*) is a metric quantifying the computational complexity of deep learning models, defined as the total number of floating-point operations (in billions) required to perform one forward pass of the model. Frames per second (*FPS*) measures real-time inference speed, providing practical guidance for algorithm deployment.

## 3. Experimental results and analysis

### 3.1. The k-fold cross-validation experiment

To validate the robustness and generalization capability of the improved model across diverse datasets, this study employed k-fold cross-validation for statistical verification of the proposed algorithm. The dataset was partitioned into k subsets, with (k-1) subsets utilized for training and the remaining subset for validation. This process was iterated k times, where k was set to 9 in this experiment. Detailed experimental results are presented in Table 4. The final model evaluation outcomes were derived from the average of these nine experimental groups. As observed in Table 4, the improved YOLOv8n model achieved a mean precision of 94.1 %, recall of 97.6 %, and mAP@50 of 96.6 % for detecting stalks of mature tomatoes. For stalks of immature tomatoes, the mean precision, recall, and mAP@50 reached 94.7 %, 75.2 %, and 86.8 %, respectively. The results demonstrate that the GCD-YOLO model exhibits stable and superior detection accuracy across diverse test sets compared to the original YOLOv8n, despite minor performance fluctuations caused by dataset partitioning. These acceptable deviations further validate the model's reliability in agricultural applications.

Note: "ripe-stalk" indicates stalks of mature tomatoes, while "unripe-stalk" denotes stalks of immature tomatoes.

**Table 4**
The k-fold cross-validation experiment.

| Class | Group | Precision / % | Recall / % | mAP@50/ % |
|---|---|---|---|---|
| ripe-stalk | 1 | 94.0 | 97.7 | 96.8 |
| | 2 | 93.8 | 97.4 | 95.6 |
| | 3 | 93.9 | 98.0 | 96.6 |
| | 4 | 94.4 | 97.9 | 96.8 |
| | 5 | 94.2 | 96.9 | 96.9 |
| | 6 | 93.6 | 97.6 | 96.5 |
| | 7 | 94.4 | 97.8 | 97.2 |
| | 8 | 94.1 | 97.5 | 96.4 |
| | 9 | 94.5 | 97.7 | 96.5 |
| | **Average** | **94.1** | **97.6** | **96.6** |
| | YOLOv8n | 93.5 | 97.5 | 97.1 |
| unripe-stalk | 1 | 94.7 | 75.3 | 86.8 |
| | 2 | 94.5 | 75.6 | 86.7 |
| | 3 | 95.2 | 75.2 | 87.0 |
| | 4 | 95.0 | 75.4 | 86.8 |
| | 5 | 94.6 | 74.8 | 86.9 |
| | 6 | 94.6 | 74.9 | 86.7 |
| | 7 | 94.7 | 75.1 | 86.2 |
| | 8 | 94.1 | 75.3 | 87.5 |
| | 9 | 94.8 | 75.6 | 86.7 |
| | **Average** | **94.7** | **75.2** | **86.8** |
| | YOLOv8n | 88.7 | 76.8 | 83.4 |

### 3.2. Ablation study on enhanced modules for object detection

To validate the effectiveness of the module improvements, ablation studies were conducted on GAM, CCFM, and Dyhead. The experimental results are presented in Table 5. As shown in the table, although the

**Table 5**
Ablation study experimental results of the GCD-YOLO model.

| | Class | P/ % | | R/ % | | mAP@50/ % | | Parameters(M) | GFLOPs(G) | FPS |
|---|---|---|---|---|---|---|---|---|---|---|
| Case1 (YOLOv8n) | Ripe | 91.1 | 93.5 | 87.1 | 97.5 | 90.3 | 97.1 | 3.011 | 8.2 | 140.0 |
| | Unripe | | 88.7 | | 76.8 | | 83.4 | | | |
| Case2 | Ripe | 92.8 | 93.7 | 86.8 | 96.5 | 91.9 | 97.4 | 3.037 | 8.2 | 131.6 |
| | Unripe | | 91.9 | | 77.1 | | 86.4 | | | |
| Case3 | Ripe | 89.6 | 93.7 | 84.2 | 95.3 | 90.0 | 97.3 | **1.969** | 6.7 | **164.7** |
| | Unripe | | 85.5 | | 73.2 | | 82.6 | | | |
| Case4 | Ripe | 92.4 | 94.1 | **87.9** | 97.6 | 91.8 | 96.3 | 2.509 | 7.8 | 102 |
| | Unripe | | 90.6 | | **78.1** | | 87.3 | | | |
| Case5 | Ripe | 89.9 | 94 | 85.5 | 96.2 | 90.8 | **97.5** | 1.991 | **6.6** | 157.7 |
| | Unripe | | 85.7 | | 74.7 | | 84.1 | | | |
| Case6 | Ripe | 90.8 | 92.7 | 86.2 | 97.5 | 91 | 96.6 | 2.532 | 7.9 | 96.0 |
| | Unripe | | 88.9 | | 75 | | 85.4 | | | |
| Case7 | Ripe | 92.9 | 93.8 | 87.6 | **98.1** | **92.0** | 96.8 | 2.362 | 7.5 | 109.3 |
| | Unripe | | 91.9 | | 77.1 | | **87.3** | | | |
| Case8 | Ripe | **94.4** | **94.1** | 86.4 | 97.6 | 91.7 | 96.6 | 2.389 | 7.5 | 105.4 |
| (GCD-YOLO) | Unripe | | **94.7** | | 75.2 | | 86.8 | | | |

Note: “ripe-stalk” indicates stalks of mature tomatoes, while “unripe-stalk” denotes stalks of immature tomatoes.

original YOLOv8 model (Case1) exhibits higher FPS compared to the improved model, it demonstrates no advantage in detection accuracy for tomato fruit stalks. However, with the incorporation of GAM, CCFM, and Dyhead, the optimized YOLOv8 model (Case8) achieves significant improvements in fruit stalk detection accuracy. Specifically, the GCD-YOLO model increases detection precision by 0.6 % for the stalks of mature tomatoes and by 6.0 % for those of immature tomatoes. In the model Case2, detection precision for the stalks of mature and immature tomatoes improves by 0.2 % and 3.2 %, respectively, while maintaining nearly identical computational complexity compared to the baseline. Compared to the original YOLOv8 model, the model Case3 exhibits a reduction in detection accuracy for the stalks of immature tomatoes. However, it achieves significant efficiency improvements with 34.6 % fewer parameters, 18.2 % lower GFLOPs, and a 24.7FPS increase in inference speed. In the model Case4, while achieving 0.6 % and 1.9 % improvements in detection accuracy for the stalks of mature and immature tomatoes, respectively, the architecture suffers a 38.0FPS reduction in inference speed compared to the baseline. Analysis of experimental results from Cases6-8 indicates that the incorporation of the Dyhead module increases model inference time, thereby reducing the FPS compared to the baseline architecture. However, the model with Dyhead achieves a significant improvement in tomato stalk detection accuracy relative to the original YOLOv8n framework. Furthermore, experimental results from Case5 demonstrate that removing the Dyhead module from the GCD-YOLO model reduces the precision for the stalks of immature tomatoes to 85.7 %, marking an 8.4 % decrease compared to the YOLOv8 framework. Compared to the model Case8, the model Case6 exhibits a slight decrease in detection precision for the stalks of tomatoes and a significantly lower inference speed of only 96.0FPS. In comparison with the GCD-YOLO model, the model Case7 improves FPS while reducing the parameter count and decreasing detection precision for tomato fruit stalks. To further validate the effectiveness of the proposed GCD-YOLO model, eight different models were evaluated under various environmental conditions, with the detection results illustrated in Fig. 8.

As illustrated in Fig. 8, the baseline model (YOLOv8n) exhibits missed detections of tomato fruit stalks under normal lighting conditions, indicating that its foundational feature extraction network lacks sufficient sensitivity to small, slender targets. After incorporating the GAM attention module (Case 2), the model shows false detections by misidentifying leaves as fruit stalks in occluded environments, suggesting that while the module enhances global contextual awareness, it also amplifies background interference that shares partial similarities with the target. The CCFM module (Case 3) results in duplicate detections of the same mature tomato fruit stalk under strong lighting, implying that its multi-scale feature fusion mechanism is prone to overfitting feature responses under extreme illumination. Although the Dyhead module (Case 4) is designed to improve the detection head’s adaptability to objects at various scales, its standalone use still fails to address missed detections of immature tomato stalks caused by insufficient foundational features. Results from Cases 5–7 demonstrate that pairwise combinations of the modules remain inadequate for effective detection of tomato fruit stalks. For instance, the combination of GAM and CCFM (Case 5) still yields duplicate detections under bright light; the integration of GAM and Dyhead (Case 6) continues to fail in suppressing false positives on background clutter in occluded settings; and the combination of CCFM and Dyhead (Case 7) still does not effectively resolve missed detections of immature targets. Ultimately, the GCD-YOLO model (Case 8), which integrates GAM, CCFM, and Dyhead, achieves accurate detection across all tested environmental conditions. Furthermore, compared to the YOLOv8n baseline, the GCD-YOLO model shows significantly higher confidence scores in detecting fruit stalks. The effective detection of the GCD-YOLO model stems from synergistic interactions among the modules: GAM localizes potential target regions, CCFM provides discriminative features integrating multi-scale context, and Dyhead dynamically adjusts the detection strategy based on image content, effectively suppressing both false positives and false negatives. These results fully validate the practicality and reliability of the proposed model in complex agricultural environments.

### 3.3. Visualization and recognition performance analysis of Grad-CAM

Grad-CAM (gradient-weighted class activation mapping) is a visualization method that highlights image regions deemed crucial by the model through gradient computation, without requiring network architecture modifications or retraining [42]. All visualization results in this study were generated based on the final C2f layer within the neck network. This layer is situated immediately before the detection head, and its output feature maps directly determine the final bounding boxes and confidence scores. Selecting this layer for visualization most directly reflects the key regions that the model focuses on when making its final predictions. The heatmap results of the YOLOv8n and GCD-YOLO models are illustrated in Fig. 9. The heatmap images employ color gradients to represent feature weights, where red indicates the highest weights, yellow intermediate values, and blue the lowest weights. In the heatmaps generated by the YOLOv8n model, the attention is predominantly focused on partial regions of tomato fruit stalks and irregularly distributed attention points. Although the model captures some features of the stalks, the overall attention regions are scattered and fail to concentrate on critical characteristics of the fruit stalks, resulting in inferior detection performance for YOLOv8n. As demonstrated in the detection results, YOLOv8n exhibits lower confidence scores for tomato

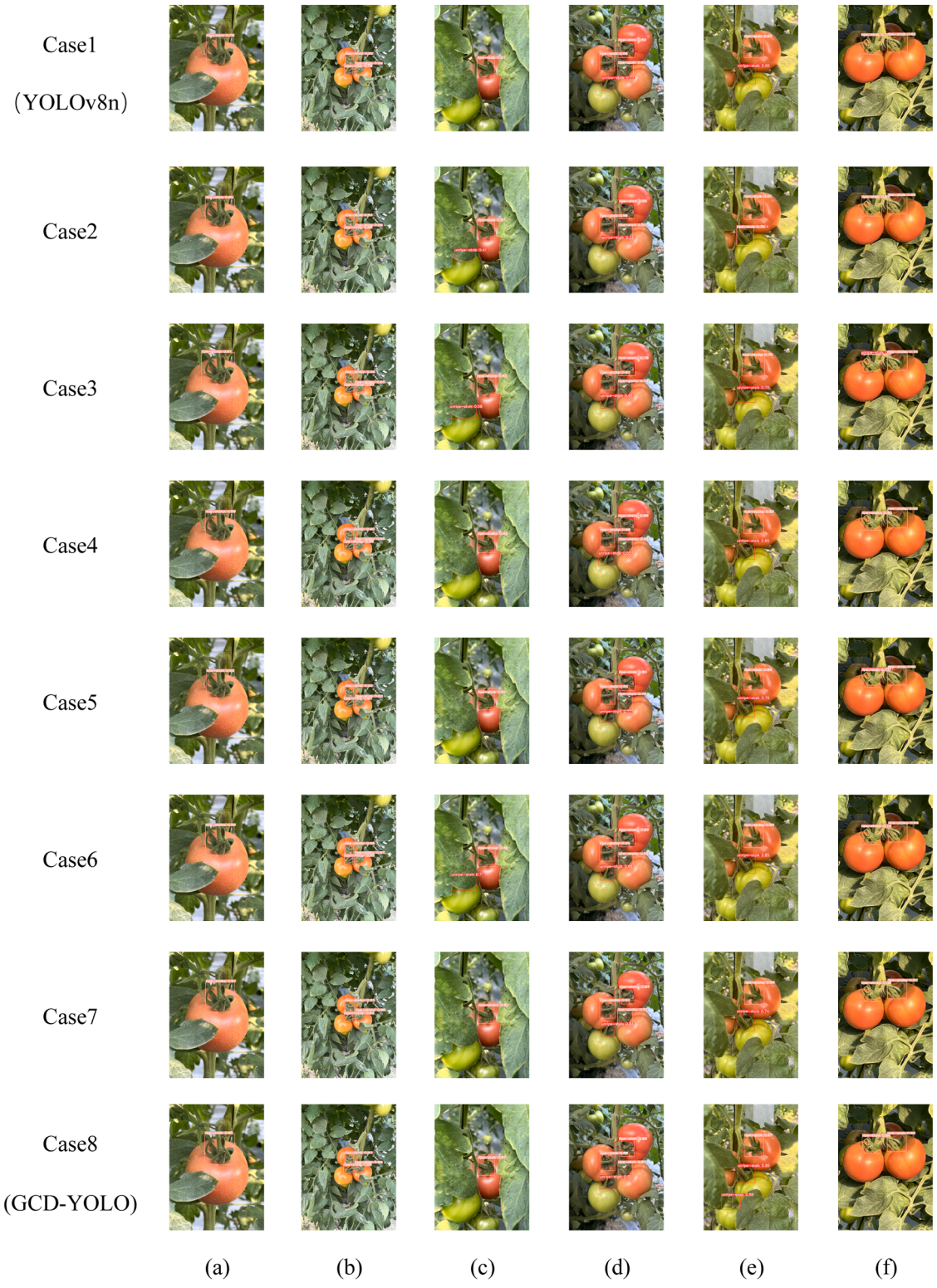


**Fig. 8.** Detection results of eight models in different environments: (a) Single fruit, (b) Multiple fruits, (c) Occlusion, (d) Dark light, (e) Normal light, (f) Intense light.

stalk detection. For instance, in single fruit scenarios, YOLOv8n achieves a confidence score of 78 %, whereas our model attains 85 %. Under occlusion scenarios, the GCD-YOLO model yields an 87 % confidence score, surpassing YOLOv8n by 4 %. Furthermore, under normal light scenarios, YOLOv8n fails to fully identify all tomato fruit stalks, with missed detection instances observed. In contrast, the heatmap results from the GCD-YOLO model demonstrate that our model focuses more on the central region of the stalk, covering more critical feature areas with broader and more unified attention distribution. Meanwhile, it pays relatively less attention to irrelevant information, enabling our model to exhibit higher confidence scores and superior detection performance.

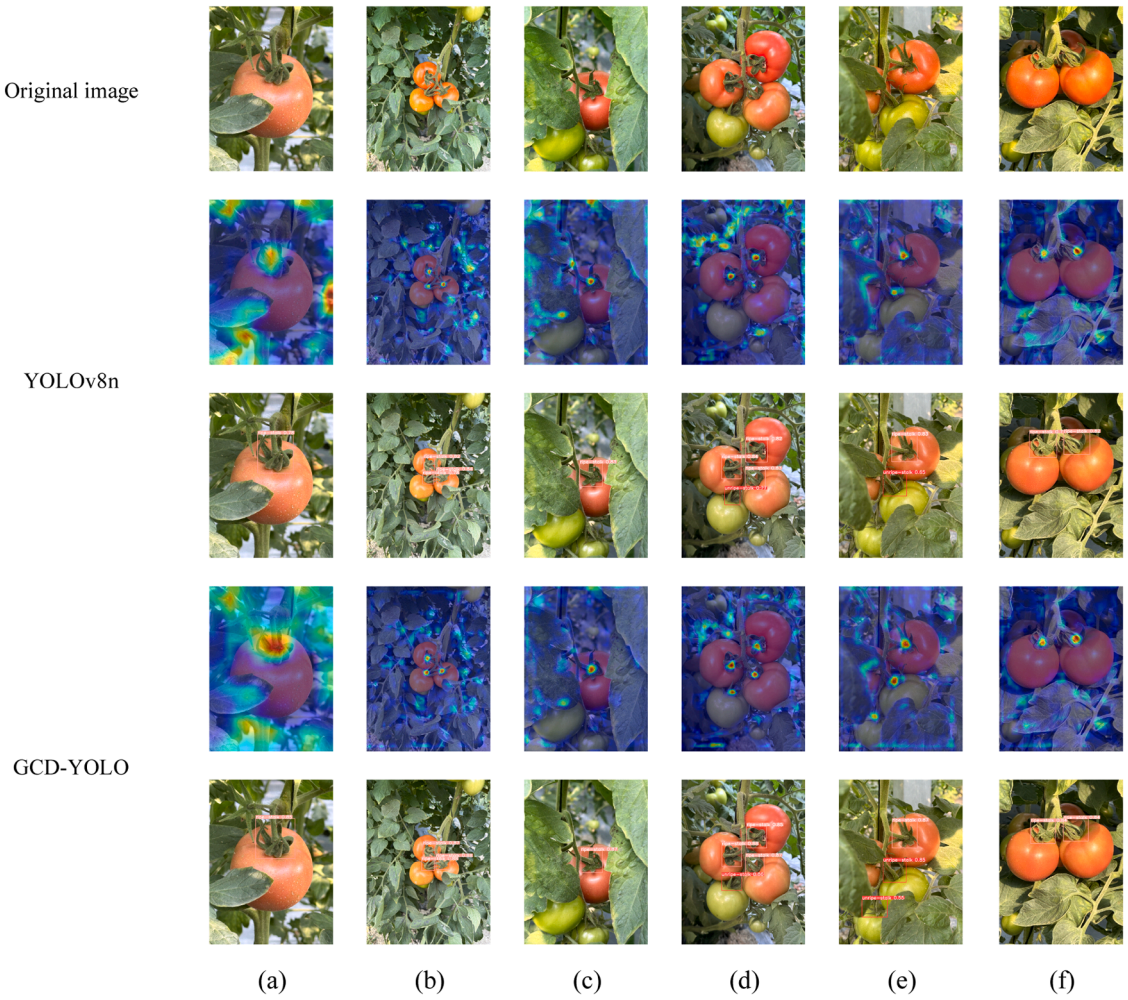


**Fig. 9.** Comparative analysis of Grad-CAM visualization results and recognition performance between GCD-YOLO and YOLOv8n in different environments: (a) single fruit, (b) multiple fruits, (c) occlusion, (d) dark light, (e) normal light, (f) intense light.

### *3.4. Comparative analysis among different object detection algorithms*

To demonstrate the advantages of the improved model in detecting tomato fruit stalks, comparative experiments were conducted on the tomato fruit stalks dataset, where the GCD-YOLO model was evaluated against SSD [43], Faster R-CNN [44], RT-DETR [45], YOLOv5n [46], YOLOv9t [47], YOLOv11n [48] and YOLOv12n [49]. The experimental results are detailed in Table 6.

From Table 5, it can be observed that GCD-YOLO demonstrates significant advantages across multiple key metrics. Particularly in terms of detection accuracy, its precision reaches 94.4 %, markedly outperforming other comparative models. Specifically, it exceeds Faster R-CNN, RT-DETR, YOLOv5n, YOLOv11n and YOLOv12n by margins of 31.5 %, 17.6 %, 4.9 %, 2.7 % and 3.0 %, respectively, which fully reflects its strong capability in accurately identifying tomato fruit stalks in complex agricultural environments. Meanwhile, GCD-YOLO achieves the best performance in mean average precision (mAP@50) at 91.7 %, while maintaining high precision with a substantially reduced parameter count (2.389 M) and computational cost (7.5 GFLOPs). With a model size of only 4.8 MB, it exhibits excellent lightweight characteristics. In terms of inference speed, GCD-YOLO reaches 105.4 FPS, far surpassing two-stage models such as Faster R-CNN (12.0 FPS) and Transformer-based RT-DETR (35.8 FPS), while also outperforming similarly scaled models like YOLOv9t (79.8 FPS), thereby meeting the requirements for real-time detection in agriculture. Given the limitations of real-time computation in tomato harvesting robots, superior real-time detection performance is essential. GCD-YOLO achieves an optimal balance among accuracy, efficiency, and lightweight design,

**Table 6**
Comparative experimental results between the GCD-YOLO model and other object detection algorithms.

| | P/ % | R/ % | mAP@50/ % | Parameters (M) | GFLOPs (G) | Model Size/MB | FPS |
|---|---|---|---|---|---|---|---|
| SSD | 84.0 | 62.7 | 64.5 | 26.301 | 62.7 | 91.1 | 57.7 |
| Faster R-CNN | 62.9 | 82.7 | 89.5 | 137.142 | 370.2 | 108.0 | 12.0 |
| RT-DETR | 76.8 | 86.7 | 83.6 | 31.987 | 103.4 | 66.1 | 35.8 |
| YOLOv5n | 89.5 | **87.4** | 91.6 | 2.503 | 7.1 | 5.0 | **136.6** |
| YOLOv9t | 89.4 | 86.6 | 90.7 | **1.971** | 7.6 | **4.4** | 79.8 |
| YOLOv11n | 91.7 | 85.9 | 91.5 | 2.624 | **6.6** | 5.19 | 130.1 |
| YOLOv12n | 91.4 | 86.0 | 91.6 | 2.602 | 6.7 | 5.5 | 103.1 |
| GCD-YOLO | **94.4** | 86.4 | **91.7** | 2.389 | 7.5 | 4.8 | 105.4 |

making it more suitable for deployment on tomato harvesting robots and ensuring rapid and efficient detection of tomato fruit stalks.

Fig. 10 demonstrates that during the training process on the tomato fruit stalk dataset, the precision, recall, and mAP@50 curves of various detection models exhibit similar trends. These curves gradually rise to their peaks and then fluctuate near these maximum values. In the precision curve, the GCD-YOLO model achieves the highest value, followed by YOLOv11n and YOLOv12n, which is consistent with the conclusions shown in Table 5. Among all models, the precision, recall, and mAP@50 curves of the SSD, Faster R-CNN, and RT-DETR models show the most significant fluctuations. Analysis of the mAP@50 curves indicates that the YOLOv5n, YOLOv9t, YOLOv11n, YOLOv12nand GCD-YOLO models tend to converge around the 50th epoch. In contrast, Faster R-CNN, SSD, and RT-DETR converge more slowly, stabilizing only around the 150th epoch. It is worth noting that in both the precision and mAP@50 curves, the GCD-YOLO model consistently maintains the highest values and demonstrates exceptional stability. These results further validate the feasibility of the improved GCD-YOLO algorithm proposed in this study.

Furthermore, the detection performance of the GCD-YOLO model was evaluated against other models under diverse environmental conditions, as shown in Fig. 11. As illustrated in Fig. 11, under occlusion scenarios with dense foliage growth, Faster R-CNN misidentifies green leaves as stalks of immature tomatoes and splits a single stalk of mature tomatoes into two false detections, while RT-DETR and YOLOv9t exhibit similar errors by incorrectly classifying foliage as tomato fruit stalks, thereby introducing erroneous detection results. The SSD model exhibits missed detection of stalks of immature tomatoes under dark light scenarios. Under normal light conditions, SSD, RT-DETR, YOLOv5n, YOLOv9t and YOLOv11n similarly demonstrate failure to detect immature tomato fruit stalks, likely due to limited model generalization capabilities. Additionally, SSD introduces misclassification errors by falsely identifying stalks of immature tomatoes as mature ones, further degrading detection reliability. As demonstrated in the detection results under intense light scenarios, YOLOv9t incorrectly splits a single stalk of mature tomato into two erroneous detections: one stalk of mature tomato and one stalk of immature tomato. In this study, only the GCD-YOLO and YOLOv12n models achieved accurate detection of tomato fruit stalks across all six environments. However, the GCD-YOLO model demonstrated higher detection confidence and more precise bounding box predictions for the peduncles. Significantly, GCD-YOLO demonstrates substantially higher confidence scores for stalk identification compared to both YOLOv5n, YOLOv9t, YOLOv11n and YOLOv12n. While SSD and Faster R-CNN occasionally exhibit higher confidence scores than GCD-YOLO in specific scenarios, their detection reliability proves highly inconsistent. For example, under intense light and single-fruit conditions, SSD and Faster R-CNN achieve near-perfect confidence scores close to 100 % for stalk detection. However, in multi-fruit environments, SSD's confidence plummets to approximately 0.60, markedly lower than GCD-YOLO's robust performance. Moreover, the Faster R-CNN and RT-DETR model exhibit recurrent erroneous detections across all six tested environmental conditions, significantly undermining their detection reliability. Comparative analysis demonstrates that GCD-YOLO outperforms other models with exceptional performance, effectively adapting to varying greenhouse lighting conditions and providing reliable technical support for in-situ tomato harvesting robots.

### 3.5. Field experiment

To validate the effectiveness of the model on edge devices, this study deployed the GCD-YOLO model on the NVIDIA Jetson Orin Nano platform and evaluated its recognition performance across multiple environments. Field experiments for tomato fruit stalks recognition were conducted on March 8, 2025, at Yinong Farm in Changle District, Fuzhou City, Fujian Province. The greenhouse environment maintained a temperature of 16 °C, with tomato plants of the Syngenta Spectrum variety cultivated at 0.15 m intra-row spacing and 0.6 m inter-row spacing. Mature tomato plants averaged 1.7 m in height. Experimental results are illustrated in Fig. 12.

This study selected tomato fruit stalks under six different environmental conditions as the identification targets. Experimental results demonstrate that the GCD-YOLO model accurately identified tomato fruit stalks across all six environments with no missed detections. When deployed on the Jetson edge device, the model achieved an average detection speed of 26.24 FPS, meeting the requirements for rapid detection. Moreover, it maintained high confidence levels under varying lighting conditions and complex scenarios, demonstrating excellent robustness. Through practical validation in greenhouse environments for tomato fruit stalks recognition, the model exhibited strong generalization capability and practicality, providing reliable technical support for precise cutting of tomato fruit stalks by harvesting robots.

## 4. Discussion

To address the challenges of labor-intensive and costly tomato harvesting in greenhouses, this study developed a tomato fruit stalks detection model based on an improved YOLOv8 architecture. The enhancements include integrating GAM into the YOLOv8 backbone network, replacing the original Neck with CCFM, and adopting Dyhead. The optimized model was tested in PyCharm for performance evaluation and deployed on edge devices. Field validations in actual field environments confirmed its robustness, providing critical technical support for advancing tomato harvesting robotics in stalk-cutting applications.

In this study, our proposed method successfully achieved rapid and efficient detection of tomato fruit stalks in field environments. Comparative analysis between the original YOLOv8n and the proposed GCD-YOLO model on the tomato fruit stalks dataset demonstrates a

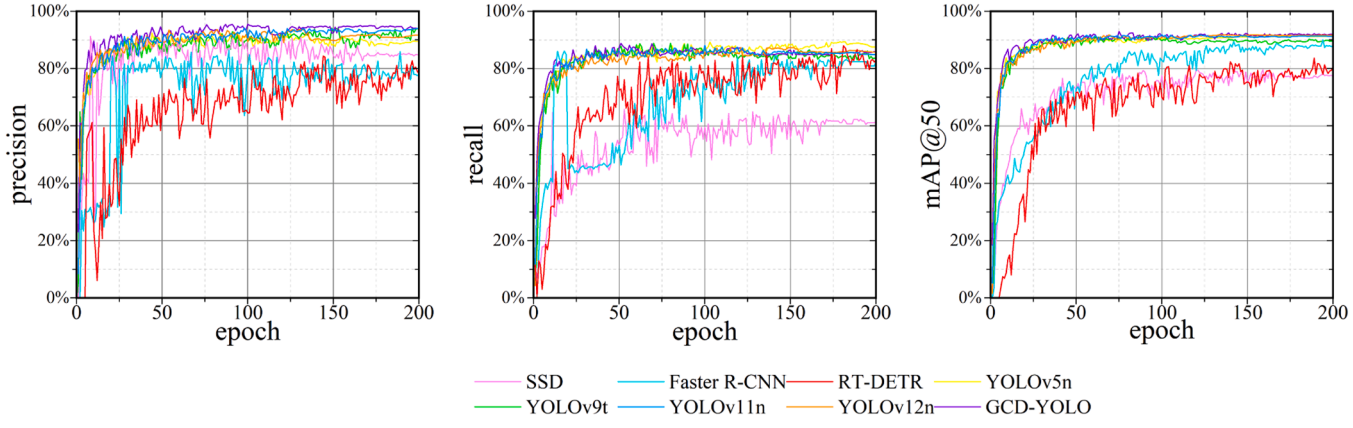


**Fig. 10.** The comparison results with the performance curve of the classical target detection model.


**Fig. 11.** Detection results of GCD-YOLO and other models in different environments: (a) single fruit, (b) multiple fruits, (c) occlusion, (d) dark light, (e) normal light, (f) intense light.

significant improvement in detection accuracy. The GCD-YOLO architecture achieves a 94.1 % precision for stalks of mature tomatoes and 94.7 % for stalks of immature tomatoes, enabling accurate identification of tomato fruit stalks with exceptional robustness across diverse agricultural scenarios. The ablation study results demonstrate that the introduction of the GAM module (Case 2) improved the detection accuracy of tomato fruit stalks by 3.3 % through enhanced global feature representation, with only a minimal increase in model complexity. However, as shown in Fig. 8, while improving sensitivity, this module also amplifies background interference, leading to false detections such as misidentifying leaves as fruit stalks in occluded scenarios. The CCFM module (Case 3), employing a multi-scale feature fusion strategy,

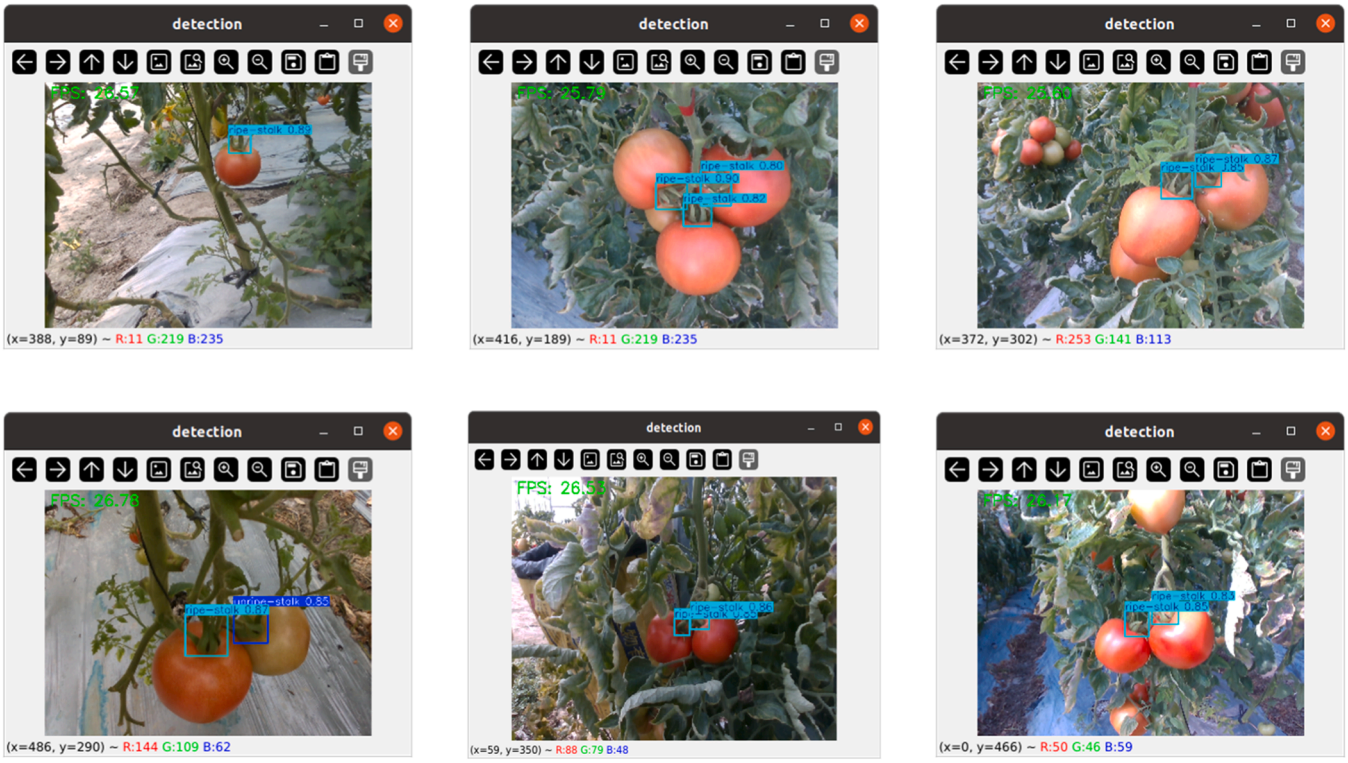


**Fig. 12.** Field experiment results of different detection models in different environments: (a) single fruit, (b) multiple fruits, (c) occlusion, (d) dark light, (e) normal light, (f) intense light.

significantly reduced model complexity (with a 34.6 % decrease in parameters and an 18.2 % reduction in GFLOPs) while increasing inference speed to 164.7 FPS. Nevertheless, detection results in Fig. 5 reveal that it tends to over-respond under strong lighting conditions, resulting in duplicate detections. The Dyhead module (Case 4), by dynamically adjusting the detection mechanism, improved the mAP for fruit stalk detection by 1.6 %, but its computational overhead caused a frame rate drop of 38 FPS. Comparisons among Cases 5, 6, 7, and 8 indicate that although the GAM+CCFM combination significantly enhanced feature fusion capability, the lack of Dyhead's dynamic reasoning ability led to a decrease in detection accuracy for immature tomato fruit stalks to 85.7 %. The GAM+Dyhead combination (Case 6) improved accuracy but substantially reduced inference efficiency (FPS dropped to 96). The CCFM+Dyhead combination (Case 7), while balancing model complexity and accuracy, still failed to overcome the limitations of basic feature extraction. In contrast, the combined GAM+CCFM+Dyhead model demonstrated superior discriminative performance compared to pairwise combinations of the modules. It achieved a tomato fruit stalk detection accuracy of 94.4 % while maintaining high inference efficiency (105.4 FPS). It is hypothesized that this improvement arises from the synergistic effects of the modules: GAM enhances focus on critical features of the fruit stalks, CCFM provides multi-scale contextual feature representation and contributes to model lightweighting, while Dyhead dynamically optimizes the detection decision process, reducing the rate of missed and false detections.

Furthermore, Grad-CAM visualization analysis reveals that compared to the YOLOv8n model, our GCD-YOLO achieves more precise feature extraction by focusing on the central regions of fruit stalks, broadly covering key features while suppressing irrelevant interference. Comparative experiments with other detection algorithms (SSD, Faster R-CNN, RT-DETR, YOLOv5n, YOLOv9t, YOLOv11n and YOLOv12n) demonstrate GCD-YOLO's accurate detection across all six complex scenarios. Baseline models exhibited either missed detections (SSD, YOLOv5n, YOLOv9t and YOLOv11n) or false positives (e.g., Faster R-CNN and RT-DETR misidentifying leaves as stalks). With a compact model size of 4.8 MB and high-frame-rate performance, GCD-YOLO fulfills the dual requirements of lightweight deployment and rapid response for tomato harvesting robots, effectively addressing multi-scenario failures caused by insufficient generalization in conventional models. The improved GCD-YOLO model was deployed on the edge device Jetson Orin Nano, where experimental results under six distinct environmental conditions confirmed its ability to correctly identify tomato fruit stalks without any missed detections. Additionally, the model maintained an average detection speed of 26.24 FPS on the Jetson edge device with high confidence levels, demonstrating excellent robustness.

In this study, the improved GCD-YOLO model successfully identified tomato fruit stalks in actual field environments by integrating color features of both the stalks and tomato fruit surfaces. K-fold cross-validation results demonstrated an average accuracy of 94.7 % for stalk recognition, highlighting the model's strong detection capabilities. Recent advancements in agricultural vision technologies have significantly improved fruit stalk recognition accuracy. Xu et al. [50] integrated RGB and depth data to enhance Mask R-CNN for cherry tomato stalk detection, achieving 89.34 % accuracy. Fu et al. [51] proposed the YOLO-Banana model, attaining 85.98 % mean precision for banana stem detection. Zhang et al. [52] incorporated a mango segmentation subtask into an improved YOLOv5s architecture, developing the YOLOMS multi-task model with 89.84 % accuracy for mango stalk recognition. In this study, the GCD-YOLO algorithm achieved an average accuracy of 94.4 % for tomato fruit stalks recognition, representing improvements of 5.06 %, 8.42 %, and 4.56 % over Mask R-CNN, YOLO-Banana, and the improved YOLOv5s methods, respectively, thereby demonstrating superior recognition capabilities. Notably, while prior studies validated models only on PCs, our approach not only demonstrates robust performance on dataset but also enables high-frame-rate tomato fruit stalks detection on edge devices. With its streamlined architecture and rapid inference speed, GCD-YOLO offers superior practicality for field-deployed agricultural robotics.

This study demonstrates that the improved GCD-YOLO model achieves high accuracy and speed in tomato fruit stalks recognition. However, the widespread adoption of machine vision technologies for crop identification faces two primary limitations. First, a primary challenge lies in the time-consuming process of dataset construction. Under manual annotation workflows, acquiring high-quality, large-scale datasets within short timeframes remains infeasible. Although automated annotation tools are available, they demonstrate slow processing speeds and low annotation accuracy for image datasets requiring extensive labeling or small-target annotations. To address this bottleneck, developing self-supervised adaptive object recognition algorithms is imperative. Future efforts will integrate advanced techniques such as reinforcement learning and multi-modal fusion to achieve annotation-efficient model training while maintaining detection robustness. Second, addressing the challenge of severe target occlusion causing machine vision failures, a multi-sensor collaborative system could enable robust detection of occluded targets through 3D spatial reconstruction and multi-source data fusion. Furthermore, in subsequent research, we will integrate the enhanced detection model into a tomato harvesting robotic system to achieve automated picking in greenhouses, advancing the sustainable development of smart agriculture through precision farming technologies.

## 5. Conclusion

This study proposes GCD-YOLO, a tomato fruit stalks detection model for unstructured and actual field environments, leveraging machine vision technologies. The following conclusions are drawn from this research:

(1) Integrating GAM, CCFM, and Dyhead modules into the YOLOv8n architecture significantly enhances the detection performance of the tomato fruit stalks recognition model. Compared to the original YOLOv8n, GCD-YOLO achieves a 3.3 % improvement in detection accuracy on the tomato fruit stalks dataset.
(2) Grad-CAM visualization and recognition efficacy analysis demonstrate that each enhancement module effectively strengthens the model's feature extraction capabilities, thereby improving GCD-YOLO's detection accuracy for tomato fruit stalks. Comparative experiments with other object detection models (SSD, Faster R-CNN, RT-DETR, YOLOv5n, YOLOv9t, YOLOv11n and YOLOv12n) reveal that GCD-YOLO consistently maintains superior detection performance, with a detection precision of 94.4 % and an mAP@50 of 91.7 % for tomato fruit stalks.
(3) Field deployment of the GCD-YOLO model on the NVIDIA Jetson Orin Nano edge device demonstrates its practical efficacy. GCD-YOLO accurately identifies tomato fruit stalks across six environmental conditions while achieving an average detection speed of 26.24FPS, fulfilling operational requirements for agricultural robotics.

The GCD-YOLO model proposed in this study effectively enables detection of tomato fruit stalks in unstructured environments. It not only meets the operational requirements for tomato harvesting robots but also provides a reference framework for the development of vision systems in other harvesting robots, thereby promoting the automation and high-quality advancement of robotic harvesting technologies.

## Ethical statement for smart agricultural technology

I testify on behalf of all co-authors that our article submitted to Smart Agricultural Technology:

1) This manuscript has not been published in whole or in part elsewhere;
2) The manuscript is not currently being considered for publication in another journal;
3) All authors have been personally and actively involved in substantive work leading to the manuscript, and will hold themselves jointly and individually responsible for its content.

## CRediT authorship contribution statement

**Wuxiong Weng:** Writing – review & editing, Supervision, Resources, Methodology, Formal analysis, Conceptualization. **Zhenhui Lai:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. **Zheming Cui:** Validation, Supervision, Project administration. **Zhixiong Chen:** Visualization, Supervision, Software. **Hongbin Chen:** Visualization, Validation, Supervision. **Tianliang Lin:** Writing – review & editing, Supervision, Resources. **Jufei Wang:** Writing – review & editing, Supervision, Software. **Shuhe Zheng:** Writing – review & editing, Resources, Project administration, Funding acquisition. **Guoqing Chen:** Writing – review & editing, Methodology, Formal analysis, Conceptualization.

## Declaration of competing interest

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

## Acknowledgements

This work was supported by Open Foundation of Fujian Key Laboratory of Green Intelligent Drive and Transmission for Mobile Machinery [grant number GIDT-202308].

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.atech.2025.101465.

## Data availability

Data will be made available on request.

## References

[1] Y. Mu, T. Chen, S. Ninomiya, W. Guo, Intact detection of highly occluded immature tomatoes on plants using deep learning techniques, Sensors 20 (10) (2020) 2984, https://doi.org/10.3390/s20102984.
[2] H. Zhou, X. Wang, W. Au, H. Kang, C. Chen, Intelligent robots for fruit harvesting: recent developments and future challenges, Precis. Agric. 23 (2022) 1856–1907, https://doi.org/10.1007/s11119-022-09913-3.
[3] G. Haidar, M. Deen, R. Achkar, M. Owayjan, R. Daou, Automated tomato inspection and harvesting system using robotic arm and computer vision in greenhouses, in: 2023 Fifth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 2023, pp. 183–1899, https://doi.org/10.1109/ACTEA58025.2023.10194232.
[4] T. Jin, X. Han, Robotic arms in precision agriculture: a comprehensive review of the technologies, applications, challenges, and future prospects, Comput. Electron. Agric. 221 (2024) 108938, https://doi.org/10.1016/j.compag.2024.108938.
[5] J. Wang, G. Chen, J. Ju, T. Lin, R. Wang, Z. Wang, Characterization and classification of urban weed species in northeast china using terrestrial hyperspectral images, Weed Sci 71 (4) (2023) 353–368, https://doi.org/10.1017/wsc.2023.36.
[6] B. Rasti, P. Ghamisi, J. Plaza, J. Plaza, A. Plaza, Fusion of hyperspectral and LiDAR data using sparse and low-rank component analysis, IEEE Trans. Geosci. Remote Sens. 55 (11) (2017) 6354–6365, https://doi.org/10.1109/TGRS.2017.2726901.
[7] T. Liu, J. Qiu, Y. Liu, J. Li, S. Chen, J. Lai, B. Mai, Research on an intelligent pineapple pre-harvest anti-lodging method based on deep learning and machine vision, Comput. Electron. Agric. 218 (2024) 108706, https://doi.org/10.1016/j.compag.2024.108706.
[8] J. Ju, G. Chen, Z. Lv, M. Zhao, L. Sun, Z. Wang, J. Wang, Design and experiment of an adaptive cruise weeding robot for paddy fields based on improved YOLOv5,

Comput. Electron. Agric. 219 (2024) 108824, https://doi.org/10.1016/j.compag.2024.108824.
[9] H. Zhang, S. Feng, D. Wu, C. Zhao, X. Liu, Y. Zhou, S. Wang, H. Deng, Hyperspectral image cassification on large-scale agricultural crops: the Heilongjiang benchmark dataset, validation procedure, and baseline results, Remote Sens. 16 (3) (2024) 478, https://doi.org/10.3390/rs16030478.
[10] B. Chen, L. Gong, C. Yu, X. Du, J. Chen, S. Xie, X. Le, Y. Li, C. Liu, Workspace decomposition based path planning for fruit-picking robot in complex greenhouse environment, Comput. Electron. Agric. 215 (2023) 108353, https://doi.org/10.1016/j.compag.2023.108353.
[11] Y. Tian, X. Lai, F. Zhang, R. Shi, W. Gu, W. Wang, Recognition method for apple stem integrity based on hyperspectral imaging (in Chinese), J. Shenyang Agric. Univ. 49 (2) (2018) 234–241, https://doi.org/10.3969/j.issn.1000-1700.2018.02.016.
[12] J. Zhuang, C. Hou, Y. Tang, Y. He, Q. Guo, Z. Zhong, S. Luo, Computer vision-based localisation of picking points for automatic litchi harvesting applications towards natural scenarios, Biosyst. Eng. 187 (2019) 1–20, https://doi.org/10.1016/j.biosystemseng.2019.08.016.
[13] J. Xiong, Z. Liu, R. Lin, R. Bu, Z. He, Z. Yang, C. Liang, Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (CCD) vision sensor with artificial illumination, Sensors 18 (4) (2018) 969, https://doi.org/10.3390/s18040969.
[14] W. Chen, M. Liu, C. Zhao, X. Li, Y. Wang, MTD-YOLO: multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection, Comput. Electron. Agric. 216 (2024) 108533, https://doi.org/10.1016/j.compag.2023.108533.
[15] A. Cardellicchio, V. Renò, S. Cellini, S. Summerer, A. Petrozza, A. Milella, Incremental learning with domain adaption for tomato plant phenotyping, Smart Agric. Technol. (2025) 101324, https://doi.org/10.1016/j.atech.2025.101324.
[16] I. Glukhikh, D. Glukhikh, A. Gubina, T. Chernysheva, Deep learning method with domain-task adaptation and client-specific fine-tuning YOLO11 model for counting greenhouse tomatoes, Appl. Syst. Innov. 8 (3) (2025) 71, https://doi.org/10.3390/asi8030071.
[17] C. Liang, J. Xiong, Z. Zheng, Z. Zhong, Z. Li, S. Chen, Z. Yang, A visual detection method for nighttime litchi fruits and fruiting stems, Comput. Electron. Agric. 169 (2020) 105192, https://doi.org/10.1016/j.compag.2019.105192.
[18] F. Wu, J. Duan, P. Ai, Z. Chen, Z. Yang, X. Zou, Rachis detection and three-dimensional localization of cut off point for vision-based banana robot, Comput. Electron. Agric. 198 (2022) 107079, https://doi.org/10.1016/j.compag.2022.107079.
[19] J. Chen, A. Ma, L. Huang, L. Huang, H. Li, H. Zhang, Y. Huang, T. Zhu, Efficient and lightweight grape and picking point synchronous detection model based on key point detection, Comput. Electron. Agric. 217 (2024) 108612, https://doi.org/10.1016/j.compag.2024.108612.
[20] W. Zhang, Y. Liu, K. Chen, H. Li, Y. Duan, W. Wu, Y. Shi, W. Guo, Lightweight fruit-detection algorithm for edge computing applications, Front. Plant Sci. 12 (2021) 740936, https://doi.org/10.3389/fpls.2021.740936.
[21] W. Ji, Y. Pan, B. Xu, J. Wang, A real-time apple targets detection method for picking robot based on ShufflenetV2-YOLOX, Agriculture 12 (6) (2022), https://doi.org/10.3390/agriculture12060856.
[22] X. Huang, W. Chen, W. Hu, L. Chen, An AI edge computing-based robotic arm automated guided vehicle system for harvesting pitaya, in: 2022 IEEE International Conference on Consumer Electronics (ICCE), 2022, pp. 1–2, https://doi.org/10.1109/ICCE53296.2022.9730442.
[23] P. Wang, Z. Ma, X. Du, W. Lu, W. Xing, F. Du, C. Wu, A binocular stereo vision system of fruits picking robots based on embedded system, in: ASABE annual international virtual meeting, 2020, https://doi.org/10.13031/aim.202000408.
[24] H. Liao, G. Wang, S. Jin, Y. Liu, W. Sun, S. Yang, HCRP-YOLO: a lightweight algorithm for potato defect detection, Smart Agric. Technol. 10 (2025) 100849, https://doi.org/10.1016/j.atech.2025.100849.
[25] X. Xia, N. Zhang, Z. Guan, X. Chai, S. Ma, X. Chai, T. Sun, PAB-Mamba-YOLO: VSSM assists in YOLO for aggressive behavior detection among weaned piglets, Artif. Intell. Agric. 15 (1) (2025) 52–66, https://doi.org/10.1016/j.aiia.2025.01.001.
[26] R. Varghese, M. Sambath, YOLOv8: a novel object detection algorithm with enhanced performance and robustness, in: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), 2024, pp. 1–6, https://doi.org/10.1109/ADICS58448.2024.10533619.
[27] Z. Gu, D. He, J. Huang, J. Chen, X. Wu, B. Huang, T. Dong, Q. Yang, H. Li, Simultaneous detection of fruits and fruiting stems in mango using improved YOLOv8 model deployed by edge device, Comput. Electron. Agric. 227 (1) (2024) 109512, https://doi.org/10.1016/j.compag.2024.109512.
[28] G. Zhang, H. Cao, Y. Jin, Y. Zhong, A. Zhao, X. Zou, H. Wang, YOLOv8n-DDA-SAM: accurate cutting-point estimation for robotic cherry-tomato harvesting, Agriculture 14 (7) (2024) 1011, https://doi.org/10.3390/agriculture14071011.
[29] G. Zhang, C. Wang, D. Xiao, A novel daily behavior recognition model for cage-reared ducks by improving SPPF and C3 of YOLOv5s, Comput. Electron. Agric. 227 (1) (2024) 109580, https://doi.org/10.1016/j.compag.2024.109580.
[30] Liu, Y., Shao, Z., Hoffmann, N., Global attention mechanism: retain information to enhance channel-spatial interactions. arXiv preprint arXiv:2112 (2021) 05561. https://doi.org/10.48550/arXiv.2112.05561.
[31] H. Yang, C. Gao, H. Ge, Y. Sang, Y. Wang, Risk assessment of aviation DC series arc based on reconstructed CBAM-CNN, J. Power Electron. 23 (5) (2023) 811–820, https://doi.org/10.1007/s43236-022-00575-y.
[32] X. Shan, Y. Shen, H. Cai, Y. Wen, Convolutional neural network optimization via channel reassessment attention module, Digit. Signal Prog. 123 (2022) 103408, https://doi.org/10.1016/j.dsp.2022.103408.
[33] X. Zhang, Z. Wang, Spatial proximity feature selection with residual spatial-spectral attention network for hyperspectral image classification, IEEE Access 11 (1) (2023) 23268–23281, https://doi.org/10.1109/ACCESS.2023.3253627.
[34] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, DETRs beat YOLOs on real-time object detection, in: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 16965–16974, https://doi.org/10.1109/CVPR52733.2024.01605.
[35] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, L. Zhang, Dynamic head: unifying object detection heads with attentions, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7369–7378, https://doi.org/10.1109/CVPR46437.2021.00729.
[36] F. Zhang, W. Cao, S. Wang, X. Cui, N. Yang, X. Wang, X. Zhang, S. Fu, Improved YOLOv4 recognition algorithm for pitaya based on coordinate attention and combinational convolution, Front. Plant Sci. 13 (2022) 1030021, https://doi.org/10.3389/fpls.2022.1030021.
[37] N. Ma, Y. Wu, Y. Bo, H. Yan, Chili pepper object detection method based on improved YOLOv8n, Plants 13 (17) (2024), https://doi.org/10.3390/plants13172402.
[38] J. Yang, T. Zhang, C. Fang, H. Zheng, C. Ma, Z. Wu, A detection method for dead caged hens based on improved YOLOv7, Comput. Electron. Agric. 226 (2024) 109388, https://doi.org/10.1016/j.compag.2024.109388.
[39] Y. Sui, L. Zhang, Z. Sun, W. Yi, M. Wang, Research on coal and gangue recognition based on the improved YOLOv7-Tiny target detection algorithm, Sensors 24 (2) (2024) 456, https://doi.org/10.3390/s24020456.
[40] A. Waleed, A. Khaled, K. Khaled, Estimating pavement roughness using a low-cost depth camera, Int. J. Pavement Eng. 23 (14) (2022) 4923–4930, https://doi.org/10.1080/10298436.2021.1984478.
[41] K. Vinoth, S. P, Lightweight object detection in low light: pixel-wise depth refinement and TensorRT optimization, Results Eng. 23 (2024) 102510, https://doi.org/10.1016/j.rineng.2024.102510.
[42] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626, https://doi.org/10.1109/ICCV.2017.74.
[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C, SSD: single shot multibox detector, in: Computer Vision-ECCV 2016: 14th European Conference, 2016, pp. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2.
[44] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2016, pp. 1137–1149, https://doi.org/10.1109/TPAMI.2016.2577031.
[45] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dan, Y. Liu, J. Chen, DETRs beat YOLOs on real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16965–16974.
[46] Jocher, G., Chaurasia, A., Qiu, J. YOLOv5: a real-time object detection model. https://github.com/ultralytics/yolov5 [Accessed 2024-06-20].
[47] Wang, C., Yeh, I., Liao, H., YOLOv9: learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402 2024 13616. https://doi.org/10.48550/arXiv.2402.13616.
[48] Ultralytics, YOLO11 NEW, (n.d.). https://docs.ultralytics.com/models/yolo11 (accessed October 21, 2024).
[49] Y. Tian, Q. Ye, D. Doermann, YOLOv12: attention-centric real-time object detectors, arXiv abs/2502. 12524, https://arxiv.org/abs/2502.12524, 2025.
[50] P. Xu, N. Fang, N. Liu, F. Lin, S. Yang, J. Ning, Visual recognition of cherry tomatoes in plant factory based on improved deep instance segmentation, Comput. Electron. Agric. 197 (2022) 106991, https://doi.org/10.1016/j.compag.2022.106991.
[51] L. Fu, Z. Yang, F. Wu, X. Zou, J. Lin, Y. Cao, J. Duan, YOLO-Banana: a Lightweight neural network for rapid detection of banana bunches and stalks in the, Natural Environ.. Agron. 12 (2) (2022), https://doi.org/10.3390/agronomy12020391.
[52] B. Zhang, Y. Xia, R. Wang, Y. Wang, C. Yin, M. Fu, W. Fu, Recognition of mango and location of picking point on stem based on a multi-task CNN model named YOLOMS, Precis. Agric. 25 (2024) 1454–1476, https://doi.org/10.1007/s11119-024-10119-y.