YieldNet: A Near-Zero-Cost YOLOv8n Enhancement for UAV-Based Real-Time Green Tomato Detection to Support Pre-Harvest Yield Forecasting

Chenyu Yu
Beijing Information Science and Technology University
Lu Li
20192380@bistu.edu.cn

Beijing Information Science and Technology University
Bolin Huang
Beijing Information Science and Technology University

Research Article

Keywords: Green tomato detection, UAV, Lightweight YOLO, ShuffleNetV2, Attention mechanism, Pre-harvest yield forecasting

Posted Date: April 28th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9533248/v1

Additional Declarations: No competing interests reported.

Highlights:

- We built a dataset of green tomatoes in greenhouses using real drone-captured field images, providing a foundation for early fruit detection.
- A new ShuffleNetV2-ECA architecture with PIoU v2 loss is proposed on top of YOLOv8n, with near-zero extra cost.
- The improved model achieves superior performance with only +0.3 M parameters and reduced GFLOPs, enabling edge deployment.
- This is the first study combining UAVs, immature green tomatoes, and lightweight models for practical pre-harvest yield forecasting in greenhouses.

IMPACT: This work provides a near-zero-cost enhancement to YOLOv8n for UAV-based green tomato detection, enabling real-time, high-precision inference on edge devices. By supporting pre-harvest yield forecasting with minimal computational overhead, it aligns with Precision Agriculture's goal of improving resource efficiency and crop management through data-driven, site-specific interventions.

YieldNet: A Near-Zero-Cost YOLOv8n Enhancement for UAV-Based Real-Time Green Tomato Detection to Support Pre-Harvest Yield Forecasting

Chenyu Yu[1], Lu Li [†1*], Bolin Huang[1]

[1]College of Automation and AI, Beijing Information Science and Technology University, No. 12 Qinghe Xiaoying East Road, Haidian District, Beijing, 100192, China

[†] ORCID: 0000-0001-9823-0565.

*Corresponding author(s). E-mail(s): 20192380@bistu.edu.cn;
Contributing authors: 2023010765@bistu.edu.cn;
2023010740@bistu.edu.cn;

Abstract

Accurate pre-harvest yield forecasting of greenhouse tomatoes is essential for reducing post-harvest losses, with reliable detection of immature green tomatoes being the core challenge. However, these fruits are small, heavily occluded, and chromatically highly similar to foliage, making real-time detection from low-altitude UAV imagery extremely difficult, while onboard edge processors impose stringent power and weight constraints. To address this, we propose YieldNet, an ultra-lightweight framework that introduces near-zero-overhead enhancements to vanilla YOLOv8n: the backbone is replaced with ShuffleNetV2 to strengthen small-object representation; Efficient Channel Attention (ECA) modules are embedded after the P3–P5 layers in the neck to suppress leaf-background interference; and PIoU v2 loss is adopted to refine bounding-box regression for densely overlapped fruits via size-adaptive and non-monotonic focusing mechanisms. The model is rigorously validated on both a self-collected real-world UAV dataset comprising 600 low-altitude green-tomato images and a public multi-ripeness benchmark. Compared with the YOLOv8n baseline, YieldNet achieves relative improvements in mAP@50-95, Recall, and F1-score by 18.9%, 6.1%, and 5.8%, respectively, on the large dataset, and enhances Recall, F1-score, and Precision by relative gains of 4.3%, 4.0%, and 3.8%, respectively, on the small dataset, while increasing parameters only from 3.0 M to 3.3 M and reducing FLOPs from 8.1 G

to 8.0 G. This work provides an efficient, readily deployable solution for high-precision real-time detection of immature green tomatoes on UAV platforms, enabling reliable pre-harvest yield estimation.

Keywords: Green tomato detection, UAV, Lightweight YOLO, ShuffleNetV2, Attention mechanism, Pre-harvest yield forecasting

1 Introduction

Tomatoes, as a very common crop, are widely cultivated around the world and possess significant economic potential. According to the analysis by Data Bridge Market Research, the global tomato market is expected to achieve a compound annual growth rate (CAGR) of 3.2% during the period from 2023 to 2030[1]. However, numerous studies have pointed out that there are huge post-harvest losses in the tomato vegetable value chain [1]. As a fruit with high water content, tomatoes are prone to rot; farmers often cannot accurately grasp market fluctuations before planting. In order to reduce post-harvest losses, it is crucial to connect the market in time before crop picking and within a short time after picking [2]. The key here lies in reliable and timely yield prediction. However, traditional prediction methods rely heavily on manual work, which requires experienced farmers to step into the planting area frequently for statistics, which is time-consuming and laborious [3]. With the development of UAV technology and machine vision, this challenge is expected to be solved [4].

UAVs, also known as drones, have been increasingly used as a new type of agricultural technical means and have a wide range of applications in precision agriculture tasks [5]. With the combination of photography technology and UAV technology, UAV agricultural monitoring technology has become a new type of UAV agricultural auxiliary technology after UAV plant protection and UAV sowing [6]. Drones can process the images they shoot in real time through their onboard camera units and computing units, so as to obtain the number and coordinates of crop targets in the shooting field of view [7]. Combined with specific flight algorithms, comprehensive shooting and identification of crops in the planting field can be carried out [8]. The application of this technology can greatly reduce the labor cost traditionally relied on in agricultural management. Through automated crop monitoring and data collection by drones, farmers no longer need to frequently enter the field for manual inspection and recording [9].

YOLO (You Only Look Once) is a single-stage detector. Its working principle is that it first divides the image into grids, and then performs one convolutional neural network forward propagation, and then it can directly output the class confidence and position coordinates of the target [10]. Different from traditional detectors, it merges the two separately processed steps of proposing candidate boxes and extracting features into one step, doing so gives it an obvious improvement in speed compared

[1]Data Bridge Market Research, Global Tomatoes Market – Industry Trends and Forecast to 2030, Market Report, 2022. Available online: https://www.databridgemarketresearch.com/reports/global-tomatoes-market (accessed on 15 August 2025).

to traditional detectors, and can also maintain a relatively high detection accuracy. YOLO has not stopped evolving since the moment it was proposed: from the original v1 version to the latest version [11], it has successively added improvements such as residual connections and feature pyramids, dealing with small objects and object occlusion and other previously challenging detection tasks has also obtained stronger performance. Moreover, considering the differences in deployment devices, it also provides five specifications of n/s/m/l/x, so users can choose a suitable version according to the different performance of different devices, among which the lightest n (nano) specification can exactly be used for the implementation of automatic green tomato recognition on UAVs [12].

In related research, Egi et al. used YOLOv5 combined with drones for greenhouse tomato detection and counting, and their key contribution lies in dataset construction and application verification [13]. Wang et al. achieved higher reliability in yield estimation by improving YOLO11n and combining it with an optimized regional tracking and counting method, emphasizing the importance of combining detection and counting [14]. The DCFA-YOLO proposed by Chai et al. achieved high-precision detection of cherry tomato clusters through a dual-channel cross-feature fusion attention module, proving the effectiveness of attention mechanisms in small target dense detection [15]. Although existing studies have made important progress, research on green tomatoes is still insufficient [16]. Green tomatoes, as the initial stage of tomato fruit growth, their timely detection and counting are of great significance for yield prediction . However, green tomato detection faces many difficulties: first, green tomatoes have high similarity with leaves in color and texture, which is prone to false detection and missed detection ; second, the field environment is complex, and factors such as branch and leaf occlusion, fruit overlap, and lighting changes bring great challenges to the stability of the model ; in addition, computational overhead must be considered, and the model's parameters and computational complexity should not be too high to meet the real-time inference needs of UAV edge computing devices [17].

To address these challenges, this paper proposes a lightweight model for green tomato detection on UAVs - YieldNet. This method is based on YOLOv8n and explores its application effect in green tomato detection through experiments on self-built and public datasets. The main contributions are as follows:

1. We construct a dedicated greenhouse immature green tomato dataset captured by low-altitude UAVs, providing a realistic and challenging benchmark for early-stage fruit detection.
2. We propose lightweight yet targeted modifications to the vanilla YOLOv8n architecture—replacing its backbone with ShuffleNetV2, integrating lightweight ECA attention modules in the neck, and adopting the PIoU v2 loss function. These changes yield substantial performance gains while increasing parameters by only 0.3 M and even reducing computational complexity.
3. To our knowledge, this study is the first to address the underexplored intersection of UAV-based detection, targeted immature green tomatoes, and lightweight model improvements. It delivers a practical, high-precision solution readily deployable for pre-harvest yield forecasting in commercial greenhouse settings.

2 Materials and Methods

2.1 Dataset and Data Collection

To evaluate both in-domain performance and cross-domain generalization of the proposed model, two datasets were used:

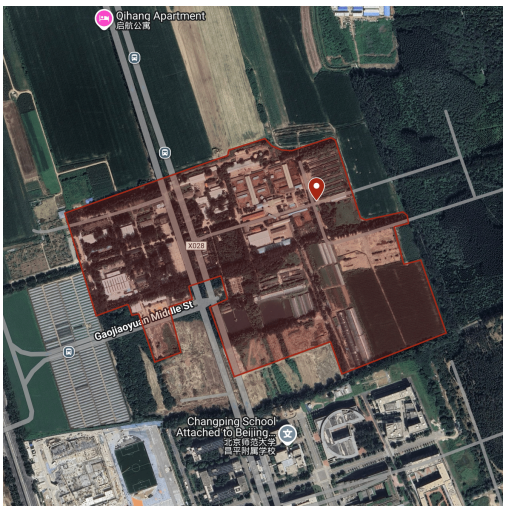
1. **GreenTomato-UAV Dataset (Self-Built)**: Images were collected in a greenhouse at the China Nongjiyuan Agricultural Machinery Test Station (shown in Fig. 1), 577M+7X2, Changping District, Beijing, China, 102206, on 19 July 2025, between 16:30 and 18:00. An F450 quadcopter equipped with a 640 × 480-pixel camera was flown at 0.5–0.8 m above the canopy (this height range corresponds to the area where tomato fruits are most densely concentrated in this greenhouse's tomato cultivation). The UAV traversed between planting beds following the serpentine pattern illustrated in Fig. 2 for image acquisition. Approximately 80,000 raw frames were captured. Scenes included soil, stakes, and weeds; most targets were green, unripe tomatoes exhibiting frequent overlapping, leaf occlusion, dense planting, and motion blur caused by UAV movement. Fig. 3 illustrates those key challenges encountered in the dataset. An open-source annotation tool, *Labelme*, was employed, with five annotators independently labeling objects using rectangular bounding boxes. After manual review and cleaning, 600 images were retained (100 without targets).
2. **Tomato-Recog Public Dataset**: A publicly available tomato dataset consisting of 1,986 images covering green, turning, and red tomatoes was used for zero-shot cross-domain evaluation. We divide all images into training and validation sets at a 7:3 ratio, which is referred to as the large dataset in subsequent sections. The dataset includes official YOLO-format annotations and is fully open-source.

2.2 Incremental Data Augmentation

To enhance model robustness under complex greenhouse conditions and mitigate the limited scale of the original dataset, a **label-driven, proportional, multi-stage augmentation strategy** was implemented using Python and the **imgaug** library[2]. The key idea is:

- **Proportional Sampling:** In each augmentation round, the number of images to be augmented is calculated as a fixed proportion (20%) of the *original dataset size*, while images are randomly sampled from the *current dataset* (i.e., the master set including all previously augmented images). By doing so, we can have more diverse image augmentation combination effects and also expand the scale of our dataset.
- **Augmentation Operations:** Each sampled image undergoes three successive augmentation steps:
  1. **Motion Blur:** Kernel size of 15, with the blur angle randomly chosen from $[-45°, 45°]$.
  2. **Affine Rotation:** Rotation angle randomly selected from the range $[-30°, 30°]$.

[2] A. Jung and others, imgaug: Image augmentation for machine learning experiments, GitHub repository, 2017–2025. Available online: https://github.com/aleju/imgaug (accessed on 17 October 2025).



(a)

(b)

Fig. 1: GreenTomato-UAV dataset collection sites. (**a**) Overview of the China Nongjiyuan Test Station, Changping District, Beijing (40.18°N, 116.27°E). (**b**) Greenhouse where green-tomato images were acquired(40.16°N, 116.28°E). Base-map imagery source: Google My Maps (Google Inc.).

3. **Lighting Adjustment:** Brightness and contrast scaling factors randomly set within $[0.7, 1.3]$.

- **Dataset Management:** The original labels of the images also change along with certain augmentations that alter the target object coordinates (such as rotation and

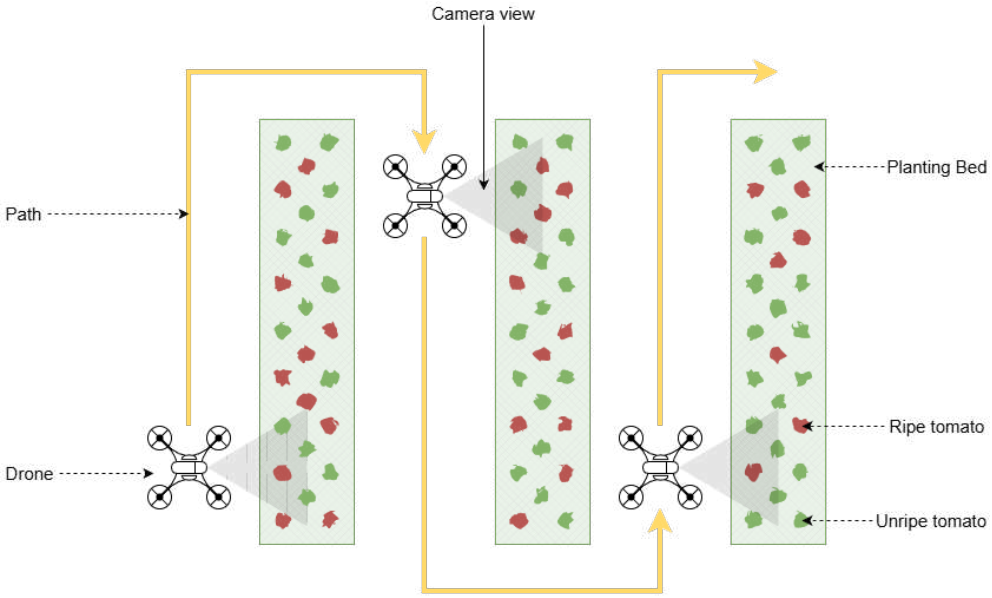


Fig. 2: UAV serpentine flight path for image acquisition over planting beds

translation), while filenames are appended with corresponding round suffixes (e.g., _r1, _r2) to record the image augmentation round. The results of the previous image augmentation will be merged into the images before the previous augmentation, entering the next round of image augmentation together.

- **Cumulative Effect:** Following three augmentation rounds, the overall dataset size grew by about 60% relative to the original, which notably improves model robustness against small objects, occlusion, and diverse lighting conditions.

The specific data augmentation workflow is illustrated in Fig. 4.

600 images from the Self-Built dataset are first randomly divided into training and validation sets at a 7:3 ratio. Then the training set images are augmented through the above process to obtain 672 images, finally resulting in 672 training images and 180 validation images, which is referred to as the large dataset in subsequent sections.

2.3 Improvement in YieldNet Network

As depicted in Figure 5, the model is constructed based on the YOLOv8 detection framework, featuring P3–P5 feature outputs. In the backbone network, the original convolutional blocks are replaced with five **ShuffleNetV2** [18] modules to reduce computational load while maintaining feature representation capability; modifications to the neck involve adding **Efficient Channel Attention (ECA)** [19] modules after each layer output to enhance inter-channel dependencies in the feature maps; finally, the head predicts bounding boxes at three scales (P3/8, P4/16, P5/32) through the **Detect** layer, enabling precise detection of objects of varying sizes.

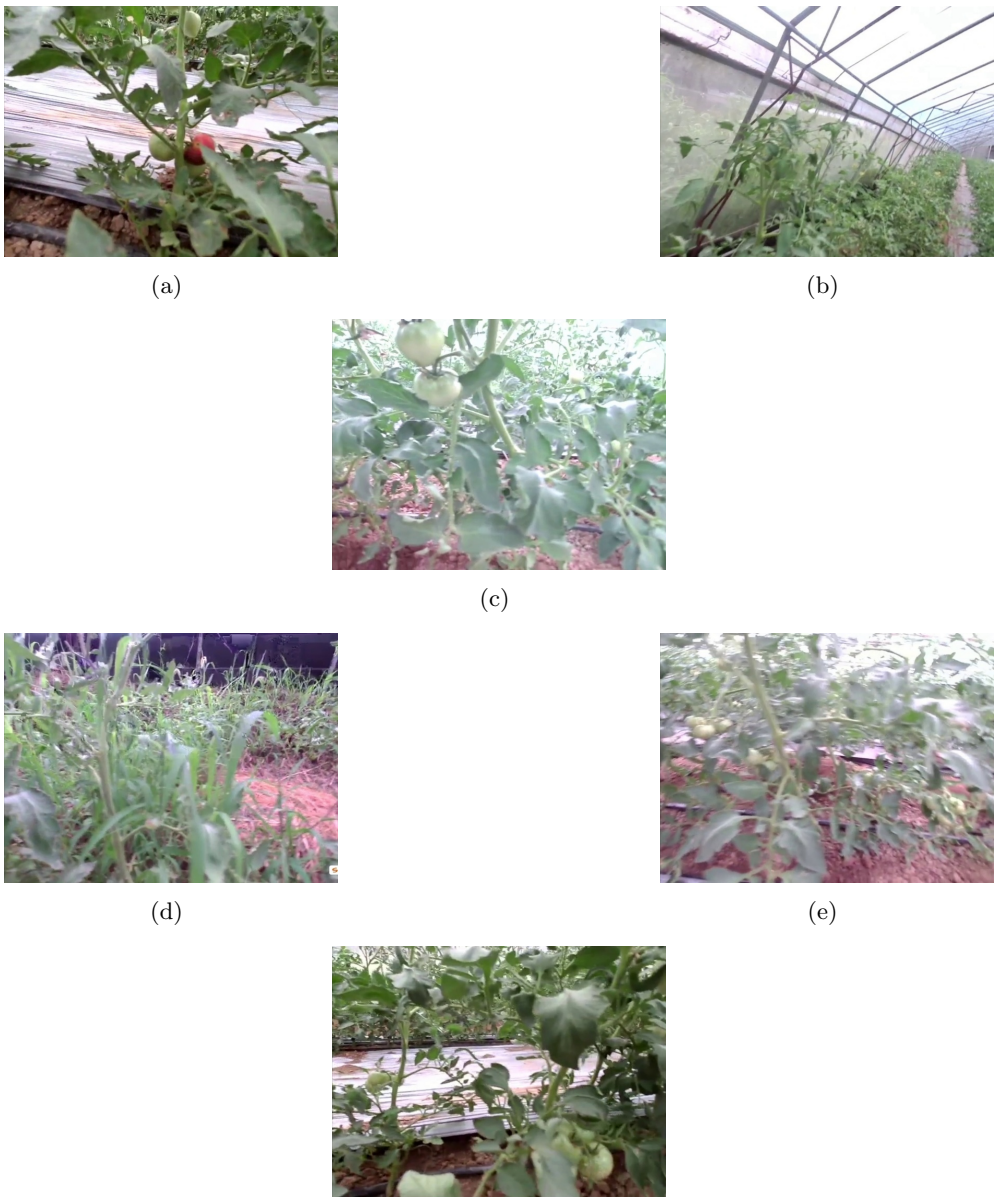


Fig. 3: Representative dataset samples illustrating key challenges in agricultural UAV imagery: (a) mature tomatoes, (b) complex greenhouse environment, (c) overexposure, (d) weed interference, (e) motion blur, and (f) leaf occlusion.

In addition to architectural enhancements, we also incorporate the **Powerful-IoU v2 (PIoU v2)** [20] strategy into the calculation of the loss function. We hope

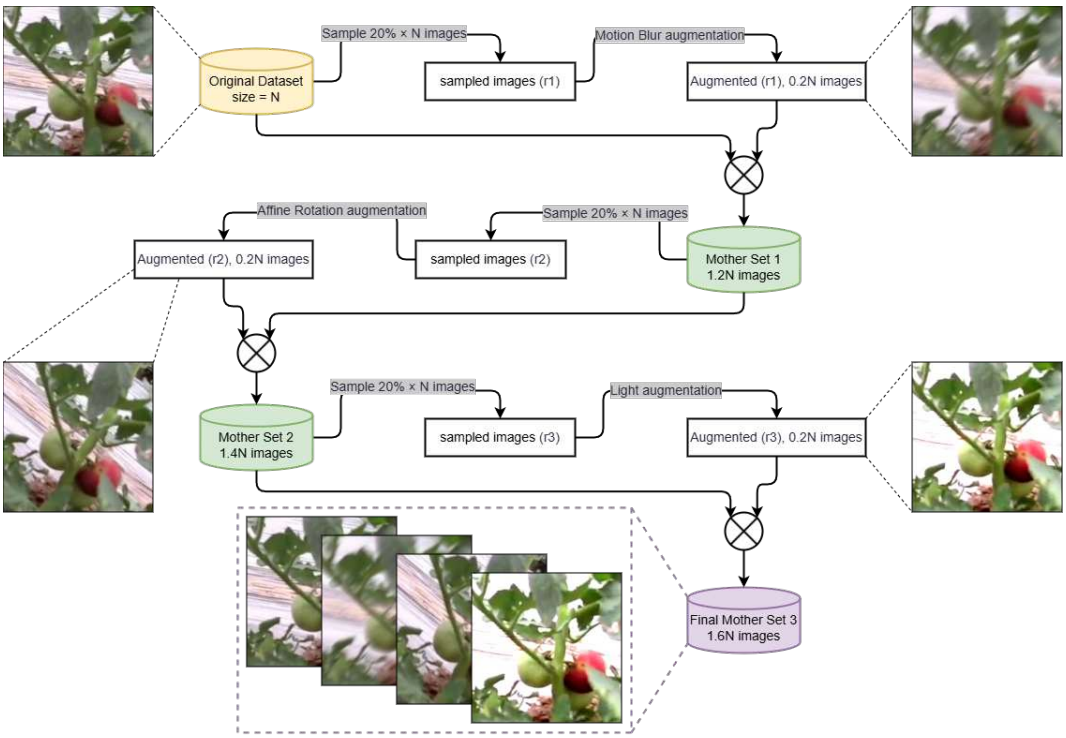


Fig. 4: Incremental data augmentation workflow (S-shaped). Nodes represent datasets/mother sets, while augmentation operations are indicated on the arrows. Each round samples $0.2N$ images from the current mother set. After three rounds, the final mother set contains $1.6N$ images.

to strengthen the model's attention to small targets through its unique size-adaptive penalty term, as well as improve bounding box accuracy and training convergence speed through the monotonic focusing mechanism. Such a design is especially effective for green tomatoes.

2.3.1 Improvement in Backbone Network

YOLOv8n baseline uses CSPDarknet for hierarchical feature extraction. However, its deep residual modules introduce too many parameters, which slows down real-time inference on resource-limited UAV platforms. To solve this problem, we replace the backbone with ShuffleNetV2—a hardware-efficient architecture originally designed for mobile use. This architecture has been successfully used in applications such as apple detection [21] and tomato detection [22], and serves as an improvement over the original ShuffleNet V1 [23].

Unlike conventional group convolutions, ShuffleNetV2 uses channel splitting and shuffling operations to enable information exchange between parallel branches while keeping memory access costs low. Its basic unit splits input features into two branches:

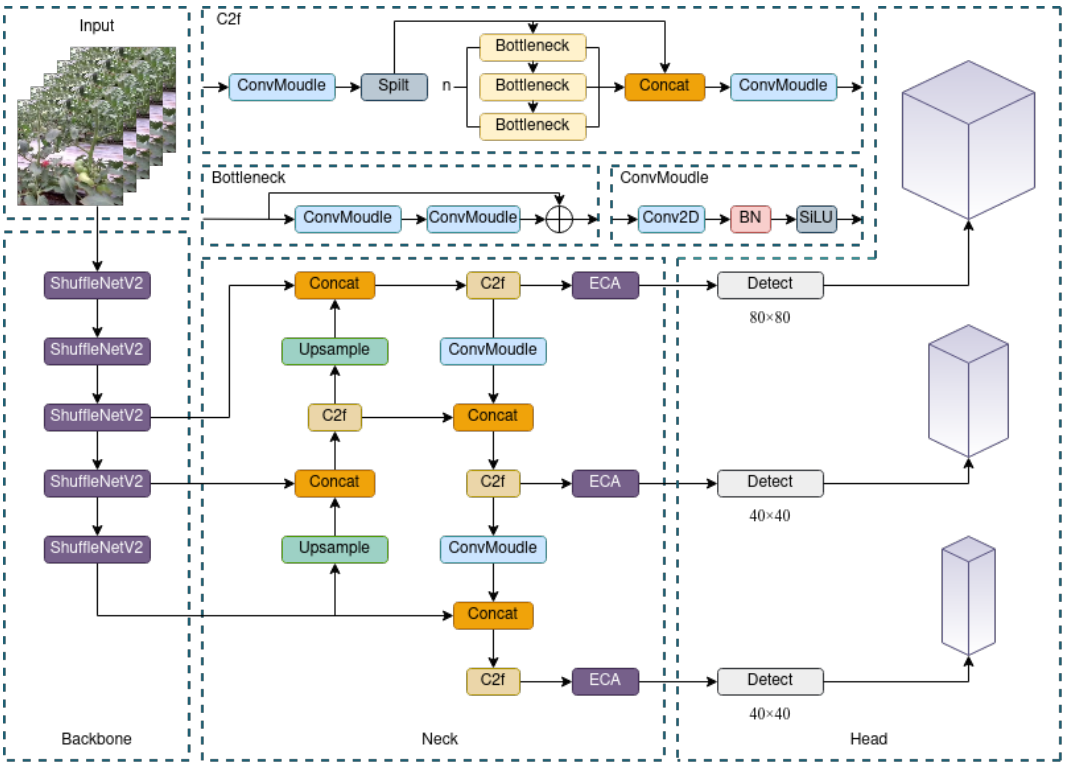
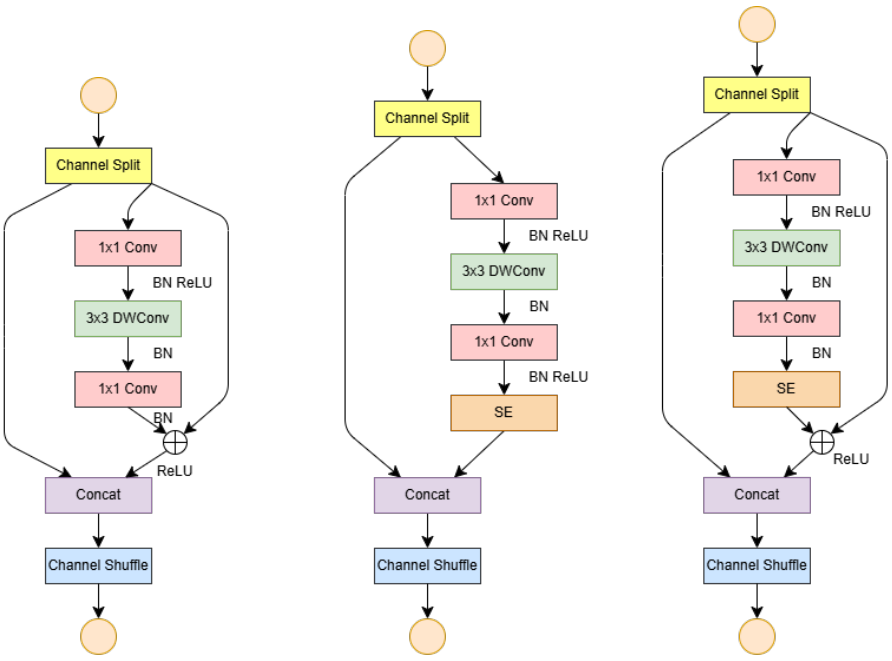


Fig. 5: Overall architecture of YieldNet. The purple module highlights the components modified or added compared to the YOLOv8n baseline.

one branch uses identity mapping, and the other uses depthwise separable convolutions for lightweight feature transformation. We further modify the downsampling module into a symmetric dual-branch structure, where both branches use parallel 3×3 depthwise convolutions with stride 2 to reduce spatial resolution and expand channel capacity at the same time. As empirically supported by the attention heatmap visualization in Section 3.2, this structural modification enables the network to extract richer multi-scale features for small green tomatoes without introducing substantial computational overhead. Experimentally measured, this fused backbone network uses about **3.3 M** parameters and **8.0 GFLOPs**, with only a very small parameter increase compared to the original model.

2.3.2 Improvement in Neck Network

After reducing backbone FLOPs with ShuffleNetV2, we turn our attention to the neck. We basically retain the complete original YOLOv8 neck framework, only directly inserting an ECA module (with kernel size 3) right after each C2f output at scales P3, P4, and P5 (80×80, 40×40, 20×20). We hope to utilize it to perform adaptive weighting (recalibration) on the final features input to the detection head,



(a) (b) (c)

Fig. 6: Evolution of ShuffleNetV2 building blocks: (**a**) original basic block with channel split, DWConv and shuffle; (**b**) SE module inserted in the residual branch; (**c**) SE module applied after concatenation for global channel recalibration. All blocks maintain channel shuffle for cross-group information exchange.

suppress noise features (possibly generated by background elements and branches and leaves), thereby enabling the detection head to focus more effectively on key features. Moreover, benefiting from the lightweight design of the ECA module (one-dimensional convolution) and the fixed kernel size parameter, the implementation of this improvement results in almost zero increase in parameters and computational cost [24].

Parameters γ and b remain as hyperparameters in the original formulation, where σ denotes the sigmoid activation function. Unlike the adaptive kernel sizing proposed in the original ECA design, we fix the 1-D convolution kernel to $k = 3$—this practice maintains attention efficacy while also reducing the computational overhead of the model.

The feature processing flow in the ECA module is as follows: the input tensor $[H, W, C]$ is first compressed to a vector of length C with shape $[1, 1, C]$ through global average pooling, then the pooled feature vector undergoes convolution operation through the 1-D convolution kernel with fixed $k = 3$ and generates per-channel weights, finally after sigmoid operation it is multiplied with the original features from the initial input to obtain the final weighted features.

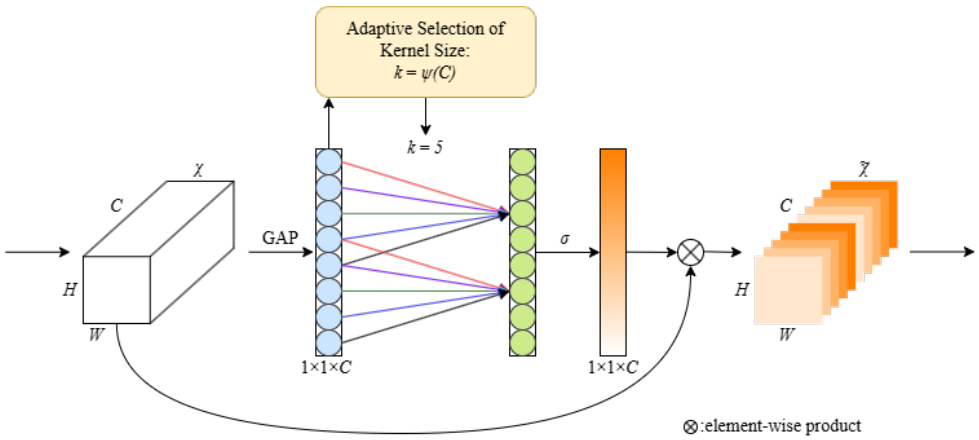


Fig. 7: Model of ECA mechanism. H is the height, W is the width, and C is the number of channels.

For the feature $y \in \mathbb{R}^C$ after average pooling (its feature dimension remains unchanged), the process of obtaining feature weights through convolution operation by fast 1-D convolution (denoted as C1D) with our fixed-size kernel can be expressed as follows:

$$\omega = \sigma(\mathrm{C1D}_{k=3}(y)) \tag{1}$$

The original ECA formulation posits an exponential mapping between kernel size k and channel dimension C:

$$C = \phi(k) = 2^{(\gamma k - b)} \tag{2}$$

We depart from this adaptive scheme. Empirical validation on our greenhouse datasets demonstrated that fixed $k = 3$ maintains strong performance. Combined with the ShuffleNetV2 backbone, the network more easily achieves higher mAP.

2.3.3 Improvement in Loss Function

To address the bounding-box regression accuracy issue for densely overlapped green tomatoes in detection, we use Powerful-IoU v2 (PIoU v2) to replace the standard CIoU loss. PIoU designs a penalty factor P that depends on the absolute distance between corresponding edges of the predicted box and the target box:

$$P = \left(\frac{dw_1 + dw_2}{w_{gt}} + \frac{dh_1 + dh_2}{h_{gt}} \right) / 4 \tag{3}$$

where dw_1, dw_2, dh_1, dh_2 are the absolute distances between corresponding edges of the predicted box and the target box, and w_{gt} and h_{gt} are the width and height of the target box.

Simultaneously, a gradient adjustment function $f(x) = 1 - e^{-x^2}$ is introduced, and the PIoU loss is defined as:

$$L_{\mathrm{PIoU}} = L_{\mathrm{IoU}} + f(P) = 2 - \mathrm{IoU} - e^{-P^2} \tag{4}$$

Furthermore, PIoU v2 introduces a non-monotonic attention mechanism controlled by the hyperparameter λ (in this experiment, $\lambda = 1.5$), defining the attention function:

$$q = e^{-P}, \quad u(x) = 3x \cdot e^{-x^2} \tag{5}$$

The final loss function is:

$$L_{\text{PIoU v2}} = u(\lambda q) \cdot L_{\text{PIoU}} = 3 \cdot (\lambda q) \cdot e^{-(\lambda q)^2} \cdot L_{\text{PIoU}} \tag{6}$$

For the green-tomato dataset, PIoU v2 increases the penalty for inaccurate localization on small fruits, thereby significantly improving the regression precision of small targets. Its quality-aware gradient modulation also accelerates training convergence, simultaneously improving the mAP of high-IoU and recall, making PIoU v2 a suitable loss function for YieldNet to perform object recognition in real field scenarios.

2.4 Experimental Environment and Model Evaluation Indicators

2.4.1 Experimental Environment

All experiments were conducted on the hardware platform specified in Table 1. To ensure fair comparisons, all models were trained under identical settings. The main hyperparameters are summarized as follows:

- Input resolution: 640×640 pixels;
- Training epochs: 100;
- Batch size: 8;
- Optimizer: SGD (initial learning rate 0.01, momentum 0.937);
- Single-class mode (`single_cls=True`) for tomato detection.

The model checkpoint from the final epoch with the best validation performance was used to report all evaluation metrics.

Table 1: Experimental Platform Configuration.

Component	Configuration
Operating System	Ubuntu 20.04.6 LTS (Kernel 5.15.0-139-generic)
CPU	13th Gen Intel Core i9-13900HX @ 2.4 GHz (Turbo Boost 5.4 GHz)
GPU	NVIDIA GeForce RTX 4060 Laptop GPU 8 GB GDDR6
GPU Accelerator	CUDA 12.1, cuDNN 8.7
Deep Learning Framework	PyTorch 2.4.1 + cu121
Hardware Configuration	15 GB RAM, 320 GB NVMe SSD
Compiler	Python 3.8.20 (Anaconda environment)
Programming Language	Python 3.8.20

2.4.2 Model Evaluation Indicators

Detection performance is assessed using:

1. *F1 Score:*
$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2-4}$$
tallied at IoU ≥ 0.5.
2. *mAP@50:*
Derived from the precision-recall curve area per category:
$$\mathrm{AP}_c = \int_0^1 P(R)\, dR \tag{5}$$
then averaged across C classes at IoU = 0.5:
$$\mathrm{mAP@50} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{AP}_c \tag{6}$$
3. *mAP@50-95:* Averaged over IoU $\in$ [0.50:0.05:0.95]:
$$\mathrm{mAP@50\text{-}95} = \frac{1}{9C} \sum_{c=1}^{C} \sum_{t} \mathrm{AP}_{c,t} \tag{7}$$
where t indexes the nine thresholds. This metric imposes stricter localization constraints than mAP@50, essential for accurate aerial yield estimation.

Model efficiency is measured by parameters (M) and GFLOPs.

3 Results

3.1 Comparison of Model Before and After Improvement

To evaluate the performance of the improved YieldNet model, this study compares it with the YOLOv8n baseline model across multiple metrics—**mAP@50**, **mAP@50-95**, **Precision(P)**,**Recall(R)**,**F1**,**Params**, and **GFLOPs**—after training on large dataset and small dataset. The comparison results are presented in Table 2 and Table 3 below.

As depicted in Table 2 and Table 3, the ShuffleNetV2-ECA model achieves significant improvements over the baseline across all metrics for all categories on both datasets. This is particularly evident in the large dataset, where the mAP@50-95 metric shows a relative gain of up to 18.92%, while mAP@50 and F1-Score improve by 4.75% and 5.82% respectively. The model also performs well on the small dataset, with mAP@50-95, mAP@50, and F1-Score improving by 3.50%, 3.51%, and 4.02% respectively.

Table 2: Detection performance on the large validation set (596 images, 2228 instances).

Model	Class	mAP @50	mAP @50-95	Precision	Recall	F1	Params (M)	
YOLOv8n baseline	All	0.926	0.666	0.921	0.868	0.894	3.00	8.1
	Ripe	0.931	0.709	0.959	0.882	0.919	3.00	—
	Semiripe	0.906	0.652	0.919	0.820	0.867	3.00	—
	Unripe	0.942	0.637	0.884	0.901	0.892	3.00	—
YieldNet (Ours)	All	**0.970**	**0.792**	**0.972**	**0.921**	**0.946**	3.30	8.0
	Ripe	**0.990**	**0.846**	**1.000**	**0.960**	**0.980**	3.30	—
	Semiripe	**0.951**	**0.765**	**0.953**	**0.883**	**0.917**	3.30	—
	Unripe	**0.969**	**0.763**	**0.964**	**0.921**	**0.942**	3.30	—

Bold indicates improvement over baseline ($\geq$1.5% absolute gain where applicable).

Table 3: Detection performance on the small validation set (180 images, 590 instances).

Model	Class	mAP @50	mAP @50-95	Precision	Recall	F1	Params (M)	
YOLOv8n baseline	All	0.884	0.457	0.872	0.820	0.845	3.00	8.1
YieldNet (Ours)	All	**0.915**	**0.473**	**0.905**	**0.855**	**0.879**	3.30	8.0

Same notation as Table 2.

These data tell us that the upgraded YieldNet model handles challenging targets in the dataset with greater ease. Behind these improvements lies the powerful performance of our original ShuffleNetV2-ECA fusion structure in feature extraction, as well as the more precise bounding-box regression for object detection boxes by the PIoU v2 loss function. What is even more surprising is that the improvement in model capability comes at the cost of only a **small increase in model parameters** (the total number of parameters increases modestly from 3.0 M to 3.3 M), and our new structure even reduces the computational complexity (FLOPs) of the model (decreasing slightly from 8.1 G to 8.0 G). This is what we have been pursuing: a high-efficiency vision model that can achieve accurate tomato identification in the field with high precision while being driven by lower computational power.

Figure 8 shows the qualitative comparison between the baseline YOLOv8n and our proposed YieldNet on random samples. It can be clearly seen from the figure that YieldNet is more capable of capturing small and occluded objects and has significantly

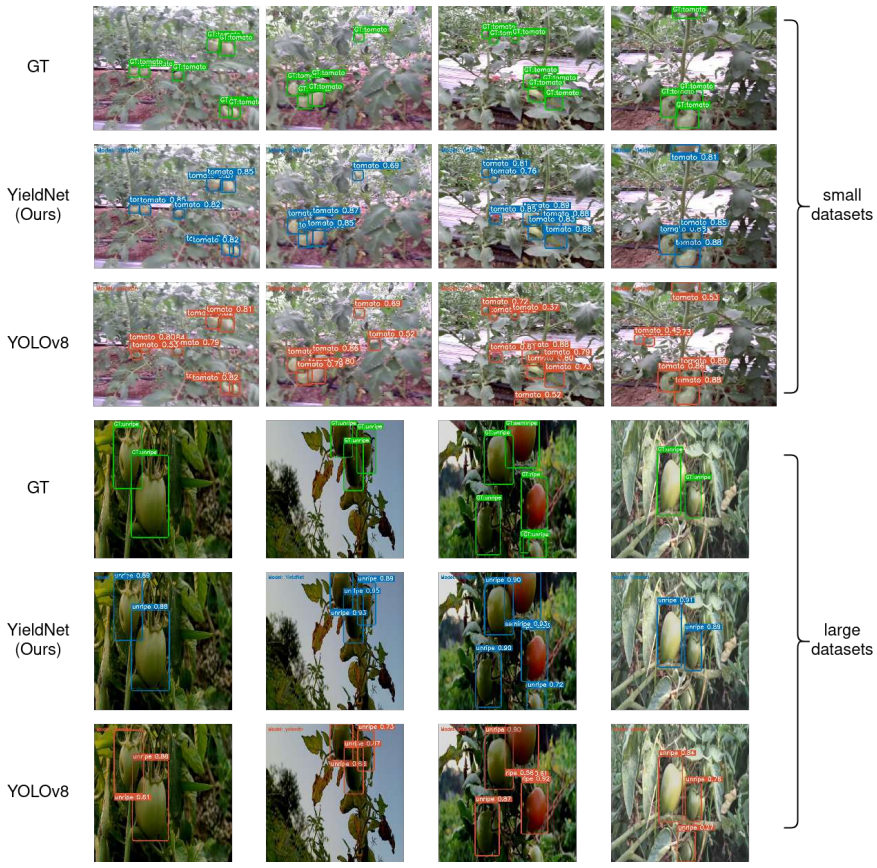


Fig. 8: Qualitative comparison of detection results on two datasets.

improved robustness. This is highly consistent with the quantitative results in Table 2 and Table 3.

3.2 Ablation Experiment

To further investigate the roles played by each improvement module, this study conducts an ablation experiment on the model: quantitative comparisons are made on small dataset between the training results of the baseline YOLOv8n model and those of various model variants after progressively incorporating each improvement module, with the comparison results presented in Table 4; on the same dataset, attention heatmaps are generated for selected images using the baseline YOLOv8n model and each variant model, and the results are shown in Figure 9. This paper aims to clearly demonstrate the effects of each module through these two experiments, while also validating the effectiveness of each improvement step.

Table 4 shows that when ECA and ShuffleNet V2 act alone, they both enhance all performance metrics, especially on mAP@50 and Precision: ECA improves mAP@50

Table 4: Ablation study on the integration of ShuffleNetV2, ECA, and PIoU v2 components on the small validation set (180 images, 590 instances).

Model	mAP @50	mAP @50-95	Precision	Recall	F1	Params (M)	GFLOPs
YOLOv8n baseline (no PIoU)	0.884	0.457	0.872	0.820	0.845	3.00	8.1
+ ShuffleNetV2 (no PIoU)	**0.912**	0.460	**0.904**	0.834	**0.867**	3.30	8.0
+ ECA (no PIoU)	**0.905**	0.459	**0.893**	0.827	0.858	3.00	8.1
+ ShuffleNetV2-ECA (no PIoU)	**0.912**	**0.475***	**0.904**	**0.839**	**0.870**	3.30	8.0
+ PIoU v2	0.896	0.456	0.862	**0.842**	0.851	3.00	8.1
+ ShuffleNetV2 + PIoU v2	**0.908**	**0.462**	**0.896**	**0.871***	**0.883***	3.30	8.0
+ ECA + PIoU v2	0.896	**0.469**	**0.888**	0.814	0.849	3.00	8.1
YieldNet (ShuffleNetV2-ECA + PIoU v2)	**0.915***	**0.473**	**0.905***	**0.855**	**0.879**	3.30	8.0

Bold indicates improvement over baseline ($\geq$1.5% absolute gain). * indicates the best value in the column. All values are reported for the "All" class on the small validation set.

by 0.021 (2.1 percentage points, reaching 0.905), with Precision also showing significant improvement; ShuffleNet V2 improves mAP@50 by 0.028 (2.8 percentage points, reaching 0.912), and Precision by 0.032 (3.2 percentage points, reaching 0.904). This indicates that the group convolution and channel shuffle operations of ShuffleNetV2 extract richer and more discriminative features, greatly reducing false positives caused by background leaves (which are highly similar in color and texture to green tomatoes). The ECA module also strengthens important features and multi-channel connections through its adaptive weighting of feature channels, similarly filtering out background noise and making the network more focused on the target fruit regions in the images.

Furthermore, when ShuffleNet V2 and ECA are combined, the ShuffleNet V2-ECA structure produces a significant synergistic effect, achieving relative improvements of +3.2% and +3.7% on mAP@50 and Precision respectively, while also improving mAP@50-95 by +3.9% (reaching 0.475), Recall by +2.3% (reaching 0.839), and driving the F1 score up by +3.0% (reaching 0.870). ShuffleNet V2, located in the network backbone, passes the richer and more discriminative features it extracts to the ECA module in the lower layers of the network for channel recalibration and adaptive weighting, further focusing the features.

When PIoU v2 acts alone, it only shows significant improvement on Recall (from 0.820 to 0.842, +2.2%). Its working principle is to make bounding boxes converge in a more natural way, but this requires a certain capability of feature extraction from the network. Only when the early-stage features are sufficiently rich and discriminative can more significant effects be produced. When it works together with ShuffleNet V2 or ECA modules, significant improvements in mAP@50-95 can be observed (ShuffleNetV2 + PIoU v2 reaches 0.462, ECA + PIoU v2 reaches 0.469), both higher than when acting alone, which best demonstrates its strengthening effect on bounding box convergence.

On this basis, strengthening multi-component collaborative design, the finally proposed YieldNet (ShuffleNetV2-ECA + PIoU v2) achieves globally optimal performance: on the strict clean validation subset, mAP@50 reaches 0.915 (3.1% improvement over baseline), mAP@50-95 reaches 0.473, Precision reaches 0.905 (3.3% improvement over baseline), and F1 score reaches 0.879, with all core indicators being the optimal or sub-optimal values in the table with almost no additional computational cost (GFLOPs ≈ 8.0, with only 3.30 M parameters). In small-sample agricultural detection tasks, the model shows excellent precision-recall balance, fully verifying the effectiveness of the multi-component collaborative design and providing a robust and efficient benchmark for lightweight object detection.

As shown in Figure 9, the original YOLOv8 model exhibits three missed detection phenomena, and the attention heatmap energy is mostly concentrated on the obvious less-occluded targets in the center of the image, with less attention distribution to highly occluded targets at the edges (such as the middle right of the image). After adding ShuffleNet V2, all targets are correctly identified, and attention spreads toward highly occluded and edge targets compared to the original YOLOv8. After adding ECA to the original YOLOv8, all targets are also correctly identified, and due to its feature weighting enhancement for each channel, the figure shows higher and brighter attention energy on each target, with very uniform distribution.

In the performance of the ShuffleNet V2-ECA structure combining ShuffleNet V2 and ECA modules, compared to when ShuffleNet V2 acts alone, each target shows relatively higher brightness on the attention heatmap (particularly noticeable for the two in the middle right of the figure), while compared to when ECA acts alone, the attention energy for each target is relatively more concentrated in the central region (especially for the four targets in the lower middle part of the figure). This also confirms our conclusion from the data in Table 4 that the combination of the two can produce a synergistic effect.

Furthermore, when PIoU v2 acts alone, the missed detection phenomenon of the original YOLOv8 is improved, but due to the general feature extraction capability of the original network, the attention energy distribution on targets in the figure is relatively scattered, which also corresponds with the data in Table 4 (Recall from 0.820 to 0.842, +2.2%, but other indicators show no significant change). When PIoU v2 is combined with other variants, it can also be observed that thanks to the powerful bounding box regression capability of PIoU v2, bounding boxes converge more closely to ground truth (particularly evident in the two targets in the middle right of the figure, where after adding PIoU v2 they tend to separate more and trend closer to ground truth bounding boxes).

However, the combination of ShuffleNetV2 + PIoU v2 produces a false detection in the middle right of the image; the combination of ECA + PIoU v2 misses the red tomato target. This indicates that the combination of PIoU v2 with a single module is insufficient to unleash its maximum benefit. The final YieldNet (ShuffleNetV2-ECA + PIoU v2) ensures accurate identification of all targets, with more concentrated attention distribution in the central region for each target, and more attention distribution for high-difficulty targets such as highly occluded targets and special red targets, while

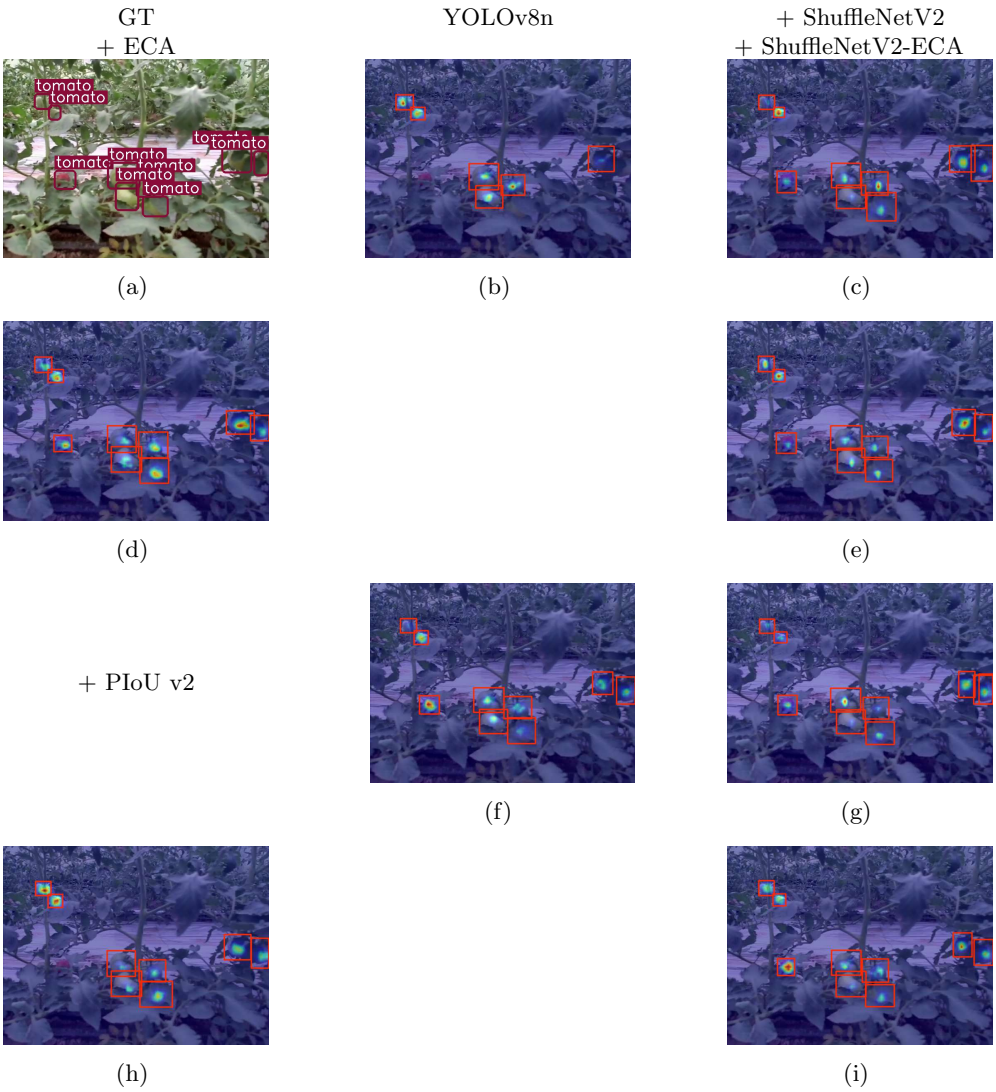


Fig. 9: Ablation visualization on the greenhouse tomato ripeness dataset. (**a**) Ground-truth boxes; (**b**) baseline YOLOv8n; (**c**) YOLOv8n+ShuffleNetV2; (**d**) YOLOv8n+ECA; (**e**) YOLOv8n+ShuffleNetV2+ECA; (**f**)–(**i**) further equipped with PIoU v2. Best viewed in color.

also achieving more precise bounding box localization through PIoU v2, making it the deservedly best-performing combination. This also confirms the excellent performance of this model shown in Table 4.

3.3 Comparison Between Different Models

To further validate the superiority of the proposed YieldNet, this experiment conducts a head-to-head comparison on the two datasets mentioned above between YieldNet and representative nano- and small-scale models from the current mainstream YOLOv5, YOLOv8, and YOLOv11 [25] families, as well as the state-of-the-art transformer-based detector, RT-DETR-n. The detailed comparison results are presented in Table 5.

Table 5: Comparison with other models on large and small validation sets.

Model	mAP@50	mAP@50-95	Precision	Recall	F1	Params (M)	GFLOPs
Large validation set (596 images, 2228 instances)							
YOLOv8n (baseline)	0.926	0.666	0.921	0.868	0.894	3.00	8.1
YieldNet (Ours)	**0.970***	**0.792***	**0.972***	**0.921***	**0.946***	3.30	8.0
YOLOv8s	**0.958**	**0.747**	**0.956**	**0.898**	**0.926**	11.10	28.4
YOLOv5n	0.915	0.641	0.872	0.865	0.868	2.50	7.1
YOLOv5s	**0.960**	**0.721**	**0.952**	**0.907**	**0.929**	9.10	23.8
YOLOv11n	0.920	0.636	0.901	0.856	0.878	2.60	6.3
YOLOv11s	**0.951**	**0.722**	**0.963**	0.868	**0.913**	9.40	21.3
RT-DETR-n	0.869	0.633	0.885	0.844	0.864	15.50	37.4
Small validation set (180 images, 590 instances)							
YOLOv8n (baseline)	0.884	0.457	0.872	0.820	0.845	3.00	8.1
YieldNet (Ours)	**0.915***	**0.473**	**0.905**	**0.855**	**0.879***	3.30	8.0
YOLOv8s	**0.901**	**0.473**	**0.894**	**0.861***	**0.877**	11.10	28.4
YOLOv5n	0.898	0.456	0.849	**0.853**	0.851	2.50	7.1
YOLOv5s	**0.900**	**0.474***	**0.910***	0.824	**0.865**	9.10	23.8
YOLOv11n	0.897	0.453	0.882	**0.834**	0.857	2.60	6.3
YOLOv11s	**0.901**	0.468	0.885	0.835	0.859	9.40	21.3
RT-DETR-n	0.862	0.430	0.824	0.816	0.820	15.50	37.4

Bold indicates improvement over YOLOv8n baseline ($\geq$1.5% absolute gain where applicable).

* indicates the best value in the column for the respective validation set.

All values are reported on the "All" class.

As shown in Table 5, on both the large and small datasets, YieldNet achieves the best results across all performance metrics among all nano-level models.

Furthermore, on the large dataset, YieldNet even achieves the best overall performance on all metrics (mAP@50 of 0.970, mAP@50-95 of 0.792, Precision of 0.972, Recall of 0.921, and F1 of 0.946). On the small dataset, YieldNet similarly achieves the best overall results on mAP@50 (0.915) and F1 (0.879), and its other metrics (mAP@50-95: 0.473, Precision: 0.905, and Recall: 0.855) are second only to YOLOv8s and YOLOv5s, ranking second overall.

In summary, whether on the large dataset or the small dataset, YieldNet is a top-tier performer: its performance on the small dataset is close to that of small-sized models, while on the large dataset it surpasses all small-sized models. With 3.3M parameters and 8.0G FLOPs, it achieves performance that exceeds its class.

3.4 Training Visualization Analysis

To further analyze the dynamic characteristics during the training process and validate the performance improvements, we visualize the mAP@50-95 curves and the training loss components of the models mentioned in the previous section (Comparison Between Different Models) on the two datasets.

Figure 10 shows the validation mAP@50 and mAP@50-95 curves during training, illustrating how mAP@50 and mAP@50-95 changes with the number of epochs; Figure 11 displays the training loss curves, including the box loss, classification loss, and DFL loss (also GIoU loss and L1 loss for RT-DETR model).

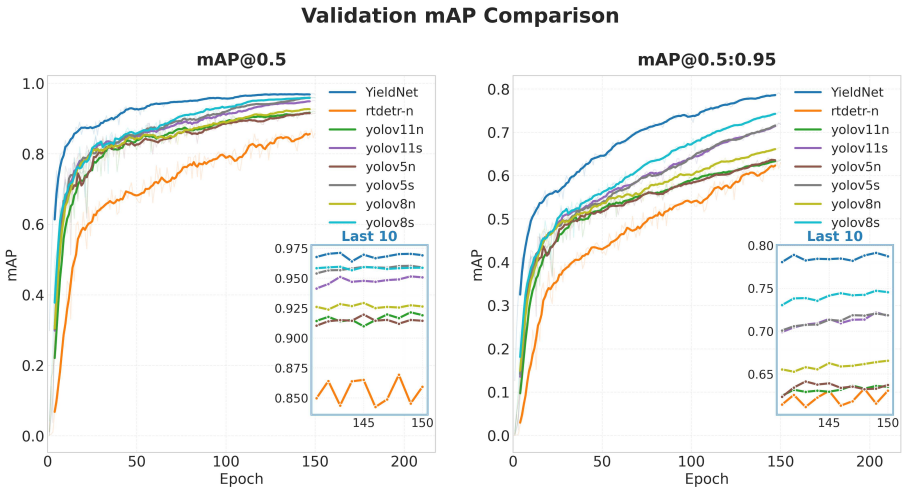
Figure 10 illustrates the validation mAP@50 and mAP@50–95 curves during the training process. It is evident that, in both datasets, YieldNet significantly outperforms other YOLO variants in the early stages of training; by the final convergence phase, it achieves the highest mAP on both datasets. This indicates that YieldNet not only converges more rapidly but also exhibits excellent robustness.

Figure 11 presents the training loss curves for each component, including Box Loss, Classification Loss (Cls Loss), and Distributed Focal Loss (DFL Loss) (RT-DETR model is not included in the comparison scope). It can be observed that, in both datasets, YieldNet's loss components decrease most rapidly and reach the lowest loss values among all models by the end of training, with a particularly notable advantage in Box Loss.

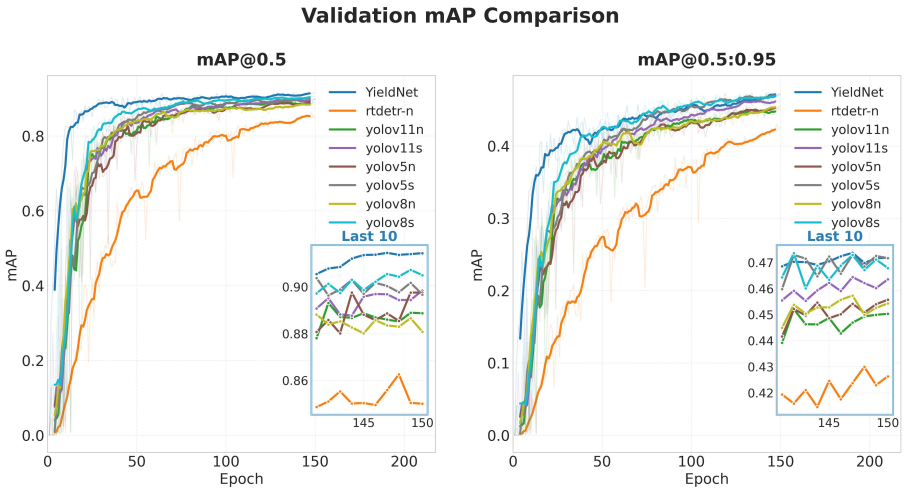
In summary, YieldNet demonstrates a significant advantage from the initial stages of training: under scenarios with limited computational resources or a restricted number of training epochs, it can achieve optimal performance with the same number of epochs; moreover, throughout the entire training period, its final convergence performance also surpasses that of most comparative models. Particularly on small datasets and with low epoch settings, YieldNet shows a remarkable balance of efficiency and accuracy, making it a cost-effective lightweight detection model.

4 Discussion

This study proposes YieldNet, a lightweight green-mature tomato detector built upon an improved YOLOv8n backbone, specifically designed for real-time UAV-based detection and counting of green tomatoes in greenhouses. Through the redesign of the backbone, neck, and loss function, the model achieves a significant performance improvement: On the large validation set, YieldNet achieves significant improvements over the YOLOv8n baseline model: mAP@50 increases from 0.926 to 0.970 (4.8% improvement), mAP@50-95 increases from 0.666 to 0.792 (18.9% improvement), Precision increases from 0.921 to 0.972 (5.5% improvement), Recall increases from 0.868 to 0.921 (6.1% improvement), and F1 score increases from 0.894 to 0.946 (5.8% improvement). On the small validation set, YieldNet similarly demonstrates obvious



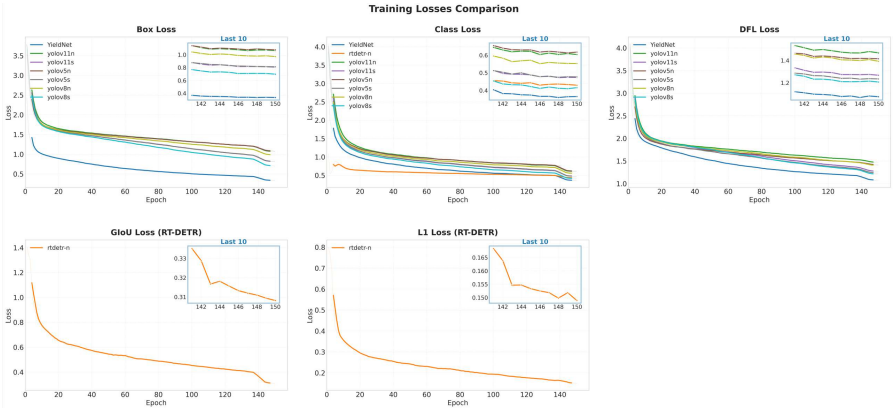
(a) Large Dataset



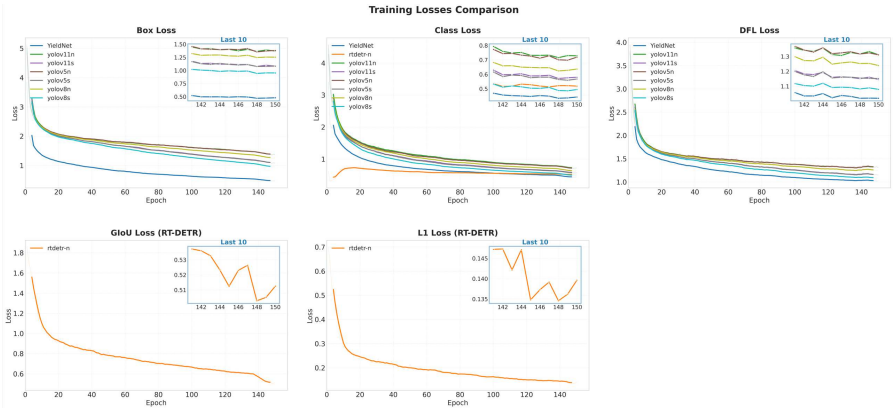
(b) Small Dataset

Fig. 10: Validation mAP@50 and mAP@50-95 curves during training on the large and small datasets.

advantages: mAP@50 increases from 0.884 to 0.915 (3.5% improvement), mAP@50-95 increases from 0.457 to 0.473 (3.5% improvement), Precision increases from 0.872 to 0.905 (3.8% improvement), Recall increases from 0.820 to 0.855 (4.3% improvement), and F1 score increases from 0.845 to 0.879 (4.0% improvement). These improvements stem from the enhanced representation of target objects by ShuffleNetV2, the ECA attention mechanism embedded in the neck that effectively separates green fruits from



(a) Large Dataset



(b) Small Dataset

Fig. 11: Validation loss curves during training on the large and small datasets.

the leaf background, and the size-adaptive, non-monotonic focusing mechanism of the PIoU v2 loss that optimizes bounding-box regression.

In comparative experiments with several mainstream lightweight detectors, YieldNet surpasses all same-size (nano) models, ranking first in mAP@50 and Precision, with other performance indicators also ranking second on the small dataset, making it the best model in comprehensive evaluation. It even further surpasses all small-sized models on the large dataset, ranking first in all performance indicators. Moreover, YieldNet achieves more than 10% higher precision than the state-of-the-art Transformer architecture RT-DETR-n with only 1/5 of the computational cost. This result fully demonstrates the excellent detection capability and competitiveness of YieldNet under lightweight constraints. Despite an increase in parameters from 3.0M to 3.3M, the FLOPs decrease from 8.1G to 8.0G, almost maintaining the computational footprint unchanged, fully meeting the real-time requirements for UAV detection in

greenhouses. Thanks to the systematic screening and fine-tuning of lightweight modules, we finally adopted the ShuffleNetV2 backbone and ECA attention, achieving a reduction in FLOPs and an improvement in accuracy with only an additional 0.3M parameters. The model significantly boosts performance with virtually no extra computational burden, truly realizing "more with less," which is the core contribution of this study.

In related research, most tomato detection efforts have primarily focused on ripe or mixed-ripe fruits. Studies targeting the unripe green tomatoes, which hold the most predictive value for yield, are still very scarce, and those that combine the use of UAVs in low-altitude scenarios are even rarer. Among the few studies that incorporate UAVs, such as the work by Egi et al. [13], although they validated the model's application effects in real greenhouse environments, they directly used the original YOLOv5 network, resulting in a clear deficiency in identifying green tomatoes. Compared to the red tomato category, the identification performance for the green tomato category has decreased by 35.0%, 31.0%, and 39.1% in Recall, mAP@50, and mAP@95, respectively. This study, by making ultra-low-cost improvements to the original YOLOv8 network, and double-verifying it with our own real UAV-captured green tomato dataset as well as public tomato datasets. The results show that the model performance is significantly enhanced, with the relative gap in key indicators such as mAP@50-95 and Recall for red and green fruits compressed to within 10%, truly filling the research gap of "UAVs + targeted green unripe tomatoes + lightweight improved models."

However, this study still has several limitations: the images in our self-built dataset were all collected from the same tomato greenhouse, which may raise concerns about excessive data similarity and insufficient generalization capability of the model structure. To address this concern, we also introduced testing on the public dataset Tomato-Recog in both baseline comparison experiments and mainstream model comparison experiments. The results show that even in cross-dataset and cross-task scenarios (single-class detection task to multi-class detection task), YieldNet still maintains excellent performance and leadership, which to a certain extent demonstrates that our structure possesses robust generalization capability across different agricultural scenarios.

Due to the lack of hardware conditions, we were unable to conduct deployment tests on real edge devices. However, the model parameters are only 3.3M, and the computational cost is strictly controlled at 8.0 GFLOPs, making it a standard nano-level model. Using the YOLOv8n model with very similar size as a reference, we estimate that YieldNet can easily run on common edge processors such as NVIDIA Jetson Nano or Orin NX within a 10W–15W power consumption range, and we estimate that it can achieve real-time inference speed exceeding 30 FPS, meeting the real-time requirements for UAV-based pre-harvest monitoring. Although the estimation is ideal, its actual performance still awaits further experimental verification.

We plan to expand the scale of the dataset in future research work, incorporating more diverse real agricultural scenarios and images under different lighting and climatic conditions, to further improve the generalization performance of the model. In addition, we also plan to deploy the model on common edge computing devices to conduct an accurate evaluation of its real-time computing performance. Furthermore,

we will also conduct field tests and compare them with manual counting results, using more comprehensive experiments to demonstrate its practical

5 Conclusions

In summary, this study proposes a novel object detection model architecture—YieldNet—based on the UAV platform and targeting immature tomato fruit detection. It is a novel object detection structure obtained by transforming the basic algorithmic structure idea of the YOLOv8n model through a series of steps: ShuffleNetV2 backbone network fusion, neck ECA attention module addition, and PIoU v2 improved loss function. These designs enable our YieldNet, compared with the original YOLOv8n model, to better capture fine-grained features, more efficiently concentrate attention on high-weight features, and possess higher bounding box localization accuracy. These upgrades mutually reinforce each other, collectively achieving YieldNet's capability to, in complex real field scenarios, distinguish complex background elements such as leaves and branches highly similar to fruits and effectively identify green tomato. This investigation leveraged both self-acquired UAV imagery of green tomatoes in greenhouses and consulted public tomato repositories to cross-validate the architecture. Empirical results reveal that our architecture surpasses the foundational YOLOv8n architecture: On the large-scale public dataset, YieldNet achieves significant relative improvements across all metrics, with Precision improving by 5.5% (from 0.921 to 0.972), F1-score improving by 5.8% (from 0.894 to 0.946), and most notably mAP@50-95 substantially improving by 18.9% (from 0.666 to 0.792). On the small-scale self-built dataset, YieldNet similarly demonstrates stable superiority, improving mAP@50 by 3.5% (from 0.884 to 0.915), mAP@50-95 by 3.5% (from 0.457 to 0.473), and F1-score by 4.0% (from 0.845 to 0.879). This substantiates YieldNet's exceptional competence in addressing the challenge of detecting green tomatoes in demanding field environments. Moreover, the architecture merely marginally expanded parameters from 3.0 M to 3.3 M, and FLOPs decreased from 8.1 G to 8.0 G, considerably elevating performance at a negligible expense. This also suggests that the architecture could potentially function as a feasible solution for accurate early-stage tomato yield forecasting.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant No. 62471049).

Declarations

The authors declare no conflicts of interest or competing interests. The dataset and codes supporting the conclusions of this article will be made available by the authors on request.

References

[1] L. Kitinoja, H.Y. AlHassan, Identification of appropriate postharvest technologies for small scale horticultural farmers and marketers in Sub-Saharan Africa and

South Asia—Part 1. Postharvest losses and quality assessments, *Acta Hortic.* **934** (2012) 31–40. https://doi.org/10.17660/actahortic.2012.934.1.

[2] F.J. Izdori, D. Mkwambisi, S.T. Karuaihe, E. Papargyropoulou, Multi-stakeholder collaboration framework for post-harvest loss reduction: the case of tomato value chain in Iringa and Morogoro regional in Tanzania, *Agricultural and Food Economics* **13** (2025) 6. https://doi.org/10.1186/s40100-025-00351-z.

[3] S.M. Shawon, F.B. Ema, A.K. Mahi, F.L. Niha, H.T. Zubair, Crop yield prediction using machine learning: An extensive and systematic literature review, *Smart Agricultural Technology* **10** (2025) 100718. https://doi.org/10.1016/j.atech.2024.100718.

[4] D.C. Tsouros, S. Bibi, P.G. Sarigiannidis, A review on UAV-based applications for precision agriculture, *Information* **10** (2019) 349. https://doi.org/10.3390/info10110349.

[5] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, I. Moscholios, A compilation of UAV applications for precision agriculture, *Computer Networks* **172** (2020) 107148. https://doi.org/10.1016/j.comnet.2020.107148.

[6] P. Hu, R. Zhang, J. Yang, L. Chen, Development status and key technologies of plant protection UAVs in China: A review, *Drones* **6** (2022) 354. https://doi.org/10.3390/drones6110354.

[7] A. Rejeb, A. Abdollahi, K. Rejeb, H. Treiblmaier, Drones in agriculture: A review and bibliometric analysis, *Computers and Electronics in Agriculture* **198** (2022) 107017. https://doi.org/10.1016/j.compag.2022.107017.

[8] H. Zhou, F. Huang, W. Lou, Q. Gu, Z. Ye, H. Hu, X. Zhang, Yield prediction through UAV-based multispectral imaging and deep learning in rice breeding trials, *Agricultural Systems* **223** (2025) 104214. https://doi.org/10.1016/j.agsy.2024.104214.

[9] T.B. Shahi, C.-Y. Xu, A. Neupane, W. Guo, Machine learning methods for precision agriculture with UAV imagery: A review, *Electronic Research Archive* **30** (2022) 4277–4317. https://doi.org/10.3934/era.2022218.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016) 779–788. https://doi.org/10.1109/cvpr.2016.91.

[11] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, arXiv preprint arXiv:1804.02767, 2018. https://doi.org/10.48550/arXiv.1804.02767.

[12] M. Sohan, T. Sai Ram, Ch.V.R. Reddy, A review on YOLOv8 and its advancements, In: *Data Intelligence and Cognitive Informatics*, Springer, Singapore, (2024) 529–545. https://doi.org/10.1007/978-981-99-7962-2_39.

[13] Y. Egi, M. Hajyzadeh, E. Eyceyurt, Drone-computer communication based tomato generative organ counting model using YOLOv5 and Deep-SORT, *Agriculture* **12** (2022) 1290. https://doi.org/10.3390/agriculture12091290.

[14] A. Wang, Y. Xu, D. Hu, L. Zhang, A. Li, Q. Zhu, J. Liu, Tomato yield estimation using an improved lightweight YOLO11n network and an optimized region tracking-counting method, *Agriculture* **15** (2025) 1353. https://doi.org/10.3390/agriculture15131353.

[15] S. Chai, M. Wen, P. Li, Z. Zeng, Y. Tian, DCFA-YOLO: A dual-channel cross-feature-fusion attention YOLO network for cherry tomato bunch detection, *Agriculture* **15** (2025) 271. https://doi.org/10.3390/agriculture15030271.

[16] 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, *Computers and Electronics in Agriculture* **216** (2024) 108533. https://doi.org/10.1016/j.compag.2023.108533.

[17] Z. Xu, T. Luo, Y. Lai, Y. Liu, W. Kang, EdgeFormer-YOLO: A lightweight multi-attention framework for real-time red-fruit detection in complex orchard environments, *Mathematics* **13** (2025) 3790. https://doi.org/10.3390/math13233790.

[18] N. Ma, X. Zhang, H.-T. Zheng, J. Sun, ShuffleNet V2: Practical guidelines for efficient CNN architecture design, In *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018) 116–131. https://doi.org/10.1007/978-3-030-01264-9_8 .

[19] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020) 11531–11539. https://doi.org/10.1109/cvpr42600.2020.01155.

[20] C. Liu, K. Wang, Q. Li, F. Zhao, K. Zhao, H. Ma, Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism, *Neural Networks* **170** (2024) 276–284. https://doi.org/10.1016/j.neunet.2023.11.041.

[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** (2022) 856. https://doi.org/10.3390/agriculture12060856.

[22] T. Zeng, S. Li, Q. Song, F. Zhong, X. Wei, Lightweight tomato real-time detection method based on improved YOLO and mobile deployment, *Computers and Electronics in Agriculture* **205** (2023) 107625. https://doi.org/10.1016/j.compag.2023.107625.

[23] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018) 6848–6856. https://doi.org/10.1109/cvpr.2018.00716.

[24] Q. Dong, L. Sun, T. Han, M. Cai, C. Gao, PestLite: A novel YOLO-based deep learning technique for crop pest detection, *Agriculture* **14** (2024) 228. https://doi.org/10.3390/agriculture14020228.

[25] M. Kotthapalli, D. Ravipati, R. Bhatia, YOLOv1 to YOLOv11: A comprehensive survey of real-time object detection innovations and challenges, arXiv preprint arXiv:2508.02067, 2025. https://doi.org/10.48550/arXiv.2508.02067.