

Article

LGD-YOLO: An Asymmetric Lightweight Network with Dynamic Feature Alignment for Greenhouse Tomato Maturity Detection

Xing Xu ^{1,2} , Aixiang Wu ^{1,2}, Yun Zhao ^{1,2,*} , Na Wu ^{1,2}, Dan Yang ^{1,2}, Yanan Mi ³ and Petr Skobelev ⁴ 

¹ School of Artificial Intelligence and Information Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; xuxing@zust.edu.cn (X.X.)

² Zhejiang-Finland Joint Laboratory of Intelligent Perception Computing and Equipment, Hangzhou 311100, China

³ Department of Business Development, Pegasor Oy, 33100 Tampere, Finland

⁴ Samara Federal Research Scientific Center, Russian Academy of Sciences, 443001 Samara, Russia

* Correspondence: yunzhao@zust.edu.cn

Abstract

Greenhouse tomato detection faces critical challenges due to dense fruit occlusion, background interference, and the stringent computational constraints inherent to edge-deployed harvesting robots. Resolving these bottlenecks requires efficient architectures. We propose LGD-YOLO as an asymmetric lightweight network adapted from YOLOv10n specifically for edge-based tomato maturity detection. The architecture integrates a C2f-GMKSF module utilizing grouped multi-kernel convolutions to extract multi-scale textures with limited computational overhead. Precise feature alignment under occluded conditions is subsequently achieved through a Dy-HSFPN structure, accompanied by a C2f-CFCGLU module that expands the receptive field while preserving linear complexity. Furthermore, replacing the traditional detection head with a partial convolution head reduces memory access costs. A Focaler-Wise-SIoU loss function is utilized to stabilize bounding box regression against the lightweight penalty without introducing inference latency. Performance evaluations on a custom three-class dataset with a 180-image test set yield an 88.0% mAP@50. Relative to the baseline model, LGD-YOLO improves detection accuracy by 0.6 percentage points while shrinking the parameter volume by 37.6% to 1.41 M and lowering computational demand by 41.5% to 3.8 GFLOPs. Hardware deployment on an NVIDIA Jetson AGX Orin achieves a sustained processing speed of 40.6 FPS, while the weight file is 2.99 MB, supporting its feasibility for real-time agricultural robotics.

Keywords: tomato detection; maturity detection; YOLOv10; lightweight model; edge deployment; greenhouse agriculture



Academic Editor: Hongbin Pu

Received: 8 June 2026

Revised: 4 July 2026

Accepted: 10 July 2026

Published: 14 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Tomato represents one of the most extensively cultivated and economically vital solanaceous vegetables in global protected agriculture, playing a crucial role in the international vegetable trade [1,2]. The rapid expansion of protected horticulture over the past two decades has popularized large-scale intensive greenhouse tomato cultivation. Unlike traditional open-field farming, greenhouse tomatoes are typically grown using high-density hanging vine systems for year-round production, resulting in clustered fruit distributions and extended ripening periods where immature, turning, and fully ripe fruits

frequently coexist on the same truss [3]. This growth pattern renders harvesting operations highly dependent on human experience, requiring intensive labor with low efficiency. Given the continuous agricultural labor shortage and escalating labor costs [4], developing automated fruit detection and selective harvesting technologies serves as a critical pathway for upgrading the greenhouse tomato industry.

Precise maturity assessment forms the foundation for targeted robotic harvesting. Different maturity stages exhibit significant variations in quality metrics, including firmness, soluble solids content, and acidity, and accurate recognition directly informs optimal harvest timing, minimizes post-harvest losses, and improves yield estimation reliability [5]. In this study, three maturity labels were defined based on visually distinguishable surface coloration (unripe, semi-ripe, and ripe), following the three-stage scheme commonly adopted in automated harvesting studies to balance annotation efficiency and production decision needs. Detailed annotation criteria and class labels are provided in Section 2.1.1. Greenhouse environments, nevertheless, introduce compounded visual detection obstacles. Immature tomatoes share high visual similarity with background foliage, creating severe green-on-green interference [6]. Furthermore, dense clustering and mutual occlusion, combined with overlapping stems and leaves, cause blurred edge contours and localized information loss [7]. Compounding these visual challenges, agricultural robots equipped with vision systems operate under strict power and computational constraints. Conventional deep convolutional neural networks provide high accuracy but often suffer from substantial inference latency and energy consumption, rendering them unsuitable for real-time field operations [8,9]. Achieving high-precision and low-latency tomato detection under the dual constraints of complex greenhouse environments and limited edge resources remains a primary challenge for autonomous agricultural harvesting.

Existing deep-learning studies for agricultural object detection can be organized according to two coupled requirements: reliable perception in visually complex crop canopies and computational feasibility on mobile platforms [10]. Two-stage algorithms represented by Faster R-CNN [11] and Mask R-CNN [12] establish high-precision benchmarks in complex scenarios. In tomato-related applications, Faster R-CNN has been adapted for green and red tomato maturity assessment, achieving 96.14% average precision in its experimental setting [13], while Mask R-CNN has been used to segment mature green greenhouse tomatoes with an F1 score of 92.0% [14]. These methods demonstrate the value of deep feature extraction for maturity perception; however, their region-proposal pipelines, large parameter volumes, and computational overhead restrict deployment on resource-constrained mobile robotic platforms requiring real-time responsiveness.

For real-time harvesting scenarios, one-stage detectors are generally more suitable because they eliminate explicit region proposal and offer faster inference [15]. The YOLO series has therefore been widely used for agricultural fruit detection [16,17], particularly for small targets, occlusion, and maturity-related visual variation. Lawal employed an improved YOLOv3 with a MixNet backbone for tomato counting in natural scenes [18]; Miao et al. designed a YOLOv5-based method for tomato maturity estimation and stem localization [19]; Zhao et al. proposed Ta-YOLO, integrating multi-dimensional attention and an improved bounding box loss to mitigate missed detections of small tomatoes under occlusion [1]. Other tomato-oriented variants, such as SE-YOLO and GAE-YOLO, further introduced edge enhancement, Ghost convolution, and activation-function redesign to improve accuracy and deployment potential [20,21]. Because these reported metrics were obtained under different crop environments, maturity definitions, IoU thresholds, and hardware platforms, they are summarized here to indicate methodological trends rather than direct rankings. Overall, existing YOLO-based studies show that targeted architectural

improvements can enhance feature discrimination, but many still emphasize detection accuracy more strongly than lightweight edge deployment.

To address computational redundancy, lightweight backbone networks have been developed by replacing standard convolutions with depthwise separable convolutions, low-cost feature generation, or channel shuffling, as represented by MobileNetV3 [22], GhostNet [23], and ShuffleNetV2 [24]. These designs provide foundational support for compact agricultural perception models. In facility and robotic agriculture, lightweight perception has also been pursued in crop-specific tasks, ranging from vision-based banana rachis detection and cut-off-point localization for robotic operation [25] to tomato maturity detection models such as YOLOv8n-FDE [26] and SWMD-YOLO [27]. The latter two methods improve efficiency through feature-fusion redesign, adaptive sampling, lightweight detection heads, and compact convolutional operators. Nevertheless, many lightweight detectors still rely on conventional post-processing and detection-head structures, leaving memory access and terminal deployment efficiency insufficiently explored.

End-to-end detection architectures provide a complementary route for reducing deployment complexity. Most YOLO-based lightweight improvements retain non-maximum suppression (NMS) post-processing; although the exact latency contribution of NMS depends on candidate-box density, confidence thresholds, and hardware implementation, it can introduce additional post-processing cost in densely clustered crop scenes [28]. YOLOv10, proposed in 2024, removes NMS post-processing through consistent dual assignments and achieves end-to-end detection [29], and has recently been adopted in smart-agriculture applications. Lu et al. proposed PGLD-YOLO for lightweight pomegranate localization and recognition [30], Li et al. developed D3-YOLOv10 for facility-scenario tomato detection [31], and Wu et al. proposed YOLO-PGC for tomato maturity detection based on YOLOv11 [32]. These studies validate the potential of end-to-end or recent YOLO architectures for agricultural detection. However, agricultural applications of NMS-free architectures remain limited, and the detection head often still follows traditional convolutional designs, which may constrain the inference-efficiency advantage expected from NMS-free deployment.

Based on the above analysis, this study proposes LGD-YOLO, a lightweight greenhouse tomato maturity detection model based on the YOLOv10n framework for edge deployment. The main contributions are as follows:

1. A multi-maturity tomato dataset was constructed under real greenhouse conditions, covering three operational maturity groups: unripe, semi-ripe, and ripe. Practical factors such as illumination variation, shooting distance, fruit count, and occlusion degree were considered, and multiple data augmentation strategies were applied to improve sample diversity.
2. An asymmetric lightweight detection architecture was designed around three core components. The C2f-GMKSF module uses grouped heterogeneous convolution to strengthen fruit texture and edge extraction in shallow layers for green-on-green interference. The C2f-CFCGLU module uses large-kernel separable convolution to expand the receptive field in deeper layers for structural perception under occlusion. The Dy-HSFPN combines DySample dynamic upsampling with high-level semantic screening to improve feature alignment and reduce the influence of visually similar background textures.
3. Hardware-aware lightweight and training-stage optimization strategies were incorporated. A PConv head based on partial convolution was used to reduce memory access cost in the detection head, while the Focaler-Wise-SIoU loss function was introduced to improve bounding-box optimization during training without changing the inference structure.

4. The transferability of the proposed modules was evaluated by migrating the core components to the YOLOv8n framework, providing evidence of their plug-and-play potential under lightweight deployment settings.

2. Materials and Methods

2.1. Dataset Construction

2.1.1. Image Acquisition and Preprocessing

The greenhouse tomato cultivation facility at the Wuzhen International Agricultural Internet Exposition Park in Zhejiang Province, China, was selected as the experimental site for data collection, as illustrated in Figure 1a,b. The cultivated tomato variety was ‘Jiahong No. 1’, managed under a substrate culture system with a row spacing of approximately 1.4 m and a row length of roughly 20 m. Data acquisition was conducted from 3 April to 30 May 2024, during which tomatoes at different growth stages were cultivated for market supply. The experimental area contained 160 plants arranged in two rows, each extending approximately 20 m. To replicate the operational perspective of an autonomous harvesting robot, a SCOUT 2.0 (AgileX Robotics, Shenzhen, China) mobile robot platform equipped with an iPhone 14 Pro (Apple Inc., Cupertino, CA, USA) mounted horizontally was deployed for image acquisition. During collection, the robot moved at a constant speed along each row, sequentially photographing plants from left to right and then returning from right to left to capture the opposite viewing direction. The acquired images were subsequently uploaded to a PC for standardization, annotation, and further processing. The camera was securely mounted in a horizontal orientation with an adjustable height mechanism, as depicted in Figure 2. Image collection was conducted to encompass various growth phases of the crop. All images were captured under ambient natural illumination, incorporating direct sunlight, partial shadows, and dynamic lighting fluctuations to enhance the diversity of the collected dataset. The imaging process therefore reflected practical greenhouse robotic perception scenarios rather than a clean-background laboratory setup.

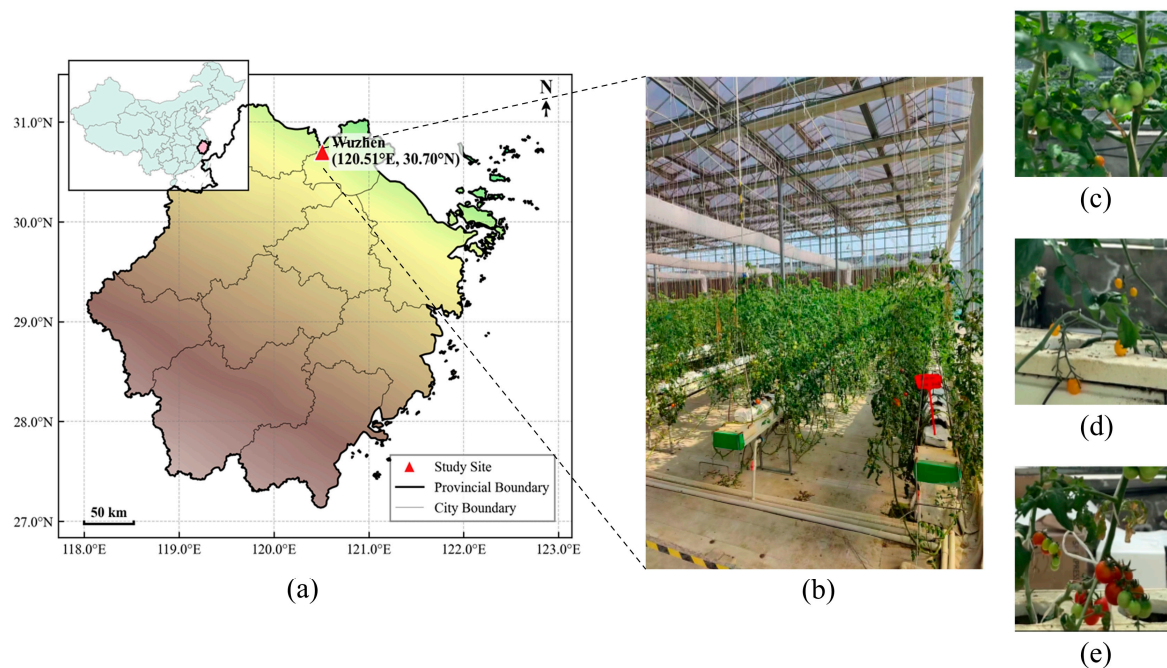


Figure 1. Overview of the data acquisition environment and tomato fruit samples. (a) Geographic location of the tomato data collection site; (b) greenhouse environment used for tomato cultivation and image acquisition; (c–e) representative examples of tomato fruits at different maturity stages (unripe, semi-ripe, and ripe).



Figure 2. The SCOUT 2.0 autonomous mobile platform operating in the greenhouse environment. The platform traverses the ridges to simulate the trajectory of robotic harvesting under real-world unstructured conditions.

A total of 3205 raw images were collected and uniformly resized to 640×640 pixels using letterbox padding to preserve the original aspect ratio. Manual annotation was performed using the Labelling tool (version 1.8.6). Annotation categories were defined primarily according to external fruit color and the practical harvesting and yield-forecasting requirements of the cooperating greenhouse farm. Specifically, fruits with predominantly green surfaces were labelled unripe, fruits showing visible yellow-to-red transition but not dominant red coloration were labelled semi-ripe, and fruits with dominant red surface coloration were labelled ripe, as demonstrated in Figure 1c–e. Borderline samples were assigned according to the dominant visible surface color and then checked during annotation review to maintain label consistency. Fruit size was not used as an independent maturity criterion; it was considered only when separating visible fruit instances during bounding-box annotation. All visible fruits, including those partially occluded by leaves or stems, were annotated using minimum bounding boxes.

2.1.2. Dataset Partitioning and Augmentation

The collected dataset was partitioned into training, validation, and testing sets containing 2826, 199, and 180 images, respectively, based on chronological acquisition order and environmental diversity. Images from the same acquisition sequence were assigned to the same subset where possible to reduce leakage caused by visually similar adjacent frames. Table 1 details the distribution of images and annotated bounding boxes across different maturity categories within each subset. The statistical analysis reveals a pronounced class imbalance, with unripe tomatoes predominating the dataset with 82,710 instances, whereas semi-ripe and ripe instances are substantially less frequent, totaling 1682 and 6157 instances, respectively. Furthermore, a significant proportion of the targets exhibit small spatial scales coupled with partial occlusions, presenting an additional confound for precise maturity recognition. To avoid masking minority-class behavior, class-specific precision, recall, mAP@50 and mAP@50:95, together with confusion-matrix analysis, are reported in Section 3.3.

To mitigate the effects of class imbalance and enhance model adaptability to variable imaging conditions, several offline data augmentation techniques were applied to both the training and validation sets after the train and validation split. These operations included

random rotation and translation, horizontal flipping, illumination adjustment, Gaussian noise injection, and localized spatial cutout. For geometric transformations, bounding-box coordinates were transformed consistently with the image transformation; boxes were clipped to the valid image region, and severely truncated boxes were removed when the remaining visible area was insufficient for reliable annotation. Photometric transformations, including brightness variation and Gaussian noise, preserved the original box coordinates. The reported test metrics were calculated only on the original, non-augmented test images. Representative examples of the augmented images are illustrated in Figure 3.

Table 1. Composition of the Tomato Maturity Dataset.

Dataset	Number of Images	Number of Bounding Box			
		Total	Unripe	Ripe	Semi-Ripe
Training set	2826	80,324	73,449 (91.44%)	5415 (6.74%)	1460 (1.82%)
Validation set	199	5568	5074 (91.13%)	414 (7.44%)	80 (1.44%)
Test set	180	4657	4187 (89.91%)	328 (7.04%)	142 (3.05%)
Complete dataset	3205	90,549	82,710 (91.34%)	6157 (6.80%)	1682 (1.86%)

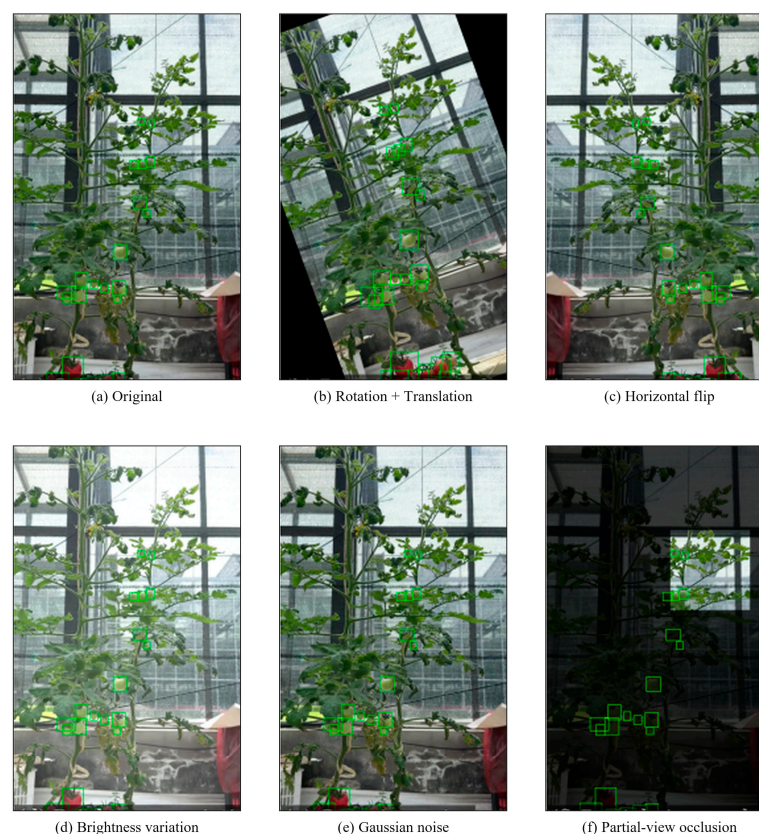


Figure 3. Visualization of data augmentation strategies applied to the training set. (a) Original image; (b) rotation and translation; (c) horizontal flip; (d) brightness variation; (e) Gaussian noise injection; (f) partial-view occlusion. For geometric transformations, bounding boxes were transformed consistently with the image geometry; photometric transformations preserved the original box coordinates.

2.2. LGD-YOLO Lightweight Detection Architecture

To improve tomato detection on resource-constrained edge computing platforms, this study develops LGD-YOLO, an asymmetric lightweight model built upon YOLOv10n. The method is motivated by the practical conflict between preserving fine-grained fruit boundaries under occlusion and reducing computation and memory access for edge-

deployed agricultural robots. Therefore, the architecture is organized around task-specific modifications to the shallow backbone, neck, detection head, and training loss.

Compared with YOLOv10n, C2f-GMKSF is inserted into the shallow backbone to preserve texture and edge cues, Dy-HSFPN replaces the original feature-fusion path to improve content-aware alignment, C2f-CFCGLU is used at selected neck fusion nodes to enlarge contextual modeling, and PConv is applied to the P3/P4/P5 detection heads to reduce prediction-stage computation. The Focaler-Wise-SIoU loss is used only during training to adjust the bounding-box optimization objective and does not change the inference graph.

The novelty of this design lies not in claiming each operator as a new standalone module, but in the task-oriented integration and placement of these techniques within an NMS-free lightweight framework for greenhouse tomato maturity detection. Grouped convolutions, large-kernel depthwise convolutions, dynamic upsampling, and partial convolution have been explored in other contexts; in LGD-YOLO, they are reorganized according to an asymmetric principle that allocates shallow computation to texture preservation, neck computation to feature alignment and semantic screening, and head computation to memory-efficient prediction. This design choice is intended to balance green-on-green discrimination, occlusion-aware feature fusion, and edge-device efficiency. The overall architecture is illustrated in Figure 4, with each module detailed in the subsequent subsections.

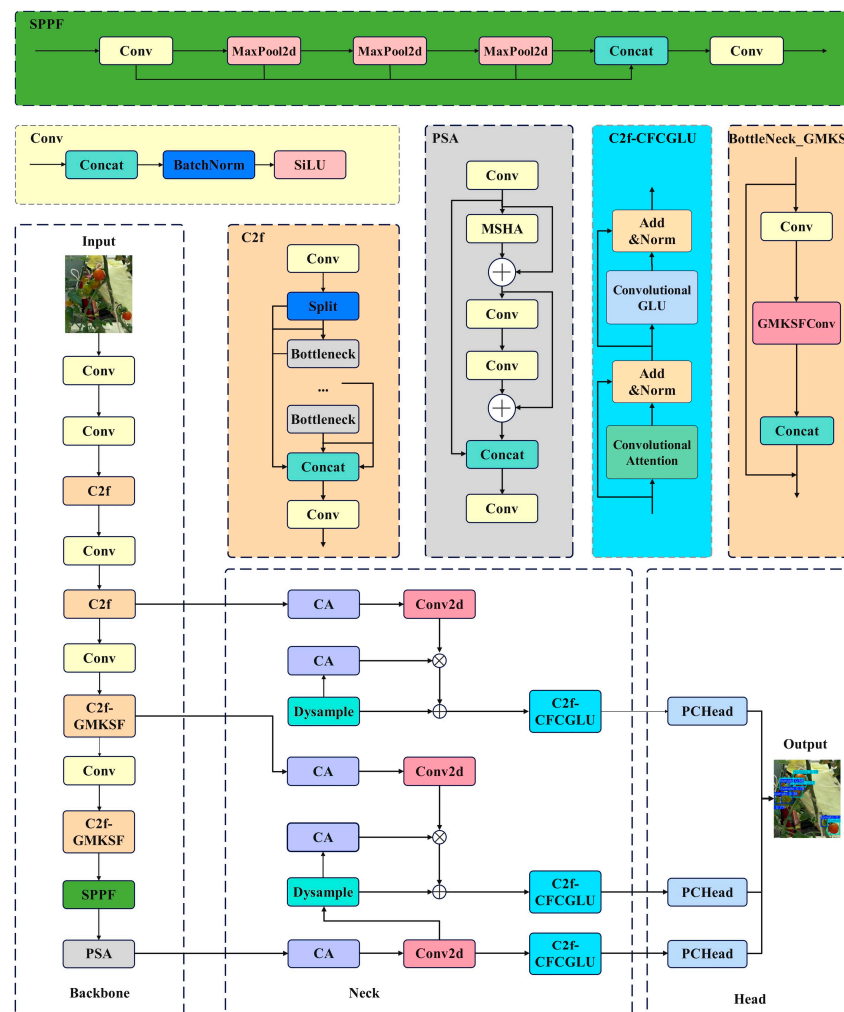


Figure 4. Overall architecture of the proposed LGD-YOLO model. The backbone combines C2f-GMKSF for shallow texture extraction and C2f-CFCGLU for deep contextual encoding, the neck uses Dy-HSFPN for dynamic feature alignment, and the P3/P4/P5 detection heads use PConv for memory-efficient prediction.

2.2.1. C2f-GMKSF: Shallow Multi-Scale Feature Extraction Module

Greenhouse tomato images often contain unripe fruits whose boundaries and color-texture cues are close to surrounding leaves and stems; therefore, the shallow backbone must preserve fine fruit-edge information without relying on dense high-resolution convolution [33]. C2f-GMKSF is inserted into the shallow backbone to replace the corresponding standard C2f block in YOLOv10n, as illustrated in Figure 5. Different from the original homogeneous C2f processing, C2f-GMKSF groups channels and applies heterogeneous kernel sizes to different branches. The key mechanism is to represent local fruit boundaries and nearby cluster context in parallel while avoiding the use of large kernels on all channels.

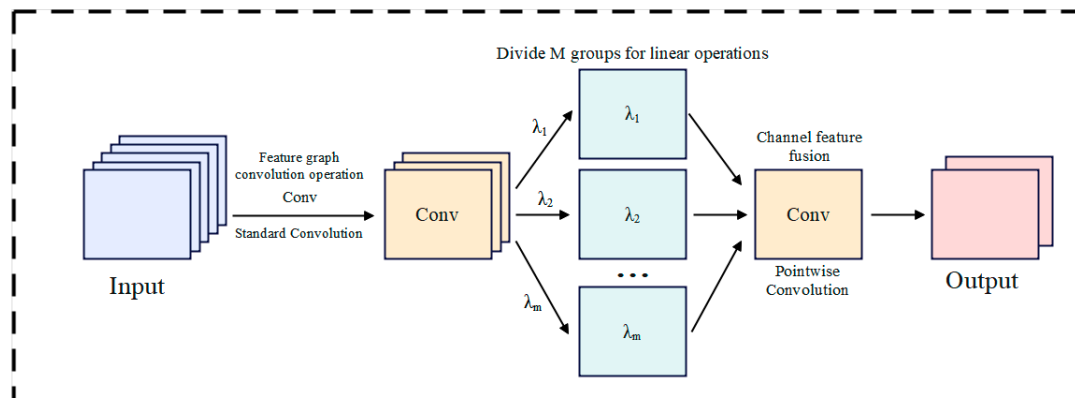


Figure 5. Schematic structural topology of the Grouped Multi-Kernel Spatial Fusion (GMKSF) unit.

Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$, GMKSF first partitions the channels into non-overlapping subgroups:

$$X = [X_1, X_2, \dots, X_m], X_i \in \mathbb{R}^{\frac{C}{m} \times H \times W}, \quad (1)$$

Each subgroup is processed by a convolutional branch with a different kernel size, allowing the shallow backbone to represent fruit textures and local structures at multiple scales [34]:

$$Y_i = \mathcal{F}_{k_i}(X_i), i = 1, \dots, m, \quad (2)$$

Small kernels, such as 3×3 , emphasize local fruit boundaries and surface texture, whereas larger kernels, such as 5×5 and 7×7 , introduce neighborhood context for clustered or partially occluded fruits. The branch outputs are concatenated and fused by pointwise convolution:

$$Z = \mathcal{P}(\text{Concat}(Y_1, \dots, Y_m)). \quad (3)$$

In Equations (1)–(3), X denotes the input feature map with C channels and spatial size $H \times W$; m is the number of channel groups; X_i is the i -th grouped feature; $\mathcal{F}_{k_i}$ denotes the convolution branch with kernel size k_i ; Y_i is the branch output; $\mathcal{P}(\cdot)$ is the 1×1 pointwise fusion operation; and Z denotes the fused GMKSF output.

In the overall architecture, C2f-GMKSF contributes shallow texture preservation for green-on-green scenes. The ablation study in Section 3.1 evaluates this role together with parameter volume, GFLOPs, and FPS under the same protocol. These results support the use of C2f-GMKSF as a lightweight shallow-feature extractor within the complete LGD-YOLO structure, rather than as a standalone accuracy source.

2.2.2. Dy-HSFPN: Dynamic High-Level Screening Feature Pyramid Network

In dense tomato canopies, fixed upsampling and direct feature concatenation in the original YOLOv10n neck can misalign deep semantic features with shallow fruit boundaries

and may pass foliage-like texture into the detection head. Dy-HSFPN therefore replaces the original feature-fusion path of YOLOv10n. Compared with the baseline neck, it combines DySample-based dynamic upsampling with a high-level semantic screening branch so that feature alignment and background filtering are performed before the fused features are sent to the detection head [35,36].

As shown in Figure 6, this design is not intended as a new general-purpose pyramid network; rather, it is used here to make neck-level fusion more compatible with overlapping tomato fruits, where small spatial misalignments can lead to missed detections or background confusion.

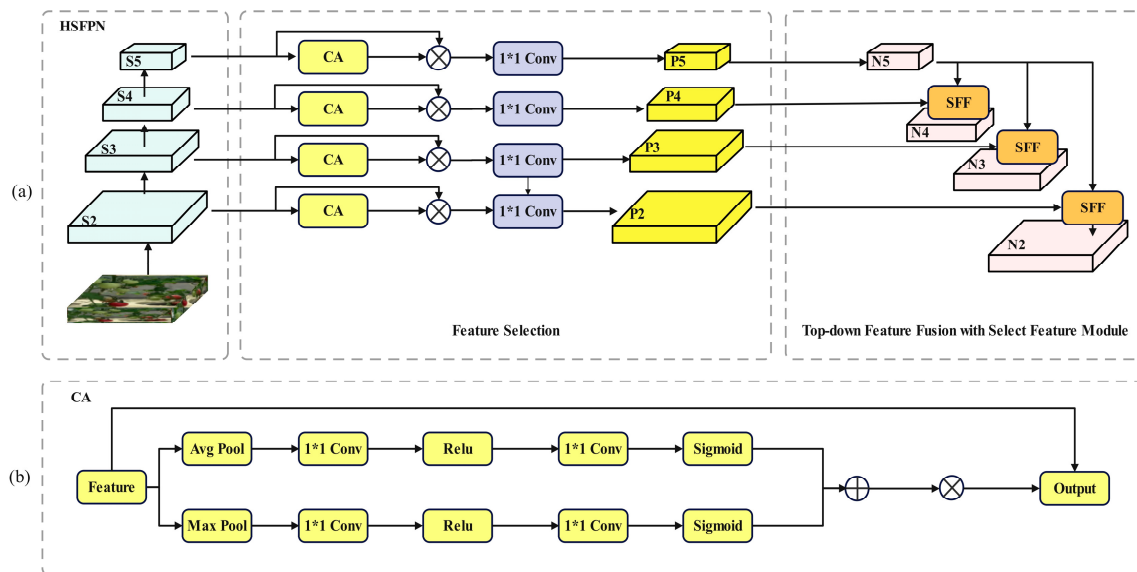


Figure 6. High-Level Screening Feature Pyramid Network. (a) Overall structure of HSFPN; (b) Structure of the Channel Attention (CA) module.

In the implemented neck, DySample replaces static interpolation by predicting sampling offsets and then resampling the feature map with bilinear interpolation, as shown in Figure 7a [37,38]:

$$X_{up} = \text{GridSample}(X, G + \text{Linear}(X)), \quad (4)$$

In Equation (4), X denotes the input feature map, G is the regular sampling grid, $\text{Linear}(\cdot)$ predicts the sampling offset field, and $\text{GridSample}(\cdot)$ performs bilinear resampling. This operation allows the upsampling grid to adapt to visible fruit structures while keeping the operator lightweight.

The semantic screening branch then uses high-level responses to weight shallow lateral features before fusion [39]. After resolution and channel alignment, global average pooling and max pooling generate a channel attention vector, and the screened feature is fused as follows:

$$f_{out} = f_{high} + f_{low} \otimes \sigma(\text{MLP}(\text{Avg}(f_{high}) + \text{Max}(f_{high}))), \quad (5)$$

In Equation (5), f_{high} and f_{low} denote the aligned high-level and low-level features, respectively; σ is the sigmoid activation; $\text{MLP}(\cdot)$ is the channel projection; $\text{Avg}(\cdot)$ and $\text{Max}(\cdot)$ denote global average and max pooling; and the product operator denotes element-wise multiplication. In the complete model, Dy-HSFPN mainly contributes occlusion-aware feature alignment and provides screened features for the following C2f-CFCGLU context module, as reflected in the ablation study.

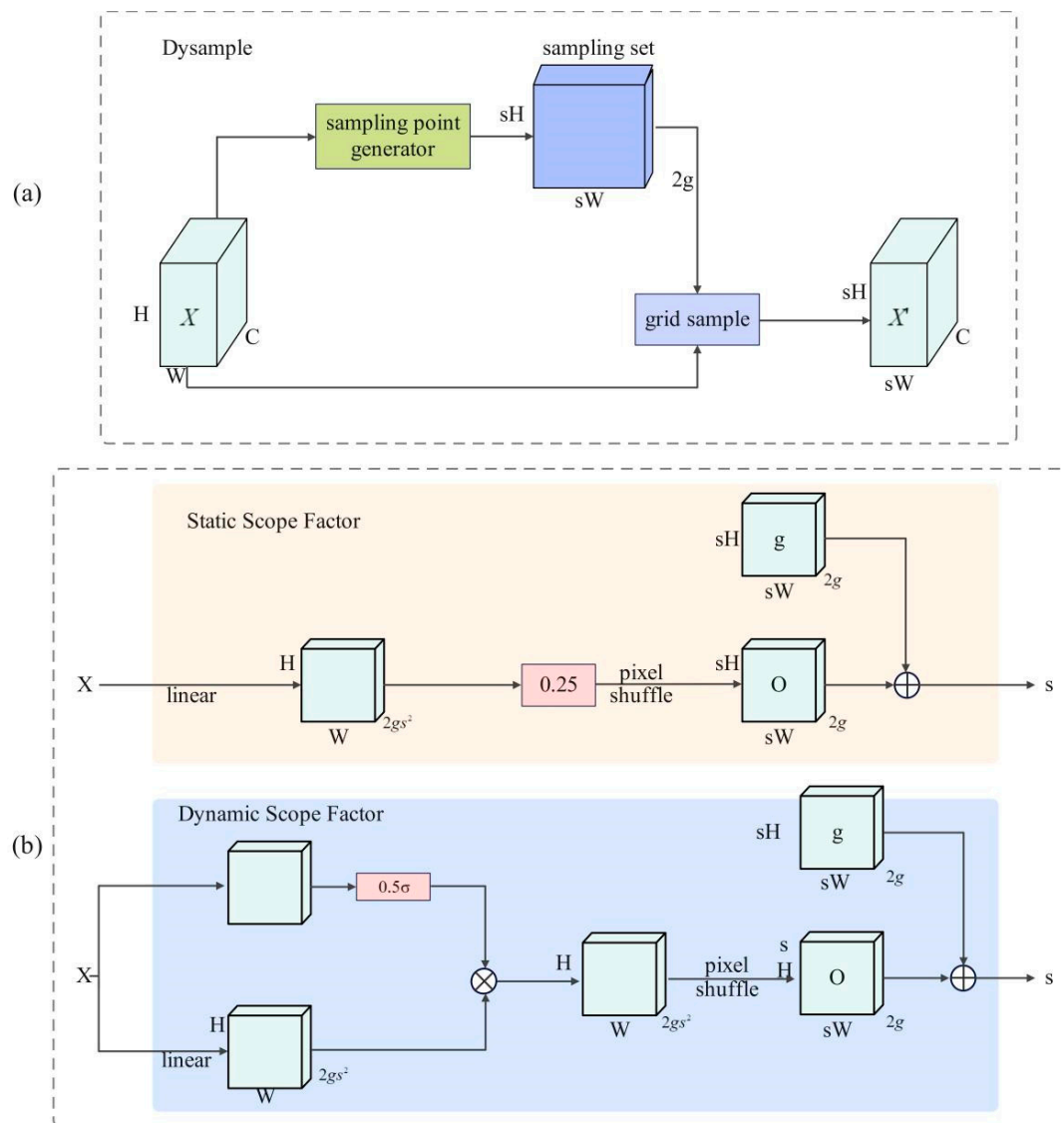


Figure 7. Dynamic Upsampling component. (a) Overall structure of DySample; (b) Details of the Sampling Point Generator within DySample.

2.2.3. C2f-CFCGLU: Mixed Context Encoding Module

After feature alignment, greenhouse tomato detection still requires contextual encoding because clustered fruits, stems, and occluding leaves often appear within the same local region. C2f-CFCGLU is deployed at selected neck fusion nodes to replace the corresponding C2f blocks in YOLOv10n. Compared with the original C2f module, it introduces a large-kernel separable spatial mixer and a convolutional gated channel mixer. The key mechanism is to combine broader fruit-cluster context with local channel modulation without using quadratic self-attention [40,41].

As shown in Figure 8, the ConvFormer branch uses large-kernel separable convolution as a lightweight spatial mixer [42]. This design broadens the receptive field and helps encode fruit-cluster morphology and nearby stem context while avoiding the computational cost of applying self-attention to high-resolution neck feature maps.

For channel mixing, CGLU is embedded after the ConvFormer branch [43], as shown in Figure 9. Unlike a standard linear feed-forward layer, the gating branch includes depthwise convolution so that channel modulation retains local spatial information:

$$\text{CGLU}(X) = (XW_1 + B_1) \odot \sigma(\text{DWConv}(XW_2 + B_2)). \quad (6)$$

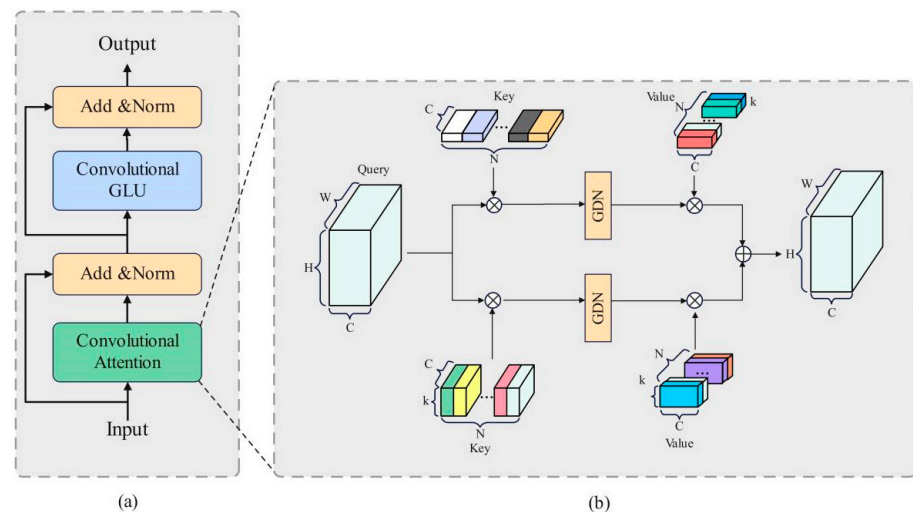


Figure 8. Spatial mixer within the mixed context encoding module. (a) Overall architecture of the ConvFormer module. (b) Details of the convolutional attention mechanism. GDN denotes Grouped Double Normalization; $\otimes$ represents the convolution operation; $\oplus$ indicates element-wise addition.

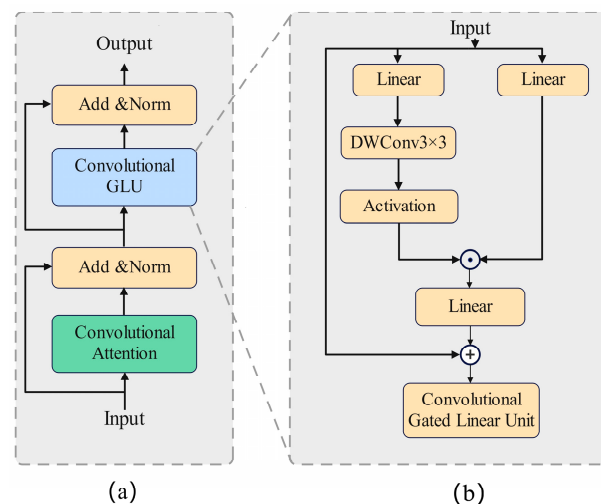


Figure 9. Channel mixer within the mixed context encoding module. (a) Structural topology of the CFormerCGLU unit integrating CGLU. (b) Internal details of the Convolutional Gated Linear Unit (CGLU). The gating branch explicitly incorporates depthwise convolution to enhance local perception.

In Equation (6), X denotes the input feature tensor; W_1 and W_2 are projection weight matrices; B_1 and B_2 are bias vectors; $DWConv(\cdot)$ denotes the depthwise convolution in the gating branch; the multiplication operator denotes element-wise multiplication; and sigma is the activation function. This operation introduces a local context prior into channel modulation and is intended to reduce responses from texture-similar background regions.

Within the complete LGD-YOLO structure, C2f-CFCGLU is therefore interpreted as a complementary contextual integration module. The ablation results indicate that its value is expressed mainly when combined with C2f-GMKSF and Dy-HSFPN, rather than as an isolated source of accuracy gain.

2.2.4. Lightweight Detection Head Based on PConv

At the prediction stage, the original YOLOv10n decoupled head uses repeated dense convolutions for classification and regression at multiple scales, which can increase feature-map movement and memory read/write cost on edge hardware. In LGD-YOLO, the detection head is reconstructed with PConv-based blocks at the P3, P4, and P5 detection

scales. Here, MemAC denotes the cost of memory read/write operations and feature-map movement, which is distinct from multiply–accumulate operations.

Compared with the dense convolutional branches in YOLOv10n, each PConv block applies spatial convolution only to one-quarter of the channels and keeps the remaining channels on an identity path [44]. The processed and bypassed features are then fused by 1×1 pointwise convolution, normalization, and activation. Given an input feature tensor, the output of the PConv module is expressed as:

$$Y = \text{Concat}(\text{Conv}_{3 \times 3}(X_p), X_{id}). \quad (7)$$

In Equation (7), X_p denotes the channel subset processed by the 3×3 convolution, X_{id} denotes the identity branch, and Y is the concatenated output before subsequent pointwise fusion. This mechanism reduces the number of spatially processed channels while retaining cross-channel interaction after fusion, as depicted in the PConv Head design in Figure 10b.

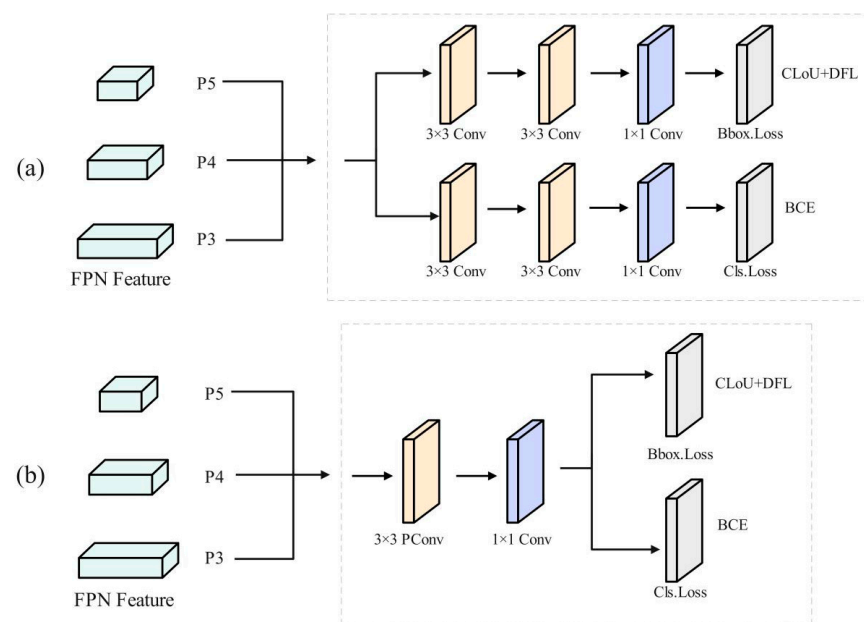


Figure 10. Structural comparison between the traditional decoupled head and the proposed PConv head. (a) Traditional decoupled head utilizing dense 3×3 convolutions with a high memory footprint. (b) PConv head applied at the P3/P4/P5 detection scales, where one-quarter of the channels are spatially processed and the remaining channels follow an identity path before pointwise fusion.

In the complete ablation sequence, PConv mainly contributes to lightweight prediction and lower computational cost. Its effect depends on the preceding backbone and neck redesign, so its contribution is interpreted as part of the coordinated lightweight architecture rather than as an independent accuracy improvement.

2.2.5. Focaler-Wise-SIoU Bounding Box Regression Loss Function

The lightweight PConv head reduces prediction-stage computation but may also weaken fine-grained spatial regression. To compensate during training, Focaler-Wise-SIoU replaces the default CIoU loss of YOLOv10n [45]. This modification is not inserted into the inference graph; it only changes the bounding-box optimization objective by combining angle-aware regression, overlap-gradient rescaling, and dynamic sample weighting.

Tomato fruits typically grow in dense clusters along the stem, where adjacent boxes may overlap and produce unstable center-distance gradients. The angle-aware term from Siou is therefore introduced to replace the center-distance term of CIoU [46]. Denoting

$b = (x, y)$ and $b^{gt} = (x^{gt}, y^{gt})$ as the center coordinates of the predicted box and the ground-truth box, respectively, and σ as the Euclidean distance between them, the angle-aware cost Λ is defined as follows:

$$\Lambda = \sin\left(2\arcsin\left(\frac{\min(|x - x^{gt}|, |y - y^{gt}|)}{\sigma}\right)\right). \quad (8)$$

This angle-aware term constrains the regression direction and helps reduce interference from adjacent fruits during training.

The Focaler-IoU term is used to rescale the overlap gradient [47]. The lower and upper focusing bounds were set to $d = 0.25$ and $u = 0.80$ according to the recommended empirical range, and no hyperparameter was tuned on the test set. The reconstructed overlap loss is expressed as:

$$\mathcal{L}_{IoU}^{Focaler} = 1 - \text{Clip}\left(\frac{IoU - 0.25}{0.8 - 0.25}, 0.1\right). \quad (9)$$

This setting reduces the influence of very low-overlap outliers and focuses on medium-difficulty samples. Because the final model emphasizes lightweight deployment, the trade-off between mAP@50 and mAP@50:95 is analyzed in the Results and Discussion Sections.

Wise-IoU is further incorporated to adjust the contribution of boxes with different regression qualities [48], as geometrically compared with CIoU in Figure 11. The Focaler-Wise-SIoU loss is formulated as follows:

$$\mathcal{L}_{FW-SIoU} = r \times \mathcal{L}_{SIoU}(\Lambda, \mathcal{L}_{IoU}^{Focaler}), \quad (10)$$

$$r = \frac{\beta}{\delta \alpha^{\beta - \delta}}, \quad (11)$$

$$\beta = \frac{\mathcal{L}_{IoU}^{Focaler}}{\bar{\mathcal{L}}_{IoU}}, \quad (12)$$

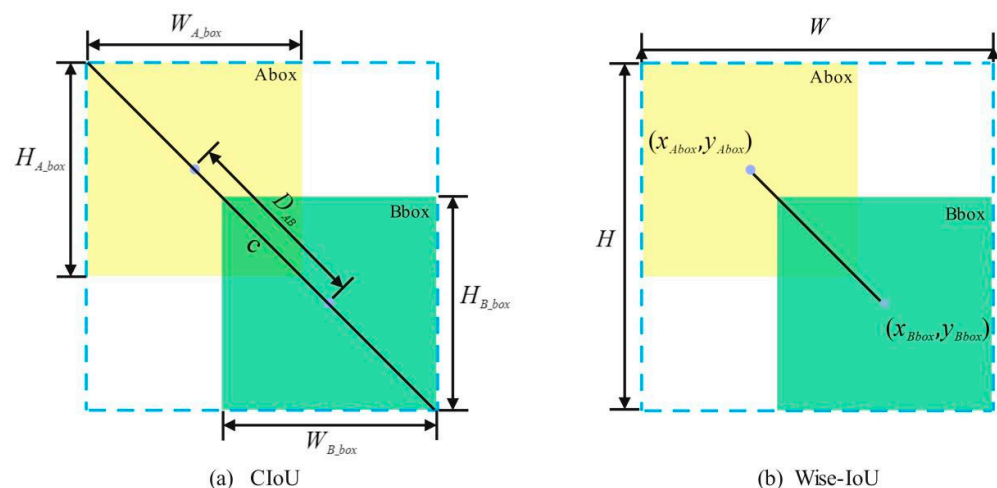


Figure 11. Geometric metric comparison of bounding box regression loss functions. (a) CIoU incorporates diagonal and center distance penalties treating samples indiscriminately, rendering it susceptible to outliers. (b) Wise-IoU leverages dynamic focusing based on anchor box quality to actively suppress interference from low-quality annotations.

In Equations (8)–(12), d and u are the lower and upper IoU focusing bounds; $\text{Clip}(\cdot)$ limits the normalized IoU term to the interval $[0, 1]$; LSIoU and LIoU denote the SIoU and IoU loss terms; r is the dynamic scaling factor; β denotes the outlier degree constructed from the ratio between the current predicted-box loss and the average loss of the current

batch; and alpha and delta are the fixed Wise-IoU hyperparameters used throughout all comparative experiments.

In the ablation study, this loss is interpreted mainly as a training-stage compensator for coarse detection and recall under the lightweight final architecture. Because the final mAP@50:95 remains lower than that of YOLOv10n, the method is not claimed to provide a general improvement in strict localization.

2.3. Experimental Environment and Evaluation Metrics

2.3.1. Experimental Environment

All training experiments in this study were executed on a unified high-performance computing platform. The hardware configuration comprised an Intel Xeon Silver 4210 processor (Intel Corporation, Santa Clara, CA, USA) coupled with an NVIDIA GeForce RTX 2080 Ti graphics processing unit (NVIDIA Corp., Santa Clara, CA, USA) utilizing 11 GB of dedicated memory. The deep learning framework was established upon PyTorch 1.10.1 using the Python 3.8 programming language. Model training was conducted under the CUDA toolkit compatible with this PyTorch release, while CUDA 12.6 corresponds to the installed driver environment used on the workstation. The model training spanned 350 epochs. The hyperparameter configuration included a batch size of 16, an initial learning rate of 0.01, a momentum factor of 0.93, and a weight decay of 0.0005. The Adam optimizer was deployed for gradient descent, and all input images were uniformly resized to a spatial resolution of 640×640 pixels. To guarantee the fairness and comparability of the experimental outcomes, identical dataset splits, input size, epoch number, and evaluation thresholds were maintained across comparative and ablation studies. The primary hyperparameter configurations are summarized in Table 2.

Table 2. Hyperparameter Configurations.

Parameter	Value	Parameter	Value
Learning rate	0.01	Weight decay	0.0005
Batch size	16	Momentum	0.93
Input image size	640×640	Total epochs	350
Number of workers	8	Optimizer	Adam

2.3.2. Evaluation Metrics

To comprehensively evaluate the perceptual performance of LGD-YOLO under complex greenhouse environments, a multi-dimensional assessment framework encompassing both detection accuracy and computational efficiency was established, predicated on the practical operational requirements of facility agricultural robots.

Regarding detection accuracy, Precision (P), Recall (R), and mean Average Precision (mAP) were adopted as the core quantitative metrics. Contextualized within agricultural detection scenarios, P measures the reliability of the model in averting false positives, such as misclassifying background illumination spots or clutter as tomatoes. Conversely, R reflects the effective detection capability of the model for occluded and small-target tomatoes. The mathematical formulations are defined as follows:

$$P = \frac{TP}{TP + FP}, \quad (13)$$

$$R = \frac{TP}{TP + FN}, \quad (14)$$

where TP, FP, and FN denote true-positive, false-positive, and false-negative samples, respectively. The F1-score is introduced as the harmonic mean of Precision and Recall, pro-

viding a combined indicator for balancing false positives and false negatives in greenhouse tomato detection:

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (15)$$

To comprehensively evaluate the model performance across all maturity categories (totaling C classes), mAP is introduced to calculate the average area under the Precision–Recall curve:

$$mAP = \frac{1}{C} \sum_{i=1}^C \int_0^1 P(R) dR. \quad (16)$$

This study uses two mAP thresholds. Specifically, mAP@50 evaluates coarse-grained target localization and maturity classification, whereas mAP@50:95 applies stricter IoU thresholds from 0.5 to 0.95 and is used to assess fine-grained boundary regression in densely overlapping scenarios.

Considering the computational and storage constraints of agricultural edge terminals, the evaluation further incorporates Parameter volume (Params), floating-point operations (GFLOPs), frames per second (FPS), and Model Size. The Model Size characterizes the storage overhead of the network and is used as one indicator of edge deployment feasibility:

$$ModelSize = \sum_i P_i \times b, \quad (17)$$

where P_i denotes the parameter count of the i -th layer, and b represents the byte size occupied by a single parameter. Concurrently, FPS is introduced to quantify the processing throughput of the model during actual execution:

$$FPS = \frac{N}{T}, \quad (18)$$

where N denotes the number of processed frames, and T denotes the total inference time. FPS is assessed under two protocols: desktop evaluation speed on the RTX 2080 Ti for fair architectural comparison, and edge inference speed on the Jetson AGX Orin (64GB Developer Kit, NVIDIA Corporation, Santa Clara, CA, USA) for deployment evaluation.

3. Results and Analysis

3.1. Ablation Study of Core Components

To quantitatively assess the individual contributions of the proposed modules to tomato detection performance and hardware efficiency, an ablation study was conducted utilizing YOLOv10n as the baseline architecture. Under identical experimental configurations, the following components were sequentially integrated: the dynamic feature alignment network Dy-HSFPN (A), the multi-scale kernel fusion module C2f-GMKSF (B), the global context gating module C2f-CFCGLU (C), the partial convolution detection head PConvHead (D), and the Focaler-Wise-SIoU loss function (E). The comprehensive ablation results are detailed in Table 3.

The baseline YOLOv10n model yielded an mAP@50 of 87.4% with a parameter volume of 2.26 M. Upon the independent incorporation of Dy-HSFPN (A), the parameter count decreased to 1.74 M while the mAP@50 improved to 87.9%, indicating that the content-aware dynamic upsampling effectively mitigates feature aliasing. The independent C2f-GMKSF configuration (B) achieved an mAP@50 of 87.6% with 2.19 M parameters and 6.4 GFLOPs. When Dy-HSFPN and C2f-GMKSF were combined (A + B), the mAP@50 increased to 88.3% and the computational load decreased to 5.3 GFLOPs, indicating that dynamic feature alignment and shallow multi-scale feature extraction are complementary. Building upon A + B, the addition of the C2f-CFCGLU module (A + B + C) resulted in

a compact parameter volume of 1.56 M and 5.0 GFLOPs while maintaining an mAP@50 of 88.1%, suggesting that the context module mainly contributes to compact semantic integration rather than an isolated large accuracy gain.

Table 3. Ablation results of the proposed components under the same tomato dataset split and desktop evaluation protocol.

Model	P (%)	R (%)	mAP@50	mAP@50:95	Params (M)	GFLOPs	FPS
YOLOv10n	83.3	78.5	87.4	52.2	2.26	6.5	38.17
YOLOv10 + A	85.4	79.3	87.9	49.5	1.74	5.4	42.98
YOLOv10 + B	82.3	80.8	87.6	50.6	2.19	6.4	46.80
YOLOv10 + C	82.9	79.4	87.4	49.8	1.89	5.4	42.92
YOLOv10 + D	80.4	78.5	84.8	47.9	3.41	7.8	30.53
YOLOv10 + A + B	83.7	80.9	88.3	50.1	1.67	5.3	46.63
YOLOv10 + A + C	83.0	78.7	87.6	49.1	1.70	5.2	32.71
YOLOv10 + A+B + C	85.8	77.2	88.1	49.4	1.56	5.0	28.76
YOLOv10 + A + B + D	81.6	78.9	86.5	48.1	1.44	4.0	37.43
YOLOv10 + A + B + C+D	85.7	77.6	87.0	48.2	1.41	3.8	19.33
YOLOv10 + A + B + C + E	85.0	78.9	88.3	50.2	1.64	5.1	29.88
LGD-YOLO	84.4	79.9	88.0	48.3	1.41	3.8	34.09

It is important to note that certain intermediate configurations achieved slightly higher accuracy metrics than the final model. For example, A + B + C + E yielded an mAP@50 of 88.3% and an mAP@50:95 of 50.2%. However, the primary design criterion for this agricultural application is edge deployment efficiency rather than absolute accuracy maximization. The isolated insertion of PConvHead (D) into the original YOLOv10n structure increased parameters to 3.41 M and GFLOPs to 7.8, showing that PConv is not beneficial when inserted without coordinated backbone and neck redesign. In contrast, when PConvHead was integrated after A + B + C, the A + B + C + D configuration compressed the model to 1.41 M parameters and 3.8 GFLOPs, although mAP@50 decreased to 87.0% and mAP@50:95 dropped to 48.2% due to the loss of spatial channel resolution. The subsequent use of Focaler-Wise-SIoU restored mAP@50 to 88.0% and provided a marginal recovery in mAP@50:95 to 48.3%, without changing the inference graph. While it does not fully surpass the baseline's strict localization performance of 52.2%, the loss function acts as a necessary compensator to mitigate the severe localization degradation caused by the lightweight head. The FPS difference between A + B + C + D and LGD-YOLO is therefore not attributed to the loss function itself, but to measurement variability and implementation-level inference conditions; accordingly, the loss-function comparison is discussed mainly in terms of accuracy and recall.

The final LGD-YOLO configuration was selected as a compromise between detection accuracy and edge deployment cost. Compared with YOLOv10n, LGD-YOLO achieved a 0.6 percentage-point increase in mAP@50, while reducing parameter volume by 37.6% and computational load by 41.5%. At the same time, the stricter mAP@50:95 decreased from 52.2% to 48.3%, indicating that the lightweight compression improves coarse detection and maturity recognition but sacrifices part of the fine-grained localization quality. This trade-off is explicitly considered in the following class-wise and deployment analyses, where the emphasis is placed on real-time perception for greenhouse harvesting rather than high-precision boundary regression alone.

3.2. Comprehensive Performance Comparison with State-of-the-Art Algorithms

All models compared in this section were retrained and evaluated on the same tomato dataset split with the same input resolution. Table 4 summarizes the overall compari-

son, including detection accuracy, parameter volume, computation cost, model size, and desktop evaluation speed. Empirical evaluations indicate that LGD-YOLO provides a favorable trade-off between lightweight architecture and detection accuracy. Compared to the YOLOv10n baseline, LGD-YOLO reduces the parameter volume by 37.6% to 1.41 M, decreases the computational load to 3.8 GFLOPs, and compresses the model size to 2.99 MB. Although YOLOv12n and YOLOv13n attain desktop FPS values close to 40 under the present evaluation protocol, their mAP@50 scores are only 74.5% and 74.7%, respectively, indicating insufficient feature discrimination for small and visually similar tomato targets in this dataset. In contrast, LGD-YOLO sustains 34.09 FPS on the desktop evaluation protocol while achieving an mAP@50 of 88.0%.

Table 4. Performance comparison with state-of-the-art detectors retrained on the same tomato dataset split.

Model	P (%)	R (%)	mAP@50	Params (M)	GFLOPs	Size (MB)	FPS
YOLOv5s	83.5	78.6	86.5	9.11	23.8	17.7	43.17
YOLOv5m	84.2	77.3	85.7	21.10	64.0	48.1	17.92
YOLOv7x	88.0	77.6	84.0	37.21	105.18	142	9.8
YOLOv8n	83.6	78.8	85.4	6.40	8.1	5.96	41.5
YOLOv8s	81.4	80.5	86.4	11.12	28.4	21.4	27.38
YOLOv8m	86.4	77.7	87.3	25.84	78.7	49.6	18.18
YOLOv9t	84.0	71.6	79.5	1.97	7.6	4.44	45.2
YOLOv9s	83.8	75.0	80.1	7.16	26.7	14.5	31.6
YOLOv10n	83.3	78.5	87.4	2.26	6.5	5.5	38.17
YOLOv10s	78.4	85.2	88.1	7.21	21.4	40.5	27.16
YOLOv10m	82.6	81.4	88.5	15.31	58.9	72.1	23.03
YOLOv11n	83.4	78.8	87.1	2.58	6.3	5.23	40.8
YOLOv11s	83.7	79.1	87.4	9.41	21.3	19.6	30.4
YOLOv12n	79.3	69.4	74.5	2.55	6.3	5.28	39.6
YOLOv12s	83.9	75.6	80.7	9.23	21.2	18.0	28.9
YOLOv13n	79.3	71.0	74.7	2.44	6.2	5.17	41.2
YOLOv13s	86.2	79.3	85.2	9.00	20.7	35.4	29.7
GoldYOLO	84.8	78.5	85.8	5.97	10.2	13.8	34.6
Ta-YOLO	87.3	76.9	84.3	10.58	14.5	24.6	31.1
LGD-YOLO *	83.7	78.5	87.0	1.48	3.9	3.1	33.2
LGD-YOLO	84.4	79.9	88.0	1.41	3.8	2.99	34.09

* 1 LGD-YOLO* represents the architectural variant migrated into the YOLOv8n baseline.

The agriculture-specific model Ta-YOLO and the general-purpose advanced model GoldYOLO were also retrained on the present dataset for a fair comparison. Ta-YOLO contains 10.58 million parameters, which is about 7.5 times that of LGD-YOLO, while its mAP@50 is 84.3% under the same dataset split. These results indicate that directly increasing model capacity does not necessarily improve detection performance for this greenhouse maturity dataset. By adopting an asymmetric computational paradigm, LGD-YOLO integrates context modeling only at selected stages and avoids a network-wide increase in parameters, thereby improving deployment feasibility on edge devices.

To evaluate the architecture-agnostic and plug-and-play characteristics of the proposed core modules, C2f-GMKSF, C2f-CFCGLU, and Dy-HSFPN were migrated into the YOLOv8n framework to construct the variant LGD-YOLO*. The variant was retrained on the same dataset split and evaluated under the same protocol. As detailed in Table 4, the parameter volume of LGD-YOLO* is reduced from 6.40 M (YOLOv8n) to 1.48 M, and the computational load drops from 8.1 GFLOPs to 3.9 GFLOPs, while the mAP@50 improves from 85.4% to 87.0%. These findings suggest that reducing channel redundancy and improving semantic alignment can improve parameter efficiency across different YOLO backbones.

Figure 12 illustrates the Pareto distribution of efficiency and accuracy for various models on the tomato dataset, indicating that LGD-YOLO is located in the high-accuracy and low-computation region.

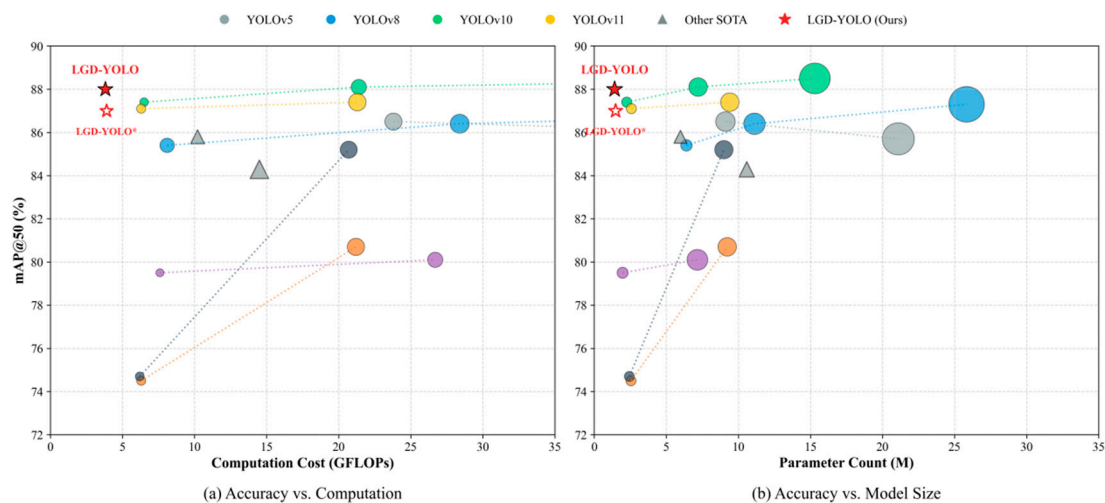


Figure 12. Trade-off between efficiency and accuracy on the tomato dataset. (a) Accuracy vs. Computation Cost (GFLOPs). (b) Accuracy vs. Parameter Count (M).

3.3. Class-Specific Detection Performance and Confusion Analysis

To comprehensively evaluate the discriminative capability of the proposed model under extreme class imbalance, particularly for visually ambiguous transitional stages, a fine-grained assessment across maturity categories was conducted. Table 5 details the precision, recall, F1-score, mAP@50, and mAP@50:95 metrics for each class compared to the YOLOv10n baseline.

Table 5. Detection performance comparison across maturity stages.

Model	Class	P%	R%	mAP@50	mAP@50:95	F1-Score
YOLOv10n	all	83.3	78.5	87.4	52.2	80.8
	Unripe	85.5	78.7	87.6	49.4	82.0
	Ripe	79.9	78.9	87.0	56.8	79.4
	Semi-ripe	84.1	78.0	87.6	50.4	80.9
LGD-YOLO	all	84.4	79.9	88.0	48.3	82.1
	Unripe	86.1	77.2	86.7	45.7	81.4
	Ripe	81.2	80.4	87.2	51.2	80.8
	Semi-ripe	86.0	82.2	90.2	47.5	84.1

As presented in Table 5, the most notable gain of LGD-YOLO is observed on the semi-ripe class. However, the stricter mAP@50:95 metric decreased from 52.2% in YOLOv10n to 48.3% in LGD-YOLO, indicating a trade-off between coarse detection robustness and fine-grained localization quality under lightweight compression. To further unpack the diagnostic performance, the class-wise Precision–Recall curves and the normalized confusion matrix are illustrated in Figures 13 and 14, respectively.

As shown in Figure 14, LGD-YOLO correctly predicts 84%, 83%, and 83% of true unripe, ripe, and semi-ripe tomatoes, respectively. The primary error source for the challenging semi-ripe category is inter-class confusion rather than background misidentification, with 4% misclassified as unripe and 7% as ripe. From an agricultural perspective, this distribution is interpretable: the semi-ripe stage is a transient physiological period characterized by uneven patchy pigmentation, which naturally overlaps visually with adjacent stages depending on fruit orientation and viewing angle. Despite this inherent ambiguity

and severe class imbalance, an 83% true-positive rate and 90.2% mAP@50 were observed for the semi-ripe class. Furthermore, 16% of unripe and 11% of ripe tomatoes were predicted as background, predominantly attributable to severe structural occlusions caused by overlapping foliage and dense clustering in unstructured greenhouse environments.

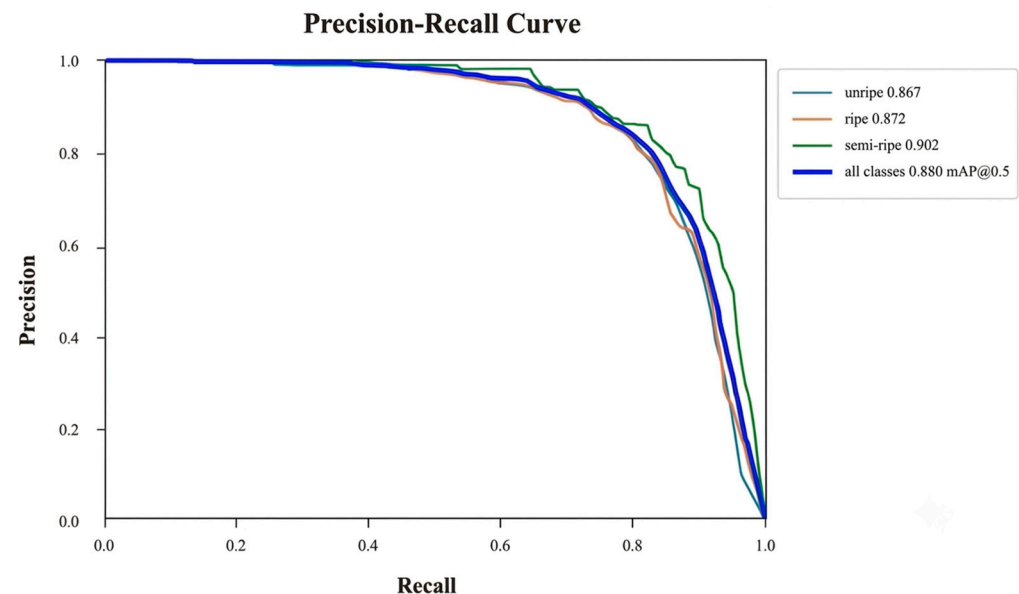


Figure 13. Class-wise Precision–Recall curves of LGD-YOLO on the tomato maturity dataset.

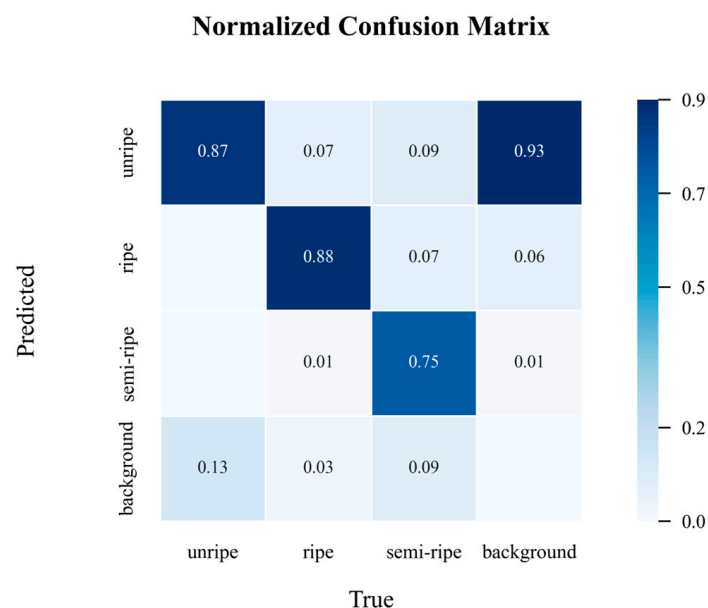


Figure 14. Normalized confusion matrix of LGD-YOLO across maturity stages.

3.4. Comparative Evaluation of Bounding Box Regression Loss Functions

To assess the effect of the proposed Focaler-Wise-SIoU loss function for greenhouse tomato detection, a comparative analysis was conducted against representative bounding box regression loss functions. Because loss functions affect only the training objective and do not change the inference graph, Table 6 is used primarily to compare accuracy-related metrics. The FPS column is retained only as a record of the measured evaluation runs and is not used to infer that a loss function directly accelerates inference.

Table 6. Comparison of bounding box regression losses under the same LGD-YOLO architecture; FPS is reported only as measured evaluation throughput.

Loss	P (%)	R (%)	mAP@50	mAP@50:95	FPS
GIoU	82.0	79.1	86.6	48.5	28.29
EIoU	82.8	78.6	86.6	48.2	26.92
DIoU	85.6	76.1	86.6	47.7	25.19
PIoU	82.8	80.3	86.4	47.9	26.12
ShapeIoU	82.0	78.7	86.4	47.9	25.89
MPDIoU	87.8	75.5	86.8	48.2	24.99
CIoU	85.7	77.6	87.0	48.2	19.33
SIoU	86.1	75.3	86.7	47.8	24.49
Wise-SIoU	86.6	76.1	87.7	49.1	26.79
Focaler-Wise-SIoU(Ours)	84.4	79.9	88.0	48.3	34.09

Both GIoU and DIoU yield an identical mAP@50 of 86.6%. While CIoU marginally elevates the mAP@50 to 87.0%, the proposed Focaler-Wise-SIoU achieves the highest mAP@50 of 88.0% by balancing the gradient contributions of hard and easy samples through a dynamic focusing mechanism. Since the loss function is not executed during inference, the analysis focuses on convergence behavior and detection metrics rather than attributing runtime differences to the loss formulation.

Regarding the Recall metric, MPDIoU and SIoU register at 75.5% and 75.3%, respectively. The proposed Focaler-Wise-SIoU attains a Recall of 79.9%, which is marginally surpassed only by PIoU (80.3%). Despite this slight recall deficit, Focaler-Wise-SIoU outperforms PIoU by 1.6 percentage points in mAP@50. The mAP@50:95 value of Focaler-Wise-SIoU is 48.3%, which is lower than Wise-SIoU (49.1%) but comparable to several alternative losses. Therefore, the proposed loss should be interpreted as improving coarse detection and recall under the lightweight final architecture, while fine-grained localization remains a limitation that requires further optimization.

3.5. Comparative Evaluation of Lightweight Detection Head Architectures

To compare detection head architectures for edge computing scenarios, five representative lightweight head schemes were reconstructed upon the LGD-YOLO baseline. These configurations include the depthwise separable convolution-based DConv + Conv, the multi-branch re-parameterization module Diverse Branch Block, the spatial and channel reconstruction convolution DScConv, and the lightweight shared detail enhancement convolution LSDECD. These variants were systematically compared against the proposed PConv head, with the quantitative results detailed in Table 7.

The traditional Conv + Conv structure achieves an mAP@50 of 86.3%; however, it incurs a computational load of 5.0 GFLOPs and a restricted inference speed of 26.73 FPS. The integration of depthwise separable convolutions (DConv + Conv) reduces the GFLOPs to 4.4, yet the FPS exhibits no proportional improvement. This result suggests that reduced FLOPs alone may not guarantee acceleration, possibly because fragmented memory access and operator implementation also affect runtime. Although the Diverse Branch Block employs multi-branch training and structural re-parameterization, its FPS declines to 24.67 and the model size increases to 4.21 MB, indicating that overly wide network topologies may encounter bandwidth bottlenecks on edge-oriented inference. Additionally, while DScConv compresses redundancy through spatial and channel reconstruction and reduces the computational cost to 3.9 GFLOPs, its mAP@50 drops to 85.4%. This decline suggests that excessive structural compression may weaken fine-grained texture representation for small tomato targets. The LSDECD variant exhibits moderate performance in both accuracy and speed metrics.

Table 7. Comparison of detection head designs reconstructed on the same LGD-YOLO backbone and neck.

Head	P (%)	R (%)	mAP@50	Params (M)	GFLOPs	Size (MB)	FPS
Conv + Conv	85.8	75.8	86.3	1.61	5.0	3.38	26.73
DConv + Conv	85.3	76.7	86.9	1.51	4.4	3.19	26.44
DiverseBranchBlock + DiverseBranchBlock	84.3	77.6	86.3	1.61	5.0	4.21	24.67
DScConv + DConv	80.2	78.3	85.4	1.42	3.9	3.03	26.33
LSDECD	80.6	77.9	85.9	1.46	5.0	3.49	26.50
PConv + Conv(Ours)	84.4	79.9	88.0	1.41	3.8	2.99	34.09

In contrast, the proposed PConv head applies spatial convolutions exclusively to a partial subset of channels, capitalizing on the high inter-channel similarity of feature maps to lower the frequency of memory read–write operations. As demonstrated in Table 7, LGD-YOLO achieves the highest mAP@50 of 88.0% and an inference speed of 34.09 FPS under the desktop evaluation protocol. This represents a 27.5% improvement in speed over the conventional Conv + Conv scheme. However, Table 7 does not yet include mAP@50:95 for each head variant; therefore, the present conclusion is limited to mAP@50, parameter/computation cost, model size, and FPS, and the effect on strict localization will be further evaluated in future work.

3.6. HiResCAM-Based Heatmap Analysis

To intuitively elucidate the feature extraction behavior of the models in unstructured greenhouse scenarios, High-Resolution Class Activation Mapping (HiResCAM) is employed to conduct a qualitative visual analysis across various models, as illustrated in Figure 15.

In green-on-green interference scenarios where immature tomato fruits share high visual similarity with the surrounding foliage, the attention distributions of YOLOv5 and YOLOv8 exhibit pronounced dispersion. Their high-response regions overlap with background leaves and supporting stems, which is consistent with their higher tendency to produce false positives under complex backgrounds. Conversely, supported by large-kernel convolutions and gating units within the C2f-CFGLU module, LGD-YOLO concentrates more of its high-activation regions within visible fruit contours and suppresses part of the response triggered by leaf textures.

Regarding dense clustering and mutual occlusion, although YOLOv9 and YOLOv10 localize salient targets, their feature activation intensities attenuate for fruits with blurred edges or severe foliage occlusion. LGD-YOLO shows more continuous activation within several occluded fruit regions. This phenomenon is consistent with the adaptive feature-alignment role of DySample in Dy-HSFPN, although the heatmap evidence remains qualitative. The heatmap results therefore provide visual support for the quantitative results, while more rigorous subset-level evaluation by occlusion degree and object size will be considered in future work.

3.7. Quantitative Error and Qualitative Failure Case Analysis

To explicitly address the detection challenges under unstructured greenhouse conditions, a quantitative error analysis was conducted utilizing the test set confusion matrices. Statistical evaluation of the dataset's spatial distribution indicates that the vast majority of target bounding boxes occupy approximately 2% to 10% of the image dimensions, corroborating the inherent challenge of small-object detection.

A key evaluation of the proposed architecture concerns its behavior under green-on-green background interference. As illustrated in Figure 16, the confusion matrix of the baseline YOLOv10n reveals susceptibility to visual homogenization, erroneously classifying 4378 background instances, such as visually similar green foliage and supporting stems, as unripe tomatoes. By integrating large-kernel spatial context modeling and dynamic feature screening, the proposed LGD-YOLO reduced these false positives to 938, as shown in Figure 17, achieving a 78.6% quantitative reduction in background misclassifications.

Regarding missed detections under dense foliage occlusion, LGD-YOLO still failed to detect 661 severely occluded tomato instances, corresponding to a false-negative proportion of approximately 16% for unripe tomatoes in the confusion matrix. This result indicates that although the model reduces background false positives, recall under extreme occlusion remains a limitation. Inter-class confusion between the visually transitional semi-ripe and ripe stages is constrained below 7% and 3%, respectively, suggesting that the three maturity classes remain reasonably separable under the proposed feature representation.

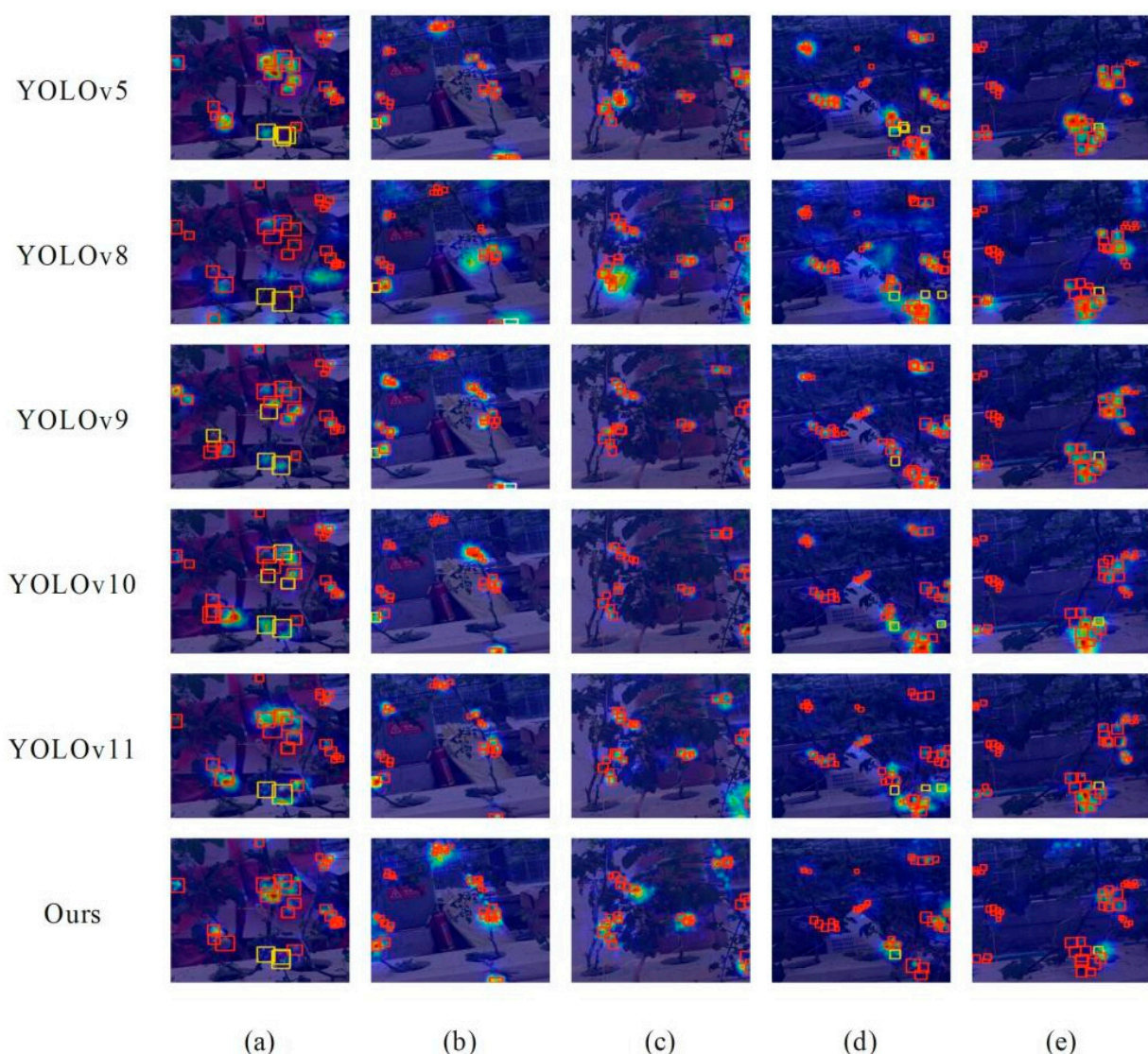


Figure 15. Visual comparison of feature extraction behaviors across different detection models using HiResCAM heatmaps. Panels (a–e) depict various greenhouse scenarios ranging from simple backgrounds to dense occlusions and clusters of small objects. Red regions indicate areas where the model focuses highly, while blue regions represent suppressed background features.

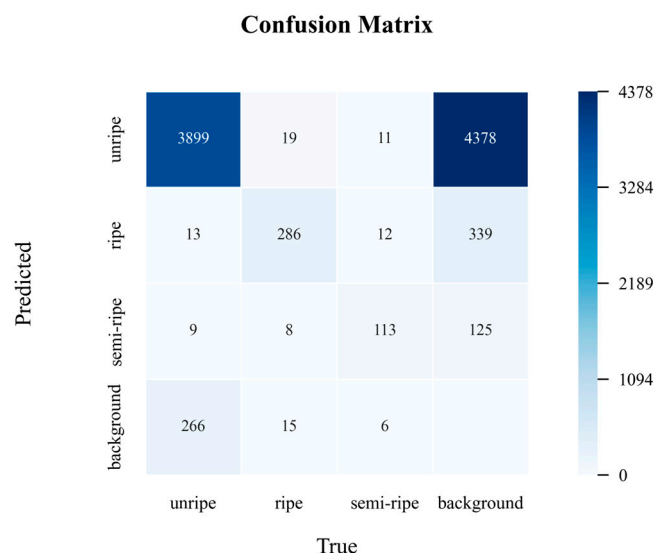


Figure 16. Confusion matrix of the baseline YOLOv10n on the test set, illustrating severe green-on-green background interference with 4378 false positives.

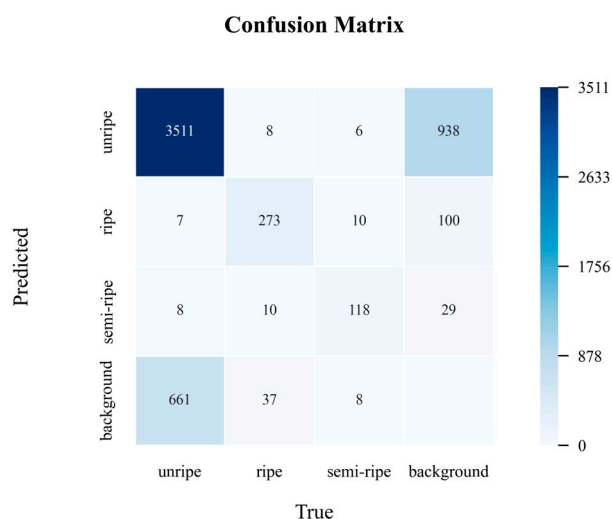


Figure 17. Confusion matrix of the proposed LGD-YOLO on the test set, demonstrating suppression of background false positives to 938 and inter-class maturity discrimination.

Visual comparisons in the detection results are consistent with these quantitative findings. As illustrated in Figure 18, the baseline model tends to miss small targets and deeply occluded instances, as highlighted by orange circles, under challenging illumination and dense foliage. It also generates some false-positive errors on background structures, as indicated by red circles. In contrast, LGD-YOLO shows improved detection of several obscure targets and fewer classification errors in these examples.

3.8. Edge Device Deployment and Evaluation

To evaluate the deployment performance of LGD-YOLO on physical hardware, inference tests were conducted on both a desktop computing platform and an embedded edge platform. The comparative results are summarized in Table 8. Under FP32 precision on the desktop platform, LGD-YOLO achieves a single-frame inference latency of 3.8 ms. Compared to the 4.6 ms latency of the YOLOv10n baseline, this constitutes a 17.4% reduction, correspondingly boosting the frame rate to 263 FPS. On the Jetson AGX Orin edge platform, with TensorRT acceleration and FP16 half-precision inference, the latency of LGD-YOLO decreases to 24.6 ms, compared to 31.4 ms for YOLOv10n, and the operating speed reaches

40.6 FPS. Latency measurements were conducted using a batch size of 1 over 1000 continuous frames following 50 warm-up iterations. Based on three independent repetitions, the standard deviation for latency remained below 0.5 ms across all tested models on both platforms. These measurements represent pure inference latency under the deployment protocol, whereas the 34.09 FPS reported in Tables 3–7 corresponds to the desktop evaluation protocol that includes dataset-level evaluation overhead. The two FPS values therefore describe different measurement settings and should not be directly compared.

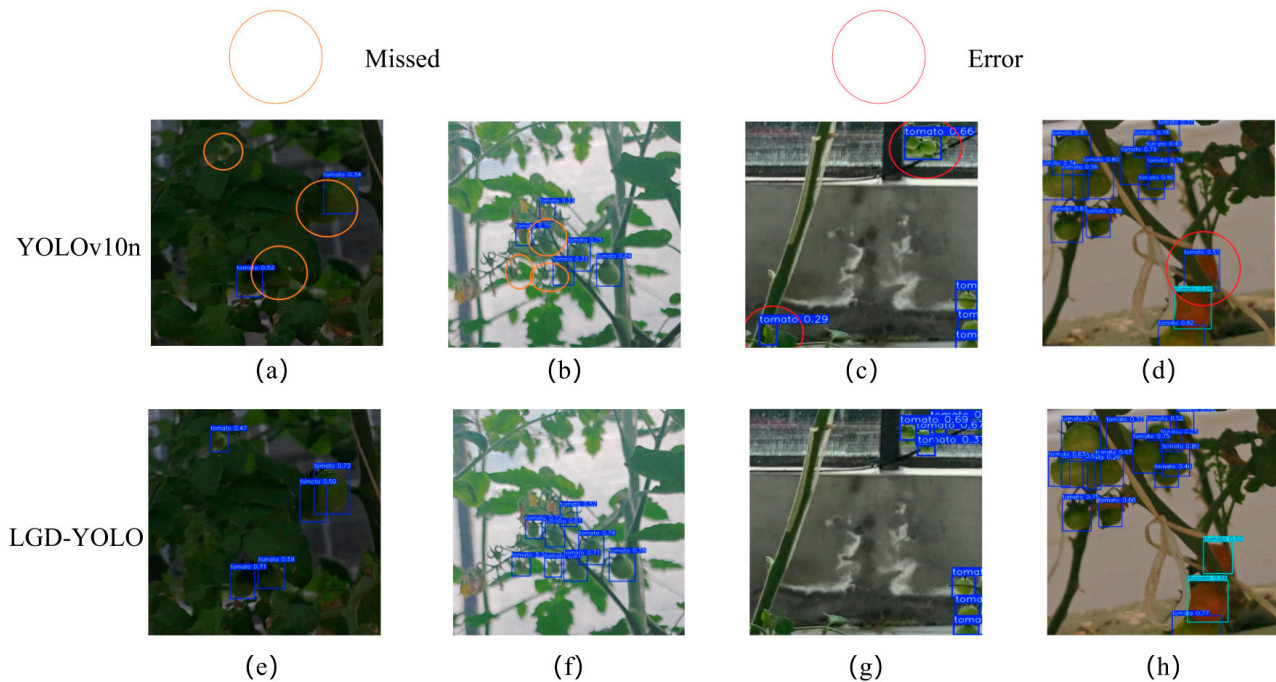


Figure 18. Qualitative comparison of tomato detection results under diverse challenging greenhouse scenarios. Detection results of representative baseline methods (top row) and LGD-YOLO (bottom row) are shown for different conditions: (a,e) backlit and low-illumination scenes, (b,f) small-object scenarios, (c,g) densely distributed fruits, and (d,h) overlapping fruits with severe occlusion. Yellow circles indicate missed detections, and red circles denote false positives.

Table 8. Inference latency and frame-rate comparison under desktop FP32 and Jetson AGX Orin TensorRT-FP16 deployment protocols.

Model	Platform	Precision	Latency (ms)	FPS	Speedup vs. Baseline
YOLOv8n	Desktop(2080Ti)	FP32	5.2	192	-
YOLOv10n	Desktop(2080Ti)	FP32	4.6	217	Reference
LGD-YOLO	Desktop(2080Ti)	FP32	3.8	263	0.174
YOLOv8n	Edge(Jetson AGX Orin)	TensorRT-FP16	34.5	29	-
YOLOv10n	Edge(Jetson AGX Orin)	TensorRT-FP16	31.4	31.8	Reference
LGD-YOLO	Edge(Jetson AGX Orin)	TensorRT-FP16	24.6	40.6	0.216

To evaluate practical applicability, a physical testing system based on the Jetson AGX Orin was assembled. High-resolution video streams collected from real greenhouse scenarios were used together with a lightweight monitoring interface developed using PyQt5. As illustrated in Figure 19, when processing video frames characterized by densely overlapping tomato foliage, the system maintains an average processing latency between 24 and 26 ms, sustaining a real-time frame rate of approximately 40 FPS. This result suggests that LGD-YOLO can satisfy the real-time perception requirement of continuous automated harvesting.



Figure 19. Real-time execution and monitoring interface of LGD-YOLO on the edge computing device.

4. Discussion

LGD-YOLO provides a practical trade-off between detection accuracy and computational efficiency in greenhouse tomato detection tasks. This performance enhancement stems primarily from three architectural innovations.

The asymmetric computational paradigm is based on the heterogeneous allocation of resources according to the frequency-domain characteristics of feature maps. In the shallow layers, the C2f-GMKSF module employs grouped multi-kernel convolutions to extract fruit textures and edges with limited computational overhead, helping to mitigate false positives against green-on-green backgrounds. Conversely, the deep-layer C2f-CFCGLU module uses large-kernel separable convolutions and gated linear units to expand the receptive field, enabling the model to capture fruit-cluster morphology across localized occlusions. Ablation studies show that the independent integration of C2f-GMKSF achieves an mAP@50 of 87.6%, whereas the A + B+C configuration sustains an 88.1% mAP@50 with a parameter volume of 1.56 M. These results support the design philosophy that prioritizes fine-grained details in shallow layers and robust semantics in deep layers within lightweight architectures.

Dynamic feature alignment is associated with improved recall under occluded conditions. Traditional FPNs often rely on nearest-neighbor interpolation, which performs simple spatial magnification and may not reconstruct geometric information lost during downsampling. The DySample operator integrated into Dy-HSFPN facilitates content-based dynamic point sampling, allowing the model to adaptively adjust the sampling grid according to the localized visible morphology of the fruits. Heatmap analysis indicates that LGD-YOLO produces stronger activation within some occluded fruit regions than YOLOv9 and YOLOv10. This architectural design improves the recall rate from 78.5% to 79.3%, which further reaches 80.9% upon the integration of C2f-GMKSF.

Hardware-aware lightweight strategies are intended to reduce memory bandwidth pressure on edge devices. Existing methodologies frequently overemphasize FLOPs reduction while neglecting MemAC. The PConv detection head applies spatial convolutions exclusively to a subset of channels, reducing the frequency of memory read–write operations. Deployed on the Jetson AGX Orin platform, LGD-YOLO reduces inference latency by

21.6% compared to YOLOv10n, achieving a frame rate of 40.6 FPS. The accuracy decrease associated with lightweight compression is partly offset by the Focaler-Wise-SIoU loss function, which improves the mAP@50 to 88.0% while sustaining a recall rate of 79.9%. Notably, in comparative experiments, although the DConv + Conv architecture reduces FLOPs, it yields no improvement in FPS. This finding suggests that fragmented memory access in depthwise convolutions may constrain empirical inference speeds. Furthermore, theoretical GFLOPs reduction does not always guarantee linear acceleration in unfused environments. Despite having fewer FLOPs, LGD-YOLO exhibits a slightly lower desktop PyTorch FPS than YOLOv10n. This suggests that custom operators, such as dynamic sampling, introduce fragmented memory access and higher kernel launch overheads compared to standard cuDNN-optimized convolutions. However, TensorRT deployment on the Jetson AGX Orin can fuse parts of these computational graphs, which helps reduce runtime overhead and supports real-time performance.

Nevertheless, this study presents several limitations regarding the dataset and evaluation scale. First, the current dataset was constructed based on a single tomato cultivar, ‘Jiahong No. 1’, cultivated in a single greenhouse facility. This limits the evaluation of the model’s cross-varietal and cross-domain generalization capabilities under varying facility structures. Second, a severe class imbalance inherently exists within the dataset due to the natural growth cycle of greenhouse tomatoes, where unripe instances overwhelmingly dominate. This imbalance still constrains comprehensive representation of transitional maturity stages. Finally, with only 180 images, the test set is relatively limited and may not encompass all extreme edge cases encountered in unstructured agricultural environments. Future work will focus on constructing a large-scale multi-cultivar dataset across diverse lighting conditions and greenhouse layouts, addressing the category distribution imbalance, and establishing a more comprehensive evaluation benchmark to further enhance model robustness in diverse agricultural scenarios.

5. Conclusions

To address the coupled challenges of green-on-green background interference, dense occlusion, and constrained edge computing resources in greenhouse tomato maturity detection, this study proposes LGD-YOLO, a lightweight object detection network. Built upon the YOLOv10n framework, the proposed model preserves the low-latency NMS-free characteristics while introducing architectural improvements in feature extraction, feature fusion, and memory-efficient prediction.

Specifically, within the backbone network, the C2f-GMKSF module is employed in shallow layers to extract texture edges, whereas the C2f-CFCGLU module is embedded in deep layers to expand the receptive field. The neck incorporates the Dy-HSFPN dynamic feature pyramid, utilizing the DySample operator to mitigate feature aliasing. The detection head leverages PConv to reduce memory access costs, and the Focaler-Wise-SIoU loss function acts as a compensator to stabilize the training objective for bounding-box regression without changing the inference graph. Experiments show that LGD-YOLO achieves an mAP@50 of 88.0% with 1.41 M parameters and 3.8 GFLOPs. Compared with YOLOv10n, it reduces parameter volume by 37.6% and computational load by 41.5%, while improving mAP@50 by 0.6 percentage points. However, mAP@50:95 decreases from 52.2% to 48.3%, indicating that the lightweight final model improves coarse detection and maturity recognition while sacrificing part of the strict localization performance. Furthermore, the validation of the model’s stability and repeatability across multiple independent training runs is detailed in Supplementary Note S1. On the NVIDIA Jetson AGX Orin platform, LGD-YOLO reaches 40.6 FPS under TensorRT-FP16 pure inference, whereas the 34.09 FPS reported in the comparative tables corresponds to the desktop evaluation protocol. The

trained model weight file is 2.99 MB. The real-time detection performance on this platform is demonstrated in Video S1. Furthermore, the plug-and-play capability of the core modules was evaluated through migration to the YOLOv8n architecture. Future work will focus on improving strict localization, expanding multi-cultivar and multi-greenhouse validation, and incorporating more detailed deployment profiling.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/agriculture16141517/s1>: Video S1: Real-time tomato maturity detection using LGD-YOLO on the Jetson AGX Orin platform; Supplementary Note S1: Validation of Model Stability and Repeatability.

Author Contributions: X.X.: Conceptualization, Data curation, Investigation, Writing—review and editing. A.W.: Methodology, Visualization, Writing—original draft. Y.Z.: Writing—review and editing. N.W.: Conceptualization, Supervision, Writing—review and editing. D.Y.: Writing—review and editing. Y.M.: Conceptualization, Project administration, Writing—review and editing. P.S.: Conceptualization, Data curation, Investigation, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors declare that financial support was received for the research and publication of this article. This work was supported by the National Natural Science Foundation of China (62476251, 32401708) and the National Key Research and Development Program of China (2019YFE0126100).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. Codes and datasets are also available at <https://github.com/wuaixiang/LGD-YOLO> (accessed on 4 July 2026).

Conflicts of Interest: Author Yanan Mi was employed by the company Pegasor Oy during the conduct of this research. The remaining authors declare that the research was conducted in the absence of any commercial, financial, or personal relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CA	Channel Attention
CGLU	Convolutional Gated Linear Unit
CIoU	Complete Intersection over Union
DIoU	Distance Intersection over Union
DWConv	Depthwise Separable Convolution
Dy-HSFPN	Dynamic High-Level Screening Feature Pyramid Network
EIoU	Efficient Intersection over Union
FLOPs	Floating-Point Operations
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frames Per Second
GDN	Grouped Double Normalization
GFLOPs	Giga Floating-Point Operations
GIoU	Generalized Intersection over Union
GMKSF	Grouped Multi-Kernel Spatial Fusion
HiResCAM	High-Resolution Class Activation Mapping
IoU	Intersection over Union
LSDECD	Lightweight Shared Detail Enhancement Convolution
MemAC	Memory Access Cost

mAP	Mean Average Precision
MCU	Micro-Controller Units
MHSA	Multi-Head Self-Attention
MLP	Multi-Layer Perceptron
NMS	Non-Maximum Suppression
P	Precision
PANet	Path Aggregation Network
Params	Parameter Volume
PConv	Partial Convolution
PIoU	Pixels Intersection over Union
R	Recall
RGB	Red, Green, Blue
SFF	Selective Feature Fusion
SIoU	SCYLLA-Intersection over Union
TP	True Positive
YOLO	You Only Look Once

References

1. Zhao, Y.; Chen, Y.; Xu, X.; He, Y.; Gan, H.; Wu, N.; Wang, Z.; Sun, X.; Wang, Y.; Skobelev, P.; et al. Ta-YOLO: Overcoming Target Blocked Challenges in Greenhouse Tomato Detection and Counting. *Front. Plant Sci.* **2025**, *16*, 1618214. [\[CrossRef\]](#)
2. Sun, H.; Zheng, Q.; Yao, W.; Wang, J.; Liu, C.; Yu, H.; Chen, C. An Improved YOLOv8 Model for Detecting Four Stages of Tomato Ripening and Its Application Deployment in a Greenhouse Environment. *Agriculture* **2025**, *15*, 936. [\[CrossRef\]](#)
3. Gao, X.; Li, F.; Yan, J.; Sun, Q.; Meng, X.; Liu, P. A Lightweight Greenhouse Tomato Fruit Identification Method Based on Improved YOLOv11n. *Agriculture* **2025**, *15*, 1497. [\[CrossRef\]](#)
4. Zhang, X.; Adli, H.K.; Abdullah, N.; Ridzuan, F.; Wu, P. Brief Review on Deep Learning-Based Tomato Ripeness Detection Techniques for Smart Agriculture. *AIP Conf. Proc.* **2025**, 3338, 020004. [\[CrossRef\]](#)
5. Yu, J.; Liu, J.; Sun, C.; Wang, J.; Ci, J.; Jin, J.; Ren, N.; Zheng, W.; Wei, X. Sensing Technology for Greenhouse Tomato Production: A Systematic Review. *Smart Agric. Technol.* **2025**, *11*, 101020. [\[CrossRef\]](#)
6. Li, R.; Sun, X.; Yang, K.; He, Z.; Wang, X.; Wang, C.; Wang, B.; Wang, F.; Liu, H. A Lightweight Wheat Ear Counting Model in UAV Images Based on Improved YOLOv8. *Front. Plant Sci.* **2025**, *16*, 1536017. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Chen, C.; Zheng, Z.; Xu, T.; Guo, S.; Feng, S.; Yao, W.; Lan, Y. YOLO-Based UAV Technology: A Review of the Research and Its Applications. *Drones* **2023**, *7*, 190. [\[CrossRef\]](#)
8. Tang, Y.; Dananjayan, S.; Hou, C.; Guo, Q.; Luo, S.; He, Y. A Survey on the 5G Network and Its Impact on Agriculture: Challenges and Opportunities. *Comput. Electron. Agric.* **2021**, *180*, 105895. [\[CrossRef\]](#)
9. Xiao, F.; Wang, H.; Xu, Y.; Zhang, R. Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review. *Agronomy* **2023**, *13*, 1625. [\[CrossRef\]](#)
10. Attri, I.; Awasthi, L.K.; Sharma, T.P.; Rathee, P. A Review of Deep Learning Techniques Used in Agriculture. *Ecol. Inform.* **2023**, *77*, 102217. [\[CrossRef\]](#)
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
12. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. *Mask R-CNN*; The Computer Vision Foundation (CVF): Scarsdale, NY, USA, 2017; pp. 2961–2969.
13. Wang, Z.; Ling, Y.; Wang, X.; Meng, D.; Nie, L.; An, G.; Wang, X. An Improved Faster R-CNN Model for Multi-Object Tomato Maturity Detection in Complex Scenarios. *Ecol. Inform.* **2022**, *72*, 101886. [\[CrossRef\]](#)
14. Zu, L.; Zhao, Y.; Liu, J.; Su, F.; Zhang, Y.; Liu, P. Detection and Segmentation of Mature Green Tomatoes Based on Mask R-CNN with Automatic Image Acquisition Approach. *Sensors* **2021**, *21*, 7842. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Liu, J.; Wang, X. EggplantDet: An Efficient Lightweight Model for Eggplant Disease Detection. *Alex. Eng. J.* **2025**, *115*, 308–323. [\[CrossRef\]](#)
17. Paul, A.; Machavaram, R.; Ambuj, Kumar, D.; Nagar, H. Smart Solutions for Capsicum Harvesting: Unleashing the Power of YOLO for Detection, Segmentation, Growth Stage Classification, Counting, and Real-Time Mobile Identification. *Comput. Electron. Agric.* **2024**, *219*, 108832. [\[CrossRef\]](#)

18. Lawal, O.M. Development of Tomato Detection Model for Robotic Platform Using Deep Learning. *Multimed. Tools Appl.* **2021**, *80*, 26751–26772. [[CrossRef](#)]
19. Miao, Z.; Yu, X.; Li, N.; Zhang, Z.; He, C.; Li, Z.; Deng, C.; Sun, T. Efficient Tomato Harvesting Robot Based on Image Processing and Deep Learning. *Precis. Agric.* **2023**, *24*, 254–287. [[CrossRef](#)]
20. Deng, X.; Huang, T.; Wang, W.; Feng, W. SE-YOLO: A Sobel-Enhanced Framework for High-Accuracy, Lightweight Real-Time Tomato Detection with Edge Deployment Capability. *Comput. Electron. Agric.* **2025**, *239*, 110973. [[CrossRef](#)]
21. Liu, X.; Teng, W.; Yu, H.; Yao, Z.; Wang, C.; Peng, Y.; Han, X.; Liu, J. GAE-YOLO: A Lightweight Multimodal Detection Framework for Tomato Smart Agriculture with Edge Computing. *Front. Plant Sci.* **2025**, *16*, 1712432. [[CrossRef](#)] [[PubMed](#)]
22. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
23. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
24. Ji, W.; Pan, Y.; Xu, B.; Wang, J. A Real-Time Apple Targets Detection Method for Picking Robot Based on ShufflenetV2-YOLOX. *Agriculture* **2022**, *12*, 856. [[CrossRef](#)]
25. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis Detection and Three-Dimensional Localization of Cut off Point for Vision-Based Banana Robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [[CrossRef](#)]
26. Gao, X.; Ding, J.; Bie, M.; Yu, H.; Shen, Y.; Zhang, R.; Xi, X. YOLOv8n-FDE: An Efficient and Lightweight Model for Tomato Maturity Detection. *Agronomy* **2025**, *15*, 1899. [[CrossRef](#)]
27. Wang, Q.; Hua, Y.; Lou, Q.; Kan, X. SWMD-YOLO: A Lightweight Model for Tomato Detection in Greenhouse Environments. *Agronomy* **2025**, *15*, 1593. [[CrossRef](#)]
28. Gong, M.; Wang, D.; Zhao, X.; Guo, H.; Luo, D.; Song, M. A Review of Non-Maximum Suppression Algorithms for Deep Learning Target Detection. In Proceedings of the SPIE, Kunming, China, 12 March 2021; Volume 11763, pp. 821–828.
29. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
30. Lu, J.; Zhao, Y.; Yu, M. PGLD-YOLO: A Lightweight Algorithm for Pomegranate Fruit Localisation and Recognition. *PeerJ Comput. Sci.* **2025**, *11*, e3307. [[CrossRef](#)]
31. Li, A.; Wang, C.; Ji, T.; Wang, Q.; Zhang, T. D3-YOLOv10: Improved YOLOv10-Based Lightweight Tomato Detection Algorithm under Facility Scenario. *Agriculture* **2024**, *14*, 2268. [[CrossRef](#)]
32. Wu, Q.; Huang, H.; Song, D.; Zhou, J. YOLO-PGC: A Tomato Maturity Detection Algorithm Based on Improved YOLOv11. *Appl. Sci.* **2025**, *15*, 5000. [[CrossRef](#)]
33. Zeng, T.; Li, S.; Song, Q.; Zhong, F.; Wei, X. Lightweight Tomato Real-Time Detection Method Based on Improved YOLO and Mobile Deployment. *Comput. Electron. Agric.* **2023**, *205*, 107625. [[CrossRef](#)]
34. Tang, Y.; Zhou, H.; Wang, H.; Zhang, Y. Fruit Detection and Positioning Technology for a *Camellia oleifera* C. Abel Orchard Based on Improved YOLOv4-Tiny Model and Binocular Stereo Vision. *Expert Syst. Appl.* **2023**, *211*, 118573. [[CrossRef](#)]
35. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
36. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
37. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. *arXiv* **2023**. [[CrossRef](#)]
38. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Seoul, Republic of Korea, 2019; pp. 3007–3016.
39. Chen, Y.; Zhang, C.; Chen, B.; Huang, Y.; Sun, Y.; Wang, C.; Fu, X.; Dai, Y.; Qin, F.; Peng, Y.; et al. Accurate Leukocyte Detection Based on Deformable-DETR and Multi-Level Feature Fusion for Aiding Diagnosis of Blood Diseases. *Comput. Biol. Med.* **2024**, *170*, 107917. [[CrossRef](#)] [[PubMed](#)]
40. Han, K.; Wang, Y.; Guo, J.; Tang, Y.; Wu, E. Vision GNN: An Image Is Worth Graph of Nodes. In Proceedings of the Advances in Neural Information Processing Systems 35; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2022.
41. Liu, G.; Nouaze, J.C.; Mbouembe, P.L.T.; Kim, J.H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. *Sensors* **2020**, *20*, 2145. [[CrossRef](#)] [[PubMed](#)]
42. Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; Yan, S. MetaFormer Is Actually What You Need for Vision. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Piscataway, NJ, USA, 2022.
43. Shazeer, N. GLU Variants Improve Transformer. *arXiv* **2020**. [[CrossRef](#)]

44. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2023.
45. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*; PKP Publishing Services Network: Burnaby, BC, Canada, 2019.
46. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**. [[CrossRef](#)]
47. Zhang, H.; Zhang, S. Focaler-IoU: More Focused Intersection over Union Loss. *arXiv* **2024**. [[CrossRef](#)]
48. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.