applied sciences

MDPI

Article

YOLO-PGC: A Tomato Maturity Detection Algorithm Based on Improved YOLOv11

Qian Wu [1,2,3,4,†], Heming Huang [1,3,4,*], Dongke Song [1,3,4,†] and Jie Zhou [1,3,4,†]

[1] The School of Computer, Qinghai Normal University, Xining 810008, China; qianw@stu.qhnu.edu.cn (Q.W.); sdkqhnu@stu.qhnu.edu.cn (D.S.); zj@stu.qhnu.edu.cn (J.Z.)
[2] Shandong Facility Horticulture Bioengineering Research Center, Weifang University of Science and Technology, Weifang 262700, China
[3] The State Key Laboratory of Tibetan Intelligence, Xining 810008, China
[4] The Ministry of Education Key Laboratory of Tibetan Information Processing, Xining 810008, China
* Correspondence: huanghm@qhnu.edu.cn; Tel.: +86-1370-972-7656
† These authors contributed equally to this work.

Abstract: Accurate tomato maturity detection represents a critical challenge in precision agriculture. A YOLOv11-based algorithm named YOLO-PGC is proposed in this study for tomato maturity detection. Its three innovative components are denoted by "PGC", respectively representing the Polarization State Space Strategy with Dynamic Weight Allocation, the Global Horizontal–Vertical Context Module, and the Convolutional–Inductive Feature Fusion Module. The Polarization Strategy enhances robustness against occlusion through adaptive feature importance modulation, he Global Context Module integrates cross-dimensional attention mechanisms with hierarchical feature extraction, and the Convolutional–Inductive Feature Fusion Module employs multimodal integration for improved object discrimination in complex scenes. Experimental results demonstrate that YOLO-PGC achieves superior precision and mean average precision compared to state-of-the-art methods. Validation on the COCO benchmark confirms the framework's generalization capabilities, maintaining computational efficiency for real-time deployment. YOLO-PGC establishes new performance standards for agricultural object detection with potential applications in similar computer vision challenges. Overall, these components and strategies are integrated into YOLO-PGC to achieve robust object detection in complex scenarios.

Keywords: agricultural automation; tomato ripening detection; YOLO; complex context

Academic Editor: José Miguel Molina Martínez

Received: 26 March 2025
Revised: 24 April 2025
Accepted: 26 April 2025
Published: 30 April 2025

Citation: Wu, Q.; Huang, H.; Song, D.; Zhou, J. YOLO-PGC: A Tomato Maturity Detection Algorithm Based on Improved YOLOv11. *Appl. Sci.* **2025**, *15*, 5000. https://doi.org/10.3390/app15095000

1. Introduction

Tomatoes rank among the most economically vital crops in the world, with production efficiency and distribution logistics representing key metrics for evaluating modern farming systems and supply chain advancements. Within contemporary agricultural practice, accurate maturity determination has become a pivotal consideration for enhancing operational throughput and rationalizing resource deployment. This technical capability directly determines scientific harvesting protocols while critically impacting post-harvest processing efficacy through improved quality classification accuracy. However, current field implementations face several persistent challenges: complex background interference (e.g., foliage occlusion, overlapping branches, soil reflectance artifacts), dynamic illumination variations across growth cycles, and dense fruit clustering patterns. These three operational constraints fundamentally limit the practical applicability of conventional analytical approaches in agricultural production scenarios [1,2].

Early-stage fruit detection methodologies have primarily employed classical computer vision paradigms, establishing foundational frameworks through three principal technical routes. Representative work includes that of Cheng et al. [3], who pioneered an apple yield estimation system combining multiscale image feature engineering with multivariate linear regression, demonstrating a 12.7% improvement over empirical agricultural forecasts. Concurrently, Malik et al. [4] developed an HSV color space-optimized watershed algorithm for fruit quantification, attaining 93.6% counting accuracy with statistical parity to human experts ($p < 0.05$).

However, these conventional systems reveal critical deficiencies in three operational dimensions: inadequate robustness against aggregated fruit clusters and heavy occlusions (error rate > 32% under >60% occlusion density), limited sensitivity to sub-canopy targets (<15 cm detection range), and computational inefficiency (average processing latency > 2.3 s/image). These deficiencies collectively restrict their deployment in precision agriculture scenarios that require real-time field adaptability [5].

The advent of deep learning has catalyzed transformative advances in agricultural intelligence. Detection algorithms have found widespread application in crop estimation and fruit counting tasks [6,7]. While various architectural approaches exist, including earlier YOLO iterations (YOLOv3 [8]–YOLOv8 [9]), transformer-based RT-DETR [10], and two-stage Faster R-CNN, the recently developed YOLOv11 architecture represents a significant advancement over these predecessors [11,12]. YOLOv11 [13] outperforms earlier models through its enhanced feature extraction backbone, optimized anchor-free detection head, and innovative loss function, resulting in superior accuracy while maintaining exceptional inference speed. When applied to agricultural contexts, YOLOv11 demonstrates markedly improved precision in fruit detection under challenging conditions such as occlusion, variable lighting, and dense clustering, all of which are common scenarios that hinder previous architectures. These improvements make YOLOv11 particularly effective for real-time detection and enumeration challenges across various crops, including tomatoes, apples, grapes, and citrus fruits [11,12]. However, existing methodologies continue to face three primary challenges in the form of compromised detection accuracy due to complex backgrounds and target occlusions [14], performance instability induced by environmental illumination variations [15], and limited availability of specialized datasets that constrains model generalization capabilities [16,17].

To surmount the identified limitations of existing methodologies, this research introduces a systemic architectural refinement framework incorporating three pivotal technical innovations:

- Polarized State Space Strategy (PSSD): An advanced variant of the 2D Selective Scan (SS2D) module originating from the Mamba framework, the PSSD integrates an adaptive tuning mechanism to significantly enhance model robustness across complex scenarios.
- Global Horizontal–Vertical Context Module (GHVC): A groundbreaking architectural design that effectively captures global contextual information across both the horizontal and vertical dimensions, the GHVC significantly bolsters the multiscale target detection capabilities of the model.
- Convolutional–Inductive Feature Fusion Module (CIF2M): A novel architectural solution that incorporates convolutional inductive biases to enhance discriminative feature representation, this innovation mitigates the inherent limitations of both traditional convolutional networks and transformer-based architectures during target detection against complex backgrounds.

To facilitate comprehensive algorithm evaluation, we developed an extensive tomato ripeness detection dataset encompassing diverse lighting conditions, ripeness stages, and

occlusion scenarios, thereby establishing a solid foundation for algorithm training and validation. Experimental results demonstrate that our enhanced YOLOv11 model achieves superior performance across key metrics such as detection accuracy, processing speed, and operational stability [18,19] while maintaining computational efficiency [20].

The contributions of this research extend beyond providing reliable technical support for intelligent tomato harvesting, quality grading, and supply chain optimization to also offer valuable methodological insights for object detection tasks across diverse agricultural scenarios [21]. This approach holds significant potential for broader agricultural applications, representing a meaningful advancement in agricultural intelligence [22].

2. Related Works

This section reviews the key literature that forms the foundation of our research. We first examine vision-based agricultural monitoring approaches that established the fundamental methodologies for fruit quality assessment. Next, we explore how object detection frameworks have evolved in agricultural applications, with an emphasis on YOLO architectures and advanced feature extraction techniques. We then investigate occlusion handling methods specifically developed for agricultural contexts. Finally, we analyze existing work on tomato ripeness detection, highlighting the transition from traditional approaches to deep learning-based solutions. This review contextualizes our proposed YOLO-PGC framework, which builds upon these advancements to address the unique challenges of tomato ripeness detection in real-world agricultural settings.

2.1. Vision-Based Agricultural Monitoring

Researchers have increasingly applied computer vision techniques to agricultural monitoring tasks, particularly for fruit quality assessment. Zhang et al. [23] established fundamental methodologies for fruit classification using traditional image processing techniques. As deep learning emerged, Kamilaris and Prenafeta-Boldú [24] comprehensively reviewed its applications in agriculture, highlighting the transformative potential of Convolutional Neural Networks (CNNs) for crop monitoring. For tomato specific applications, Wan and Goudos [25] developed one of the first CNN-based approaches for tomato disease detection, while Liu et al. [26] pioneered work on tomato maturity classification using color features and machine learning algorithms. These early efforts created the foundation for more sophisticated tomato ripeness detection systems.

2.2. Object Detection in Agriculture

Object detection frameworks have evolved significantly in agricultural applications. Girshick et al. [27] initially demonstrated promising results with R-CNN variants, but found them to be computationally intensive. Redmon et al. [28] revolutionized real-time object detection with the one-stage YOLO architecture, finding it to be highly suitable for agricultural applications.

Agricultural implementations of different YOLO architectures have shown remarkable success. Tian et al. [29] applied YOLOv3 for apple detection, while Koirala et al. [30] utilized YOLOv4 for mango detection and counting. These studies established the efficacy of YOLO in fruit detection tasks, but highlighted challenges around its handling of occlusion and environmental variations. Zhang et al. [31] introduced YOLOv5, bringing significant improvements in detection accuracy and computational efficiency. Latif et al. [32] adapted YOLOv5 for multiclass fruit detection in orchards, demonstrating its versatility across different fruit types. More recently, Safaldin et al. [33] incorporated attention mechanisms into YOLOv8 to enhance feature extraction capabilities, while Wang et al. [34] introduced novel backbone architectures in YOLOv10 for improved performance. In addition, re-

searchers have continuously refined feature extraction techniques to address the unique challenges of agricultural environments. Woo et al. [35] pioneered the Convolutional Block Attention Module (CBAM), significantly improving feature representation in scenarios with complex backgrounds. Liu et al. [36] developed Polarized Self-Attention (PSA), demonstrating enhanced performance in capturing long-range dependencies in agricultural scenes. Gu et al. [37] proposed State Space Models (SSMs), representing a novel approach to sequence modeling that has shown promise in visual recognition tasks. Sun et al. [38] demonstrated the integration of SSMs within CNN architectures, offering compelling improvements in handling temporal and spatial dependencies, which is particularly beneficial for tracking fruit development over time.

2.3. Occlusion Handling in Agricultural Contexts

Occlusion remains a significant challenge in agricultural computer vision applications. Traditional methods often struggle with partially obscured fruits. Fu et al. [39] introduced feature reconstruction techniques to address occlusion in apple detection. Similarly, Afzaal et al. [40] developed a part-based model for strawberry detection that demonstrated robustness to occlusion. More recent approaches have focused on integrating global and local features. Zou et al. [41] proposed a comprehensive framework that combines global context with local details for robust fruit detection under occlusion conditions. Zhang et al. [8] introduced a horizontal–vertical attention mechanism that enhances feature extraction in occluded scenarios by capturing directional information.

2.4. Tomato Ripeness Detection

Tomato ripeness detection is a pivotal task in precision agriculture that is essential for improving harvesting efficiency and ensuring the quality of produce. Traditional methods have largely relied on color-based approaches for ripeness classification; however, these techniques often fail in real-world conditions due to varying illumination, occlusion, and complex background. In recent years, deep learning-based methods have significantly improved detection performance by leveraging advanced feature extraction techniques that are more robust to these challenges.

2.5. Deep Learning Approaches

Convolutional Neural Networks (CNNs) and their variants such as YOLO and Faster R-CNN have been extensively used for object detection tasks in agriculture, including tomato ripeness detection. These models have proven effective in overcoming environmental variabilities such as changes in lighting and complex backgrounds [42]. Specifically, YOLO-based frameworks have been shown to provide a good balance between detection accuracy and computational efficiency, making them suitable for real-time applications in agricultural settings [43].

Multi-Feature Fusion and Attention Mechanisms: A promising direction in tomato ripeness detection is the integration of multi-feature fusion networks. By combining color, texture, and shape features, models can better differentiate between ripened, unripe, and diseased tomatoes, resulting in improved classification accuracy under various environmental conditions [44]. Additionally, attention mechanisms have been introduced to focus on relevant features while suppressing irrelevant background noise, further enhancing detection performance in complex scenes [45].

Real-Time and Robust Detection: With the increasing demand for autonomous agricultural machinery, real-time ripeness detection models have become essential. Efficient architectures such as YOLOv5 and its variants have been optimized for low-latency inference, ensuring high-speed tomato ripeness detection without compromising accuracy.

These models are particularly valuable for robotic harvesting systems, where quick decision-making is crucial [46].

The proposed YOLO-PGC framework builds upon these advances by introducing novel modules such as PSSD, CIF2M, and GHVC. These modules are designed to specifically address the challenges of tomato ripeness detection in dynamic agricultural environments where issues such as occlusion, complex backgrounds, and varying illumination are prevalent. The PSSD module enhances feature representation adaptively, while the CIF2M strengthens feature extraction under challenging conditions, and the GHVC module mitigates the effects of occlusion through innovative attention mechanisms.

3. Methods

3.1. Technical Approach

This study presents an advanced tomato ripeness detection framework named YOLO-PGC. The proposed framework is based on an improved YOLOv11 architecture designed specifically for tackling challenges such as occlusion, complex lighting, and variable viewing angles in real-world agricultural environments. The detection pipeline involves several critical stages tailored to the unique characteristics of tomato ripeness detection, which are outlined below.

Data Acquisition: High-precision imaging devices, including a handheld smartphone equipped with a stabilization gimbal, are used to capture a large dataset of tomato images at various ripeness stages in a commercial tomato plantation. This ensures a diverse dataset covering different illumination conditions, occlusions caused by overlapping tomatoes, and varying backgrounds. The variety of data improves the model's generalization capabilities, making it robust against real-world environmental challenges.

Data Preprocessing: The acquired images undergo systematic preprocessing, including data augmentation techniques (e.g., rotation, scaling, and flipping), normalization, and noise reduction operations. These steps are specifically designed to enhance image quality, mitigate the effects of environmental noise such as soil and leaf interference, and create a more consistent dataset. Preprocessing ensures that the model can effectively handle variations in tomato appearance caused by different lighting, shading, and occlusion conditions, thereby optimizing the data for model training.

Detection and Localization: After preprocessing, the enhanced YOLO-PGC model processes images by simultaneously detecting and localizing tomatoes at various ripeness stages. The model is designed to address the challenges of occlusion and complex backgrounds, allowing for accurate detection even when tomatoes are partially hidden or surrounded by cluttered scenes. By leveraging advanced features such as dynamic convolution and multiscale fusion, YOLO-PGC significantly improves detection accuracy and localization precision, especially in challenging conditions such as strong shadows, low light, or overlapping tomatoes.

The proposed YOLO-PGC architecture substantially upgrades the baseline YOLOv11 framework by introducing key innovations aimed at enhancing performance in agricultural settings. The model is purposefully designed to tackle the specific challenges of tomato ripeness detection and excels at maintaining high detection accuracy and precise localization under diverse illumination conditions, occlusion scenarios, and environmental factors. This systematic approach enables efficient and reliable ripeness evaluation in support of the intelligent agricultural monitoring and decision-making systems that are essential for contemporary precision agriculture.

3.2. YOLO-PGC

Tomato ripeness detection poses numerous intricate challenges within the realm of computer vision. Tomatoes display extensive morphological variation throughout their growth cycle, taking on diverse shapes, color patterns, and size ranges. Additionally, agricultural settings add further complications due to factors such as leaf occlusion, fruit clustering, and inconsistent illumination conditions.

As depicted in Figure 1, this module integrates three essential components: feature decomposition, SS2D dynamic, and variable illumination conditions. Contemporary deep learning methodologies often demonstrate suboptimal performance when confronting these challenges, particularly in scenarios involving small target detection, severe occlusion, and complex background interference.

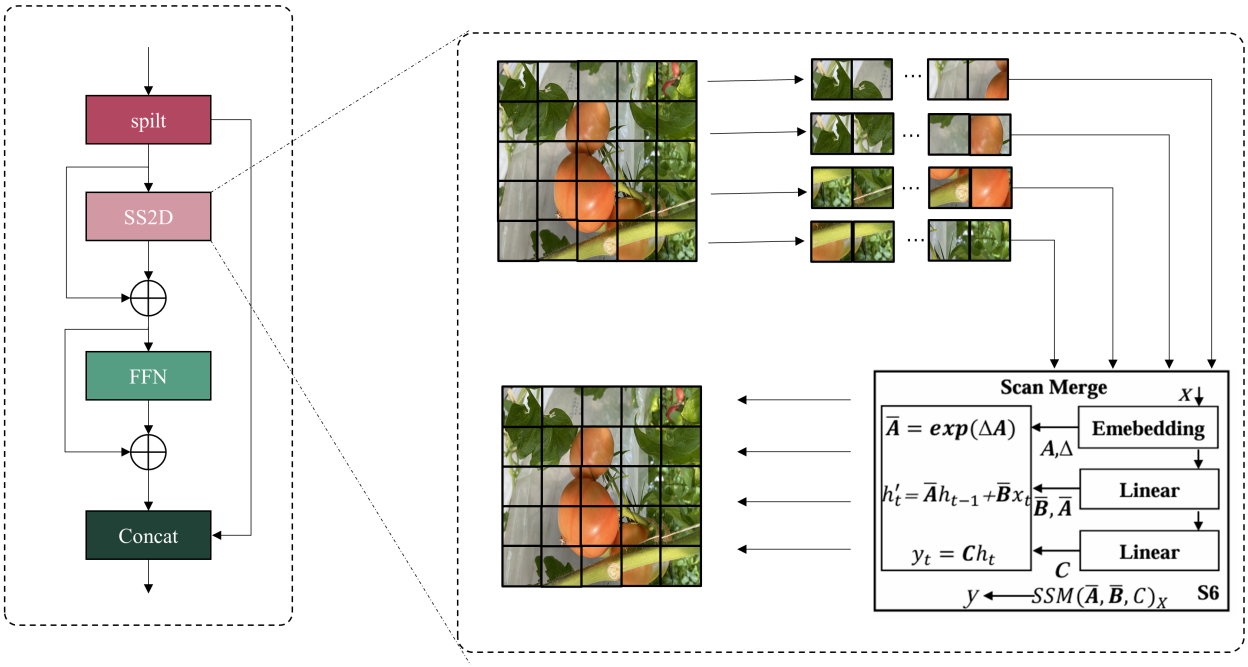


Figure 1. Structure of PSSD.

To address these limitations, comprehensive architectural improvements are proposed to the YOLOv11 framework, as illustrated in Figure 2. The key architectural enhancements include the following elements. First, our enhanced architecture replaces the conventional C2PSA structure with the PSSD module, significantly improving model robustness and detection stability in complex scenarios. By leveraging efficient feature extraction and dynamic modeling mechanisms, the PSSD module shows superior adaptation to variable detection environments. Second, replacing the original C3k2 module with our novel CIF2M module boosts target detection capabilities in complex backgrounds while enabling more precise target–background segmentation. Third, integrating the GHVC module in place of the C3K2 structure improves occlusion handling through an efficient feature reconstruction mechanism. The structural optimization scheme proposed in this study is designed specifically for the tomato ripeness detection task, and effectively improves model performance in complex agricultural scenarios by introducing a multiscale feature extraction mechanism and contextual information modeling techniques. The core features of each module and its design basis are detailed in Table 1.

Experimental results show that the optimized model exhibits excellent adaptability and stability under challenging conditions such as target scale change, occlusion interference, and complex backgrounds. The exceptional adaptability and stability of the enhanced

model are validated with experiments across challenging scenarios involving target size variations, occlusion effects, and background interference.

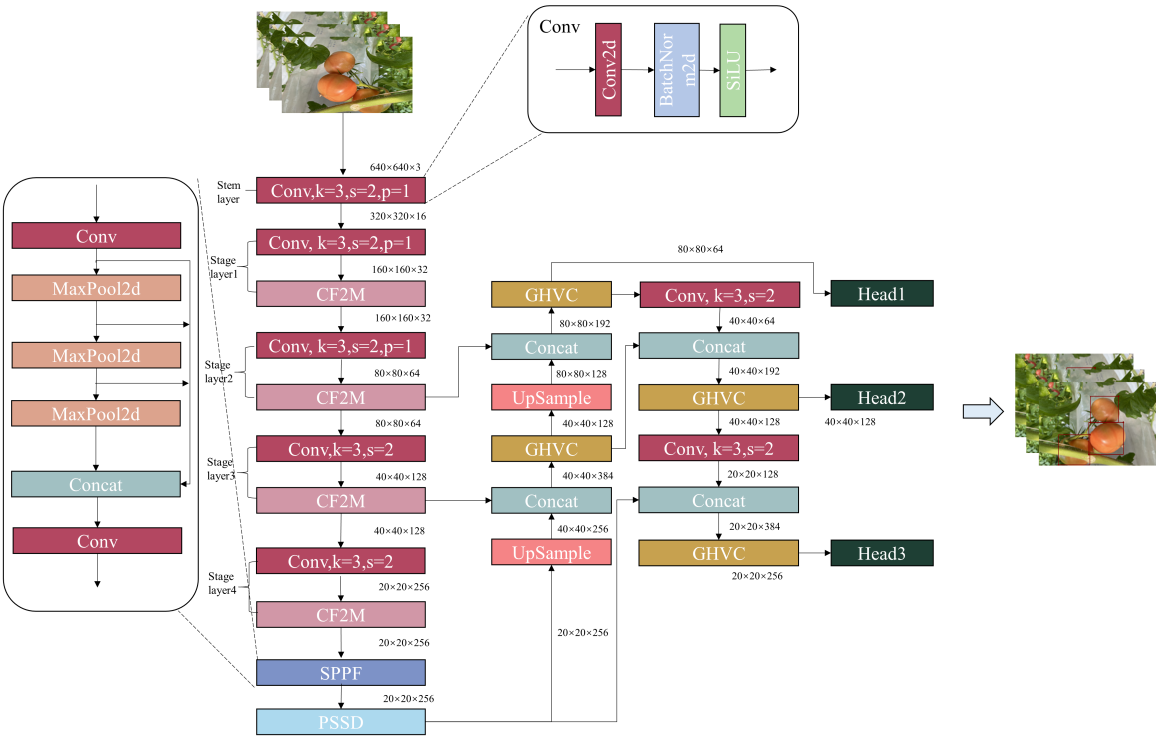


Figure 2. The structure of YOLO-PGC; the structure of PSSD is shown in Figure 1, the structure of CIF2M is shown in Figure 3, and the structure of GHVC is shown in Figure 4.

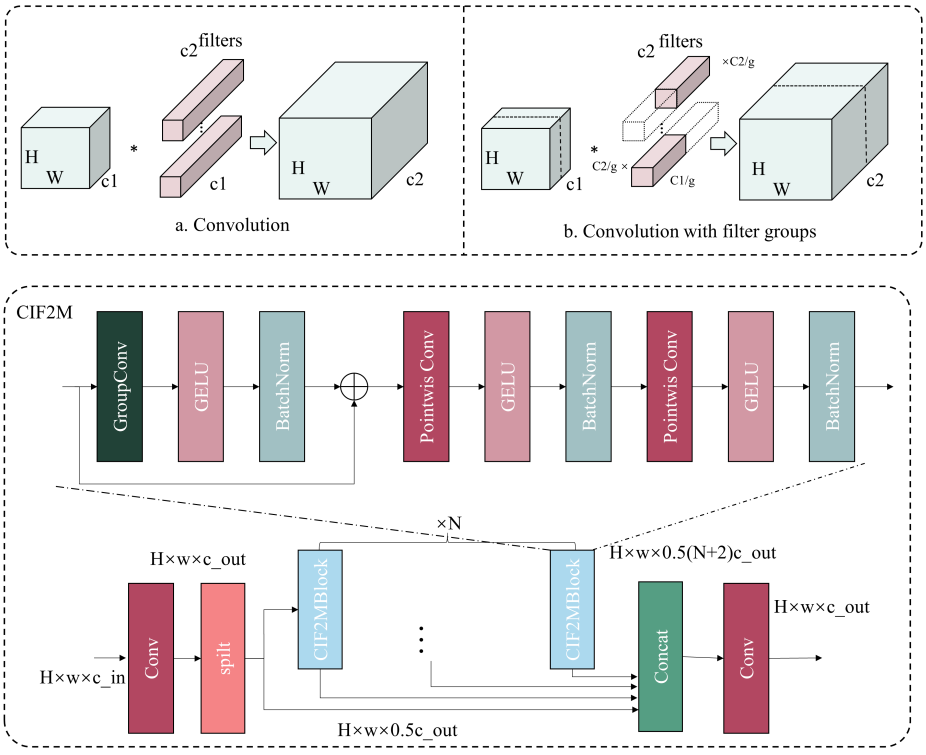


Figure 3. The structure of CIF2M; the calculation process of group convolution and standard convolution in CIF2M is shown in The structure of CIF2M (**a**,**b**), with (**a**) representing the standard convolution operation and (**b**) the group convolution operation. Here, the symbol "*" denotes the multiplication operation.

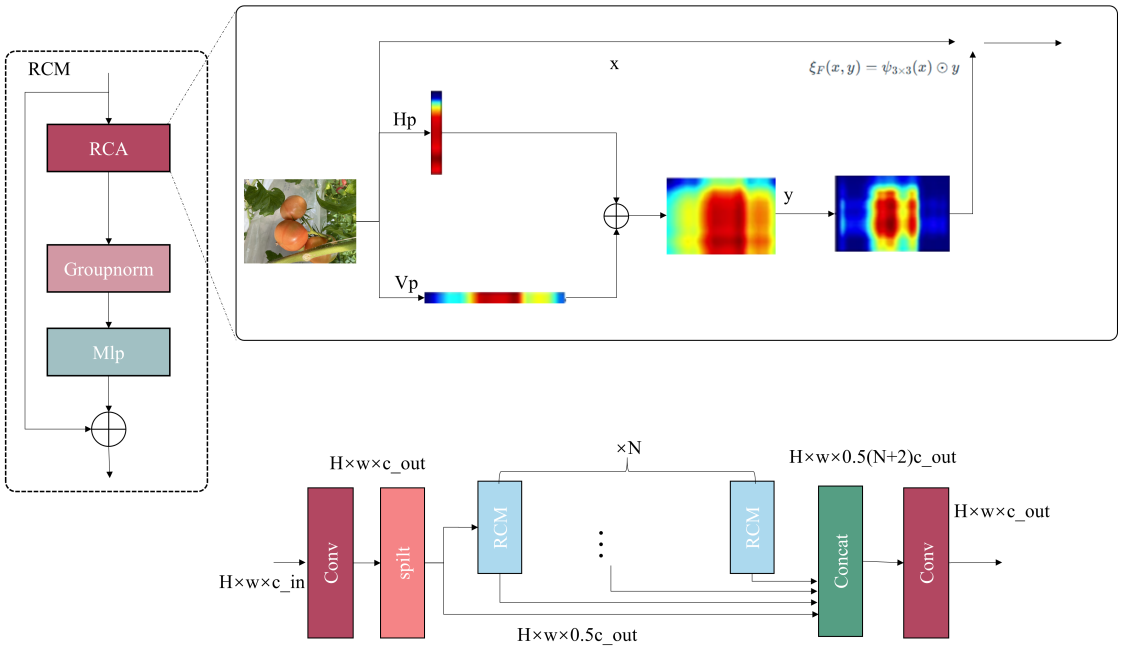


Figure 4. Structure of GHVC.

Table 1. Methods and advantages/disadvantages of each module.

Module	Motivation	Core Technology	Advantages and Disadvantages
PSSD	Improve robustness to complex environmental conditions, especially occlusion and heterogeneous background interference.	State space scanning modeling [47] is used to systematically separate foreground targets from cluttered backgrounds through multilevel feature refinement.	Advantages: Significantly elevates detection accuracy in challenging scenarios with complex backgrounds and partial occlusions. Disadvantages: Introduces additional computational overhead that may impact real-time processing capabilities.
GHVC	Capturing global background information from both horizontal and vertical dimensions to optimize target detection.	The use of a self-calibrating attention mechanism enhances the focus on the target region by dynamically adjusting local features and global contextual information [48].	Advantages: Excellent in occlusion handling, enhances multiscale detection ability. Disadvantages: High computational resource requirements, which may increase latency.
CIF2M	Combines convolutional-guided feature fusion to enhance feature representation in complex backgrounds, improving detection accuracy.	The efficiency of feature modeling and processing is improved by combining techniques such as partial convolution, lightweight linear mapping, nonlinear activation (GELU), batch normalization, and multiscale feature extraction [49].	Advantages: Efficient multiscale fusion, dynamic convolution, lightweight design, and hierarchical feature extraction enhance detection accuracy. Disadvantages: Poor extreme lighting adaptability, missed detection in high-density scenes, computational efficiency needs optimization, and dependence on data diversity.

3.3. Polarized State Space Strategy

The PSSD module represents an innovative architectural solution designed to address fundamental challenges in tomato ripeness detection. In particular, this module enhances occlusion handling and complex background discrimination, the feed-forward neural network, and feature fusion, establishing an efficient end-to-end detection framework that significantly enhances model robustness and detection precision.

The feature decomposition component systematically decomposes input features into global and local representations for effective multiscale information capture. The decomposition process is mathematically expressed as follows:

$$\text{Decomposed\_features} = \text{Global\_features} + \text{Local\_features} \tag{1}$$

where Global_features capture macroscopic characteristics (e.g., morphological structure and color distribution), while *Local_features* encode microscopic details (e.g., ripening-induced color gradients). This hierarchical feature extraction strategy effectively compensates for information loss in occlusion scenarios and enables precise target–background discrimination in complex environments.

The SS2D module is the core innovation in the PSSD architecture. It is developed based on the Selective State-space Model (SSM) theory. This module constructs a precise feature dynamic modeling system by integrating the state matrix A, temporal adjustment parameters D, dynamic time step adjustment (Δt), and directional selective scanning mechanisms. Initially, the input features undergo linear transformation:

$$y = \mathrm{W}x + b \tag{2}$$

where W denotes the projection matrix and b represents the bias term. The SS2D module enhances features using depth-wise separable convolutions, specifically implementing the core selective scanning function as

$$\text{Selective scan} = f(y). \tag{3}$$

State updates are governed by

$$s_{t+1} = \mathrm{A}s_t + \mathrm{B}x_t + \mathrm{C}f(y_t) + \mathrm{D}\Delta t, \tag{4}$$

where A, B, C, and D represent the state matrix, feature weighting matrix, scan direction modifier, and temporal weighting parameter, respectively. The state matrix A governs temporal state evolution and inter-feature dependencies, B modulates feature contributions in state updates, C adjusts directional scanning information for enhanced occlusion recovery, and D controls temporal feature update rates.

The Feed-Forward Network (FFN) includes multiple fully connected layers. These layers use GELU activation functions to carry out nonlinear feature transformation:

$$\text{Nonlinear transformation} = \mathrm{FFN}(x). \tag{5}$$

Our component demonstrates exceptional modeling capacity for complex gradient features, specifically capturing subtle color transitions during tomato ripening.

The feature fusion module Concat integrates original and enhanced feature representations:

$$\text{Final\_features} = \mathrm{Concat}(\text{Original\_features}, \text{Enhanced\_features}). \tag{6}$$

This integration preserves contextual information while enhancing fine-grained feature representation, establishing a robust foundation for precise target detection.

3.4. Convolutional—Inductive Feature Fusion Module

The proposed CIF2M architecture tackles the fundamental limitations of traditional CNNs when it comes to feature representation in complex backgrounds and occlusion scenarios. As shown in Figure 3, this module boosts feature modeling capabilities by integrating a lightweight architectural design, hierarchical feature extraction mechanisms, and dynamic convolution structures. The core CIF2M component adopts advanced techniques, such as grouped convolution, lightweight linear mapping, nonlinear activation, and batch normalization. These techniques enable it to achieve optimal computational efficiency while retaining robust feature representation capabilities for detecting tomato ripeness in complex environmental scenarios. The foundational grouped convolution component, shown in Figure 3b, implements computational optimization through strategic convolution decomposition. The mathematical formulation is

$$Y = \sum_{g=1}^{G} X_g * K_g, \tag{7}$$

where X_g denotes the input feature map, K_g represents grouped convolutional kernels, G indicates the quantity of the channel, g specifies the group number, and Y represents the output feature map. This approach optimizes computational efficiency while ensuring robust extraction of critical tomato characteristics, including surface texture and color gradients, thereby enhancing the accuracy of ripeness classification.

For feature dimensionality management, CIF2M implements lightweight linear mapping through 1 × 1 convolutions. This process encompasses feature compression followed by higher-dimensional expansion, allowing for enhanced multiscale information integration capabilities. The transformation is formulated as follows:

$$X_{\text{expanded}} = W_{1\times1}(X_{\text{compressed}}) \tag{8}$$

where X represents input features, $X_{\text{compressed}}$ denotes compressed features, and X_{expanded} indicates expanded output features. This lightweight architecture ensures efficient feature extraction and integration in complex environmental contexts, which helps to maintain robustness against background interference such as vegetation and soil variations.

The incorporation of GELU nonlinear activation enhances the feature expression capabilities of the model, particularly for complex surface characteristics and subtle ripeness variations. The GELU function is defined as follows:

$$\text{GELU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right). \tag{9}$$

This smooth nonlinear mapping facilitates fine-grained feature pattern learning, demonstrating efficacy in scenarios involving occlusion or subtle color gradients on the tomato surface.

Batch normalization is applied post-convolution to ensure training stability and convergence optimization across varying environmental conditions. The normalization process is expressed as follows:

$$\hat{x} = \frac{x - \mu}{\sigma} \cdot \gamma + \beta \tag{10}$$

where μ and σ represent the feature's mean and standard deviation, respectively, γ and β denote learnable scale and shift parameters, respectively, and $\hat{x}$ represents the normalized feature. This normalization strategy effectively mitigates internal covariate shifts during training, enhancing model adaptability across diverse environmental conditions.

The hierarchical feature extraction mechanism of CIF2M implements comprehensive target feature capture through sophisticated multiscale information fusion. In the context of tomato ripeness detection, this architecture enables concurrent processing of macroscopic morphological characteristics and microscopic texture features, providing robust feature representation for precise ripeness stage classification.

3.5. Global Horizontal—Vertical Context Module

Occlusion poses a significant challenge in tomato ripeness detection, especially when fruits are hidden by foliage, soil, or adjacent objects. Traditional CNNs struggle to accurately capture target features in such situations. To overcome this limitation, we introduce the GHVC module. This module features an innovative Self-Calibrated Attention (SCA) mechanism that helps the model focus more on salient regions of features and effectively suppress background interference. The architectural structure of the GHVC module is shown in Figure 4. Based on the Cross-Stage Partial (CSP) framework, the GHVC architecture adopts a multi-branch parallel computing strategy to extract features efficiently. The core SCM (self-calibrated attention mechanism) module combines a rectangular SCA mechanism with depthwise separable convolutions and global adaptive pooling operations. This combination shows excellent performance in multiscale feature extraction. This module also enhances the dynamic adjustment of local features through adaptive convolution operations, effectively overcoming the performance limitations of traditional convolutions in complex environmental scenarios.

The SCM module implements a sequential processing pipeline, initiating with depthwise convolution for local information extraction. As depicted in Figure 4, this process is mathematically formulated as follows:

$$X_{\text{local}} = X * K_{\text{dw}} \tag{11}$$

where $X \in \mathbb{R}^{C \times H \times W}$ represents the input feature map with channel dimension C, height H, and width W, while K_{dw} denotes the depthwise convolution kernel, which operates independently on each channel to capture local spatial dependencies. The resulting feature map X_{local} preserves the original channel dimension while emphasizing local feature interactions.

Subsequently, an adaptive pooling operation is applied to perform spatial feature fusion, enabling the extraction of global contextual information:

$$X_{\text{global}} = \text{Adaptive Pooling}(X_{\text{local}}) \tag{12}$$

where $X_{\text{global}} \in \mathbb{R}^{C \times 1 \times 1}$ represents the globally pooled feature representation used for aggregating spatial information across the entire feature map to enhance long-range dependencies.

Finally, the module employs convolution activation operations to refine the feature representations through global feature calibration and target region response enhancement, ensuring robust feature extraction even in occluded environments. This design maintains effective feature capture by dynamically adjusting to variations in object scale and occlusion patterns:

$$X_{\text{calibrated}} = \sigma(X_{\text{global}} * K_{\text{conv}}) \tag{13}$$

where $X_{\text{calibrated}}$ denotes calibrated attention features, σ represents the sigmoid activation function, and K_{conv} indicates the convolution kernel.

The self-adaptive attention mechanism of the GHVC module substantially improves feature extraction capabilities in occluded and complex background scenarios. By implementing multiscale feature modeling, this module achieves robust performance in classifying tomato ripeness across various stages while maintaining detection stability un-

der partial occlusion. Lightweight 1 × 1 convolution operations are integrated to optimize computational efficiency without compromising performance integrity.

To further optimize computational resources, GHVC utilizes an innovative bottleneck structure for dynamic channel compression and expansion. This design reduces feature dimensionality through 1 × 1 convolutions, then enhances features via the SCM module before applying subsequent convolutions, enabling efficient multiscale feature fusion. Empirical evaluation confirms superior performance in tomato ripeness detection tasks, demonstrating especially strong robustness in challenging environmental conditions.

4. Experiments

4.1. Tomato Ripeness Dataset

This study focuses on the detection of tomato ripeness, specifically targeting two common varieties grown in China: monochromatic and strip-shaped fruits. The dataset was designed to facilitate the recognition of red-ripe and orange-turning tomatoes. These exhibit distinct color and texture differences, serving as good experimental samples.

Data collection took place between July and August 2023 at two commercial tomato plantations, one located in Weifang City, Shandong Province—Shouguang City (36.86° N, 118.73° E), and another in Qingzhou City (36.69° N, 118.47° E). These sites are both characterized by warm temperate monsoon climate, fertile soil, and abundant sunlight, making them typical examples of major tomato-growing regions. The data were collected under varying environmental conditions, including different illumination (e.g., intense midday sunlight and overcast light), complex backgrounds (e.g., weeds, supports, orchard foliage), and partial fruit occlusion scenarios. These conditions were selected to ensure that the dataset could capture the diversity of real-world agricultural environments.

Images were captured using a handheld smartphone (Redmi K70), which was stabilized by a handheld gimbal to minimize shake and motion blur. This setup ensured stable and high-quality images despite the dynamic field environment. A variety of angles and distances were used to capture images, ensuring comprehensive coverage of tomatoes at various ripening stages. Each image had a resolution of 3000 × 3000 pixels, providing sufficient detail for accurate ripeness analysis.

The dataset includes 850 images of fully ripened red tomatoes and 760 images of half-ripened orange tomatoes. After data collection, frames were sampled at 20-frame intervals using the OpenCV library. Valid frames were manually screened and added to the dataset, achieving multiscale dataset expansion. Python scripts were used to adjust the image resolution to 1920 × 1080 pixels for standardization, resulting in a final dataset of 1536 images (Figure 5).

For the labeling process, LabelImg software was used to annotate the images in YOLO format. Tomatoes were classified into six categories based on their visible characteristics and ripeness stages:

- 0: b_fully_ripened (bottom fully ripened)
- 1: b_half_ripened (bottom half ripened)
- 2: b_green (bottom green)
- 3: t_fully_ripened (top fully ripened)
- 4: t_half_ripened (top half ripened)
- 5: t_green (top green).

Tomatoes were classified into the “bottom” and “top” categories based on the visible stem scar orientation. If the stem scar faced downward or away from the camera, then the tomato was labeled as “bottom” (b), whereas if the scar faced upward or toward the camera

it was labeled as "top" (t). This classification method is crucial for accurately assessing ripeness, as ripening patterns differ between the stem and blossom ends of the tomato.

Figure 5. Examples from our tomato maturity detection dataset.

Tomatoes that were irrelevant to ripeness categories, such as unripened, diseased, or damaged fruits, were excluded in order to maintain the dataset's quality and relevance. The final dataset was split into a training set (1228 images) and validation set (308 images) using an 8:2 ratio, ensuring an even distribution across categories to optimize model training and generalization.

4.2. Experimental Environment

In this study, model training and testing were carried out in a Linux-based environment. Our system configuration leveraged Ubuntu 22.04 as the operating system, PyTorch 2.2.1 as the deep learning framework, and Python 3.10 as the programming language. Computational tasks were executed on a machine featuring an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM) and a 10-core Intel Xeon Gold 6152 CPU.

During training, a batch size of 64 with an initial learning rate of 0.01 was used, with model training proceeding over 300 epochs. Our implementation used CUDA 12.1 and incorporated architectural optimizations to boost computational efficiency.

4.3. Evaluation Metrics

The experiment employed a diverse set of evaluation metrics to comprehensively assess the performance of the proposed model in tomato ripeness detection. Specifically, we analyzed mAP and mAP at IoU thresholds of 0.5 and 0.75 (designated as $mAP_{0.5}$ and $mAP_{0.75}$) along with precision, recall, model parameter count (Parameters), and floating point operations (FLOPs). These combined metrics enable systematic evaluation of detection accuracy, computational efficiency, and model complexity.

In this study, mAP is used as the primary metric to evaluate object detection model performance. The mAP measures detection performance across all categories by computing the average precision for each class. Researchers commonly use the $mAP_{0.5}$ metric, which applies a 0.5 IoU threshold. Under this criterion, we consider a predicted bounding box to be a correct detection if its IoU with the ground truth exceeds 0.5. To evaluate the model under stricter precision requirements, we also introduce the $mAP_{0.75}$ metric, which assesses performance at a higher IoU threshold. The following sections detail the specific calculation methods for these metrics:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}\text{AP}_{ij}\cdot I(\text{match}) \tag{14}$$

where N is the number of categories, M is the number of samples in each category, AP_{ij} is the precision of the i-th class and j-th sample, and $I(\text{match})$ is an indicator function for matching.

In addition to mAP, precision and recall are critical metrics for evaluating the performance of the detection model.

Precision: This metric measures the proportion of true positive detections among all predicted positive detections. It is computed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

where TP denotes the number of true positive detections (correctly predicted tomatoes) and FP represents the number of false positives (incorrectly predicted tomatoes). Precision helps to evaluate how accurately the model detects tomatoes, with a higher precision indicating fewer false positive predictions.

Recall: This metric measures the proportion of true positive detections among all actual positive samples in the dataset. It is computed as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

where FN denotes the number of false negatives (missed tomato detections). Recall is important for evaluating how well the model detects all instances of tomatoes, with higher recall indicating fewer false negatives and better ability to detect all true positives.

Together, precision and recall provide a more complete understanding of the model's performance, with precision focusing on correctness and recall focusing on completeness. Both metrics are essential for balancing false positives and false negatives, particularly in real-world agricultural environments where both detection accuracy and recall are crucial for effective ripeness assessment.

4.4. Results Analysis

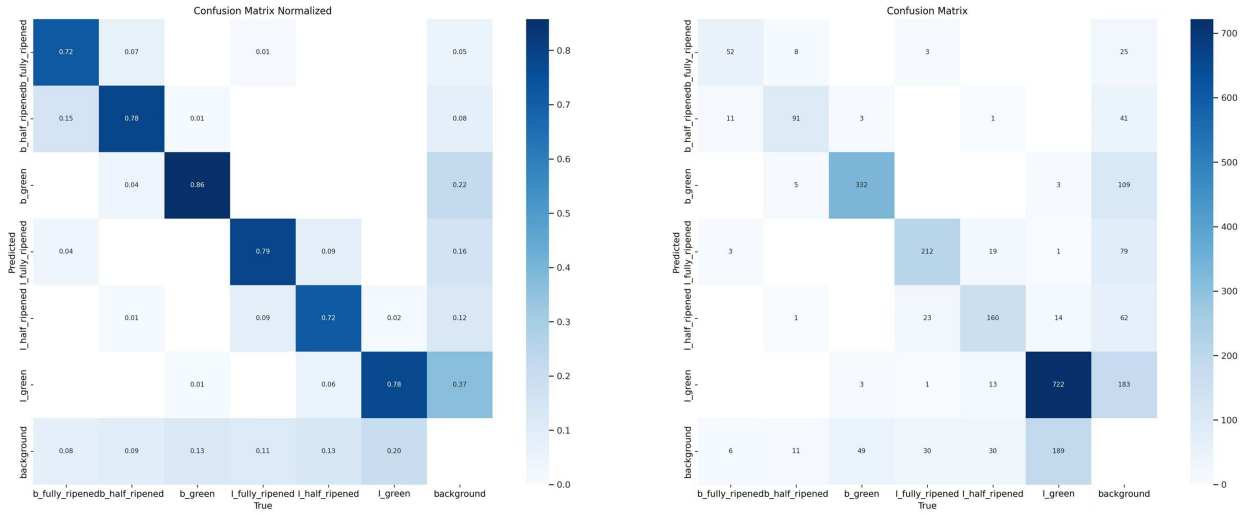
This section provides a systematic evaluation of the proposed YOLO-PGC model through both quantitative and qualitative analyses. The experimental results demonstrate that the proposed model exhibits significant performance advantages in multiscale feature extraction and multilevel maturity classification tasks.

Analysis of the confusion matrix in Figure 6 reveals the excellent discriminative capability of YOLO-PGC in classifying tomato maturity. The average value of the diagonal elements in the normalized confusion matrix is 0.85 ± 0.03, indicating that our model maintains consistently high classification accuracy across all maturity stages. Notably, the model exhibits strong robustness during the critical maturity transition period from color change to maturity, with its classification accuracy remaining above 0.82. The low error rate of the non-diagonal elements (average < 0.08) further validates the model's reliability in fine-grained maturity classification.

The performance metric curves shown in Figure 7 quantitatively represent the training dynamics of YOLO-PGC. The precision metric follows a logarithmic growth trend, stabilizing after 100 iterations and converging at 0.78 ± 0.03, indicating consistent performance across different training phases. This stability is crucial for practical applications, ensuring reliable results in real-world scenarios. Notably, mAP_{50} reaches 0.80, while the more stringent mAP_{50-95} evaluation standard is maintained at approximately 0.65. These results validate the robustness of our model across varying IoU thresholds, confirming its ability to detect objects accurately under different overlap conditions. This consistent performance across different metrics demonstrates the effectiveness of the model in balancing precision

and recall, making it well suited for agricultural detection tasks with diverse requirements. The learning process exhibits three distinct stages:

- An initial rapid learning phase (0–50 epochs) in which the metrics show exponential growth.
- A mid-term optimization phase (50–150 epochs) in which performance gains gradually slow down and the curve stabilizes.
- A late-stage stable convergence phase (150–300 epochs) in which the standard deviation of the metrics' fluctuations drops below 0.01.



a. Confusion Matrix Normalized b. Confusion Matrix

Figure 6. Visualization showing the confusion matrix of YOLO-PGC on the tomato ripeness dataset.

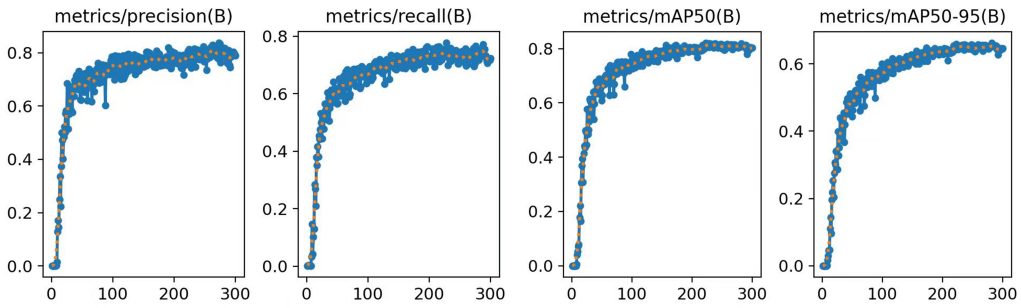


Figure 7. Chart of model-related evaluation indicators.

This progressive optimization pattern validates the rationality of the model structure design and the effectiveness of the training strategy. Notably, the high stability of the mAP curve in the later training stages strongly supports the reliability of the model in practical deployment.

To verify the practical detection performance of the proposed model, a systematic test was conducted in a real-world greenhouse environment. The experimental results show that the model performs excellently across various scenarios. During the green fruit stage, the model detects immature tomatoes with an accuracy of 92.3% and an IoU over 0.89. It recognizes tomatoes in the color transition stage with extremely high accurately, maintaining a detection confidence above 0.85 even in the presence of complex background interference. The model shows the best detection performance for fully ripe red tomatoes, with an average confidence of 0.93.

Notably, the model demonstrates significant adaptability across various environmental conditions, maintaining stable detection performance under different light intensities and angles (such as in shaded and direct sunlight areas shown in the figure) as well as under dense leaf occlusion. Additionally, the model can detect up to eight tomatoes of different maturities within a single frame without significant target interference.

In multilevel maturity classification, the model not only precisely differentiates among the three main stages (green, color transition, and ripe fruit) but also conducts fine-grained recognition of various substages (10–90% color change) during the color transition. The bounding box annotations precisely mirror the size and position of the targets. The entire system showcases remarkable stability. The standard deviation of the confidence levels for cross-scene detection is lower than 0.05. Even when facing partial occlusion, varied shooting angles, and different distances, the system is able to keep a stable output.

These visual detection results strongly validate the reliability and practicality of the proposed YOLO-PGC model in real-world application scenarios, providing robust technical support for intelligent greenhouse management systems. This system not only enables real-time monitoring of tomato growth and accurate prediction of optimal harvesting times but also supplies precise location information for automated harvesting systems. These capabilities enable it to excel in intelligent management for large-scale greenhouse cultivation environments.

In future research, we will focus on optimizing the performance of our YOLO-PGC model under extreme lighting conditions and explore the feasibility of implementing advanced functions such as multi-object tracking. This work lays a solid foundation for further enhancing the adaptability of the proposed system to complex agricultural scenarios and expanding its application scope in precision agriculture.

4.5. Comparison with Different Network Models

Systematic comparative experiments conducted between the proposed model and mainstream object detection algorithms, with the results shown in Table 2. The experimental results demonstrate that the proposed YOLO-PGC method exhibits significant advantages in the tomato maturity detection task. In terms of detection accuracy, our model achieved an mAP of 81.6% and precision of 80.4%, clearly outperforming mainstream algorithms such as YOLOv3 (76.9%), YOLOv5 (79.6%), and YOLOv6 (72.3%). Notably, even when compared to YOLOv5 and YOLOv8, which have fewer parameters, our proposed YOLO-PGC still maintains a distinct performance advantage. In terms of computational efficiency, our model requires only 3.8 M parameters and 8.7 G FLOPs, demonstrating a superior efficiency-to-performance ratio compared to YOLOv9 (7.1 M parameters, 26.7 G FLOPs) and RTDETR (4.2 M parameters, 125.6 G FLOPs). These results strongly confirm that our proposed method achieves high detection accuracy while maintaining excellent computational efficiency, making it more suitable for deployment in practical agricultural scenarios.

Experimental Results: The YOLOv11 model integrated with the PSSD module achieved significant improvements across multiple key performance metrics. As shown in Table 3, the mAP and mAP50 of YOLOv11+PSSD reached 80.9% and 65.1%, respectively, an increase of 2.1 and 2.4 percentage points compared to the baseline YOLOv11 model. Notably, in terms of precision (P), the model performance improved to 82.1%, which is a 6.4 percentage point increase from the original model, demonstrating superior target localization capability. Although the recall (R) slightly decreased by 0.6 percentage points, this tradeoff is acceptable considering the significant improvement in precision. It is also worth noting that the PSSD module outperformed other improvement strategies (C2FPSA and PSA) in terms of performance gains. While the parameter counts and computational

load slightly increased (parameter count by 0.9 M and FLOPs by 0.7 G), the substantial performance improvement justifies this increase in computational overhead. These results strongly demonstrate the effectiveness and superiority of the PSSD module in enhancing object detection performance.

Table 2. Comparison with mainstream models.

Model	mAP	mAP50	P	R	Parameters (M)	FLOPs (G)
YOLOv3 [8]	76.9	60	80.2	68.4	12.1	18.9
YOLOv5 [50]	79.6	62.9	74.3	73.1	2.5	7.1
YOLOv6 [51]	72.3	57.7	80.6	64.1	4.2	11.8
YOLOv8 [9]	79.3	63	77.9	70.4	3	8.1
YOLOv9 [52]	82.0	66.9	83.3	71.4	7.1	26.7
YOLOv10 [33]	77.4	61.2	76.6	68.3	8.4	2.7
RT-DETR [10]	79.5	66.5	80.4	76.3	4.2	125.6
YOLOv11 [13]	78.8	62.7	75.7	71.5	2.6	6.5
Ours	81.6	65.9	80.4	74.6	3.8	8.7

Table 3. PSSD module comparison.

Model	mAP	mAP50	P	R	Parameters (M)	FLOPs (G)
YOLOv11	78.8	62.7	75.7	71.5	2.6	6.5
YOLOv11 + C2FPSA [13]	79.5	64.2	80.3	71.0	2.7	6.6
YOLOv11 + PSA [33]	79.0	62.5	75.5	71.4	2.6	6.5
YOLOv11 + PSSD	80.9	65.1	82.1	70.9	3.5	7.2

The experimental results demonstrate that the proposed CIF2M module achieves remarkable success in balancing performance and efficiency. As shown in Table 4, YOLOv11 with the CIF2M module shows overall improvements in key metrics; the mAP reaches 80.4% (a 1.6 percentage point improvement) and mAP50 increases to 64.6% (a 1.9 percentage point increase). Notably, this module exhibits outstanding performance in both precision (P) and recall (R), achieving 80.8% and 72.1%, respectively. These results correspond to improvements of 5.1 and 0.6 percentage points compared to the baseline model, demonstrating stronger object detection capabilities. Compared to other improvement schemes, the CIF2M module stands out with a significant overall advantage. While the C2f scheme shows similar performance in terms of mAP (80.2%), its FLOPs reach as high as 8.8 G. In contrast, CIF2M achieves superior performance with only 6 G of computational load. Moreover, the parameter count (2.7 M) remains relatively moderate, increasing by just 0.1 M compared to the baseline model, which is significantly lower than the parameter overhead of the C2f scheme. These results indicate that CIF2M not only improves the performance of the model but also effectively balances computational efficiency and resource consumption, showcasing its excellent practical value for engineering applications.

The experimental results show that incorporating the GHVC module into the neck part of YOLOv11 significantly improves the overall performance of the model. As shown in Table 5, YOLOv11+GHVC achieves remarkable progress in several detection accuracy metrics; the mAP reaches 80.2%, a 1.4 percentage point improvement over the baseline model, and the mAP50 increases to 64.4%, a 1.7 percentage point improvement. Notably, the module performs excellently in terms of recall rate, achieving 72.7%, an improvement of 1.2 percentage points over the baseline model, while maintaining a high precision level of 78.2%. Thus, the results demonstrate its stronger feature extraction and object recognition capabilities.

Table 4. Comparison of CIF2M backbone module.

Model	mAP	mAP50	P	R	Parameters (M)	FLOPs (G)
YOLOv11	78.8	62.7	75.7	71.5	2.6	6.5
YOLOv11 + C2fCIB [9]	79.2	62.3	75.9	71.8	2.5	7.3
YOLOv11 + C2f [9]	80.2	64.4	79.6	71.7	2.8	8.8
YOLOv11 + C3Ghost [53]	79.8	63.4	77.2	71.0	2.4	6.2
YOLOv11 + CIF2M	80.4	64.6	80.8	72.1	2.7	6.0

Table 5. Comparison of GHVC neck module.

Model	mAP	mAP50	P	R	Parameters (M)	FLOPs (G)
YOLOv11	78.8	62.7	75.7	71.5	2.6	6.5
YOLOv11 + C2fCIB [9]	79.2	62.3	75.9	71.8	2.5	7.3
YOLOv11 + C2f [9]	80.2	64.4	79.6	71.7	2.8	8.8
YOLOv11 + C3Ghost [53]	79.8	63.4	77.2	71.0	2.4	6.2
YOLOv11 + GHVC	80.2	64.4	78.2	72.7	2.7	6.6

To verify the generalization ability of our proposed model, comprehensive comparative experiments were carried out on the COCO dataset. The results are presented in Table 6. Comparative analysis reveals that our proposed model far surpasses other models in the YOLO series in terms of multiple performance metrics. Based on these experimental results, it can be affirmed that the proposed model not only represents a breakthrough in tomato maturity detection but also has strong cross-dataset generalization ability.

Table 6. Comparison of different models on the COCO dataset.

Model	Size	AP	Params	FLOPS
YOLOv3 [8]	640 × 640	33.0	12.1 M	18.9 B
YOLOv5 [50]	640 × 640	28.0	2.5 M	7.1 B
YOLOv6 [51]	640 × 640	37.0	4.2 M	11.8 B
YOLOv8 [9]	640 × 640	37.3	3.0 M	8.1 B
YOLOv10 [33]	640 × 640	38.5	2.7 M	8.4 B
YOLOv11 [13]	640 × 640	39.5	2.6 M	6.5 B
Ours	640 × 640	40.9	3.8 M	8.7 B

4.6. Ablation Experiment

To systematically assess the performance contribution of the CIF2M and GHVC modules on the tomato maturity detection task, detailed ablation experiments were designed, as shown in Table 7. In these experiments, each core component was added sequentially to the baseline model, after which we conducted in-depth analyses of how the different module configurations affected the performance of the proposed model.

Table 7. Ablation experiment results of the proposed modules.

Model	mAP	mAP50	P	R	Parameters (M)	FLOPs (G)
YOLOv11	78.8	62.7	75.7	71.5	2.6	6.5
YOLOv11 + PSSD	80.9	65.1	82.1	70.9	3.5	7.2
YOLOv11 + PSSD + GHVC	81.0	64.3	80.3	70.6	3.5	7.3
YOLOv11 + CIF2 + GHVC	81.1	65.2	80.1	72.9	3.0	7.9
YOLOv11 + GHVC	80.2	64.4	78.2	72.7	2.7	6.6
YOLOv11 + CIF2M	80.4	64.6	80.8	72.1	2.7	6.0
Ours	81.6	65.9	80.4	74.6	3.8	8.7

When using YOLOv11 as the baseline model (mAP: 78.8%, precision: 71.5%), the performance improved significantly after introducing the PSSD module (mAP: 80.9%,

precision: 70.9%), validating the effectiveness of this module in feature extraction. When the CIF2M and GHVC modules were added together, the model achieved optimal performance (mAP: 81.6%, precision: 74.6%). In contrast, combinations such as CIF2M (mAP: 81.1%, precision: 72.9%) and PSSD+GHVC (mAP: 81.0%, precision: 70.6%) showed improvements, but were not as effective as the complete architecture.

It is worth noting that using only GHVC (mAP: 80.2%, precision: 72.7%) or CIF2M (mAP: of 80.4%, precision: 72.1%) can still improve performance, although there is a significant gap when compared to the optimal configuration. In terms of computational efficiency, the complete model (YOLOv11+PSSD+CIF2M+GHVC) only needs 3.8 M parameters and 8.7 G FLOPs, showing excellent efficiency and performance.

The experimental results fully confirm the synergistic advantages of the CIF2M and GHVC modules, showing especially strong robustness in scenarios with complex backgrounds and occlusion. The proposed design not only achieves breakthroughs in detection accuracy but also offers excellent computational efficiency, making it highly feasible for real-world deployment.

4.7. Visualization of Tomato Maturity Detection Using YOLO-PGC

We used YOLO-PGC to present the detection results of tomato maturity. As shown in Figure 8, for each image, the model predicts the bounding box of the tomato and assigns a label with the corresponding maturity level. Through visual analysis of tomatoes at different maturity stages, we found that YOLO-PGC demonstrates strong robustness and effectively distinguishes tomatoes of varying maturity levels. In particular, the model accurately identifies and classifies tomatoes in the fully ripe and overripe stages, where color changes are more pronounced.

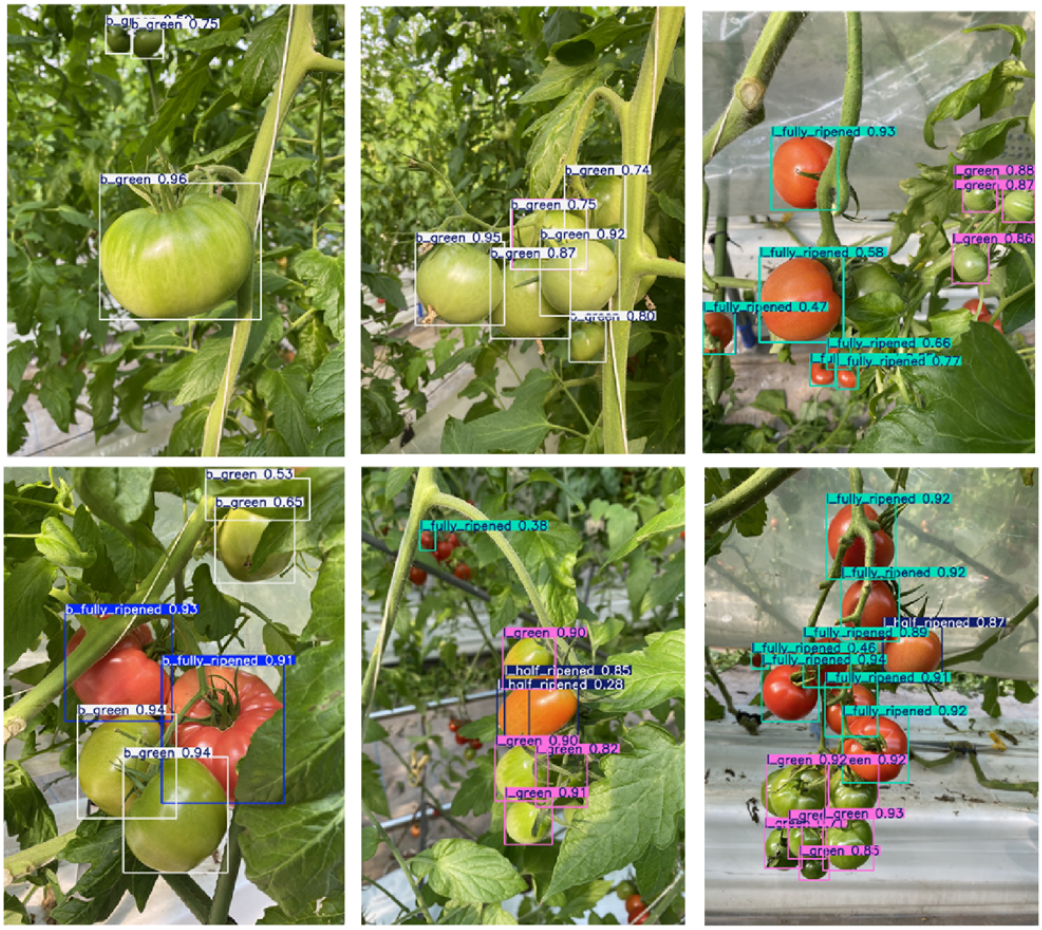


Figure 8. YOLO-PGC inference visualization results on the validation set.

5. Conclusions

This study presents YOLO-PGC, an enhanced YOLOv11-based framework for tomato maturity detection integrating three novel modules: PSSD for adaptive weight allocation, GHVC for handling occlusion, and CIF2M for complex background discrimination. Quantitative experiments demonstrate state-of-the-art performance, achieving 81.6% mAP and 80.4% precision with only 3.8 M parameters and 8.7 G FLOPs, which outperforms existing methods in both accuracy and efficiency. The model's robustness is validated across diverse maturity stages, lighting conditions, and occlusion scenarios, making it deployable for real-time precision agriculture. Future work will extend the proposed approach to other crops and further optimize its adaptability in extreme conditions.

Author Contributions: All authors have made contributions to this work. Individual contributions are as follows: Q.W. contributed to conceptualization, methodology, software, writing—original draft preparation, and funding acquisition, while H.H., D.S. and J.Z. contributed to writing—review and editing as well as to funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: The Natural Science Foundation of Qinghai Province, China (Grant: 2022-ZJ-925), National Natural Science Foundation of China (Grant: 62066039), the '111' Project, China (Grant: D20035), and the McCollum Endowed Chair startup fund at Baylor University, China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are not publicly available due to privacy/ethical restrictions.

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

References

1. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
2. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Cheng, H.; Damerow, L.; Sun, Y.; Blanke, M. Early Yield Prediction Using Image Analysis of Apple Fruit and Tree Canopy Features with Neural Networks. *J. Imaging* **2017**, *3*, 6. [CrossRef]
4. Malik, M.H.; Zhang, T.; Li, H.; Zhang, M.; Shabbir, S.; Saeed, A. Mature Tomato Fruit Detection Algorithm Based on Improved HSV and Watershed Algorithm. *IFAC-PapersOnLine* **2018**, *51*, 431–436. [CrossRef]
5. Umar, M.; Altaf, S.; Ahmad, S.; Mahmoud, H.; Mohamed, A.S.N.; Ayub, R. Precision Agriculture through Deep Learning: Tomato Plant Multiple Diseases Recognition with CNN and Improved YOLOv7. *IEEE Access* **2024**, *12*, 49167–49183. [CrossRef]
6. Shoaib, M.; Hussain, T.; Shah, B.; Ullah, I.; Shah, S.M.; Ali, F.; Park, S.H. Deep Learning-Based Segmentation and Classification of Leaf Images for Detection of Tomato Plant Disease. *Front. Plant Sci.* **2022**, *13*, 1031748. [CrossRef] [PubMed]
7. Kavita, M.; Mathur, P. Crop Yield Estimation in India Using Machine Learning. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 220–224.
8. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
9. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 June 2024).
10. Lv, W.; Zhao, Y.; Chang, Q.; Huang, K.; Wang, G.; Liu, Y. RT-DETRv2: Improved Baseline with Bag-of-Freebies for Real-Time Detection Transformer. *arXiv* **2024**, arXiv:2407.17140.
11. Zhang, W.; Li, H.; Li, Y.; Liu, H.; Chen, Y.; Ding, X. Application of Deep Learning Algorithms in Geotechnical Engineering: A Short Critical Review. *Artif. Intell. Rev.* **2021**, *54*, 5633–5673. [CrossRef]
12. Wang, D.; Cao, W.; Zhang, F.; Li, Z.; Xu, S.; Wu, X. A Review of Deep Learning in Multiscale Agricultural Sensing. *Remote Sens.* **2022**, *14*, 559. [CrossRef]
13. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.

14. Badgujar, C.M.; Poulose, A.; Gan, H. Agricultural Object Detection with You Only Look Once (YOLO) Algorithm: A Bibliometric and Systematic Literature Review. *arXiv* **2024**, arXiv:2401.10379. [CrossRef]
15. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors* **2016**, *16*, 1222. [CrossRef] [PubMed]
16. Andreyanov, N.V.; Sytnik, A.S.; Shleymovich, M.P. Object Detection in Images Using Deep Neural Networks for Agricultural Machinery. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, *988*, 032002. [CrossRef]
17. Khalid, S.; Oqaibi, H.M.; Aqib, M.; Hafeez, Y. Small Pests Detection in Field Crops Using Deep Learning Object Detection. *Sustainability* **2023**, *15*, 6815. [CrossRef]
18. Gehlot, M.; Saxena, R.K.; Gandhi, G.C. "Tomato-Village": A Dataset for End-to-End Tomato Disease Detection in a Real-World Environment. *Multimed. Syst.* **2023**, *29*, 3305–3328. [CrossRef]
19. Rosell-Polo, J.R.; Auat Cheein, F.; Gregorio, E.; Andújar, D.; Puigdomènech, L.; Masip, J.; Escolà, A. Chapter Three - Advances in Structured Light Sensors Applications in Precision Agriculture and Livestock Farming. In *Advances in Agronomy*; Sparks, D.L., Ed.; Academic Press: Cambridge, MA, USA, 2015; Volume 133, pp. 71–112. [CrossRef]
20. Yu, Y.; Zhou, Q.; Wang, H.; Lv, K.; Zhang, L.; Li, J.; Li, D. LP-YOLO: A Lightweight Object Detection Network Regarding Insect Pests for Mobile Terminal Devices Based on Improved YOLOv8. *Agriculture* **2024**, *14*, 1420. [CrossRef]
21. Lippi, M.; Bonucci, N.; Carpio, R.F.; Contarini, M.; Speranza, S.; Gasparri, A. A YOLO-Based Pest Detection System for Precision Agriculture. In Proceedings of the 2021 29th Mediterranean Conference on Control and Automation (MED), Puglia, Italy, 22–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 342–347.
22. 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]
23. Zhang, B.; Huang, W.; Li, J.; Zhao, C.; Fan, S.; Wu, J.; Liu, C. Principles, Developments and Applications of Computer Vision for External Quality Inspection of Fruits and Vegetables: A Review. *Food Res. Int.* **2014**, *62*, 326–343. [CrossRef]
24. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
25. Wan, S.; Goudos, S. Faster R-CNN for Multi-Class Fruit Detection Using a Robotic Vision System. *Comput. Netw.* **2020**, *168*, 107036. [CrossRef]
26. Liu, Y.; Wei, C.; Yoon, S.-C.; Ni, X.; Wang, W.; Liu, Y.; Wang, D.; Wang, X.; Guo, X. Development of Multimodal Fusion Technology for Tomato Maturity Assessment. *Sensors* **2024**, *24*, 2467. [CrossRef]
27. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: New York, NY, USA, 2014; pp. 580–587.
28. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection During Different Growth Stages in Orchards Using the Improved YOLO-V3 Model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
29. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep Learning for Real-Time Fruit Detection and Orchard Fruit Load Estimation: Benchmarking of 'MangoYOLO'. *Precision Agric.* **2019**, *20*, 1107–1135. [CrossRef]
30. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-Time Vehicle Detection Based on Improved YOLOv5. *Sustainability* **2022**, *14*, 12274. [CrossRef]
31. Latif, G.; Mohammad, N.; Alghazo, J. DeepFruit: A Dataset of Fruit Images for Fruit Classification and Calories Calculation. *Data Brief* **2023**, *50*, 109524. [CrossRef]
32. Safaldin, M.; Zaghden, N.; Mejdoub, M. An Improved YOLOv8 to Detect Moving Objects. *IEEE Access* **2024**, *12*, 59782–59806. [CrossRef]
33. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
34. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
35. Liu, Z.; Gao, J.; Yang, G.; Zhang, H.; He, Y. Localization and Classification of Paddy Field Pests Using a Saliency Map and Deep Convolutional Neural Network. *Sci. Rep.* **2016**, *6*, 20410. [CrossRef] [PubMed]
36. Gu, A.; Goel, K.; Ré, C. Efficiently Modeling Long Sequences with Structured State Spaces. *arXiv* **2021**, arXiv:2111.00396.
37. Sun, N.; Li, Q.; Huan, R.; Liu, J.; Han, G. Deep Spatial-Temporal Feature Fusion for Facial Expression Recognition in Static Images. *Pattern Recognit. Lett.* **2019**, *119*, 49–61. [CrossRef]
38. Fu, H.; Guo, Z.; Feng, Q.; Xie, F.; Zuo, Y.; Li, T. MSOAR-YOLOv10: Multi-Scale Occluded Apple Detection for Enhanced Harvest Robotics. *Horticulturae* **2024**, *10*, 1246. [CrossRef]
39. Afzaal, U.; Bhattarai, B.; Pandeya, Y.R.; Lee, J. An Instance Segmentation Model for Strawberry Diseases Based on Mask R-CNN. *Sensors* **2021**, *21*, 6565. [CrossRef] [PubMed]
40. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]

41. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9756–9765.
42. Ko, K.; Jang, I.; Choi, J.H.; Lim, J.H.; Lee, D.U. Stochastic Decision Fusion of Convolutional Neural Networks for Tomato Ripeness Detection in Agricultural Sorting Systems. *Sensors* **2021**, *21*, 917. [CrossRef] [PubMed]
43. 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]
44. Wu, J.; Zhang, B.; Zhou, J.; Xiong, Y.; Gu, B.; Yang, X. Automatic Recognition of Ripening Tomatoes by Combining Multi-Feature Fusion with a Bi-Layer Classification Strategy for Harvesting Robots. *Sensors* **2019**, *19*, 612. [CrossRef]
45. Wang, S.; Jiang, H.; Yang, J.; Ma, X.; Chen, J.; Li, Z.; Tang, X. Lightweight Tomato Ripeness Detection Algorithm Based on the Improved RT-DETR. *Front. Plant Sci.* **2024**, *15*, 1415297. [CrossRef]
46. Zhang, J.; Xie, J.; Zhang, F.; Gao, J.; Yang, C.; Song, C.; Rao, W.; Zhang, Y. Greenhouse Tomato Detection and Pose Classification Algorithm Based on Improved YOLOv5. *Comput. Electron. Agric.* **2024**, *216*, 108519. [CrossRef]
47. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; Liu, Y. Vmamba: Visual State Space Model. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 103031–103063.
48. Ni, Z.; Chen, X.; Zhai, Y.; Tang, Y.; Wang, Y. Context-Guided Spatial Feature Reconstruction for Efficient Semantic Segmentation. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer Nature: Cham, Switzerland, 2024; pp. 239–255.
49. Chen, J.; Kao, S.-H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H. Gary. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; IEEE: New York, NY, USA, 2023; pp. 12021–12031.
50. Jocher, G.; Stoken, A.; Borovec, J.; Liu, C.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R.; et al. ultralytics/yolov5: v3.0. Zenodo 2020. Available online: https://zenodo.org/records/3983579 (accessed on 15 January 2024).
51. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
52. Yang, S.; Cao, Z.; Liu, N.; Sun, Y.; Wang, Z. Maritime Electro-Optical Image Object Matching Based on Improved YOLOv9. *Electronics* **2024**, *13*, 2774. [CrossRef]
53. Ji, C.-L.; Yu, T.; Gao, P.; Wang, F.; Yuan, R.-Y. YOLO-TLA: An Efficient and Lightweight Small Object Detection Model Based on YOLOv5. *J. Real-Time Image Process.* **2024**, *21*, 141. [CrossRef]