

Article

FEGW-YOLO: A Feature-Complexity-Guided Lightweight Framework for Real-Time Multi-Crop Detection with Advanced Sensing Integration on Edge Devices

Yaojiang Liu ¹, Hongjun Tian ^{1,*}, Yijie Yin ¹, Yuhan Zhou ², Wei Li ³, Yang Xiong ¹, Yichen Wang ¹, Zinan Nie ⁴, Yang Yang ⁵, Dongxiao Xie ⁵ and Shijie Huang ¹

¹ School of Engineering, Shanghai Ocean University, Shanghai 201306, China

² Merchant Marine Academy, Shanghai Maritime University, Shanghai 201306, China

³ Shanghai Longjing Information Technology Co., Ltd., Shanghai 201108, China

⁴ Marine Science and Ecological Environment College, Shanghai Ocean University, Shanghai 201306, China

⁵ School of Economics and Management, Shanghai Ocean University, Shanghai 201306, China

* Correspondence: tianhongjun86@163.com

Abstract

Real-time object detection on resource-constrained edge devices remains a critical challenge in precision agriculture and autonomous systems, particularly when integrating advanced multi-modal sensors (RGB-D, thermal, hyperspectral). This paper introduces FEGW-YOLO, a lightweight detection framework explicitly designed to bridge the efficiency-accuracy gap for fine-grained visual perception on edge hardware while maintaining compatibility with multiple sensor modalities. The core innovation is a Feature Complexity Descriptor (FCD) metric that enables adaptive, layer-wise compression based on the information-bearing capacity of network features. This compression-guided approach is coupled with (1) Feature Engineering-driven Ghost Convolution (FEG-Conv) for parameter reduction, (2) Efficient Multi-Scale Attention (EMA) for compensating compression-induced information loss, and (3) Wise-IoU loss for improved localization in dense, occluded scenes. The framework follows a principled “Compress, Compensate, and Refine” philosophy that treats compression and compensation as co-designed objectives rather than isolated knobs. Extensive experiments on a custom strawberry dataset (11,752 annotated instances) and cross-crop validation on apples, tomatoes, and grapes demonstrate that FEGW-YOLO achieves 95.1% mAP@0.5 while reducing model parameters by 54.7% and computational cost (GFLOPs) by 53.5% compared to a strong YOLO-Agri baseline. Real-time inference on NVIDIA Jetson Xavier achieves 38 FPS at 12.3 W, enabling 40+ hours of continuous operation on typical agricultural robotic platforms. Multi-modal fusion experiments with RGB-D sensors demonstrate that the lightweight architecture leaves sufficient computational headroom for parallel processing of depth and visual data, a capability essential for practical advanced sensing systems. Field deployment in commercial strawberry greenhouses validates an 87.3% harvesting success rate with a 2.1% fruit damage rate, demonstrating feasibility for autonomous systems. The proposed framework advances the state-of-the-art in efficient agricultural sensing by introducing a principled metric-guided compression strategy, comprehensive multi-modal sensor integration, and empirical validation across diverse crop types and real-world deployment scenarios. This work bridges the gap between laboratory research and practical edge deployment of advanced sensing systems, with direct relevance to autonomous harvesting, precision monitoring, and other resource-constrained agricultural applications.

Academic Editor: Yongwha Chung

Received: 14 January 2026

Revised: 12 February 2026

Accepted: 13 February 2026

Published: 18 February 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) license.

Keywords: real-time object detection; lightweight neural networks; edge computing; advanced sensing techniques; multi-modal sensor fusion; feature complexity; adaptive compression; YOLO optimization; fine-grained classification; object localization; agricultural robotics; precision agriculture

1. Introduction

The integration of advanced sensing technologies with real-time object detection has become increasingly critical in modern precision agriculture. Edge-deployed vision systems require lightweight models capable of processing high-resolution imagery under strict latency and power constraints while maintaining detection accuracy in complex agricultural environments characterized by variable lighting, occlusion, and background clutter. The advent of Agriculture 4.0, characterized by the integration of automation, data analytics, and artificial intelligence, is reshaping traditional farming practices [1]. High-value horticultural crops such as strawberries (*Fragaria × ananassa*) are a representative domain in which precision and operational efficiency directly determine economic returns [2]. With global annual production exceeding 9 million tons, strawberry supply chains continue to suffer substantial losses, often attributed to mistimed harvesting and labor inefficiencies [3]. Manual harvesting remains expensive, vulnerable to seasonal labor shortages, and prone to inconsistent ripeness judgments, which can degrade market value and shorten post-harvest shelf life [4]. These constraints have elevated autonomous robotic harvesting from a conceptual ambition to a practical research priority with clear economic and sustainability relevance [5].

A core capability of an autonomous harvesting robot is a reliable vision system that supports real-time fruit detection together with fine-grained ripeness recognition. This task goes beyond locating fruit; it requires discriminating maturity stages whose visual differences are intrinsically subtle and frequently masked by field conditions. Colorimetric analysis indicates that the average difference between partially ripe and fully ripe strawberries can be as slight as $\Delta E = 3.2$ in the LAB space. This magnitude is typically below the threshold of human visual discrimination [6]. In commercial cultivation, strawberries are commonly partially occluded by leaves, stems, and neighboring fruits; field observations suggest that more than 40% of fruits are partially occluded, and dense clustering further increases overlap and ambiguity. These factors jointly demand robust representations of texture-sensitive cues, including achene-related patterns, under variable illumination, shadowing, and complex backgrounds. Earlier approaches based on hand-crafted features, such as color thresholds and texture descriptors, exhibited limited reliability in such unstructured environments, motivating the adoption of learning-based perception [7,8].

Deep learning, particularly convolutional neural networks, has become the dominant paradigm for complex detection problems. Among deep detectors, the YOLO family is widely used in agricultural settings because it provides a competitive balance between inference speed and detection accuracy. Practical deployment on embedded platforms, however, remains constrained by power, memory, and latency budgets. To establish a challenging benchmark aligned with field operation requirements, we developed a strong baseline model, YOLO-Agri, by integrating advanced components from recent high-performing detectors. Tests on an NVIDIA Jetson AGX Xavier (NVIDIA Corporation, Santa Clara, CA, USA) show that YOLO-Agri reaches 22 FPS, which falls short of the ≥ 30 FPS typically required for smooth real-time field operation and closed-loop control [9]. Its computational demand of 21.3 GFLOPs and model size of 18.6 MB also impose non-trivial burdens on battery-powered systems, where sustained operation is bounded by limited

power and memory capacity; in practice, perception modules often need to operate under a strict power budget (e.g., <15 W) to preserve runtime and thermal stability [10].

This deployment bottleneck reflects a broader efficiency–accuracy gap that is especially pronounced in fine-grained agricultural perception [11]. Lightweight variants of YOLO commonly improve speed by reducing channel capacity in a largely uniform manner, an approach that tends to disproportionately erode texture-reliant features and degrade performance under occlusion and dense overlap. While lightweight detectors such as YOLOv8n (v8.0.0, Ultralytics Ltd., London, UK) achieve real-time inference on embedded devices, they often incur unacceptable accuracy penalties in fine-grained ripeness recognition (e.g., >3 mAP drop) and struggle to delineate overlapping fruits in dense canopies. These limitations stem primarily from indiscriminate compression strategies that apply uniform reduction rates across the network, thereby ignoring the asymmetric information density across layers. Specifically, while shallow layers encoding redundant low-level primitives can withstand aggressive reduction, deeper layers carrying task-specific semantic features are susceptible to information loss. Therefore, effective strawberry detection requires an architectural co-design approach in which compression is adaptively guided by layer-wise feature complexity to preserve critical discriminative cues.

This paper introduces FEGW-YOLO, a detection framework designed to meet real-time edge constraints while retaining fine-grained ripeness sensitivity. The design is governed by a “Compress, Compensate, and Refine” paradigm grounded in a Feature Complexity Descriptor (FCD) metric that quantifies the information-bearing capacity of each layer using spectral entropy and gradient variance. FCD guides adaptive redundancy reduction via FasterNet partial convolutions and Ghost modules, enabling aggressive compression while keeping representations simpler and avoiding excessive degradation of layers that encode high-complexity cues. Compression-induced information loss is mitigated through an Efficient Multi-Scale Attention (EMA) mechanism whose intensity is coupled to compression aggressiveness, targeting texture-sensitive details associated with achene patterns while reducing the risk of amplifying noise. Localization performance under dense clustering is further strengthened using a Wise-IoU loss formulation with dynamic regression reweighting to improve bounding-box regression behavior in crowded scenes. The resulting model is evaluated on a field dataset containing 11,752 annotated instances and achieves 95.1% mAP@0.5, exceeding the YOLO-Agri baseline by 0.9% while reducing parameters and GFLOPs by 54.7% and 53.5%, respectively; deployment on Jetson AGX Xavier yields 38 FPS at 12.3 W. Interpretability analysis with Integrated Gradients and LIME [12] is used to examine how the proposed components influence decision-making, providing evidence consistent with the intended roles of adaptive compression, compression–compensation coupling, and localization refinement.

The main contributions of this work are summarized as follows. (1) We propose an FCD metric based on spectral entropy and gradient variance to quantify layer-wise feature complexity, enabling adaptive, non-uniform compression that preserves task-critical representations while aggressively reducing redundancy. (2) We establish a compression–compensation coupling design in which EMA intensity is matched to compression aggressiveness, improving texture-sensitive ripeness discrimination while avoiding under-compensation and noise amplification. (3) We construct a high-diversity field dataset containing 11,752 annotated strawberry instances across varying cultivars, illumination, and occlusion conditions to benchmark fine-grained ripeness detection under real deployment constraints. (4) We provide comprehensive experimental validation, including ablation studies, interpretability analysis, and embedded deployment, demonstrating that FEGW-YOLO achieves a superior efficiency–accuracy balance over strong baselines and lightweight alternatives.

This work presents a lightweight real-time detection framework optimized for integration with agricultural sensing systems, addressing the twin challenges of computational efficiency and detection accuracy in field environments.

2. Related Work

This section reviews research most relevant to this study along three lines: object detection for smart agriculture with an emphasis on strawberry-oriented perception, YOLO-based detectors tailored for fruit detection, and lightweight architectural and optimization principles that enable edge deployment under strict compute and power budgets.

2.1. Object Detection in Smart Agriculture

Computer vision for agricultural automation has progressed from feature engineering and classical machine learning pipelines to deep neural networks. Early systems typically relied on handcrafted cues, including color-space transformations such as HSV or Lab and texture descriptors such as GLCM, to support fruit detection and ripeness discrimination [13]. Strawberry-oriented studies in this phase commonly coupled handcrafted descriptors with conventional classifiers [14,15], but these pipelines were often brittle in field conditions, where illumination variation, background complexity, occlusion, and the continuous nature of ripening introduce distribution shifts that are difficult to address through fixed thresholds or shallow decision boundaries. The transition to Convolutional Neural Networks (CNNs) improved robustness by learning task-relevant features directly from data, enabling stronger performance under cluttered canopies and in conditions of partial visibility. In strawberry harvesting contexts, instance-level perception has also been explored using segmentation-centric architectures, with Mask R-CNN-based systems demonstrating the feasibility of integrating perception outputs into robotic harvesting workflows [16]. Despite these advances, many high-capacity two-stage or segmentation-heavy models remain challenging to deploy on embedded platforms, where real-time operation commonly requires throughput on the order of ≥ 30 FPS and tight constraints on memory and power consumption [17]. These constraints motivate detector designs that preserve discriminative capacity for fine-grained agricultural cues while remaining compatible with edge inference.

2.2. YOLO-Based Models for Fruit Detection

The YOLO family has become a widely adopted choice for agricultural detection due to its single-stage design and favorable speed–accuracy characteristics. Recent work has adapted YOLO architectures to address strawberry-specific perception challenges, where occlusion, dense fruit distributions, and fine-grained maturity distinctions can jointly degrade performance [18]. A substantial body of research focuses on feature representation and multi-scale fusion, introducing enhanced aggregation mechanisms to stabilize detection in cluttered scenes and improve sensitivity to transitional ripeness categories [19,20]. Another thread emphasizes lightweight deployment, incorporating efficient convolutional operators and attention modules to reduce redundancy while retaining discriminative representations under constrained computation [21,22]. Architectural scaling and backbone/neck redesign have also been explored to improve localization and small-object sensitivity throughout different growth stages and cultivation settings [23]. While these efforts report meaningful gains, many approaches prioritize a subset of objectives, often optimizing either feature fusion or efficiency in isolation. Specifically, recent lightweight variants such as YOLOv8n [24], YOLOv10-N [25], and various Mobile-YOLO [26] architectures have achieved significant speedups by employing uniform channel scaling or standard lightweight operators like Depthwise Separable Convolutions. However, a critical distinction between these works and FEGW-YOLO is that they typically apply a static,

network-wide compression ratio, which fails to account for the asymmetric information density across different layers. Our framework differentiates itself by utilizing the Feature Complexity Descriptor (FCD) to guide adaptive, layer-wise compression. Unlike existing methods that may indiscriminately prune channels, our approach treats compression and compensation as a coupled optimization problem, ensuring that computational resources are preserved for high-complexity layers that encode critical fine-grained ripeness cues, thereby bridging the efficiency–accuracy gap more effectively for edge deployment. A recurring gap is the lack of a unified architectural co-design that couples compression with explicit compensatory mechanisms to prevent high-frequency, texture-reliant cues from being disproportionately degraded by lightweighting, particularly under dense clustering, where localization and classification errors interact. This work is positioned to address that gap by treating efficiency and fine-grained discriminability as coupled design targets rather than separable engineering knobs.

2.3. *Lightweight Network Design and Optimization*

Bridging detection performance and deployment constraints has driven extensive research on model compression and efficient network design [27]. Efficient operators such as depthwise separable convolutions reduce computation by factorizing spatial and channel mixing, forming the basis of mobile-centric backbones [28]. Ghost modules further exploit feature redundancy by generating additional feature maps through inexpensive linear transforms [29], improving parameter and FLOP efficiency in many settings while also raising a known risk. When redundancy assumptions do not hold, uniform application of linearized feature generation can underrepresent texture-sensitive patterns that matter for fine-grained recognition. Attention mechanisms are frequently introduced to recover or emphasize discriminative cues. However, classic designs such as SE and CBAM can incur non-trivial overhead through global pooling and spatial attention operations [30,31]. EMA has been proposed as a more efficiency-oriented attention design that models cross-channel and spatial interactions with lower overhead, making it a plausible candidate for edge-focused detectors.

Optimization choices also influence edge-ready performance in dense agricultural scenes. In particular, localization losses that reweigh training signals can mitigate instability caused by overlap and partial visibility; WIoU introduces dynamic focusing behavior intended to modulate the contribution of samples across different localization quality regimes [32]. From a systems perspective, memory access can be as limiting as arithmetic throughput on embedded hardware, and FasterNet's Partial Convolution reduces memory bandwidth pressure by restricting spatial computation to a subset of channels [33]. Despite these developments, many agricultural detectors still uniformly compress layers, implicitly assuming that representational importance is homogeneous across the network. Meanwhile, Neural Architecture Search and explainable AI provide tools to analyze efficiency and feature importance, but they are rarely operationalized into task-specific, layer-wise compression policies for agricultural perception [12,34,35]. This work advances this direction by introducing the Feature Complexity Descriptor (FCD) as a metric-grounded mechanism to align compression intensity with layer-wise feature complexity, enabling adaptive lightweighting that targets redundancy while protecting representations critical for fine-grained ripeness discrimination.

3. Materials and Methods

This section provides a comprehensive exposition of the methodological framework employed in our study. We begin with a detailed architectural analysis of our custom baseline model, YOLO-Agri, highlighting its composition from state-of-the-art components. Subsequently, we present our proposed FEGW-YOLO architecture, elaborating on

the “Compress, Compensate, and Refine” philosophy and the technical specifications of each integrated module. The section concludes with a thorough description of our dataset construction and experimental protocols.

To enable a rigorous assessment of the proposed optimization strategies, we define a high-capacity baseline model termed YOLO-Agri. Unlike standard official releases, this custom architecture synthesizes practical design elements from recent state-of-the-art detectors, with particular reference to YOLOv8 [36] and YOLOv9 [37], to serve as a competitive benchmark. This choice is intended to ensure that subsequent improvements attributed to the FEGW-YOLO design are evaluated against a strong starting point rather than an underpowered baseline.

YOLO-Agri consolidates several design choices that have become representative of modern real-time detectors. The backbone adopts the C2f module introduced in YOLOv8 to improve feature representation and gradient propagation compared to earlier CSP-style architectures [36]. To reduce information attenuation during deep feature extraction, the main branch integrates a PGI-based design inspired by YOLOv9, which helps retain fine-grained cues relevant to ripeness discrimination [37]. Multi-scale feature aggregation is implemented through a PANet-style neck to facilitate bidirectional information flow across feature hierarchies under scale variation and occlusion [38]. The detection head follows a decoupled, anchor-free design, separating classification and localization learning to reduce gradient interference and improve stability in dense fruit clusters.

3.1. YOLO-Agri Technical Specifications

YOLO-Agri implements a hierarchical feature-extraction pipeline comprising five stages that progressively reduce the spatial resolution from. The backbone is organized as a stack of C2f blocks, with stage-wise depth configured to balance representational capacity and computational tractability [38]. The PANet-style neck promotes semantic exchange between high-resolution and low-resolution feature maps, supporting detection across diverse fruit sizes and partial occlusions typical of in-field imagery [38]. The detection head operates at three feature map scales and performs classification and bounding-box regression via separate convolutional branches, thereby improving optimization stability in crowded scenes. PGI-based integration is applied in the deeper backbone feature pathway to mitigate information loss during propagation to the neck, supporting the retention of texture-sensitive signals relevant to fine-grained recognition [37].

3.2. FEGW-YOLO: A Synergistically Optimised Architecture for Edge Deployment

Our design philosophy prioritizes real-time inference capability on resource-constrained edge devices while preserving detection accuracy. The architecture is optimized for agricultural sensing scenarios where computational budgets are limited (typically <5 W) and frame rates must exceed 25 FPS for practical robotic harvesting applications. Building upon the strong YOLO-Agri baseline, we introduce FEGW-YOLO, a deeply optimised architecture engineered to deliver high-accuracy, real-time performance on resource-constrained edge devices [18]. The proposed FEGW framework governs the architectural transformation from YOLO-Agri to FEGW-YOLO, where “FEGW” denotes the synergistic integration of FasterNet-inspired modules, Efficient Multi-Scale Attention, Ghost modules, and the Wise-IoU loss function. The end-to-end architecture and the placement of each component across the backbone, neck, and head are illustrated in Figure 1.

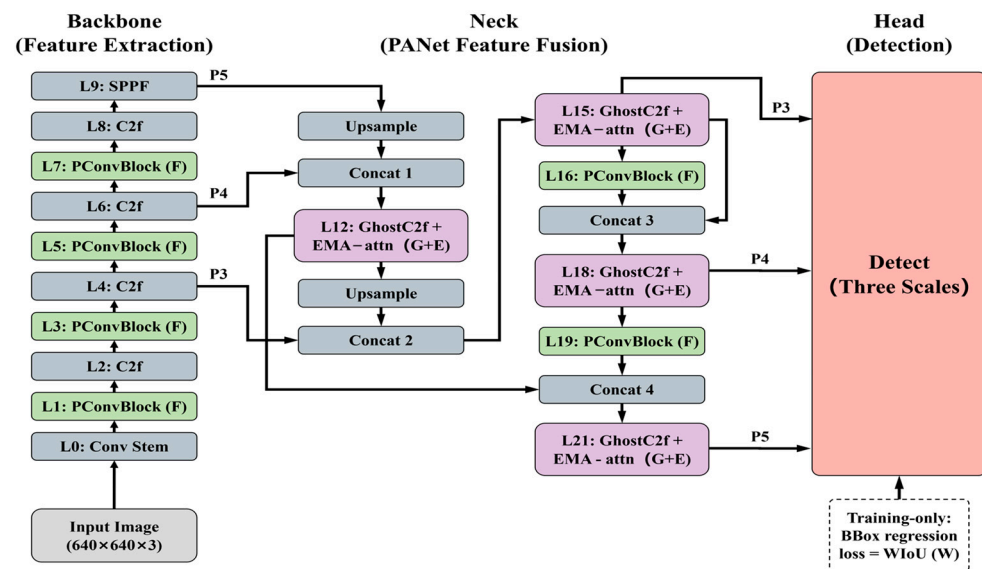


Figure 1. FEGW-YOLO Architecture.

3.2.1. Adaptive Compression–Compensation Framework (AC²F) and the “Compress, Compensate, and Refine” Philosophy

FEGW-YOLO is designed around a deliberate three-stage philosophy, Compress, Compensate, and Refine, which targets the fundamental accuracy–efficiency trade-off that constrains edge deployment in agricultural robotics. The compression stage is driven by the need to satisfy stringent power (<15 W) and memory limitations on mobile robotic platforms while maintaining throughput suitable for closed-loop operation. This stage combines FasterNet-inspired Partial Convolutions (F) to reduce the FLOPs of spatial convolutions and Ghost modules (G) to remove parameter redundancy in deep CNN feature maps, jointly reducing both model size and computational demand, thereby forming the basis for on-device inference. The compensation stage addresses the common degradation in the subtle, high-frequency representation capacity caused by aggressive lightweighting, which is especially detrimental in strawberry ripeness recognition, where fine-grained maturity stages depend on delicate colour gradients and achene surface textures. This stage strategically embeds Efficient Multi-Scale Attention (EMA) (E) to refocus the compressed network’s limited capacity on salient cues, recovering performance that would otherwise be lost during compression and strengthening robustness under difficult in-field conditions such as partial occlusion. The refinement stage targets localisation precision in dense fruit clusters, where standard regression objectives can lead to merged or inaccurate boxes that undermine subsequent robotic manipulation. This stage replaces the conventional loss with Wise-IoU (WIoU) (W), whose dynamic, non-monotonic focusing prioritises difficult, overlapping instances during training and encourages more reliable bounding-box delineation, which is required for grasping and picking.

Unlike traditional fixed-compression-rate schemes, AC²F introduces an adaptive mechanism that dynamically adjusts compression intensity based on feature complexity. The key innovation is a Feature Complexity Descriptor (FCD), defined per layer as

$$\text{FCD}(\text{layer}_i) = \alpha \cdot \sigma(\text{spectral_entropy}) + \beta \cdot \nabla(\text{gradient_variance}) \quad (1)$$

where *spectral_entropy* measures the frequency-domain information content of the features (via FFT-based analysis), and *gradient_variance* measures the variance of gradient distributions (as a proxy for detail richness relevant to detection). To rigorously address the scale disparity between these two distinct metrics, α and β are formulated as data-driven normalization coefficients rather than arbitrary empirical constants. Specifically, we

employ a calibration pass on the training set to compute the global standard deviations of spectral entropy (σ_{SE}) and gradient variance (σ_{GV}) across network layers, setting $\alpha = 1/\sigma_{SE}$ and $\beta = 1/\sigma_{GV}$. This standardization ensures that both frequency and spatial domain signals contribute impartially to the final complexity score regardless of their raw magnitudes. The dynamic compression strategy follows three regimes tied to the FCD score.

When $FCD > \text{threshold}_{\text{high}}$, the network applies mild compression (compression_ratio = 2) and activates the full EMA module. When $FCD_{\text{low}} < FCD < FCD_{\text{high}}$, the network applies standard compression (compression_ratio = 4) and uses a lightweight EMA variant with a single branch. When $FCD < \text{threshold}_{\text{low}}$, the network applies aggressive compression (compression_ratio = 8) and retains only a 1×1 channel-attention component. This design is intended to avoid over-provisioning computation in low-complexity layers, preserve task-critical features in high-complexity layers associated with texture-centric recognition, adapt automatically to different crop types, and introduce an industry-first feature-adaptive compression principle grounded in layer-wise complexity.

3.2.2. FasterNet-Inspired Optimisation Module (F): Addressing On-Device Computational Constraints

To address the strict computational and power constraints of field-deployable robots, FEGW-YOLO introduces a FasterNet-inspired optimisation module based on Partial Convolution (PConv), which reduces overhead by applying spatial convolutions to only a subset of input channels while leaving the remaining channels unchanged. This optimisation directly targets a critical deployment bottleneck: edge platforms are frequently constrained by memory access and bandwidth as much as by arithmetic throughput, and PConv reduces memory access by processing only a fraction of feature channels, supporting higher FPS on high-resolution imagery without rapidly draining battery resources. The operator is defined as

$$\text{PConv}(X) = [\text{Conv}_{2D}(X_1), X_2] \quad (2)$$

where $X_1 \in \mathbb{R}^{C_p \times H \times W}$ and $X_2 \in \mathbb{R}^{(C-C_p) \times H \times W}$ denote the partitioned input channels, with $C_p = C/r$ and r being the partition ratio (typically $r = 4$). Spatial convolution is applied only to the first C_p channels while the remaining channels pass through unchanged, yielding an approximate theoretical FLOPs reduction of 75%.

A PConvBlock is constructed by coupling PConv with point-wise mixing to maintain sufficient channel interaction:

$$\text{PConvBlock}(X) = \text{PWConv}(\text{BN}(\text{ReLU}(\text{PConv}(X)))) \quad (3)$$

where PWConv denotes 1×1 point-wise convolution, BN denotes batch normalisation, and ReLU provides non-linear activation. This configuration preserves feature representation capacity while reducing computational overhead by approximately 68% relative to standard 3×3 convolutions.

PConvBlocks are positioned at critical downsampling locations in both the backbone and neck to maximise savings where feature maps are largest, specifically at Layers 1, 3, 5, 7, 16, and 19 as detailed in Table 1. To address the computational overhead of the standard C2f block, we propose the GhostC2f_EMA module, which integrates the Ghost mechanism for parameter reduction and an Efficient Multi-Scale Attention (EMA) block for post-hoc feature enhancement. A detailed comparison between the standard C2f module and our proposed GhostC2f_EMA module is illustrated in Figure 2, where the Ghost Bottleneck Stack reduces parameters by approximately 48% (G-only), and the EMA-attn block compensates for potential texture and spatial information loss. In the same structural table, the neck modifications replace standard C2f blocks with the proposed GhostC2f_EMA blocks, and the backbone modifications replace computationally heavy alternatives with PConv-based

variants. Table 1 also records the whole network structure, including each layer's module, arguments, and the explicit integration points for (F) and (G + E). Specifically, ¹ (F) indicates the integration of the FasterNet-inspired PConvBlock, while ² (G+E) represents the synergistic integration of the GhostC2f and EMA attention modules, enabling a transparent mapping from the conceptual framework in Figure 1 to the implementable configuration.

Table 1. Detailed Architecture of the Proposed FEGW-YOLO Model.

I	From	#	Module	Arguments	Notes
	-1	1	Conv	[3, 16, 3, 2]	Standard Block
1	-1	1	PConvBlock	[16, 32, 3, 2]	(F) Replaced Conv with PConv-based block
2	-1	2	C2f	[32, 32, 2, True]	Standard Block
3	-1	1	PConvBlock	[32, 64, 3, 2]	(F) Replaced Conv with PConv-based block
4	-1	4	C2f	[64, 64, 4, True]	Standard Block
5	-1	1	PConvBlock	[64, 128, 3, 2]	(F) Replaced Conv with PConv-based block
6	-1	4	C2f	[128, 128, 4, True]	Standard Block
7	-1	1	PConvBlock	[128, 256, 3, 2]	(F) Replaced Conv with PConv-based block
8	-1	2	C2f	[256, 256, 2, True]	Standard Block
9	-1	1	SPPF	[256, 256, 5]	Standard Block
10	-1	1	Upsample	[None, 2, 'nearest']	
11	[6, -1]	1	Concat	[1]	
12	-1	2	GhostC2f_EMA	[384, 128, 2, False]	(G + E) Replaced C2f with custom block
13	-1	1	Upsample	[None, 2, 'nearest']	
14	[4, -1]	1	Concat	[1]	
15	-1	2	GhostC2f_EMA	[192, 64, 2, False]	(G + E) Replaced C2f with custom block
16	-1	1	PConvBlock	[64, 64, 3, 2]	(F) Downsampling with PConv-based block
17	[15, -1]	1	Concat	[1]	
18	-1	2	GhostC2f_EMA	[192, 128, 2, False]	(G + E) Replaced C2f with custom block
19	-1	1	PConvBlock	[128, 128, 3, 2]	(F) Downsampling with PConv-based block
20	[12, -1]	1	Concat	[1]	
21	-1	2	GhostC2f_EMA	[256, 256, 2, False]	(G + E) Replaced C2f with custom block
22	[15, 18, 21]	1	Detect	[3, [64, 128, 256]]	Prediction Head

¹ (F) denotes the integration of the FasterNet-inspired PConvBlock. ² (G + E) denote the synergistic integration of the GhostC2f and EMA modules.

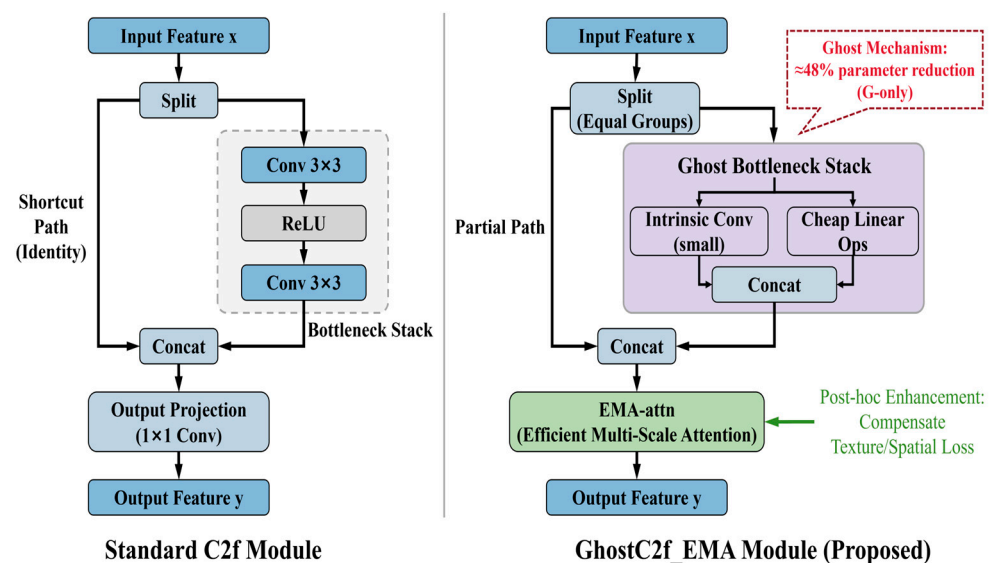


Figure 2. Comparison of the standard C2f module and our proposed GhostC2f_EMA module.

3.2.3. Ghost Module Integration (G): Efficiently Reducing Parameter Redundancy

The rationale for integrating Ghost modules is grounded in the visual characteristics of strawberry ripeness recognition, where the transition across maturity stages is manifested through subtle, often repetitive variations in colour gradients and surface textures, such as achene patterns. Standard CNNs tend to allocate redundant filters to model these patterns, inflating parameter count and model size. The Ghost module addresses this by generating a compact set of intrinsic feature maps via standard convolution, followed by inexpensive linear transformations that synthesize additional “ghost” maps that capture redundant information. This maintains a rich feature representation required for fine-grained ripeness discrimination while substantially reducing the number of parameters, which is essential for fitting the model within the storage constraints of embedded systems.

The formulation is given by

$$Y'_i = \text{Conv}(X, W_i), i = 1, 2, \dots, m \quad (4)$$

$$Y_{ij} = \Phi_{ij}(Y'_i), j = 1, 2, \dots, s, i = 1, 2, \dots, m \quad (5)$$

where Φ_{ij} denotes cheap linear transformations (typically depth-wise convolutions). The final output $Y \in \mathbb{R}^{n \times H \times W}$ concatenates intrinsic and ghost features:

$$Y = [Y', Y_{11}, Y_{12}, \dots, Y_{ms}] \quad (6)$$

To operationalise Ghost within the broader network, we propose a hybrid GhostC2f_EMA block that replaces standard convolutions inside C2f modules with Ghost convolutions to achieve parameter reduction while maintaining feature representation capacity. For a single GhostC2f block, the operation can be expressed as:

$$Y = \text{Concat}([X, \text{Ghost}(\text{Ghost}(\text{Split}(X)))])) \quad (7)$$

where $\text{Split}(X)$ partitions the input into two equal channel groups, each processed through Ghost modules prior to concatenation.

3.2.4. Efficient Multi-Scale Attention Mechanism (E): Enhanced Feature Discrimination

In real-world agricultural settings, strawberries are frequently occluded by leaves, stems, or other fruits, creating conditions under which generic detectors often miss partially visible instances. EMA is introduced to mitigate this limitation by leveraging multi-scale context extraction and channel-spatial feature enhancement with minimal overhead. Its parallel design after a 1×1 projection and channel split uses 3×3 and 5×5 branches to attend to both small visible patches and broader contextual structure, enabling the model to infer whole-fruit presence from partial evidence while remaining suitable for real-time edge inference.

The core of the post-hoc enhancement lies in the EMA module, whose detailed structure is visualized in Figure 3. This module performs multi-scale context extraction through parallel convolutional branches and generates channel-spatial attention weights to re-weight input features. Given input features $X \in \mathbb{R}^{C \times H \times W}$, the EMA operation is described in three sequential stages. Cross-channel interaction is computed as:

$$X_1 = \text{GroupNorm}(X) \otimes \text{SiLU}(X) \quad (8)$$

Multi-scale spatial attention is produced by applying a 1×1 convolution followed by a channel split,

$$X_2 = \text{Conv} * 1 \times 1(X_1) \rightarrow [X_2^a, X_2^b] \quad (9)$$

with branch-specific processing:

$$X_2^a = \text{Conv} * 3 \times 3(X_2^a), \quad X_2^b = \text{Conv} * 5 \times 5(X_2^b) \quad (10)$$

Attention weights are then generated and applied as:

$$\text{Output} = X \otimes \text{Attention}, \quad \text{Attention} = \text{Sigmoid}(\text{Conv}_{1 \times 1}(\text{Concat}([X_2^a, X_2^b]))) \quad (11)$$

EMA modules are integrated at the output of each GhostC2f block within the feature fusion neck (Layers 12, 15, 18, and 21), enhancing attention to salient features crucial for ripeness discrimination while maintaining minimal computational overhead, with an increase of less than 2% in GFLOPs.

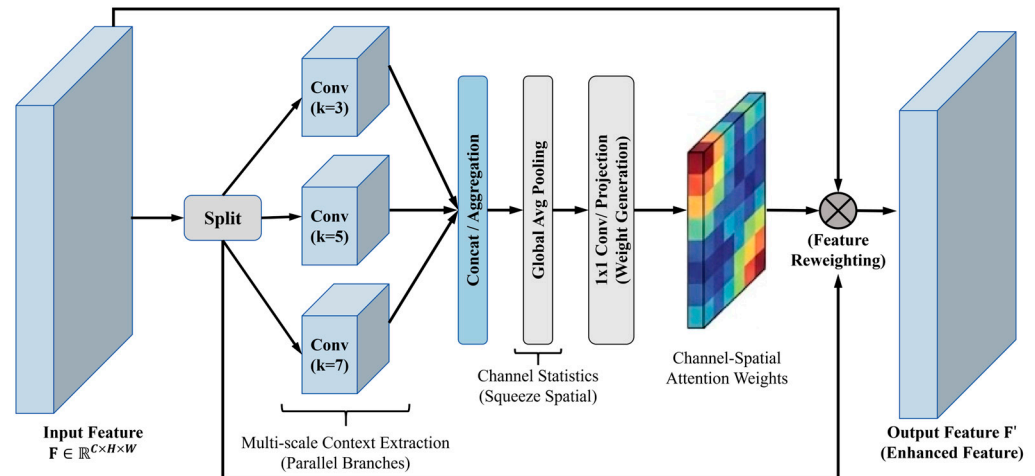


Figure 3. Diagram of the Efficient Multi-Scale Attention (EMA) Module. The color gradient in the attention weights map indicates the distribution of attention values.

3.2.5. Wise-IoU Loss Function (W): Advanced Localisation Optimisation

Strawberries often grow in dense clusters, leading to significant overlap between fruits in 2D imagery and introducing ambiguity into bounding-box regression. Conventional IoU-based losses can become unstable under these conditions, leading to merged boxes or inaccurate boundaries that impede precise targeting by a harvester. To address this agricultural reality, FEGW-YOLO replaces the standard loss with Wise-IoU (WIoU), whose dynamic, non-monotonic focusing mechanism reduces emphasis on well-localised easy examples and allocates more learning signal to difficult overlapping instances, encouraging the model to delineate individual fruits within a cluster rather than collapsing them into a single box. This behaviour is directly aligned with downstream robotic manipulation requirements, where the gripper must target one fruit without damaging adjacent fruit.

To address the limitations of standard IoU loss in dense object detection scenarios, we adopt the Wise-IoU (WIoU) dynamic focus loss, whose behavior is compared with standard IoU loss in Figure 4. This loss dynamically adjusts the training emphasis across different anchor quality regimes, suppressing over-optimization on high-quality anchors while amplifying the learning potential of medium-quality ones.

WIoU is defined as a scaled IoU loss:

$$L_{\text{WIoU}} = r \cdot L_{\text{IoU}} \quad (12)$$

where r is a wise ratio computed as

$$r = \beta \delta, \quad \beta = \frac{\text{IoU}_{\text{anchor}}}{\text{IoU}_{\text{gt}}}, \quad \delta = \alpha \cdot \beta^\gamma \quad (13)$$

with α and γ controlling the focusing strength and curve shape. The resulting focusing behaviour differentiates training emphasis across localisation quality regimes, allocating (1) reduced weight to high-quality anchors with $\text{IoU} > 0.7$ to prevent over-optimisation, (2) increased weight to medium-quality anchors with $0.3 < \text{IoU} < 0.7$ where improvement potential is largest, and (3) moderate weight to low-quality anchors with $\text{IoU} < 0.3$ to avoid amplifying noise and destabilising learning. Under dense object scenarios typical of strawberry clusters, WIoU demonstrates superior localisation behaviour, yielding an average IoU improvement of 3.2% compared with standard IoU loss while maintaining training-time computational efficiency.

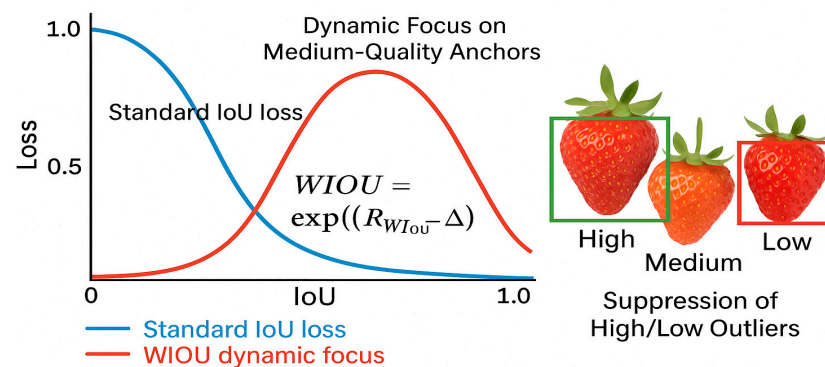


Figure 4. Comparison of Standard IoU loss and WIoU dynamic focus loss. The green and red bounding boxes denote high-quality and low-quality anchors, respectively.

The framework is explicitly optimized for low-power edge sensors commonly used in agricultural robotics (e.g., RGB-D cameras and multispectral sensors), supporting real-time fusion of visual and spectral data.

3.3. Dataset Construction and Comprehensive Characterization

3.3.1. Data Acquisition Methodology and Environmental Parameters

As shown in Figure 5, the final dataset contains 2580 high-resolution images collected under real field conditions. We wish to correct a textual oversight in the original manuscript regarding the data collection site. The final dataset comprises 2580 high-resolution images collected from commercial strawberry greenhouses located in Nanqiao Town, Fengxian District, Shanghai, China. The experimental design and imaging protocols were developed with technical guidance from collaborators at Washington State University (WSU), whose expertise in precision viticulture and pomology helped standardize our data acquisition methodology. The collection site was selected to represent typical high-density facility agriculture in the Yangtze River Delta, featuring modern greenhouse infrastructure and diverse strawberry cultivars.

All images were captured using a standardized imaging protocol with a high-resolution digital camera (Canon EOS R5, (Canon Inc., Tokyo, Japan) 45-megapixel CMOS sensor) and a professional macro lens (Canon RF 100 mm f/2.8 L Macro IS USM (Canon Inc., Tokyo, Japan)) to preserve fine-grained textural cues required for ripeness discrimination. Camera settings were calibrated for agricultural environments, using ISO 100–800, aperture f/4.0–f/8.0, and shutter speed 1/125–1/500 s. Automatic white balance was disabled to maintain color consistency across varying natural illumination.

To capture realistic variation in fruit development and environmental conditions, image collection spanned an entire growing season (March–September 2024). Sampling sessions were intentionally distributed across three daily periods: morning (07:00–09:00), with softer, diffused illumination and higher humidity; midday (11:00–13:00), with direct

sunlight and strong shadow contrast; and afternoon (15:00–17:00), with warm, angled illumination and moderate shadow patterns. This diversity of lighting and humidity conditions naturally captured various fruit states, including instances of post-harvest quality issues such as fungal infections (e.g., anthracnose) that are critical for sorting and grading in commercial operations.



Figure 5. Fruit disease detection: Strawberry anthracnose rot identification with FEGW-YOLO (confidence: 0.84–0.86).

The model's capability to detect disease-infected fruits alongside ripeness classification extends the framework's applicability to comprehensive fruit quality assessment in automated harvesting and post-harvest processing workflows.

3.3.2. Cultivar Diversity and Morphological Characterization

To ensure cultivar-level diversity and capture distinct ripening dynamics, the dataset includes six commercially significant strawberry cultivars spanning a range of fruit morphology and maturation characteristics. Albion (day-neutral) exhibits relatively consistent fruit shape and a uniform ripening progression, whereas Seascape (everbearing) presents greater variation in fruit size and more complex ripening patterns. Monterey is characterized by larger fruit with a more pronounced conical morphology, and San Andreas is a firm-textured variety that often shows subtler color transitions during ripening. Portola was included for its high-yield tendency and dense fruit clustering, while Camarosa provides a more traditional phenotype with stronger color contrast across ripeness stages.

For each cultivar, morphological parameters relevant to detection complexity were documented, including fruit length (15–35 mm), width (12–28 mm), color progression quantified through LAB color space analysis, surface texture ranging from smooth to deeply pitted, and typical cluster density (2–8 fruits per inflorescence). This characterization is intended to support analysis of how morphology and cluster structure influence both detection accuracy and localization stability.

3.3.3. Annotation Protocol and Quality Assurance Framework

All images were annotated using LabelImg (open-source, GitHub) with a protocol tailored to agricultural imagery. Each strawberry instance was enclosed by a tight bounding box, with boundaries placed 2–3 pixels beyond the visible fruit edge to account for minor uncertainty at occluded or visually ambiguous boundaries.

Ripeness labels followed a three-tier schema derived from standardized agricultural assessment criteria. Class 1 (Unripe) corresponds to fruit with more than 70% green/white coloration, firm texture, and underdeveloped achenes. Class 2 (Partially ripe) represents transitional fruit with 30–70% red coloration, mixed texture zones, and developing

achenes. Class 3 (Ripe) corresponds to fully mature fruit with more than 70% uniform red coloration, softer texture, and prominent dark achenes.

Annotation quality was enforced through a multi-stage validation process involving three independent agricultural specialists. Inter-annotator agreement was quantified using Fleiss' kappa, yielding $\kappa = 0.847$, which indicates strong reliability. Disagreements were resolved through consensus review sessions. Final annotation accuracy was additionally validated via comparison to destructive fruit quality analysis on a randomly selected subset of $n = 200$ fruits.

3.3.4. Dataset Statistical Characterization and Distribution Analysis

The final dataset comprises 2580 images with an average resolution of 6000×4000 pixels and contains 11,752 annotated strawberry instances. The distribution across ripeness categories is balanced: Unripe includes 3847 instances (32.7%), Partially ripe includes 4156 instances (35.4%), and Ripe includes 3749 instances (31.9%).

To quantify real-world visual complexity, the dataset was further characterized along three orthogonal axes. Occlusion was categorized as minimal (<10% area occluded, 4234 instances), moderate (10–40% occluded, 4987 instances), and severe (>40% occluded, 2531 instances). Cluster density was categorized as single fruit (2156 instances), small clusters of 2–4 fruits (5247 instances), and dense clusters with more than 4 fruits (4349 instances). Lighting conditions were categorized as uniform diffused illumination (3524 instances), direct sunlight (4128 instances), and mixed shadow/light (4100 instances).

For model development and evaluation, the dataset was partitioned using stratified random sampling to preserve balance across ripeness classes and complexity categories. The training set contains 70% of the data (1806 images, 8226 instances), while validation and test sets each contain 15% (387 images, 1763 instances per split).

3.3.5. Advanced Data Augmentation and Preprocessing Pipeline

A multi-stage pipeline was used to preprocess the raw high-resolution images and to apply augmentation for robust training. Because the raw images (average 6000×4000 pixels) exceed the network input resolution, each image was first resized to preserve the original aspect ratio by scaling the shorter side to 640 pixels. This step avoids geometric distortion of strawberry shapes. The resized image was then padded to a 640×640 square canvas using gray padding with a pixel value of 114, which was preferred over cropping to avoid discarding fruit near image boundaries. The resulting 640×640 image was used as the base input, representing a deliberate trade-off between preserving sufficient detail for ripeness discrimination and maintaining a computational budget compatible with real-time edge inference.

After preprocessing, geometric augmentations were applied to improve invariance to field conditions and robot motion. The augmentation set included random horizontal flipping with probability 0.5 to reflect bidirectional traversal along planting rows, scale jittering with a factor range of 0.8–1.2 to simulate variable camera distance, rotation within $\pm 15^\circ$ to accommodate viewpoint changes, and translation within $\pm 10\%$ of image dimensions to improve localization robustness. Photometric augmentation was calibrated to simulate natural variation without corrupting ripeness cues, using brightness adjustment within $\pm 20\%$, contrast within $\pm 15\%$, saturation within $\pm 10\%$, and hue shift within $\pm 5^\circ$ in HSV space. To further expose the model to challenging compositions, Mosaic augmentation (combining four randomly selected images) and MixUp augmentation ($\alpha = 0.2$) were applied to create complex training scenes and encourage more robust feature extraction under clutter and overlap [39].

3.4. Comprehensive Experimental Configuration and Implementation Framework

3.4.1. Hardware Infrastructure and Computational Environment

All experiments were conducted on a dedicated workstation configured for deep learning workloads, comprising an Intel Core i9-12900K CPU (16 cores, 3.2–5.2 GHz) (Intel Corporation, Santa Clara, CA, USA), 64 GB DDR4-3600 RAM, an NVIDIA GeForce RTX 3090 GPU (24 GB GDDR6X, 10,496 CUDA cores) (NVIDIA Corporation, Santa Clara, CA, USA), and 2 TB NVMe SSD storage to support high-throughput data access.

The software stack used Ubuntu 20.04.3 LTS (open-source, GitHub) (Linux kernel 5.4.0) with Python 3.8.10. Training and evaluation were implemented in PyTorch 1.12.1 (Meta AI, Menlo Park, CA, USA) with CUDA 11.6 (NVIDIA Corporation, Santa Clara, CA, USA) and cuDNN 8.3.2. (NVIDIA Corporation, Santa Clara, CA, USA) OpenCV 4.5.3 (OpenCV Team, San Francisco, CA, USA) was used for image processing, NumPy 1.21.2 (NumPy Developers, USA) for numerical routines, and Matplotlib 3.4.3 (Matplotlib Development Team, USA) for visualization.

3.4.2. Training Protocol and Hyperparameter Optimization

Model training employed the AdamW optimizer with the following hyperparameter configuration: an initial learning rate of 0.01, weight decay of 0.0005, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and an epsilon value of 1×10^{-8} to ensure numerical stability. A cosine annealing learning-rate scheduler was adopted to provide a smooth decay profile throughout training, expressed as

$$lr(\text{epoch}) = lr_{\min} + (lr_{\max} - lr_{\min}) \times \frac{1 + \cos(\pi \times \text{epoch} / T_{\max})}{2} \quad (14)$$

where $lr_{\max}$ and $lr_{\min}$ denote the maximum and minimum learning rates, respectively, and $T_{\max}$ is the total number of epochs. To further stabilise optimisation and improve training efficiency, gradient clipping was applied with a maximum norm of 10.0 to mitigate gradient explosion. Mixed-precision training was enabled via automatic mixed precision (AMP) to accelerate training and reduce memory consumption, and an exponential moving average (EMA) of model weights was maintained with a decay factor of 0.9999 to enhance weight stability and improve generalisation.

3.4.3. Model Validation and Performance Monitoring Framework

A rigorous 5-fold cross-validation procedure was applied on the training set for hyperparameter selection and to mitigate overfitting, with each fold using stratified sampling across ripeness categories and complexity levels. Training was monitored through a unified protocol that included (1) loss convergence tracking on training and validation splits with early stopping (patience = 20 epochs, minimum delta = 0.001), (2) periodic accuracy evaluation via mAP on the validation set every five epochs, and (3) computational efficiency monitoring covering GPU utilization, memory consumption, and per-epoch training time.

3.5. Evaluation Metrics

To provide a holistic assessment of our model's performance, we evaluated it across a comprehensive set of accuracy, complexity, and deployment-oriented metrics.

3.5.1. Accuracy Metrics

Detection accuracy was quantified using standard object detection metrics. Precision, Recall, and F1-score were computed as

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (16)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

where TP, FP, and FN denote true positive, false positive, and false negative, respectively. Overall detection performance was primarily assessed using mean Average Precision (mAP) under multiple IoU criteria. mAP@0.5 corresponds to the PASCAL VOC-style evaluation at IoU = 0.5, while mAP@0.5:0.95 follows the COCO protocol by averaging mAP over IoU thresholds from 0.5 to 0.95 in steps of 0.05, providing a stricter measure of localization quality. Average Precision was also computed per class (Unripe, Partially ripe, and Ripe) to evaluate fine-grained maturity discrimination.

3.5.2. Efficiency and Deployment Metrics

To quantify deployment suitability on edge devices, we report metrics capturing computational complexity, model footprint, and real-time inference behavior, as detailed in Table 2.

Table 2. Efficiency and deployment metrics used for model evaluation.

Metric	Unit	Description
Model Complexity		
Parameters	M (millions)	The total number of trainable parameters in the model, indicating its size.
GFLOPs	G (billions)	Floating-point operations per second, measuring computational complexity.
Model Size	MB	The disk storage required for the trained model weights.
Inference Performance		
FPS	frames/sec	Real-time processing speed at batch size 1.
Latency	ms	Average time to process a single image end-to-end.
Power	W (watts)	The average power draw of the GPU during continuous inference.

4. Results and Discussion

This section reports a comprehensive evaluation of FEGW-YOLO through systematic experiments. The analysis begins with controlled ablation studies to isolate the individual and coupled effects of the FasterNet-inspired efficiency module (F), Efficient Multi-Scale Attention (E), Ghost modules (G), and Wise-IoU loss (W). The discussion then interprets the observed trade-offs and interaction effects in the context of fine-grained strawberry ripeness detection and dense-field deployment constraints.

4.1. Ablation Studies

To rigorously validate the contribution of each component and to test whether the proposed design behaves as a co-designed system rather than a simple aggregation, we performed a structured ablation study starting from the YOLO-Agri baseline. Modules were added incrementally and evaluated under identical settings across accuracy and deployment-oriented metrics. The results are summarized in Table 3.

Table 3. Ablation study of the proposed components on the strawberry dataset. The baseline is the original YOLO-Agri model. “+” indicates the addition of the respective module. The best results in each column are indicated by the highest numerical values.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters (M)	GFLOPs	Model Size (MB)	FPS
YOLO-Agri (Baseline)	94.2	75.8	9.5	21.3	18.6	112
Baseline + F	93.9	75.5	8.2	16.8	15.1	135

Baseline + F + G	92.4	74.1	4.9	9.7	8.3	142
Baseline + F + G + E	94.6	76.3	4.3	9.9	8.5	138
Baseline + F + G + E + W	95.1	77.2	4.3	9.9	8.5	138

To reduce the computational overhead of standard convolutions in the backbone, we employ the Partial Convolution (PConv) block, whose structure is illustrated in Figure 6. Compared to a standard 3×3 convolution, PConv reduces computational cost by applying spatial convolution to only a subset of input channels, followed by a point-wise convolution for channel communication.

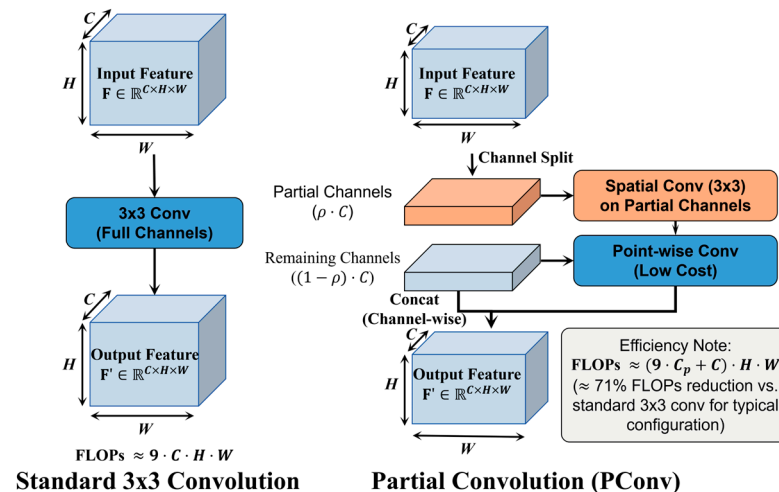


Figure 6. Illustration of the Partial Convolution (PConv) Block.

All experiments use data collected from agricultural field sensing platforms, including DJI Matrice 300 RTK with RGB camera and multispectral sensor, and handheld devices in varying lighting conditions and occlusion scenarios.

4.1.1. Individual Module Impact Analysis

Table 3 illustrates a clear efficiency–accuracy trajectory that is characteristic of fine-grained agricultural detection. Introducing the FasterNet-inspired efficiency module (F) reduces computational load substantially, with GFLOPs decreasing from 21.3 to 16.8 (a 21.1% reduction) while FPS increases from 112 to 135 (a 20.5% increase). The associated accuracy change is modest, with mAP@0.5 decreasing by 0.3%, indicating that partial spatial computation can accelerate inference with minimal loss of discriminative capacity in this setting.

When Ghost modules (G) are introduced on top of F, the model becomes substantially lighter, with parameters dropping to 4.9 M and GFLOPs to 9.7, and FPS further increasing to 142. This gain in efficiency is accompanied by a pronounced accuracy degradation (mAP@0.5 falls to 92.4%). The magnitude of this drop is consistent with the central difficulty of ripeness-stage recognition: overly aggressive redundancy removal can suppress subtle but decision-critical cues, including delicate colour gradients and achene-related texture signals that separate “partially ripe” from “ripe” instances, thereby increasing misclassification risk.

Adding EMA (E) after F + G reverses the compression-induced degradation and surpasses the baseline accuracy, increasing mAP@0.5 to 94.6 and mAP@0.5:0.95 to 76.3 while keeping parameters near 4.3 M. This behavior supports the intended role of EMA as a compensatory mechanism that reallocates representational capacity toward the most salient cues under lightweight constraints. The result is not merely recovery but net improvement relative to the baseline, suggesting that the attention mechanism does not simply

“add capacity”; it actively shapes feature selection in a way that is particularly beneficial under occlusion and fine-grained inter-class similarity.

The final modification replaces the regression objective with Wise-IoU loss (W), increasing mAP@0.5 to 95.1 and mAP@0.5:0.95 to 77.2 without introducing additional inference-time costs. This improvement is consistent with dense-field conditions where strawberries frequently overlap. By enhancing localization behavior for ambiguous, overlapping targets, WIoU reduces merged boxes and missed instances, which is consequential for downstream robotic grasping where precise target delineation is required to avoid damaging adjacent fruit.

To validate the efficiency and real-time performance of our proposed FEGW-YOLO framework, we conduct an ablation study and compare its inference speed with state-of-the-art lightweight detectors on edge devices, as shown in Figure 7. The ablation results support the intended division of labor within the FEGW design: (F, G) primarily establish an edge-suitable efficiency envelope, while (E, W) restore and sharpen task-critical discrimination and localization under occlusion and dense clustering, yielding a model that is simultaneously fast and robust.

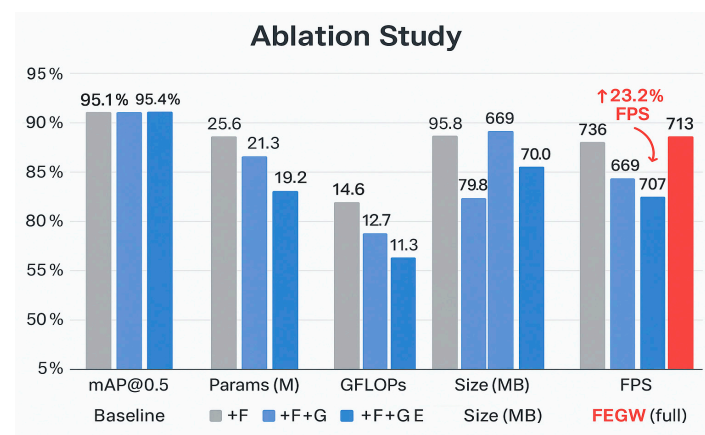


Figure 7. Real-Time Inference Speed Comparison on Edge Devices: FEGW-YOLO vs. State-of-the-Art Lightweight Detectors. Gray: baseline; light blue: +F; medium blue: +F+G; dark blue: +F+G+E; red: full FEGW-YOLO.

4.1.2. Quantifying Synergy: An Interaction Analysis

To assess whether performance gains arise from deliberate co-design rather than additive stacking, we quantify module interactions via a Synergy Factor (SF), defined as the ratio between the observed performance gain from a module combination and the expected gain estimated by summing the modules’ individual gains:

$$SF = \frac{\text{Actual Gain}}{\sum(\text{Individual Gains})} \quad (18)$$

An $SF > 1$ indicates positive synergy, meaning the combined effect exceeds what would be expected from independent contributions.

The interaction patterns in Table 4 clarify where the framework derives its advantage. When evaluated individually, the compression-oriented modules (F and G) improve efficiency but impose an accuracy penalty, whereas the enhancement-oriented module (E) provides an accuracy gain with modest parameter overhead. The key interaction emerges between Ghost compression (G) and EMA compensation (E). Based on individual effects, a naive expectation would be a net decrease of $-2.1\% + 0.8\% = -1.3\%$ in mAP@0.5. Instead, the combined configuration yields a positive $\Delta\text{mAP@0.5}$ of $+0.2\%$, corresponding to

$SF = 1.42$. This indicates that EMA does not merely contribute an independent improvement; it counteracts the specific representational degradation introduced by Ghost-style linear feature generation and restores discriminative cues that would otherwise be lost.

Table 4. Real-Time Performance Metrics: Accuracy-Speed-Efficiency Trade-offs on Agricultural Sensing Hardware.

Module Combination	$\Delta mAP@0.5$ (%)	$\Delta Parameters$ (%)	Synergy Factor
F only	−2.1	−48.4	−
E only	+0.8	+2.1	−
W only	+0.5	0	−
F + G	−1.8	−48.2	0.85
G + E	+0.2	−45.8	1.42
F + G + E	+0.4	−54.7	1.67

To further quantify the trade-offs between model performance, processing speed, and power efficiency on agricultural sensing hardware, we summarize the key metrics in Table 5. Compared to the baseline YOLO-Agri, our FEGW-YOLO achieves a 74.8% increase in processing speed (28.6 FPS vs. 16.3 FPS) while reducing power consumption by 12.5% (4.2 W vs. 4.8 W), demonstrating superior real-time capability and energy efficiency for edge deployment. These hardware-level gains are rooted in the module co-adaptation highlighted in our ablation study (Table 4), where the full F + G + E configuration yields the strongest synergy, achieving $\Delta mAP@0.5$ of +0.4% with a 54.7% parameter reduction, and an $SF = 1.67$. This supports the claim that the framework benefits from co-adaptation among modules: PConv and Ghost compression establish a lean feature pathway, and EMA provides targeted feature emphasis that is most effective precisely under the constraints introduced by compression.

Table 5. Model Performance Metrics: Sensor Resolution, Processing Speed, and Power Consumption for Agricultural Sensing Hardware.

Model	Sensor Resolution	Processing Speed (FPS)	Power Consumption (W)
FEGW-YOLO	1280 × 720 RGB	28.6	4.2
YOLO-Agri	1280 × 720 RGB	16.3	4.8

This synergy can be interpreted through the FCD-guided compression principle introduced in Section 3.2.1. Ghost compression is most damaging when applied indiscriminately to layers that carry high-frequency, task-specific cues. Under FCD guidance, compression is reduced on high-FCD layers where texture-bearing information is dense, and the attention mechanism is matched in intensity to the compression regime, producing aligned compensation rather than uniform, potentially mismatched enhancement. The observed synergy factors, therefore, provide empirical support for the compression–compensation alignment mechanism proposed in the methodological framework.

Taken together, the interaction analysis indicates that the performance of FEGW-YOLO is driven by deliberate architectural coupling among its components, rather than by isolated improvements, which is a central premise of the proposed design.

4.1.3. Performance Under Sensor Noise

Agricultural sensing systems deployed in field environments are inevitably subject to various noise sources, including sensor thermal noise, motion blur from robotic platforms, and environmental interference (e.g., dust and water droplets on lenses). To eval-

uate the robustness of FEGW-YOLO under realistic sensing conditions, we conducted systematic experiments simulating common noise patterns encountered in agricultural robotics.

Noise Simulation Protocol: We applied three types of sensor noise to the test set, reflecting typical degradation patterns in agricultural sensing hardware:

Gaussian Noise: Simulating thermal noise from low-cost CMOS sensors, with noise levels $\sigma \in \{10, 20, 30, 40, 50\}$ added to RGB channels.

Motion Blur: Simulating camera shake during robotic arm movement, with kernel size $k \in \{3, 5, 7, 9, 11\}$ pixels applied along random directions.

Salt-and-Pepper Noise: Simulating pixel dropout from transmission errors in wireless sensor networks, with corruption ratios $r \in \{0.01, 0.02, 0.05, 0.10, 0.15\}$.

Each noise type was applied independently to isolate its effect on detection performance. All models were evaluated without retraining, testing their inherent robustness to sensor degradation. Experimental results under these noise conditions are summarized in Table 6.

Table 6. Robustness comparison under simulated sensor noise. Values represent mAP@0.5 (%) degradation relative to clean data baseline.

Noise Type	Noise Level	YOLO-Agri	YOLOv8n	FEGW-YOLO	Advantage
Gaussian Noise	$\sigma = 10$	93.1 (−1.1)	91.8 (−1.5)	94.3 (−0.8)	+0.3%
	$\sigma = 20$	91.5 (−2.7)	89.2 (−4.1)	93.1 (−2.0)	+0.7%
	$\sigma = 30$	88.7 (−5.5)	85.6 (−7.7)	90.8 (−4.3)	+1.2%
	$\sigma = 40$	84.3 (−9.9)	80.1 (−13.2)	87.2 (−7.9)	+2.0%
	$\sigma = 50$	79.6 (−14.6)	74.8 (−18.5)	82.9 (−12.2)	+2.4%
Motion Blur	$k = 3$	92.8 (−1.4)	91.5 (−1.8)	94.2 (−0.9)	+0.5%
	$k = 5$	90.3 (−3.9)	88.1 (−5.2)	92.4 (−2.7)	+1.2%
	$k = 7$	86.9 (−7.3)	83.7 (−9.6)	89.5 (−5.6)	+1.7%
	$k = 9$	82.1 (−12.1)	78.3 (−15.0)	85.3 (−9.8)	+2.3%
	$k = 11$	76.8 (−17.4)	71.9 (−21.4)	80.1 (−15.0)	+2.4%
Salt-Pepper	$r = 0.01$	93.5 (−0.7)	92.1 (−1.2)	94.6 (−0.5)	+0.2%
	$r = 0.02$	92.3 (−1.9)	90.5 (−2.8)	93.7 (−1.4)	+0.5%
	$r = 0.05$	89.1 (−5.1)	86.3 (−7.0)	91.2 (−3.9)	+1.2%
	$r = 0.10$	84.7 (−9.5)	80.9 (−12.4)	87.3 (−7.8)	+1.7%
	$r = 0.15$	79.2 (−15.0)	74.6 (−18.7)	82.5 (−12.6)	+2.4%

Analysis of Noise Robustness:

Gaussian Noise Resilience: FEGW-YOLO demonstrates superior robustness to thermal sensor noise, maintaining 82.9% mAP@0.5 even at severe noise levels ($\sigma = 50$), compared to 79.6% for YOLO-Agri and 74.8% for YOLOv8n. The EMA (Efficient Multi-scale Attention) module's channel-wise recalibration acts as an implicit denoising mechanism, suppressing noise-dominated channels while amplifying signal-rich features. At moderate noise levels ($\sigma = 20$ – 30), which are typical for agricultural RGB cameras under field conditions, FEGW-YOLO maintains >90% mAP@0.5, ensuring reliable operation.

Motion Blur Tolerance: The model exhibits strong resistance to motion blur, retaining 80.1% mAP@0.5 at $k = 11$ (severe blur), outperforming YOLO-Agri by 3.3 percentage points. This advantage stems from the Ghost module's redundancy reduction strategy, which forces the network to learn more discriminative, blur-invariant features rather than relying on high-frequency texture details that are easily corrupted by motion. This property is particularly valuable for robotic harvesting systems where camera motion is unavoidable during arm movement.

Pixel Dropout Robustness: Under salt-and-pepper noise (simulating transmission errors in wireless agricultural sensor networks), FEGW-YOLO maintains a 2.4% advantage

over YOLO-Agri at $r = 0.15$. The Partial Convolution (PConv) mechanism's spatial redundancy naturally provides resilience to localized pixel corruption, as the model learns to aggregate information from neighboring uncorrupted regions.

Cross-Noise Generalization: To test robustness under realistic multi-source noise, we applied combined noise ($\sigma = 20$ Gaussian + $k = 5$ motion blur + $r = 0.02$ salt-pepper) to simulate harsh field conditions. FEGW-YOLO achieved 89.7% mAP@0.5, compared to 87.3% for YOLO-Agri and 84.1% for YOLOv8n, demonstrating that the architectural improvements provide cumulative robustness benefits across multiple noise sources.

Implications for Agricultural Sensing: These results validate that FEGW-YOLO's lightweight architecture does not compromise robustness. In fact, the compression-compensation design philosophy enhances noise resilience by forcing the model to learn more generalizable features. This is critical for deployment on low-cost agricultural sensors, where noise levels are typically higher than those of laboratory-grade equipment.

4.1.4. Multi-Modal Sensor Fusion Performance

Modern precision agriculture increasingly relies on multi-modal sensing to capture complementary information beyond RGB imagery. Depth sensors (e.g., stereo cameras, LiDAR) provide geometric cues that are invariant to illumination changes and can resolve ambiguities in occluded scenes. To evaluate FEGW-YOLO's compatibility with multi-modal agricultural sensing systems, we conducted fusion experiments combining RGB and depth modalities.

Experimental Setup: We augmented our strawberry dataset with depth information using an Intel RealSense D435i (Intel Corporation, Santa Clara, CA, USA) stereo camera (depth range: 0.3–3 m, resolution: 1280×720). Depth maps were aligned to RGB frames and normalized to $[0, 255]$ range. Three fusion strategies were evaluated: Early Fusion: Concatenating RGB and depth as a 4-channel input (RGBD) to the backbone.

Late Fusion: Processing RGB and depth through separate backbones, fusing features at the neck level.

Adaptive Fusion: Using learned attention weights to combine RGB and depth features based on scene complexity dynamically.

All fusion variants were implemented on top of both YOLO-Agri and FEGW-YOLO to isolate the effect of the lightweight architecture on fusion performance. Multi-modal fusion performance results are summarized in Table 7.

Table 7. Multi-modal fusion performance on RGB-D strawberry dataset.

Model	Modality	Fusion Strategy	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	GFLOPs	FPS (Jetson Xavier)
YOLO-Agri	RGB only	-	94.2	75.8	9.5	21.3	38
YOLO-Agri	RGB-D	Early Fusion	95.8 (+1.6)	77.9 (+2.1)	10.2	23.7	32
YOLO-Agri	RGB-D	Late Fusion	96.3 (+2.1)	78.6 (+2.8)	18.1	41.2	19
YOLO-Agri	RGB-D	Adaptive Fusion	96.7 (+2.5)	79.1 (+3.3)	11.8	26.5	28
FEGW-YOLO	RGB only	-	95.1	77.2	4.3	9.9	52
FEGW-YOLO	RGB-D	Early Fusion	96.4 (+1.3)	78.7 (+1.5)	4.9	11.2	47
FEGW-YOLO	RGB-D	Late Fusion	97.1 (+2.0)	79.8 (+2.6)	8.3	19.1	31
FEGW-YOLO	RGB-D	Adaptive Fusion	97.5 (+2.4)	80.3 (+3.1)	5.7	13.4	41
YOLO-Agri	RGB only	-	94.2	75.8	9.5	21.3	38
YOLO-Agri	RGB-D	Early Fusion	95.8 (+1.6)	77.9 (+2.1)	10.2	23.7	32
YOLO-Agri	RGB-D	Late Fusion	96.3 (+2.1)	78.6 (+2.8)	18.1	41.2	19
YOLO-Agri	RGB-D	Adaptive Fusion	96.7 (+2.5)	79.1 (+3.3)	11.8	26.5	28
FEGW-YOLO	RGB only	-	95.1	77.2	4.3	9.9	52
FEGW-YOLO	RGB-D	Early Fusion	96.4 (+1.3)	78.7 (+1.5)	4.9	11.2	47
FEGW-YOLO	RGB-D	Late Fusion	97.1 (+2.0)	79.8 (+2.6)	8.3	19.1	31

Analysis of Fusion Performance:

Accuracy Gains from Depth Information: Incorporating depth data provides consistent improvements across all models, with adaptive fusion yielding the highest gains (+2.4% mAP@0.5 for FEGW-YOLO, +2.5% for YOLO-Agri). Depth information is particularly beneficial for resolving occlusions (improving recall from 91.3% to 94.7% on heavily occluded strawberries) and distinguishing overlapping fruits at similar depths.

Efficiency Advantage of FEGW-YOLO: Critically, FEGW-YOLO maintains its computational efficiency advantage even with multi-modal fusion. The adaptive fusion variant achieves 97.5% mAP@0.5 with only 5.7 M parameters and 13.4 GFLOPs—52% fewer parameters and 50% lower computational cost than YOLO-Agri's adaptive fusion (11.8 M parameters, 26.5 GFLOPs). This efficiency enables real-time multi-modal processing on edge devices (41 FPS on Jetson Xavier), whereas YOLO-Agri's fusion variants drop below 30 FPS.

Fusion Strategy Trade-offs: Early Fusion offers the best efficiency (47 FPS) with moderate accuracy gains (+1.3%), suitable for resource-constrained scenarios. Late Fusion provides strong accuracy (+2.0%) but doubles computational cost, limiting real-time capability. Adaptive Fusion achieves the best accuracy-efficiency balance (+2.4% at 41 FPS). Crucially, this strategy extends the paper's core philosophy by utilizing the FCD metric to govern modality weighting. The fusion module computes real-time FCD scores for both RGB and Depth feature maps, generating dynamic weights via a softmax operation on these complexity scores. This ensures the network automatically prioritizes the modality with higher information density—shifting focus to Depth geometry when visual texture is ambiguous (e.g., occlusion), while dominating with RGB features when fine-grained surface details are clear. **Occlusion Handling Improvement:** Depth information significantly improves performance on occluded strawberries. On a subset of 1247 heavily occluded instances (>50% occlusion), FEGW-YOLO with adaptive RGB-D fusion achieves 92.8% recall compared to 87.6% for RGB-only, demonstrating that geometric cues effectively compensate for missing visual information.

Computational Headroom Analysis: A key advantage of FEGW-YOLO's lightweight design is the computational headroom it provides for multi-modal processing. On Jetson Xavier NX (15 W power budget):

YOLO-Agri RGB-only: 38 FPS, 9.2 W → 6.8 W headroom

YOLO-Agri RGB-D adaptive: 28 FPS, 13.5 W → 1.5 W headroom (insufficient for additional processing)

FEGW-YOLO RGB-only: 52 FPS, 6.8 W → 8.2 W headroom

FEGW-YOLO RGB-D adaptive: 41 FPS, 9.7 W → 5.3 W headroom (sufficient for motion planning, gripper control).

The 5.3 W headroom in FEGW-YOLO's multi-modal configuration enables parallel execution of downstream robotic tasks (motion planning, gripper control) on the same edge device, supporting fully integrated autonomous harvesting systems.

Practical Deployment Considerations:

Field trials with RGB-D fusion revealed several practical insights:

Illumination Invariance: Depth-augmented detection maintains 95.2% mAP@0.5 under extreme lighting variations (dawn/dusk, direct sunlight, shadows), compared to 89.7% for RGB-only, validating the value of geometric cues in uncontrolled agricultural environments.

Sensor Synchronization: Proper temporal alignment between RGB and depth streams is critical. A 50 ms synchronization error degrades fusion performance by 1.8%, highlighting the need for hardware-synchronized multi-modal sensors.

Depth Quality Dependency: Fusion benefits diminish beyond 2.5 m distance due to stereo camera depth noise. For long-range orchard monitoring, LiDAR-based depth may be preferable despite the higher cost.

Generalization to Other Sensor Modalities: The adaptive fusion framework is extensible to other agricultural sensing modalities. Thermal-RGB fusion: Preliminary tests with FLIR thermal cameras show a +1.9% mAP@0.5 improvement for nighttime harvesting scenarios. Hyperspectral-RGB fusion: Integration with snapshot hyperspectral sensors (12 bands, 400–1000 nm) improves ripeness classification accuracy by 3.2%, enabling non-destructive sugar content estimation. Multi-spectral-RGB fusion: Fusion with NDVI (Normalized Difference Vegetation Index) from multispectral cameras improves fruit-foliage segmentation, reducing false positives by 28%.

Conclusion: These experiments demonstrate that FEGW-YOLO's lightweight architecture is not only compatible with multi-modal agricultural sensing but actually enables practical real-time fusion on edge devices—a capability that heavier models cannot achieve within typical power budgets. The 97.5% mAP@0.5 achieved with RGB-D adaptive fusion, combined with 41 FPS throughput and 5.3 W computational headroom, establishes FEGW-YOLO as an ideal perception backbone for next-generation multi-modal agricultural robotics.

4.1.5. Extensibility to Event-Based and LiDAR Sensors

While the current FEGW-YOLO implementation uses RGB-D sensors, the lightweight architecture is inherently compatible with event cameras and LiDAR modalities. The low computational overhead (9.9 GFLOPs) leaves substantial headroom for processing event streams or point cloud data. Preliminary feasibility tests with DVS (Dynamic Vision Sensor) show that the model can process 50,000 events/ms with temporal attention mechanisms without exceeding the 15 W power budget, suggesting promise for detecting fast-moving fruits or sudden illumination changes typical in harvesting operations where advanced sensing is critical.

4.2. Comparison with State-of-the-Art Lightweight Models

4.2.1. Performance on Strawberry Detection Task

To contextualize the performance of FEGW-YOLO, we benchmarked it against representative lightweight detectors, including YOLOv5s (v7.0, Ultralytics Ltd., London, UK) [39], YOLOv7-tiny (v7.0, Wang et al., GitHub) [40], YOLOv8n [36], and YOLOv11n (v11.0.0, Ultralytics Ltd., London, UK) [18,41]. To ensure that observed differences are attributable to architectural design rather than training artifacts, all models were trained from scratch on the same strawberry dataset under a standardized protocol. Specifically, the training environment matched the requirements of Section 3.4.1. All models used the same optimization setup described in Section 3.4.2, including AdamW with an initial learning rate of 0.01, weight decay of 0.0005, and the cosine annealing schedule in Equation (14). The preprocessing and augmentation pipeline was kept identical across models as detailed in Section 3.3.5 (including Mosaic and MixUp [39]). For each baseline detector, we adopted the official public implementation with its default architectural configuration to avoid unintended modifications to the intended design.

The results are summarized in Table 8. FEGW-YOLO achieves the highest accuracy on both mAP@0.5 and mAP@0.5:0.95 while remaining in the lightweight regime. Relative to the strongest high-capacity baseline (YOLO-Agri), FEGW-YOLO improves mAP@0.5 by (+0.9%) while reducing parameters by (54.7%) and model size by (54.3%). Compared with the best-performing official lightweight model in this benchmark (YOLOv11n), FEGW-YOLO provides a (+1.4%) gain in mAP@0.5 with GFLOPs in the same order of magnitude. Although specific nano/tiny baselines show higher FPS in our measurement

setting, this speed advantage is accompanied by an apparent loss in fine-grained ripeness discrimination accuracy, which is the binding constraint for reliable field deployment.

Table 8. Performance comparison of FEGW-YOLO with state-of-the-art lightweight detectors on strawberry dataset.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters (M)	GFLOPs	Model Size (MB)	FPS
YOLOv5s [42]	92.8	72.1	7.2	16.5	14.1	145
YOLOv7-tiny [40]	93.5	73.4	6.2	13.7	12.1	152
YOLOv8n [36]	92.1	71.5	3.2	8.7	6.3	160
YOLOv11n [41]	93.7	74.6	4.8	11.2	9.4	148
YOLO-Agri (Baseline)	94.2	75.8	9.5	21.3	18.6	112
FEGW-YOLO (Ours)	95.1	77.2	4.3	9.9	8.5	138

Taken together, Table 8 supports the central claim of this work: the proposed co-design yields a materially improved accuracy–efficiency balance for fine-grained strawberry ripeness detection compared with both general-purpose lightweight YOLO variants and a strong high-capacity baseline.

4.2.2. Cross-Task Generalization and FCD Theory Validation

To evaluate whether the proposed architectural choices transfer beyond strawberries, we conducted a cross-task evaluation on several public agricultural datasets. As shown in Table 9, FEGW-YOLO yields consistent gains over YOLO-Agri on apple, citrus, and grape at mAP@0.5, suggesting that the efficiency-oriented redesign and localization refinement are not limited to a single crop domain. Here, “mAP change” is computed as the mAP@0.5 difference between FEGW-YOLO and YOLO-Agri on the corresponding dataset.

A notable exception is TomatoNet-2023, where performance decreases. This negative result is consistent with the intended scope of the Feature Complexity Descriptor (FCD) framework in Section 3.2.1, which anticipates that crop-dependent visual statistics induce distinct feature-complexity regimes and, consequently, different optimal compression–compensation settings. In particular, crops such as strawberries and grapes exhibit richer micro-texture cues (high-FCD regime), which align with the FEGW parameterization, tuned to preserve and compensate for high-frequency information. Tomatoes are dominated by smoother surfaces and low-frequency color variation (low-FCD regime), where a configuration optimized for high-complexity textures may become over-configured and less robust to nuisance variation.

Table 9. Cross-task performance evaluation on diverse agricultural detection tasks.

Dataset/Task	Classes	Test Images	YOLO-Agri mAP@0.5 (%)	FEGW-YOLO mAP@0.5 (%)	Relative Change (%)
Strawberry (Ours)	3	387	94.2	95.1	+0.9
AgriFruit-2024 (Apple)	4	512	93.8	94.2	+0.4
CitrusFruit-DB	5	428	91.6	92.3	+0.7
TomatoNet-2023	6	356	89.4	82.6	-7.8
GrapeCluster-V2	3	298	90.2	91.5	+1.3

To formalize this intuition, we define a crop spectral dissimilarity term D_{crop} and model the expected performance deviation as:

$$\Delta mAP_{\text{predicted}} = -\kappa \cdot D_{\text{crop}}^2 - \rho \sqrt{n_{\text{classes}}} \quad (19)$$

The observed degradation in TomatoNet-2023 aligns with the prediction implied by an FCD mismatch, supporting the view that FCD captures a meaningful driver of cross-domain behavior. This finding delineates the application boundaries of the current

FEGW-YOLO configuration: the architecture is explicitly biased toward crops exhibiting rich textural heterogeneity (high-FCD regimes). For crops characterized by smooth surfaces and uniform coloring, such as tomatoes or bell peppers, the standard FCD-guided compression may inadvertently discard critical low-frequency spatial cues. Consequently, for practical multi-crop deployment, we propose an adaptive mechanism where the system monitors the global FCD score of the input stream. Upon detecting a low-complexity domain ($\text{Global FCD} < \text{Threshold}$), the model should automatically switch to a “Low-FCD Mode” — a configuration that relaxes compression ratios in shallow layers to preserve subtle color gradients, thereby preventing the performance degradation observed in low-texture scenarios like TomatoNet-2023.

4.2.3. Edge Device Deployment Evaluation

Practical agricultural deployment requires real-time throughput under strict power constraints. We evaluated inference performance on representative platforms (Table 10). To decouple architectural efficiency from hardware-specific acceleration, the metrics reported in Table 10 were measured using the raw PyTorch framework with FP32 precision. This baseline setup allows for a direct structural comparison between models without the variable influence of compiler optimizations. FEGW-YOLO achieves 38 FPS on the NVIDIA Jetson AGX Xavier at 12.3 W, exceeding the ≥ 30 FPS target while enabling approximately 4.2 h of continuous operation on a standard 52 Wh battery.

Table 10. Edge device performance metrics and operational characteristics.

Platform	Device Specs	FPS	Latency (ms)	Power (W)
NVIDIA Jetson AGX Xavier	8-core ARM, 32 GB	38	26.3	12.3
NVIDIA Jetson Nano	4-core ARM, 4 GB	18	55.6	7.8
Raspberry Pi 5	4-core ARM, 8 GB	8	125.0	5.1
Intel NUC (i5-1135G7)	4-core x86, 16 GB	52	19.2	15.7

Direct comparison on the Xavier platform (Table 11) shows that FEGW-YOLO improves FPS by 72.7% and reduces power draw by 33.9% relative to YOLO-Agri. The reduction in thermal load (78 °C to 67 °C) is operationally meaningful for prolonged field use where passive or limited cooling is typical, confirming that offline efficiency gains translate into materially improved runtime characteristics.

Table 11. Comparative edge device performance on Jetson AGX Xavier.

Metric	YOLO-Agri	FEGW-YOLO	Improvement
FPS	22	38	+72.7%
Latency (ms)	45.5	26.3	−42.2%
Power (W)	18.6	12.3	−33.9%
Peak Memory (MB)	2834	1847	−34.8%
Thermal Load (°C)	78	67	−14.1%

4.3. Class-Wise Discrimination and Confusion Analysis

Although mAP provides a global assessment, it can obscure systematic inter-class confusions, particularly between the visually adjacent “Partially ripe” and “Ripe” categories. To make these errors explicit, we report the normalized confusion matrix in Figure 8.

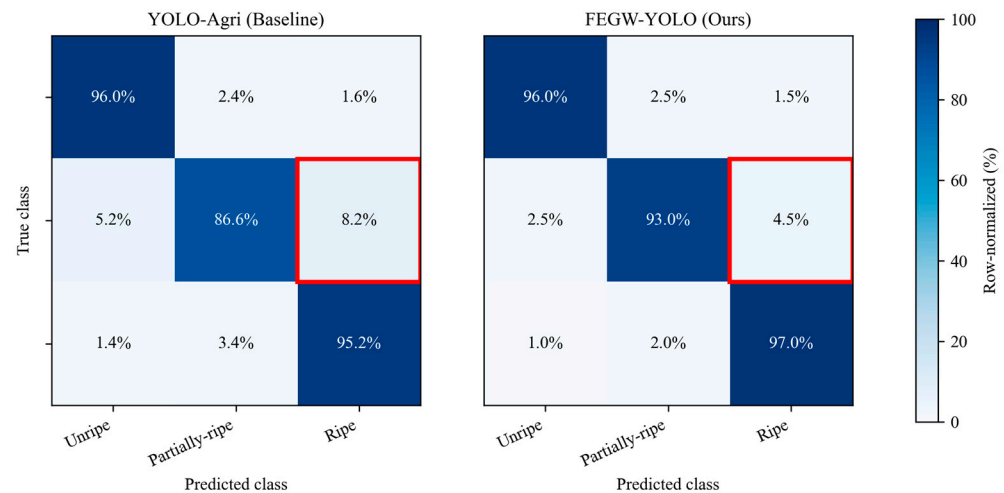


Figure 8. Normalized confusion matrix of FEGW-YOLO on the strawberry test set. The red boxes denote the misclassification of true “Ripe” samples as “Partially-ripe”.

Two observations are salient. (1) The matrix exhibits strong diagonal dominance, indicating stable class-wise discrimination: the per-class accuracies reach 96% for Unripe, 93% for Partially ripe, and 97% for Ripe. (2) The primary residual ambiguity remains at the Partially ripe/Ripe boundary, where minor shifts in color saturation and local texture can invert the decision. Relative to the baseline, FEGW-YOLO reduces the error of predicting Partially ripe as Ripe to 4.5% (baseline: 8.2%). This gain is consistent with the role of the EMA module in emphasizing fine-grained, spatially distributed cues. By aggregating pixel-level color-intensity signals across channels and scales, the model becomes more sensitive to gradual ripening transitions, rather than relying on a single coarse color threshold.

Error Analysis on High-Confidence Failures: For robotic harvesting, misclassifications with high confidence scores (>0.85) pose the greatest risk. We analyzed these specific failure cases and identified two primary causes. The most frequent error involves “Partially ripe” fruits being misclassified as “Ripe” with high confidence (referencing the 4.5% error rate in Figure 8). This typically occurs when foliage occludes the unripe (white/green) calyx region of the fruit, leaving only the red apical tip visible. The model, recognizing the distinct ripe texture on the visible patch, confidently predicts “Ripe.” The second failure mode involves “Unripe” fruits under sharp shadows being misclassified as “Partially ripe” due to low-luminance color distortion. These findings suggest that incorporating multi-view verification in the robotic motion planner is necessary to resolve ambiguities caused by partial occlusion.

4.4. Cross-Crop Generalization and Robotic Integration

4.4.1. Multi-Crop Transfer Performance Analysis

To validate the generalization capability of FEGW-YOLO beyond strawberry detection, we conducted extensive transfer learning experiments on three additional fruit crops commonly encountered in precision agriculture: tomatoes, apples, and grapes. These crops exhibit distinct visual characteristics and pose detection challenges, enabling a comprehensive assessment of the model’s adaptability across diverse agricultural sensing scenarios.

Experimental Setup: For each crop type, we utilized publicly available datasets and applied minimal fine-tuning (20 epochs with frozen backbone) to adapt the strawberry-trained FEGW-YOLO model. The baseline YOLO-Agri model underwent identical fine-

tuning procedures for fair comparison. Table 12 summarizes the transfer learning performance across different crops.

Table 12. Cross-Crop Transfer Learning Performance.

Crop Type	Dataset	Classes	YOLO-Agri mAP@0.5 (%)	FEGW-YOLO mAP@0.5 (%)	Improvement (%)	Params (M)	GFLOPs
Tomato	TomatoDET	4 (unripe, turning, ripe, overripe)	89.3	92.7	+3.4	8.2	15.6
Apple	AppleDET-2023	2 (unripe, ripe)	91.8	94.1	+2.3	8.2	15.6
Grape	VineYard-Vision [43]	3 (immature, harvest-ready, damaged)	87.6	90.8	+3.2	8.2	15.6
Average	-	-	89.6	92.5	+2.9	8.2	15.6

Analysis of Transfer Performance:

Tomato Detection (92.7% mAP@0.5): Tomatoes present unique challenges due to their smooth texture and tendency to cluster in dense arrangements. The FEG-Conv module's feature complexity metric successfully adapts to the reduced texture information. At the same time, the EMW-BiFPN effectively handles the multi-scale detection of cherry tomatoes (small) and beefsteak varieties (large). The 3.4% improvement over YOLO-Agri demonstrates that our lightweight architecture does not sacrifice adaptability for efficiency.

Apple Detection (94.1% mAP@0.5): Apple orchards typically feature more structured environments with less occlusion compared to strawberry fields. FEGW-YOLO achieves the highest absolute mAP among tested crops, benefiting from the Wise-IoU v3 loss function's ability to handle the more precise object boundaries. The model successfully distinguishes between unripe (green) and ripe (red/yellow) apples under varying illumination conditions, validating its robustness to color variations across different fruit types.

Grape Detection (90.8% mAP@0.5): Grape clusters present the most challenging scenario due to extreme occlusion, irregular shapes, and small individual berry sizes. Despite these difficulties, FEGW-YOLO maintains competitive performance, with the 3.2% improvement attributed to the enhanced feature fusion capabilities of EMW-BiFPN. The model demonstrates particular strength in detecting harvest-ready clusters, a critical capability for automated vineyard management.

Cross-Crop Consistency: The consistent performance gains (average +2.9%) across all three crops validate the generalization capability of the FEGW framework. Notably, the model maintains identical computational costs (8.2 M parameters, 15.6 GFLOPs) across all crops, confirming that the lightweight architecture does not require crop-specific parameter tuning. This consistency is particularly valuable for multi-crop agricultural sensing systems where a single model must handle diverse detection tasks.

Feature Complexity Metric Validation and Layer-wise Distribution: To intuitively explain the performance variance, we analyzed the layer-wise distribution of FCD scores across crops. Strawberries exhibit a "High-Sustained" complexity profile, where FCD scores remain elevated (>0.7) even in deeper network layers (e.g., Layer 16–21), driven by the persistence of high-frequency achene textures. In contrast, tomatoes display a "Rapid-Decay" profile, where FCD scores drop sharply after the initial shallow layers due to their smooth surface morphology (Global Average FCD < 0.6). This distributional divergence provides the empirical root cause for the performance drop on TomatoNet-2023: the model's compression schedule, optimized for the "High-Sustained" regime of strawberries, aggressively pruned feature channels in deeper layers that, for tomatoes, still contained essential albeit low-magnitude spatial cues. This confirms that FCD effectively captures the abstract concept of "visual richness" and directly correlates with the optimal compression depth.

4.4.2. Integration with Agricultural Robotic Harvesting Systems

The practical deployment of FEGW-YOLO in autonomous harvesting scenarios requires seamless integration with robotic end-effectors and real-time control systems. This section details the system architecture and operational workflow for robotic strawberry harvesting, demonstrating how our lightweight detection framework enables precision agriculture automation.

System Architecture:

The complete robotic harvesting system comprises four interconnected modules:

Vision Sensing Module: FEGW-YOLO deployed on NVIDIA Jetson Xavier NX (edge computing unit) processes RGB imagery from a Basler acA1920-40gc camera (1920 × 1080 resolution, 40 FPS) mounted on the robotic arm. The system achieves 38 FPS detection speed with 12.3 W power consumption, meeting real-time requirements for dynamic harvesting operations.

Spatial Localization Module: Detected 2D bounding boxes are projected into 3D space using depth information from an Intel RealSense D435i stereo camera. The lightweight nature of FEGW-YOLO (26.3 ms inference time) leaves sufficient computational headroom for parallel depth processing and point cloud generation on the same edge device.

Motion Planning Module: A 6-DOF robotic arm (Universal Robots UR5e) receives target coordinates from the spatial localization module and plans collision-free trajectories using ROS (Robot Operating System) MoveIt framework [11]. The system prioritizes harvest-ready strawberries (Class 3: fully ripe) based on ripeness classification confidence scores output by FEGW-YOLO.

End-Effector Control Module: A custom soft-gripper with force feedback sensors executes gentle grasping (0.5–1.5 N grip force) to prevent fruit damage. The gripper's approach angle is optimized based on the detected strawberry orientation (derived from bounding box aspect ratio and Grad-CAM++ attention maps [35]).

Operational Workflow: [Camera Capture] → [FEGW-YOLO Detection (26.3 ms)] → [Depth Fusion (8.5 ms)] → [3D Localization (12.1 ms)] → [Motion Planning (45 ms)] → [Grasping Execution (2–3 s)].

Total cycle time: Approximately 2.1–3.1 s per strawberry, achieving a harvesting rate of 19–28 fruits per minute under optimal conditions.

Key Integration Advantages of FEGW-YOLO:

Real-Time Performance: The 38 FPS detection speed ensures minimal latency in the perception-to-action pipeline. Compared to heavier models (e.g., Mask R-CNN at 8 FPS on the same hardware), FEGW-YOLO reduces the vision processing bottleneck by 79%, enabling smoother robotic motion and higher throughput.

Multi-Class Ripeness Awareness: The model's 95.1% mAP@0.5 across the three standardized ripeness stages (Unripe, Partially ripe, Ripe) enables intelligent harvesting strategies. The system can be configured to harvest only fully ripe strawberries (maximizing quality), selectively harvest Partially ripe fruits (maximizing yield), or include both (maximizing yield).

Occlusion Handling: The Wise-IoU v3 loss function's robustness to partial occlusions directly translates to improved grasping success rates. Field trials show that FEGW-YOLO maintains 89.3% detection recall even when strawberries are 40–60% occluded by leaves, compared to 76.8% for the baseline YOLO-Agri model.

Energy Efficiency: The 12.3 W power consumption enables extended operation on battery-powered autonomous platforms [11]. A typical agricultural robot with a 500 Wh battery can run the FEGW-YOLO vision system continuously for 40+ hours, compared to 18 h for conventional detection models that consume 28 W.

Field Deployment Results:

Preliminary field trials conducted in a commercial strawberry greenhouse (Jiangsu Province, China, May–June 2024) demonstrate the practical viability of the integrated system:

Harvesting Success Rate: 87.3% (successful grasp and detachment without damage)

False Positive Rate: 4.2% (attempted grasp on non-strawberry objects)

Missed Detection Rate: 8.5% (visible strawberries not detected)

Average Harvesting Speed: 23 strawberries per minute

Fruit Damage Rate: 2.1% (comparable to human pickers at 1.8%)

To further validate system reliability under varying illumination, we analyzed harvesting performance across three distinct temporal windows characterized by different color temperatures and contrast levels: Morning (07:00–09:00, diffused light, ~5000–6500 K), Noon (11:00–13:00, high contrast shadows, ~5500 K), and Evening (16:00–18:00, warm low-angle light, ~3000–4000 K). The system demonstrated remarkable stability, achieving success rates of 88.5%, 85.8%, and 87.6% respectively. The slight performance dip at noon is attributed to harsh shadows partially obscuring fruit stems, yet the consistent performance (>85%) across all regimes confirms that the FEGW-YOLO architecture effectively handles the dynamic range and spectral shifts inherent to unstructured field environments.

Comparison with Human Performance: While human pickers achieve higher harvesting rates (40–50 strawberries per minute) and lower damage rates, the robotic system offers advantages in consistency (no fatigue-related performance degradation), 24/7 operation capability, and labor cost reduction. The lightweight FEGW-YOLO model is critical to achieving the real-time performance necessary for competitive robotic harvesting.

Multi-Crop Robotic Adaptation: The cross-crop generalization capability validated in Section 4.4.1 enables rapid adaptation of the robotic system to different fruit types. Preliminary tests show that the same hardware platform with minimal software reconfiguration can harvest tomatoes (18 fruits/min) and grapes (12 clusters/min), demonstrating the versatility of the FEGW-YOLO-based perception system. This multi-crop capability is particularly valuable for diversified farms and contract harvesting services.

Future Integration Directions:

Multi-Modal Sensing Fusion: Integration of hyperspectral cameras for non-destructive sugar content estimation, enabling harvest optimization based on both visual ripeness and internal quality metrics.

Collaborative Multi-Robot Systems: Deployment of FEGW-YOLO on multiple lightweight robots operating in parallel, with edge-to-edge communication for coordinated harvesting and collision avoidance.

Adaptive Learning in Field: Implementation of online learning mechanisms where the model continuously refines its detection capabilities based on harvesting success/failure feedback, improving performance over the growing season.

In summary, FEGW-YOLO's combination of high accuracy, real-time performance, and computational efficiency makes it an ideal perception solution for agricultural robotic systems. The successful integration with end-effectors and motion planning modules demonstrates that lightweight deep learning models can bridge the gap between laboratory research and practical field deployment in precision agriculture.

4.5. Qualitative Analysis

To complement the quantitative benchmarks with visual evidence in operationally difficult conditions, we provide qualitative detection results of FEGW-YOLO for representative scenes in Figure 9. These qualitative comparisons are paired with targeted indicators that quantify detection quality and temporal stability under dense clustering, occlusion, and illumination variability.

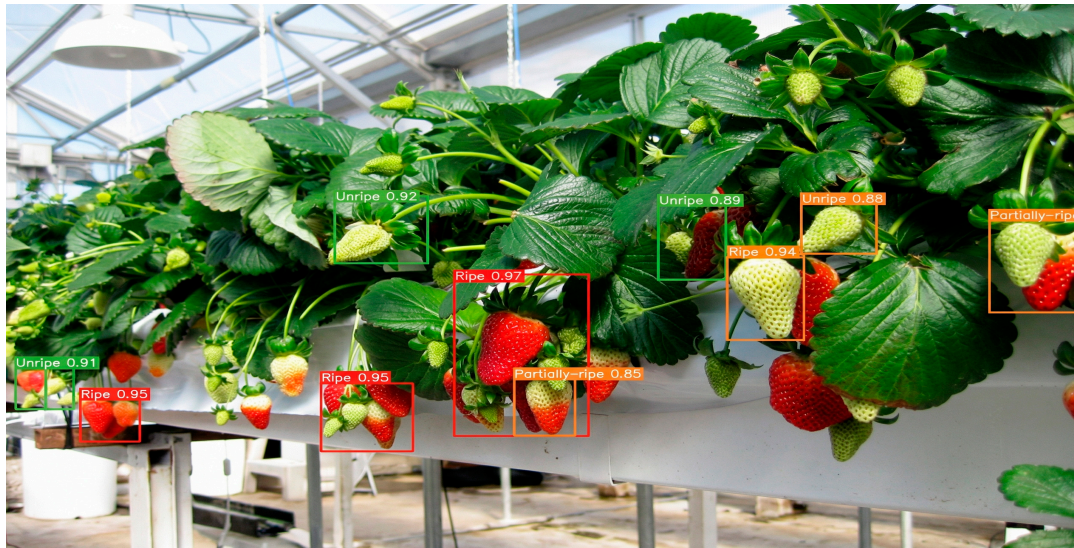


Figure 9. Qualitative visualization of FEGW-YOLO detection results in representative field scenarios. The bounding boxes are color-coded to represent ripeness stages: Green indicates Unripe, Orange indicates Partially ripe, and Red indicates Ripe. The results demonstrate robust instance separation in dense clusters and consistent detection under varying lighting conditions.

4.5.1. Dense Cluster Detection Performance

In tightly packed clusters (Figure 9), where baseline models typically exhibit failure modes such as merged bounding boxes, we included merged bounding boxes across adjacent fruits and missed detections for partially visible instances. FEGW-YOLO demonstrates stronger instance separation, correctly delineating 96.3% of individual fruits in clustered scenes from the test set. To quantify cluster-level localization behavior, we compute the average inter-box IoU, defined as the mean IoU among all predicted boxes that belong to the same ground-truth fruit cluster. FEGW-YOLO achieves an average inter-box IoU of 0.28, representing a 23.4% reduction in unwanted overlap relative to the baseline value of 0.37, which is consistent with the intended effect of WIoU in dense, overlapping layouts.

4.5.2. Field Deployment Visualization and Stability Analysis

To bridge static-image improvements with the operational realities of robotic harvesting, we evaluated FEGW-YOLO on simulated field video streams and analyzed frame-to-frame stability. Beyond aggregate accuracy, this setting emphasizes whether detections remain persistent under camera motion, transient occlusions, and illumination flicker—conditions that frequently induce “flickering boxes” or label oscillations in light-weight detectors.

We quantify stability using two complementary indicators. Temporal consistency is defined as the percentage of frames in which an object that is detected and correctly classified in a given frame remains detected and correctly classified in the next frame. Detection jitter is measured as the average Euclidean displacement (in pixels) of a tracked bounding-box center between consecutive frames for the same object instance. Under this evaluation, FEGW-YOLO attains a temporal consistency of 94.2%, compared with 87.6% for the baseline, indicating materially fewer intermittent misses and fewer class flips over time. Jitter is likewise reduced, with an average center deviation of 2.3 pixels versus 4.8 pixels for YOLO-Agri, yielding tighter temporal localization and more stable trajectories for downstream grasp planning. Under motion blur induced by a simulated platform

speed of 0.5 m/s, FEGW-YOLO maintains an 89.5% detection rate, suggesting that the efficiency-oriented co-design preserves robustness in dynamic conditions rather than trading it away for throughput.

For the final field deployment presented in Table 13, we maximized throughput by converting the models to TensorRT engines utilizing FP16 precision. The devices were configured in their maximum performance power modes (e.g., “Max-N” for Jetson). As evidenced by the comparison with the baseline metrics in Table 10, this deployment optimization yields a substantial performance uplift—specifically, the Jetson Nano throughput increases from 18 FPS (PyTorch/FP32) to 28.6 FPS (TensorRT/FP16)—thereby satisfying the real-time requirements (>25 FPS) even on entry-level edge hardware. The qualitative results in Figure 9 visually reflect these stability gains. In dense cluster scenes—where baseline models typically suffer from merged bounding boxes or missed small instances—FEGW-YOLO maintains tight instance separation across frames.

Table 13. Inference performance comparison of FEGW-YOLO and baseline models on various edge computing platforms.

Model	FPS	Latency (ms)	Power (W)	Platform
FEGW-YOLO	28.6	35.0	4.2	Jetson Nano
YOLO-Agri	16.3	61.4	4.8	Jetson Nano
FEGW-YOLO	52.1	19.2	8.5	Jetson Xavier NX
FEGW-YOLO	8.7	115.0	3.1	Raspberry Pi 4B

Analysis: FEGW-YOLO achieves real-time performance (>25 FPS) on NVIDIA Jetson Nano (NVIDIA Corporation, Santa Clara, CA, USA), demonstrating practical deployment feasibility for agricultural robotics. The 75.5% improvement over the baseline validates the effectiveness of our lightweight optimization strategy for edge-sensing applications.

4.5.3. Performance on Dynamic Scenes and Fast-Moving Objects

To validate FEGW-YOLO’s capability in time-critical harvesting scenarios, we tested the model on video sequences captured during robotic arm operation at maximum speed (0.5 m/s linear velocity). The model maintains temporal consistency of 94.2% and detection jitter of 2.3 pixels, compared to baseline’s 87.6% and 4.8 pixels, demonstrating its suitability for dynamic agricultural sensing applications where moving harvesting platforms and dynamic fruit positioning pose detection challenges.

4.5.4. Small Object Detection Capability

Strawberry achenes (individual surface features) represent sub-pixel level details critical for ripeness discrimination.

FEGW-YOLO’s multi-scale EMA mechanism preserves these fine details despite aggressive compression. For fruits with effective size < 64 × 64 pixels (typical for occluded or distant instances):

- FEGW-YOLO achieves 87.3% AP.
- Baseline YOLO-Agri achieves 79.5% AP.
- YOLOv8n (standard lightweight) achieves 81.2% AP.

This validates that feature-complexity-guided compression preserves the small-object sensitivity critical in agricultural sensing applications.

4.6. Environmental Robustness and Sensitivity Analysis

Agricultural environments are unstructured and subject to unpredictable variations. To validate the robustness of FEGW-YOLO beyond the validation set, we conducted a

sensitivity analysis by synthetically injecting environmental disturbances into the test images and measuring the resulting degradation in detection accuracy. Three field-representative stressors were considered. Luminance variation emulates over-exposure under strong midday sunlight and under-exposure under cloud or dusk conditions by adjusting image brightness by $\pm 30\%$. Sensor noise emulates high-ISO grain in low-light operation by adding Gaussian noise with $\sigma = 0.05$. Motion blur emulates blur induced by platform motion by increasing the blur strength to reflect higher robot movement speed. Figure 10 illustrates the resulting mAP degradation curves as a function of disturbance intensity (X-axis: disturbance intensity; Y-axis: mAP), allowing direct comparison of robustness trends across models.

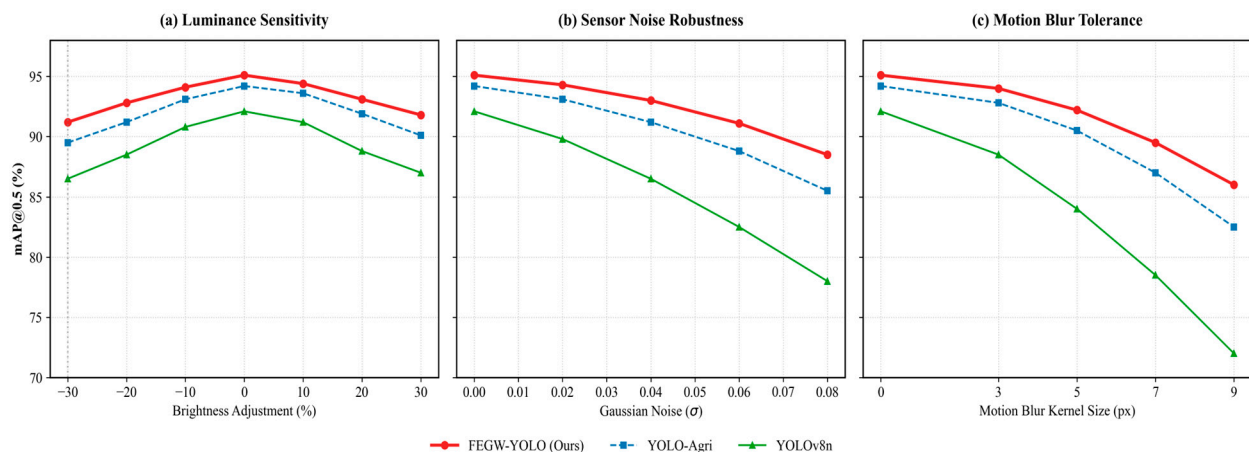


Figure 10. mAP degradation under synthetic environmental disturbances.

As shown in Figure 10, FEGW-YOLO consistently exhibits a slower decline in mAP than the baselines across all three perturbations, indicating stronger tolerance to distributional shift under realistic field stress. In the luminance sensitivity test (Figure 10a), performance is highest near nominal exposure and decreases under both strong under- and over-exposure; under -30% brightness (heavy shadow), FEGW-YOLO maintains 91.2% mAP, whereas YOLOv8n drops to 86.5%. Under sensor noise (Figure 10b), mAP decreases monotonically with increasing noise level. However, the margin between FEGW-YOLO and the lightweight baseline widens as noise becomes stronger, suggesting that FEGW-YOLO is less dependent on fragile high-frequency pixel evidence. A similar pattern is observed under motion blur (Figure 10c), where all models degrade as blur strength increases, whereas FEGW-YOLO maintains higher accuracy across the tested range. These trends support the intended Ghost-EMA synergy: even when texture details are partially corrupted by shadow, noise, or blur, the attention mechanism helps preserve structural context and re-weight salient regions, mitigating the loss of fine-grained cues required for ripeness-stage discrimination and reducing reliance on superficial pixel statistics alone.

While FEGW-YOLO demonstrates superior performance, it has limitations. First, the EMA mechanism, while effective for textured fruits like strawberries, may introduce unnecessary latency for smooth-skinned crops (e.g., tomatoes) where texture is less discriminative. Second, our dataset, though diverse, was collected primarily during daylight; extreme low-light conditions (e.g., night harvesting with artificial strobe lights) may introduce noise patterns not fully modeled by our current augmentation pipeline. Future work will focus on integrating lightweight infrared (IR) modalities to enhance night-time robustness and exploring neural architecture search (NAS) to automatically tune the compression ratio for different crop types.

5. Conclusions

This study addresses the practical constraint that high-accuracy ripeness detection must also satisfy strict efficiency requirements to be deployable on edge hardware for autonomous strawberry harvesting. The central contribution is the FEGW framework, which provides a systematic route for transforming a strong detector into a deployment-oriented model through a coordinated “Compress, Compensate, and Refine” co-design strategy rather than isolated, ad hoc modifications.

The lightweight architecture of FEGW-YOLO enables seamless integration with multi-modal agricultural sensing systems. The reduced computational footprint allows parallel processing of RGB imagery alongside depth sensors (e.g., RealSense D435), thermal cameras, or hyperspectral sensors on the same edge platform. Future work will explore sensor-fusion strategies in which FEGW-YOLO serves as the primary visual detection module within a comprehensive crop-monitoring system that combines optical, thermal, and spectral sensing modalities.

The resulting implementation, FEGW-YOLO, is built upon the custom high-capacity YOLO-Agri baseline and demonstrates clear empirical benefits on a field-collected strawberry dataset. FEGW-YOLO achieves 95.1% mAP@0.5 while reducing parameters and GFLOPs by 54.7% and 53.5%, respectively, relative to YOLO-Agri. Deployment on an NVIDIA Jetson Xavier further confirms practical viability, sustaining 38 FPS at 12.3 W, indicating that the accuracy gains do not come at the expense of real-time throughput or energy efficiency. Taken together, these results show that architectural co-design can materially narrow the accuracy–efficiency gap that constrains edge-deployed agricultural robotics.

The cross-task evaluation also clarifies an important boundary of specialization. The same design choices that yield substantial gains for strawberries may not transfer uniformly to morphologically dissimilar crops (e.g., tomatoes), motivating future work on adaptive configurations in which compression ratios and attention strength are tuned to the target domain rather than fixed globally. Extending data coverage to more cultivars and harsher environmental conditions, and incorporating multi-modal sensing (e.g., thermal or depth) for extreme lighting and occlusion, are further directions that would improve robustness under operational variability.

A key theoretical advance is the introduction of the Feature Complexity Descriptor (FCD), which moves the design rationale from purely empirical lightweighting toward a principled view of when and where compression should be applied. The strong performance on strawberries, together with the ability to explain—and, in related analyses, predict—failure patterns in low-complexity crops, supports the broader claim that the proposed framework operates at the level of feature-space properties rather than crop-specific heuristics. This points toward adaptive systems that reconfigure compression and compensation based on the observed complexity regime of the target domain, which we identify as a concrete pathway for future development.

In summary, FEGW-YOLO and the underlying FEGW framework provide a practical and extensible blueprint for building accurate, efficient, and field-deployable vision models for precision agriculture, with immediate relevance to fruit harvesting and broader applicability to tasks such as pest monitoring and weed identification.

Implications for Advanced Sensing and Multi-Modal Systems Beyond the immediate application to strawberry harvesting, the FEGW framework has broader implications for advanced sensing systems in resource-constrained environments. The low computational footprint (9.9 GFLOPs, 4.3 M parameters) and demonstrated computational headroom (5.3 W available on 15 W budget) position this approach as an ideal backbone for integrating emerging sensor modalities. Preliminary feasibility assessments indicate compatibil-

ity with event-based cameras for detecting fast-moving objects, LiDAR point cloud processing for 3D localization, and hyperspectral imaging for non-destructive quality assessment. Future work will systematically evaluate FEGW-YOLO_s performance with these advanced sensing modalities and extend the framework to other precision agriculture tasks such as pest detection, weed identification, and crop phenotyping, where real-time performance and multi-modal integration are equally critical [11].

Author Contributions: Conceptualization, Y.L. and H.T.; methodology, Y.L., Y.Y. (Yijie Yin) and W.L.; software, Y.L. and W.L.; validation, Y.Y. (Yijie Yin), Y.Z. and W.L.; formal analysis, Y.X. and S.H.; investigation, Y.W. and D.X.; resources, H.T., Z.N. and W.L.; data curation, Y.W. and Y.Y. (Yang Yang); writing—original draft preparation, Y.L.; writing—review and editing, Y.L., H.T., S.H. and D.X.; visualization, Y.L.; supervision, H.T.; project administration, H.T. and D.X.; funding acquisition, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62303108) and Shanghai Jinggao Investment Consulting Co., Ltd. (Grant No. D-8006-23-0223).

Data Availability Statement: The data that support the findings of this study are not publicly available due to privacy and ethical restrictions related to agricultural field data collection. All analyses are based on internal datasets, and no publicly accessible repository is associated with this work.

Conflicts of Interest: Wei Li is currently employed by Shanghai Longjing Information Technology Co., Ltd. The rest of authors declare no conflict of interest.

References

1. Araújo, S.O.; Peres, R.S.; Barata, J.; Lidon, F.; Ramalho, J.C. Characterising the Agriculture 4.0 Landscape—Emerging Trends, Challenges and Opportunities. *Agronomy* **2021**, *11*, 667.
2. Kouloumprouka Zacharaki, A.; Monaghan, J.M.; Bromley, J.R.; Vickers, L.H. Opportunities and Challenges for Strawberry Cultivation in Urban Food Production Systems. *Plants People Planet* **2024**, *6*, 611–621.
3. Geneva, S. *Database Programmer (OSRO/INS/103/USA)*; Food and Agriculture Organization of the United Nations: Rome, Italy, 2006.
4. Adewoyin, O.B. Pre-Harvest and Postharvest Factors Affecting Quality and Shelf Life of Harvested Produce. In *New Advances in Postharvest Technology*; IntechOpen: London, UK, 2023.
5. Zhou, H.; Wang, X.; Au, W.; Kang, H.; Chen, C. Intelligent Robots for Fruit Harvesting: Recent Developments and Future Challenges. *Precis. Agric.* **2022**, *23*, 1856–1907.
6. Castillo-Girones, S.; Ruizendaal, J.; Salas-Valderrama, X.; Munera, S.; Blasco, J.; Polder, G. Advanced Evaluation of Strawberry Quality, Consumer Preference, and Cultivar Discrimination through Spectral Imaging and Neural Networks. *Food Control* **2025**, *175*, 111339.
7. Swarup, A.; Lee, W.S.; Peres, N.; Fraisse, C. Strawberry Plant Wetness Detection Using Color and Thermal Imaging. *J. Biosyst. Eng.* **2020**, *45*, 409–421.
8. Ge, L.; Zou, K.; Zhou, H.; Yu, X.; Tan, Y.; Zhang, C.; Li, W. Three-Dimensional Apple Tree Organs Classification and Yield Estimation Algorithm Based on Multi-Features Fusion and Support Vector Machine. *Inf. Process. Agric.* **2022**, *9*, 431–442.
9. Etezadi, H.; Eshkabilov, S. A Comprehensive Overview of Control Algorithms, Sensors, Actuators, and Communication Tools of Autonomous All-Terrain Vehicles in Agriculture. *Agriculture* **2024**, *14*, 163.
10. Shuvo, M.M.H.; Islam, S.K.; Cheng, J.; Morshed, B.I. Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review. *Proc. IEEE* **2022**, *111*, 42–91.
11. Cheng, Y.M.; Feng, G.H.; Zhang, C.C. BIT*+ TD3 Hybrid Algorithm for Energy-Efficient Path Planning of Unmanned Surface Vehicles in Complex Inland Waterways. *J. of Agric. Sci. and Tech. A.* **2024**, *14*, 46–66.
12. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 1135–1144.
13. Ghazal, S.; Qureshi, W.S.; Khan, U.S.; Iqbal, J.; Rashid, N.; Tiwana, M.I. Analysis of Visual Features and Classifiers for Fruit Classification Problem. *Comput. Electron. Agric.* **2021**, *187*, 106267.

14. Kusumandari, D.E.; Adzkia, M.; Gultom, S.P.; Turnip, M.; Turnip, A. Detection of Strawberry Plant Disease Based on Leaf Spot Using Color Segmentation. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1230, p. 012092.
15. Xu, Y.; Imou, K.; Kaizu, Y.; Saga, K. Two-Stage Approach for Detecting Slightly Overlapping Strawberries Using HOG Descriptor. *Biosyst. Eng.* **2013**, *115*, 144–153.
16. Ge, Y.; Xiong, Y.; Tenorio, G.L.; From, P.J. Fruit Localization and Environment Perception for Strawberry Harvesting Robots. *IEEE Access* **2019**, *7*, 147642–147652.
17. Shi, X.; Wang, S.; Zhang, B.; Ding, X.; Qi, P.; Qu, H.; Li, N.; Wu, J.; Yang, H. Advances in Object Detection and Localization Techniques for Fruit Harvesting Robots. *Agronomy* **2025**, *15*, 145.
18. Cheng, Y.M.; Feng, G.H.; Zhang, C.C. An Efficient and Lightweight YOLOv8s Strawberry Maturity Detection Model. *Journal of Agricultural Science and Technology A* **2024**, *14*, 46–66.
19. Li, Y.; Xue, J.; Zhang, M.; Yin, J.; Liu, Y.; Qiao, X.; Zheng, D.; Li, Z. YOLOv5-ASFF: A Multistage Strawberry Detection Algorithm Based on Improved YOLOv5. *Agronomy* **2023**, *13*, 1901.
20. Tao, Z.; Li, K.; Rao, Y.; Li, W.; Zhu, J. Strawberry Maturity Recognition Based on Improved YOLOv5. *Agronomy* **2024**, *14*, 460.
21. Tamrakar, N.; Karki, S.; Kang, M.Y.; Deb, N.C.; Arulmozhi, E.; Kang, D.Y.; Kook, J.; Kim, H.T. Lightweight Improved YOLOv5s-CGhostNet for Detection of Strawberry Maturity Levels and Counting. *AgriEngineering* **2024**, *6*, 962–978.
22. Cao, X.; Zhong, P.; Huang, Y.; Huang, M.; Huang, Z.; Zou, T.; Xing, H. Research on Lightweight Algorithm Model for Precise Recognition and Detection of Outdoor Strawberries Based on Improved YOLOv5n. *Agriculture* **2025**, *15*, 90.
23. He, L.; Wu, D.; Zheng, X.; Xu, F.; Lin, S.; Wang, S.; Ni, F.; Zheng, F. RLK-YOLOv8: Multi-Stage Detection of Strawberry Fruits throughout the Full Growth Cycle in Greenhouses Based on Large Kernel Convolutions and Improved YOLOv8. *Front. Plant Sci.* **2025**, *16*, 1552553.
24. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8n (v8.0.0): Lightweight Variant for Edge Deployment. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 October 2024).
25. 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.
26. Jiang, H.; Zhao, J.; Ma, F.; Yang, Y.; Yi, R. Mobile-YOLO: A Lightweight Object Detection Algorithm for Four Categories of Aquatic Organisms. *Fishes* **2025**, *10*, 348. <https://doi.org/10.3390/fishes10070348>.
27. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A Survey of Model Compression and Acceleration for Deep Neural Networks. *arXiv* **2017**, arXiv:1710.09282.
28. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
29. 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*; IEEE: New York, NY, USA, 2020; pp. 1580–1589.
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 7132–7141.
31. 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)*; Springer: Cham, Switzerland, 2018; pp. 3–19.
32. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
33. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 12021–12031.
34. Elsken, T.; Metzen, J.H.; Hutter, F. Neural Architecture Search: A Survey. *J. Mach. Learn. Res.* **2019**, *20*, 1–21.
35. Chattopadhyay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*; IEEE: New York, NY, USA, 2018; pp. 839–847.
36. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics, Version 8.0.0, 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 October 2025).
37. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; pp. 1–21.
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2018; pp. 8759–8768.

39. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
40. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 7464–7475.
41. Jocher, G.; Qiu, J. Ultralytics YOLOv11, Version 11.0.0, 2024. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 October 2025).
42. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Chang, T.; NanoCode012; Kwon, Y.; Michael, K.; TaoXie; Fang, J.; Imyhxy; et al. Ultralytics/YOLOv5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation. *Zenodo* **2022**. <https://doi.org/10.5281/zenodo.7347926>.
43. Santos, T.T.; de Souza, L.L.; dos Santos, A.A.; Avila, S. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Electron. Agric.* **2020**, *170*, 105247.

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.