

Article

Lightweight WSS-YOLO Quince Fruit Detection Algorithm Integrating SimAM

Xingrui Wu ^{1,2}, Jinting Zou ^{1,3} and Haiwei Wu ^{1,*} 

¹ College of Engineering and Technology, Jilin Agricultural University, Changchun 130118, China; 18946532199@163.com (X.W.); zjt@tccu.edu.cn (J.Z.)

² School of Information and Digital Media, Jilin Science and Technology Vocational College, Changchun 130123, China

³ Artificial Intelligence College, Changchun University Tourism College, Changchun 130607, China

* Correspondence: haiwei@jlau.edu.cn

Abstract

Real-time fruit maturity detection in unstructured orchards remains challenging because of variable illumination, fruit occlusion, complex backgrounds, and the limited computing capacity of edge devices. To address these challenges, this study proposes WSS-YOLO, a lightweight detection framework based on YOLOv11n for quince maturity detection. The model introduces WaveletPool to reduce texture loss during downsampling, adopts a GSConv-based Slim-neck to improve feature fusion with lower computational cost, and integrates SimAM to enhance discriminative fruit-region responses without adding trainable parameters. Experiments on a multi-scenario quince maturity dataset show that WSS-YOLO achieves 86.4% precision, 87.5% recall, and 93.4% mAP@0.5, improving the YOLOv11n baseline by 2.3, 1.7, and 2.5 percentage points, respectively. The model contains only 2.23 M parameters and requires 4.1 G FLOPs. Deployment on the NVIDIA Jetson Orin Nano achieved a real-time speed of 23.0 FPS, suggesting a favorable trade-off between detection accuracy and computational efficiency under the tested conditions.

Keywords: fruit maturity detection; YOLOv11n; wavelet pooling; lightweight network; SimAM

1. Introduction

Japanese quince is a temperate fruit species characterized by aromatic, acid-rich fruits that undergo pronounced physicochemical changes during ripening, making it suitable for non-destructive maturity detection in postharvest and precision agriculture studies. With the continuous growth of the global population and the increasing demand for high-quality agricultural products, agriculture is rapidly transitioning toward digitalization and intelligent management. In orchard production systems, accurate monitoring of fruit maturity is essential for determining optimal harvest timing, improving supply chain efficiency, and ensuring fruit quality. However, traditional maturity assessment and harvesting practices largely depend on manual experience, which is not only labor-intensive and inefficient but also prone to subjective bias, often resulting in premature or delayed harvesting in large-scale orchards and ultimately reducing economic returns [1].

In recent years, computer vision technology, particularly deep learning-based object detection algorithms, has demonstrated tremendous application potential in the agricultural sector. Kamilaris and Prenafeta-Boldú systematically reviewed the research progress and application status of deep learning methods in agricultural information, indicating



Academic Editor: Antonio López-Quílez

Received: 31 May 2026

Revised: 19 June 2026

Accepted: 19 June 2026

Published: 24 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

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

that deep learning technology provides essential technical support for agricultural intelligence [2]. While early methodologies relied on two-stage mechanisms (e.g., Faster R-CNN) to ensure accuracy, the advent of single-stage paradigms—most notably the YOLO lineage—has marked a critical milestone. These modern approaches have successfully reconciled the trade-off between computational speed and detection performance [3]. Terven et al. conducted an exhaustive survey of the YOLO lineage, spanning from the inaugural v1 to the recent YOLOv8 and YOLO-NAS. Their study systematically dissects the architectural innovations and performance trajectories characterizing each iteration of the series [4]. In the field of fruit detection, Lawal proposed a modified YOLOv3 framework for tomato detection, which significantly improved detection performance in natural orchard environments by optimizing network structure [5,6]. The latest YOLOv11 model further enhances feature extraction efficiency through an enhanced backbone and neck architecture, establishing a new benchmark in real-time object detection [7].

However, directly transferring these models that perform excellently on general datasets to unstructured natural orchard environments still faces dual challenges of environmental adaptability and hardware resource limitations. On one hand, orchard environments exhibit high complexity and uncontrollability. Unlike controlled laboratory settings, dramatic fluctuations in natural lighting conditions (such as strong backlighting, dappled shadows) and random occlusion of fruits by branches and leaves can easily cause loss of visual information. Furthermore, fruits at different maturity stages often display subtle phenotypic differences, especially when unripe fruits show high color similarity with background foliage, making it difficult for general models to capture fine-grained features that distinguish maturity levels, resulting in higher miss detection and false detection rates in complex scenarios [8,9]. On the other hand, agricultural automation applications typically require deploying detection models on edge devices such as mobile robots, drones, or handheld terminals. These devices have very limited computational power (FLOPs), storage space, and power budgets. Existing high-performance detection models often come with massive parameter counts and complex network structures, leading to excessive inference latency that fails to meet the stringent “real-time” requirements of agricultural operations.

Lightweight object detection network design has gradually become a research hotspot. Addressing the challenges of detection in naturalistic agricultural backdrops, Chen et al. introduced GSBF-YOLO, a model that leverages the GSim strategy to minimize parameter overhead. Empirical results indicate its superior performance in accurately characterizing tomato ripeness despite environmental complexities [10]. Building upon the YOLO11n baseline, Li et al. engineered the YOLO11-LES framework for assessing strawberry maturity. By synergizing a lightweight adaptive weighting downsampling scheme with a spatial-enhanced attention mechanism, the model constrains its storage footprint to a mere 4.6 MB whilst securing a 2.9% gain in precision [11]. In the domain of pomegranate recognition, Chen et al. devised the PL-YOLO framework, which incorporates an edge-feature extraction unit and a context-guided attention FPN to navigate environmental complexities [12]. Parallely, Lou et al. engineered YOLO-TLA, a streamlined architecture that employs C3CrossCovn blocks and a specialized detection head for small targets [13]. While these innovations have reduced computational burdens, establishing an ideal equilibrium between granular feature preservation and inference efficiency within unstructured orchard settings remains an unresolved hurdle.

Traditional pooling often incurs a loss of high-frequency information (e.g., texture). As a remedy, Williams and Li proposed wavelet pooling, a method that compresses features by discarding specific sub-bands during decomposition, thereby solving the overfitting issues inherent in max pooling more effectively than neighborhood approaches [14]. Subsequent

studies, such as the MWCNN model by Liu et al., successfully integrated these transforms to balance resolution and receptive fields for tasks like super-resolution [15]. Additionally, Brito et al. proposed a multi-pooling network combining the advantages of max pooling and wavelet pooling, effectively changing output signal dimensions through 1×1 convolution, achieving better trade-offs in semantic segmentation tasks [16]. However, existing wavelet pooling methods are primarily applied to image restoration and classification tasks, with limited applications in agricultural object detection.

Attention mechanisms, as important techniques for enhancing network performance, have been widely applied in agricultural vision. Yang et al. formulated SimAM, a parameter-free operator that deduces 3D attention weights by minimizing an energy function. Its elegance lies in its implementation simplicity, requiring negligible code while identifying neuronal importance without adding model weight [17]. Distinctively, the CBAM module (Woo et al.) adopts a sequential inference strategy, refining features first along the channel axis and then the spatial axis to achieve adaptive calibration [18]. Furthermore, Hu et al. pioneered the Squeeze-and-Excitation (SE) paradigm, which explicitly models channel interdependencies to dynamically recalibrate feature responses, proving effective across diverse CNN backbones [19].

In network architecture optimization, Li et al. proposed GSConv as a convolution strategy in depthwise separable convolution that concatenates standard convolution results with depthwise convolution results followed by channel shuffling, significantly reducing parameters and computational costs while maintaining detection performance, with FLOPs approaching half that of standard convolution when channel numbers are large. The VoVGSCSP module, derived from the GSConv operation, represents a strategic fusion of CSP and VoVNet topologies. This hybrid architecture optimizes the speed-accuracy trade-off by mitigating computational complexity through efficient feature reuse and parallel processing mechanisms [20]. Zhou et al. proposed a lightweight real-time object detection method based on YOLOv4 for complex scenes, replacing CSPDarknet53 backbone network with MobileNetV3 and using depthwise over-parameterized convolutional layer to promote feature extraction effectiveness, achieving 41.82 fps on Titan X while maintaining competitive accuracy [21]. Targeting resource-limited computing environments, Chen et al. engineered shuffle-octave-yolo. When deployed on the NVIDIA Jetson TX2, this architecture attained a mean Average Precision (mAP) of 65.97% at a frame rate of 30.9 fps, thereby exemplifying a superior compromise between computational swiftness and detection fidelity [22].

Comprehensive analysis of existing research reveals that current fruit maturity detection methods face key challenges, including the contradiction between feature preservation and lightweighting, insufficient multi-scale information fusion, and inadequate environmental adaptability. Existing lightweight methods often sacrifice fine-grained features, and traditional feature fusion structures have incomplete information transmission between different levels and lack sufficient robustness when facing dense targets, occlusion, and natural environmental variations. Therefore, this study proposes an improved lightweight fruit maturity detection network, WSS-YOLO, based on YOLOv11n. This model is specifically designed for edge deployment in complex orchard environments, utilizing WaveletPool technology to losslessly preserve texture detail features through the multi-resolution characteristics of wavelet transforms, designing a Slim-neck lightweight architecture based on GSConv to significantly reduce parameter count and computational cost, and integrating the parameter-free attention mechanism SimAM to achieve an adaptive focus on key fruit regions and noise suppression, thereby effectively balancing computational efficiency while ensuring detection accuracy.

2. Materials and Methods

2.1. Data Materials

The dataset construction utilized the optical sensor of a Samsung Galaxy A8 smartphone (Samsung Electronics Co., Ltd., Suwon, Gyeonggi-do, Republic of Korea). This device, featuring a 16 MP rear camera with an $f/1.9$ aperture and a 27 mm equivalent focal length, was configured to output images at a raw resolution of 4608×3456 pixels. To maximize the ecological validity of the dataset within natural orchard settings, image capture spanned a broad spectrum of meteorological conditions, illumination angles, and background complexities. The imaging strategy was rigorously stratified into four spatial categories to bolster model generalization: macro shots (15–20 cm) for texture detail, partial canopy views (20–50 cm), full cluster compositions (50–70 cm), and wide-field orchard scenes (up to 1 m). This multi-scale imaging strategy helps the model learn fruit features across different object scales, from close-up texture details to wider orchard scenes [23].

Fruit maturity was classified into two categories: ripe and unripe. Unripe fruits exhibit a uniform green color, firm texture, and no noticeable aroma; ripe fruits display a yellow surface, slightly soft and glossy skin, indicating that the fruit is approaching the harvest stage. This classification criterion is based on externally visible fruit characteristics such as color, gloss, and surface texture, consistent with maturity identification standards commonly used in orchard management. As shown in Figure 1.



Figure 1. Images from the quince dataset.

To ensure annotation quality, all images were manually labeled by experienced personnel using LabelImg software. The final dataset contains 1515 images with 17,171 annotated fruit instances. Following a 7:2:1 ratio, the dataset was divided into a training set of 1061 images with 12,008 instances, a validation set of 303 images with 3366 instances, and a test set of 151 images with 1796 instances. The distribution of maturity categories was kept consistent across the three subsets to ensure stable model training and fair performance evaluation.

The dataset used in this study is publicly available, and to ensure fair comparison with previous works, no additional images were generated to expand it. However, online data augmentation was applied dynamically during model training within the YOLO framework to improve generalization. Specifically, this pipeline included HSV color perturbation (hue = 0.015, saturation = 0.7, value = 0.4), random translation (0.1), random scaling (0.5), horizontal flipping with a probability of 0.5, and Mosaic augmentation with a probability of 1.0. These augmentations simulate variations in illumination, object scale, and spatial distribution, thereby improving the model's generalization ability in complex orchard environments. To ensure stable convergence, Mosaic augmentation was disabled during the final 10 training epochs.

2.2. WSS-YOLO

As the latest iteration in the Ultralytics lineage, YOLOv11 [7] refines the training paradigms and architecture of YOLOv8 [24] to define a contemporary benchmark for real-time detection. A key structural evolution involves substituting the legacy C2f blocks with C3K2 modules within the backbone and neck. By leveraging parallel 3×3 convolutions and subsequent feature fusion, this design optimizes the equilibrium between inference speed and precision. Furthermore, the incorporation of C2PSA multi-scale attention enhances the model’s focus on occluded objects. However, notwithstanding these advancements, the baseline model exhibits vulnerability when confronted with extreme illumination variability, dense fruit occlusion, or nuanced maturity differences [25,26]. To bridge these gaps, this study proposes an enhanced framework (WSS-YOLO) that integrates specific feature enhancement strategies and optimized attention mechanisms to ensure robustness in unstructured agricultural settings.

In the WSS-YOLO model, multiple structures work synergistically to improve fruit maturity detection performance, as shown in Figure 2. The backbone network adopts the WaveletPool structure, which preserves overall structure and texture details through multi-scale wavelet decomposition, enhancing the capability to capture key features such as fruit skin color variations and spots. Slim-neck optimizes multi-scale feature fusion, improving information integration efficiency while reducing computational complexity, making it suitable for real-time detection. The SimAM attention mechanism is applied at the P5 layer before the network head, enhancing feature weights of key regions to enable the model to maintain high detection accuracy even when fruits are partially occluded or against complex backgrounds. The combination of these three components enables WSS-YOLO to achieve efficient and accurate fruit maturity detection in complex environments.

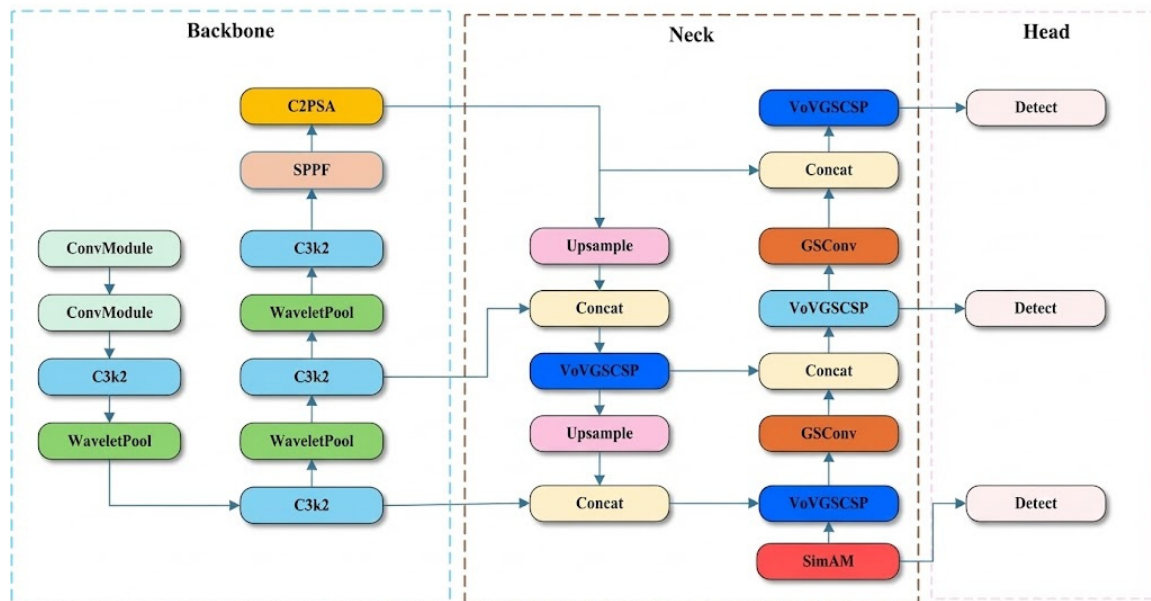


Figure 2. Improved YOLOv11n model architecture (WSS-YOLO).

2.2.1. WaveletPool

As visualized in Figure 3, the WaveletPool unit leverages the Discrete Wavelet Transform (DWT) to perform feature downsampling [13]. Unlike conventional max-pooling or average-pooling operations, WaveletPool decomposes the input feature maps into four sub-bands, including the low-frequency approximation component (LL) and three high-frequency detail components (LH, HL, HH). During the pooling process, the LL sub-band is retained as the downsampled representation, while the high-frequency sub-bands are

discarded. This design reduces feature dimensionality and introduces a regularization effect, helping to suppress redundant high-frequency responses and mitigate overfitting. In the forward propagation phase, channel-wise wavelet decomposition is implemented with a stride of 2, which integrates downsampling and signal filtering into a unified operation. Consequently, WaveletPool generates compact feature representations that preserve the dominant structural information of the input while improving computational efficiency, making it suitable for lightweight detection models requiring robust and efficient feature extraction.

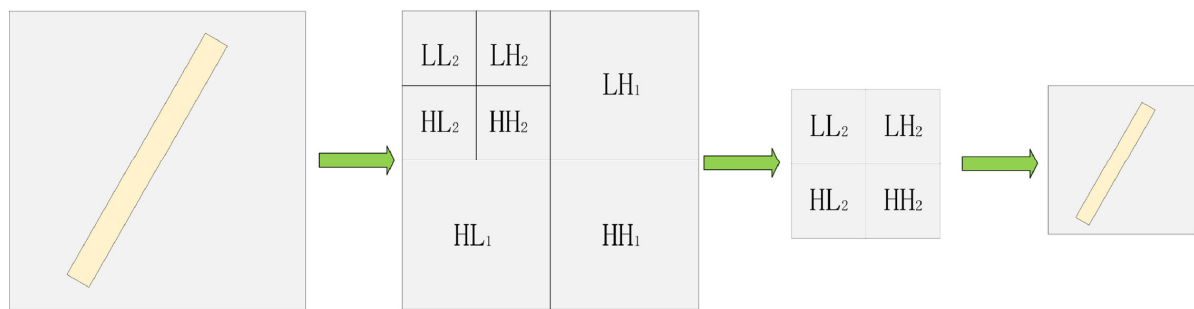


Figure 3. Detailed structure of WaveletPool.

2.2.2. Slim-Neck

GSConv (Group Shuffle Convolution) is a lightweight convolution operation that combines the advantages of Standard Convolution (SC) and Depthwise Separable Convolution (DWConv) [19]. As depicted in Figure 4, this mechanism reduces computational complexity while maintaining effective feature extraction capability. Procedurally, input feature maps are initially grouped via standard convolution, followed by the application of separable convolution to each depthwise group. This architectural design reduces computational cost and alleviates the disjointed processing of channel information, thereby preserving the integrity of extracted features. The final output is obtained by concatenating the results from both SC and DWConv branches, followed by a channel shuffle operation to enhance information exchange. By employing a dual-branch topology—where one branch performs downsampling for coarse semantic extraction and the other focuses on fine-grained spatial feature learning—GSConv integrates global and local information. This design achieves a low-complexity, high-performance structure suitable for real-time applications on resource-constrained hardware.

Focusing on the recognition of quince fruits, the VoVGSCSP component acts as a pivotal mechanism for optimizing the equilibrium between computational economy and detection accuracy. This unit is specifically engineered to capture intricate textural nuances while simultaneously handling dramatic variations in object scale. Through the strategic coupling of channel shuffling and group convolution, the module effectively filters out superfluous arithmetic operations. This design minimizes processing latency without degrading the quality of feature representation. Consequently, its streamlined topology ensures seamless compatibility with resource-limited embedded devices, translating to substantial savings in both energy consumption and memory demand. Furthermore, the module demonstrates exceptional proficiency in multi-level feature extraction, a capability essential for accurate segmentation within disordered agricultural backgrounds. Its success in resolving small-scale targets renders it an ideal candidate for implementation on hardware with strict resource limits.

Figure 5 visualizes the VoVGSCSP backbone, where efficient semantic feature transfer is facilitated by paired GSConv operations during upsampling and downsampling phases. In the neck architecture of the proposed model, the VoVGSCSP block is strategically

introduced to supersede the conventional C2f module. This substitution effectively alleviates the model's computational burden while ensuring that detection accuracy remains uncompromised.

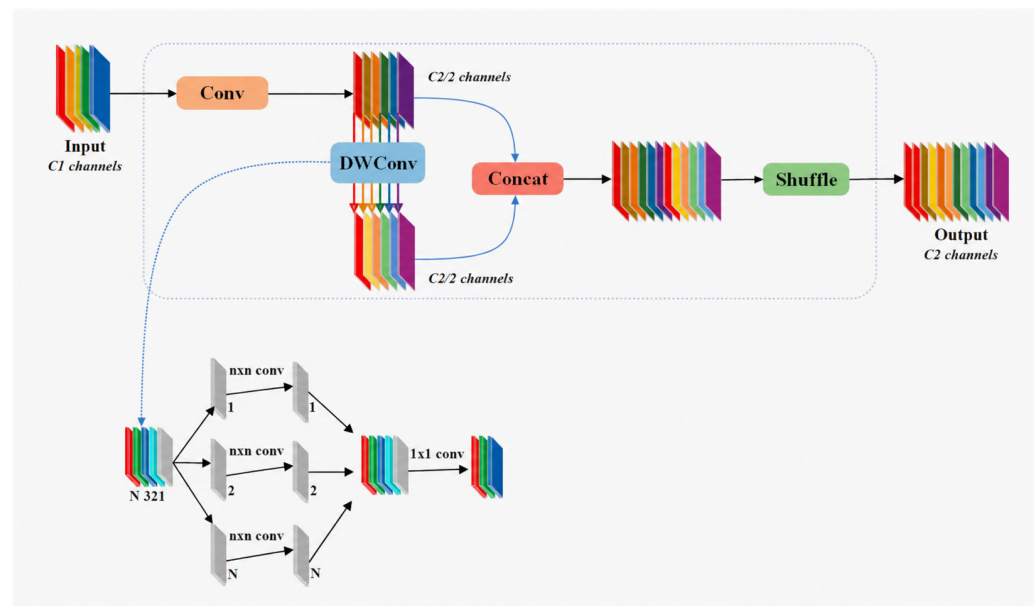


Figure 4. Structure of the GSConv module.

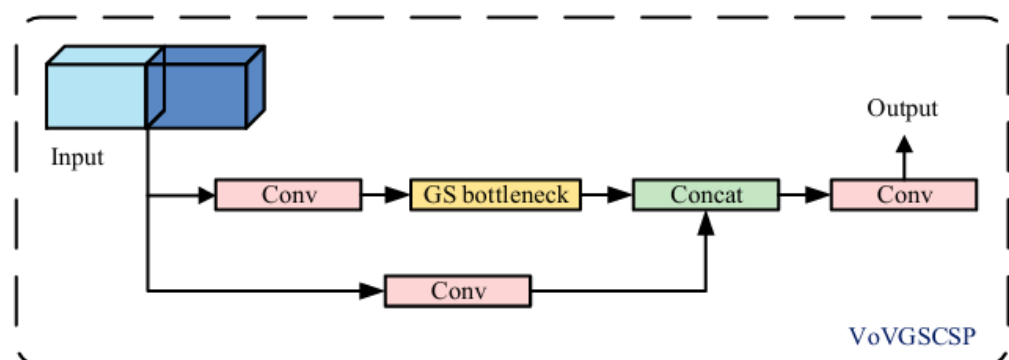


Figure 5. Structure of VoVGSCSP.

2.2.3. SimAM Attention Mechanism

This study incorporates the Simple Attention Module (SimAM) into the proposed network architecture, as illustrated in Figure 6. SimAM is a lightweight and parameter-free attention mechanism that evaluates the importance of each neuron by measuring its distinguishability from other neurons within the same channel. Unlike conventional attention mechanisms that focus solely on either channel or spatial dimensions, SimAM generates attention weights across both spatial and channel dimensions without introducing additional learnable parameters. For an input feature map X , the importance of each neuron is quantified using an energy function derived from neuroscience-inspired principles. The inverse energy of the target neuron is formulated as

$$E_t^{-1} = \frac{(x_t - \mu)^2}{4(\sigma^2 + \lambda)} + \frac{1}{2} \quad (1)$$

where x_t denotes the target neuron, μ and σ^2 represent the mean and variance of neurons within the corresponding channel, respectively, and λ is a small regularization coefficient introduced to ensure numerical stability. A higher inverse energy value indicates that the

neuron is more distinguishable from its surrounding context and therefore contains more discriminative information.

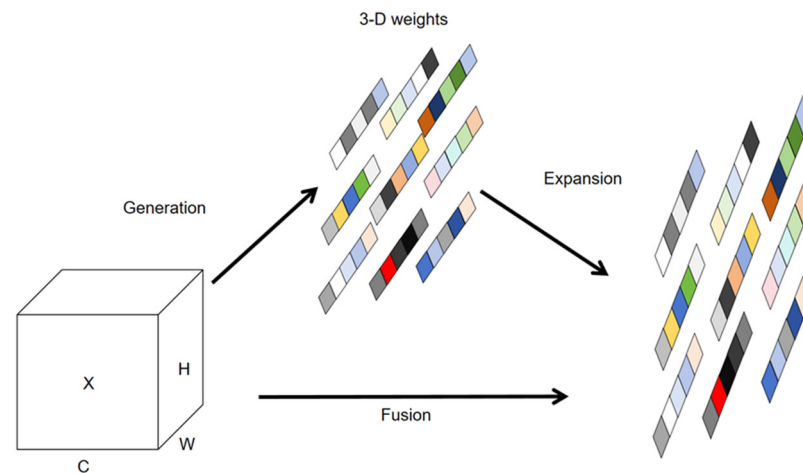


Figure 6. Structure of SimAm.

The attention-enhanced feature map is obtained by applying a sigmoid activation to the inverse energy and performing element-wise multiplication with the input feature map:

$$X_{att} = \text{Sigmoid}\left(\frac{1}{e_t}\right) \odot X \quad (2)$$

where X_{att} denotes the refined feature map, $\sigma(\cdot)$ represents the sigmoid activation function, and $\odot$ denotes element-wise multiplication.

By leveraging this energy-based attention mechanism, SimAM adaptively assigns higher weights to informative feature responses while suppressing irrelevant background noise. Since the module introduces no additional trainable parameters, it incurs negligible computational overhead and is well suited for lightweight object detection networks. The integration of SimAM improves the model's ability to extract discriminative fruit features under challenging orchard environments, including partial occlusion, illumination variations, and complex background interference.

3. Results

3.1. Experimental Environment

All empirical evaluations were executed on a workstation running the Ubuntu 18.04 operating system. The deep learning environment was configured using the PyTorch framework (version 1.8.0 + cuda11.1) compatible with Python 3.9.13. To ensure efficient computational processing, hardware acceleration was provided by an NVIDIA GeForce RTX 3090 GPU equipped with 24 GB of video memory. Table 1 outlines the specific hyperparameter settings adopted for model training.

Table 1. Detailed hyperparameters of the experiment.

Parameters	Value
Epochs	200
Batch Size	16
Optimizer	SGD
Initial Learning Rate	0.01
Optimizer momentum	0.937
Weight-Decay	5×10^{-4}
Close Mosaic	Last 10 epochs

3.2. Evaluation Criteria

To systematically evaluate the performance of YOLOv11n and its enhanced variants in fruit maturity analysis, this study employs established benchmarks: Precision, Recall, and mean Average Precision (mAP) [27]. As formalized in Equations (3)–(6), Precision gauges the reliability of positive predictions by determining the fraction of true positives (TPs) relative to the total positive predictions (the sum of TPs and false positives, FPs). In contrast, Recall assesses the model's sensitivity, indicating the percentage of actual ripe fruit successfully detected against the ground truth.

Regarding aggregate performance, mAP functions as the principal indicator. Specifically, mAP@0.5 signifies the mean precision averaged across all categories at a single Intersection over Union (IoU) threshold of 0.5. This metric underscores the model's capacity to balance accuracy and recall, where elevated scores denote stronger robustness. To offer a more rigorous analysis of localization quality, mAP@0.5:0.95 is also computed. This comprehensive metric averages mAP scores across a continuum of IoU thresholds from 0.5 to 0.95 (in 0.05 increments), yielding a granular perspective on detection fidelity under stricter overlap requirements.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$AP = \int_0^1 P \cdot RdR \quad (5)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (6)$$

3.3. Model Training Process

Figure 7 presents the training and validation loss curves together with the confusion matrix of the proposed model. The loss curves include box loss, classification loss (cls loss), and distribution focal loss (dfl loss) for both the training and validation sets. During training, all loss values decreased rapidly in the initial epoch and gradually converged to stable levels as training progressed. The trends observed in the validation set were consistent with those of the training set, indicating stable optimization and good generalization performance. To improve the visualization of the convergence process, smoothed curves were additionally provided to reduce the influence of fluctuations and highlight the overall training trends.

The confusion matrix further illustrates the classification performance of the proposed model. Most samples were correctly classified, as evidenced by the dominant diagonal entries. Specifically, 1448 ripe quinces and 1085 unripe quinces were correctly identified. Confusion between ripe and unripe quinces was limited, with only three unripe samples misclassified as ripe and two ripe samples misclassified as unripe. Most classification errors were associated with the background category. Some background regions were incorrectly recognized as ripe or unripe quinces, while a portion of fruit samples were classified as background. Nevertheless, the confusion matrix indicates that the proposed model can effectively distinguish ripe and unripe quinces while maintaining reliable performance in the presence of complex background conditions.

Figure 8 presents the comparison of precision–recall curves between the proposed WSS-YOLO model and the baseline YOLOv11n model. The PR curve of the proposed model encloses a larger area and is positioned closer to the upper-right corner compared with the baseline, indicating stronger overall detection performance. This demonstrates

that the proposed model maintains higher precision across a wider range of recall values for both ripe and unripe fruits, reflecting its improved ability to detect target instances while reducing false positives.

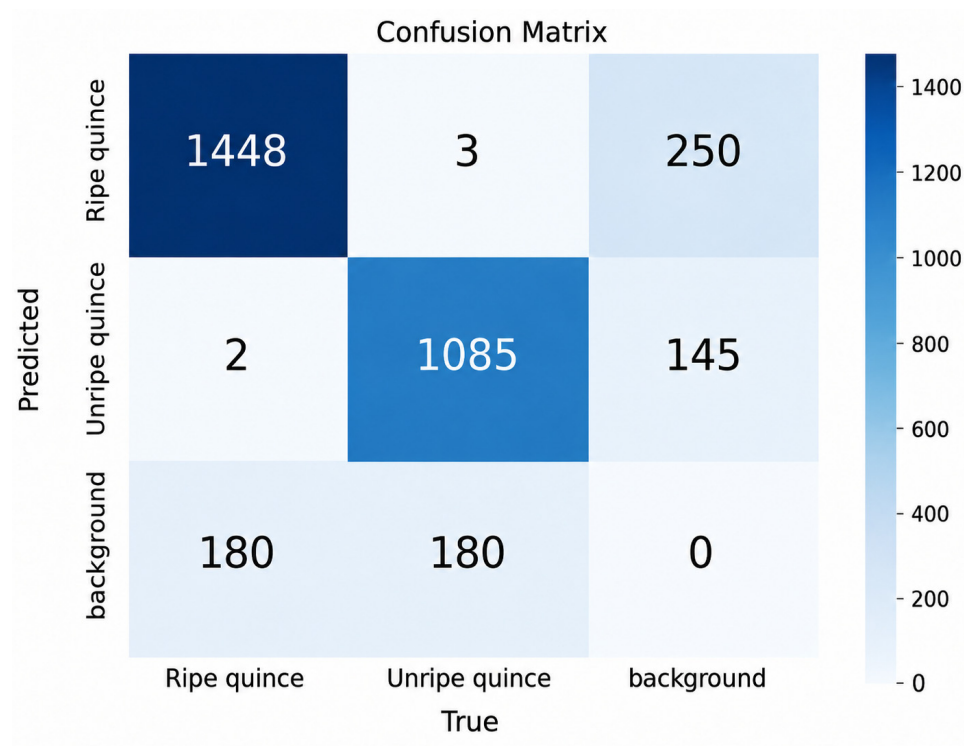
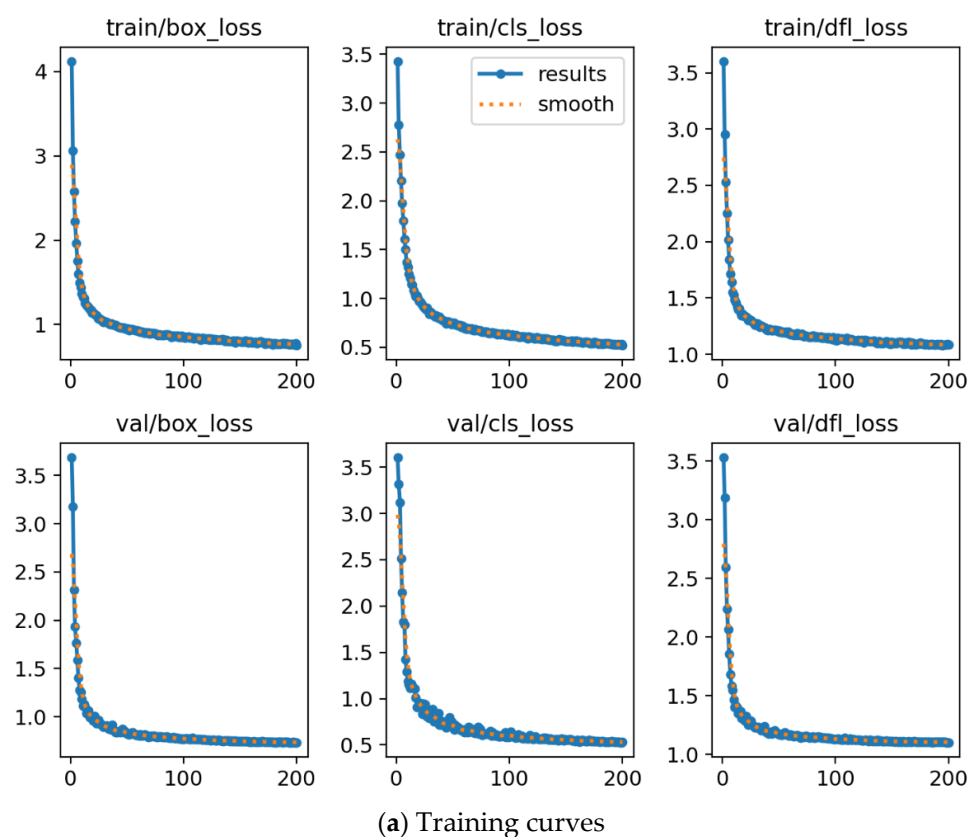


Figure 7. Training curves and confusion matrix of the proposed model.

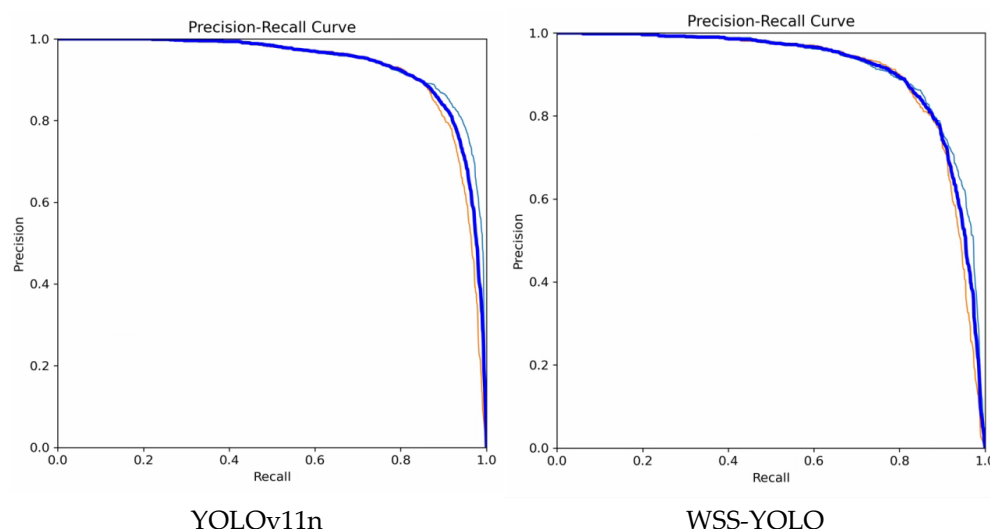


Figure 8. Precision–recall curve comparison.

3.4. Ablation Experiments

Table 2 presents the ablation study conducted on the YOLOv11n baseline to evaluate the individual and combined effects of WaveletPool, Slim-neck, and SimAM. Compared with the baseline model, which achieved 84.1% precision, 85.8% recall, 90.9% mAP50, 74.2% mAP50-95, 2.64 M parameters, and 6.5 G FLOPs, the introduction of different modules resulted in distinct performance changes in terms of detection accuracy and computational efficiency.

Table 2. Ablation experiment.

YOLOv11n	WaveletPool	Slim-Neck	SimAM	P%	R%	mAP50/%	mAP50-95/%	Parameters	FLOPs/G
✓				84.1	85.8	90.9	74.2	2.64	6.5
	✓			85.3	87.1	91.8	74.9	2.23	4.5
✓		✓		85.0	85.2	89.7	74.2	2.58	4.9
✓			✓	85.2	87.4	91.1	74.6	2.64	6.5
✓	✓	✓		85.4	87.2	90.8	73.6	2.23	4.1
✓	✓		✓	85.4	86.9	92.1	75.0	2.23	4.5
✓		✓	✓	85.8	86.7	92.3	75.2	2.58	4.9
✓	✓	✓	✓	86.4	87.5	93.4	76.0	2.23	4.1

When WaveletPool was introduced alone, the model achieved 85.3% precision, 87.1% recall, 91.8% mAP50, and 74.9% mAP50-95, while the number of parameters decreased from 2.64 M to 2.23 M and FLOPs were reduced from 6.5 G to 4.5 G. This result indicates that WaveletPool contributes to improving detection accuracy while reducing model complexity. The reduction in parameters and FLOPs suggests that the module can replace part of the conventional feature-processing operation with a more compact representation, thereby improving computational efficiency.

After introducing Slim-neck alone, FLOPs decreased from 6.5 G to 4.9 G, confirming its effectiveness in reducing computational cost. However, the mAP50 decreased from 90.9% to 89.7%, indicating that excessive feature compression in the neck structure may weaken feature representation to some extent. In contrast, introducing SimAM alone improved recall from 85.8% to 87.4% and increased mAP50 from 90.9% to 91.1% without increasing the number of parameters, suggesting that SimAM enhances feature discrimination in a parameter-free manner.

The intermediate combinations further reveal the interaction among different modules. The combination of WaveletPool and Slim-neck achieved the lowest FLOPs of 4.1 G but

showed a slight decrease in mAP50 and mAP50-95 compared with using WaveletPool alone, which may be attributed to the cumulative effect of feature compression. When SimAM was combined with WaveletPool or Slim-neck, the detection performance improved noticeably, with mAP50 reaching 92.1% and 92.3%, respectively. These results indicate that SimAM can effectively compensate for the potential loss of discriminative information caused by lightweight feature processing.

Finally, the full WSS-YOLO model integrating WaveletPool, Slim-neck, and SimAM achieved the best overall performance, with precision, recall, mAP50, and mAP50-95 reaching 86.4%, 87.5%, 93.4%, and 76.0%, respectively. Meanwhile, the model maintained only 2.23 M parameters and reduced FLOPs to 4.1 G. Compared with the baseline, WSS-YOLO improved precision by 2.3 percentage points, recall by 1.7 percentage points, mAP50 by 2.5 percentage points, and mAP50-95 by 1.8 percentage points, while reducing parameters by 15.5% and FLOPs by 36.9%. These results demonstrate that the three modules are complementary rather than simply additive, enabling WSS-YOLO to achieve a better balance between detection accuracy and computational efficiency.

3.5. Comparison of Different Attention Mechanisms

Table 3 presents the final quantitative results of different attention mechanisms in terms of precision, recall, mAP50, and mAP50-95. SimAM achieved the highest performance among all methods, with precision, recall, mAP50, and mAP50-95 of 86.4%, 87.5%, 93.4%, and 76.0%, respectively. CA and CBAM also performed competitively, while SE, EMA, and GAM showed lower results, particularly in mAP50 and mAP50-95. These results highlight the superior effectiveness of SimAM in enhancing feature representation for accurate detection.

Table 3. Comparison of different attention mechanisms.

Method	P%	R%	mAP50%	mAP50-95/%
CBAM	85.6	86.8	92.2	74.8
CA	86.0	87.1	92.8	75.4
SE	84.5	85.7	90.5	73.2
EMA	85.1	86.3	91.5	74.1
GAM	84.9	86.0	91.2	73.8
SimAM	86.4	87.5	93.4	76.0

Figure 9 provides a comprehensive comparison of mAP50 training curves, where the SimAM mechanism (blue curve) demonstrates better performance than other state-of-the-art methods. From the beginning of training, SimAM exhibited faster convergence and rapidly surpassed 0.8 mAP50 within fewer epochs compared to other methods. Throughout the stabilization phase (Epoch 50–200), SimAM consistently maintained a stable advantage over SE, EMA, and GAM. Even when compared with competitive models such as CA and CBAM, SimAM achieved higher peak accuracy at the end of training. This trajectory indicates that SimAM provides strong feature extraction capability, leading to improved overall detection performance.

3.6. Comparative Experiments

As shown in Table 4, WSS-YOLO demonstrates improved performance across all evaluation metrics. Specifically, it achieves a precision of 86.4%, recall of 87.5%, mAP50 of 93.4%, and mAP50-95 of 76.0%. Compared with YOLOv11n (precision 84.1%, recall 85.8%, mAP50 90.9%), WSS-YOLO shows improvements of 2.3%, 1.7%, and 2.5%, respectively, reflecting better detection accuracy and recall performance.

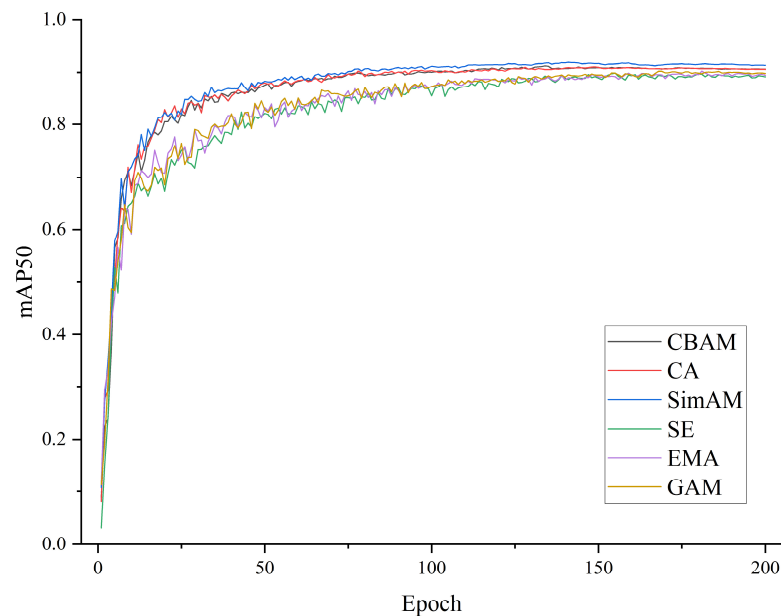


Figure 9. Comparison of mAP50 performance across different attention mechanisms.

Table 4. Comparison experiments with other models.

Model	P%	R%	mAP50%	mAP50-95/%	Parameters	Weight/MB	FLOPs/G
SSD	78.6	80.0	86.3	71.0	14.3	48.1	15.10
EfficientDet	61.5	62.4	71.8	65.3	3.9	15.3	5.23
YOLOv3	73.7	74.4	83.1	70.3	61.5	123.6	154.6
YOLOv3-Tiny	77.1	78.3	85.9	70.9	8.67	17.5	12.90
YOLOv5n	79.9	80.7	87.9	71.6	1.78	3.9	4.1
YOLOv5s	80.3	84.3	88.4	71.8	7.03	14.5	15.8
YOLOv7-Tiny	71.4	79.2	87.3	71.6	6.02	12.3	13.2
YOLOv8n	82.1	83.6	90.0	73.3	3.01	6.3	8.2
YOLOv10n	82.5	79.8	89.6	72.9	2.70	5.8	8.4
YOLOv12n	83.1	85.1	90.4	73.9	2.52	5.4	6.0
YOLOv11n + ShuffleNetV2	75.20	77.50	84.50	70.40	2.96	1.85	3.4
YOLOv11n + MobileNetV3	76.30	76.30	84.70	70.20	8.18	4.44	3.8
YOLOv11n + MobileNetV4	79.40	78.00	85.20	70.50	5.93	3.31	3.6
YOLOv11n	84.1	85.8	90.9	74.2	2.64	5.5	6.5
WSS-YOLO	86.4	87.5	93.4	76.0	2.23	4.7	4.1

In addition, when compared to YOLOv11n equipped with lightweight backbones, WSS-YOLO achieves better performance in terms of both accuracy and overall detection capability. For example, YOLOv11n + ShuffleNetV2 achieves a precision of 75.2%, recall of 77.5%, and mAP50 of 84.5%, while YOLOv11n + MobileNetV3 and YOLOv11n + MobileNetV4 achieve precision of 76.3% and 79.4%, recall of 76.3% and 78.0%, and mAP50 of 84.7% and 85.2%, respectively. This comparison indicates that although lightweight backbones reduce model parameters and computational cost, they may sacrifice detection accuracy, whereas WSS-YOLO achieves a favorable trade-off between performance and efficiency.

From the perspective of model complexity, WSS-YOLO exhibits both lightweight design and practical deployability. Its parameter count is only 2.23 M, with 4.1 G FLOPs and a model weight of 4.7 MB, which is significantly lower than YOLOv3 (61.5 M parameters, 154.6 G FLOPs) and even more efficient than other lightweight models such as YOLOv5s (7.03 M parameters, 15.8 G FLOPs). Therefore, WSS-YOLO not only achieves strong

detection performance but also maintains low computational cost and a compact model size, suggesting its potential for real-time applications on resource-constrained devices.

3.7. Model Detection and Heatmap Visualization Results

In this study, the WSS-YOLO model demonstrates significant advantages over YOLOv11n in fruit detection tasks as shown in Figure 10. First, in terms of detection accuracy, WSS-YOLO performs more consistently, particularly in recognizing fruits at different maturity stages, with generally higher confidence scores. Specifically, WSS-YOLO not only achieves high confidence for ripe fruits (up to 0.97) but also performs excellently in detecting unripe fruits (up to 0.96). Compared to YOLOv11n's confidence scores (as low as 0.59 for unripe fruits), WSS-YOLO exhibits more balanced performance when handling fruits at different maturity stages, indicating more stable behavior in the presented examples.



Figure 10. Heatmap visualization results.

In terms of accuracy of localization, WSS-YOLO shows higher precision in fruit bounding. Even in scenarios with densely packed fruits, complex backgrounds, or overlapping fruits, WSS-YOLO can more accurately bound each fruit, reducing the misidentification and bounding box deviations that may occur with the YOLOv11n model in these scenarios. Particularly when the spacing between fruits is small or the background is cluttered, WSS-YOLO shows distinct advantages, ensuring more precise fruit localization.

Through heatmap visualization results as shown in Figure 10, it can be further observed that WSS-YOLO performs exceptionally well when handling backgrounds with densely packed fruits. The heatmaps show that WSS-YOLO can generate more prominent focus regions in fruit areas, indicating that the model can more effectively focus on fruits and reduce background interference. In contrast, YOLOv11n displays more dispersed attention regions, especially in cases of unripe fruits and small inter-fruit spacing, where its focusing capability is weaker, resulting in some fruits not being accurately identified.

In terms of performance under varying illumination, fruit overlap, and complex backgrounds, WSS-YOLO performs better than YOLOv11n. Even under conditions of varying illumination, fruit overlap, or complex backgrounds, WSS-YOLO maintains high detection accuracy and localization precision, with significantly better detection accuracy for unripe fruits than YOLOv11n. YOLOv11n, on the other hand, exhibits relatively unstable performance when handling these complex environments, especially in cases of overlapping fruits or complex backgrounds.

4. Discussion

The proposed design provides a useful reference for lightweight fruit detection. By combining WaveletPool, Slim-neck, and SimAM, the model improves feature preservation and background suppression while keeping computational cost low. WaveletPool helps retain important structural information during downsampling, Slim-neck reduces redundant computation in feature fusion, and SimAM strengthens discriminative fruit-region responses without adding trainable parameters. These characteristics make WSS-YOLO suitable for real-time agricultural applications on resource-constrained devices.

Second, targeting different task characteristics, the multi-module fusion design proposed in this study can effectively avoid the introduction of redundant mechanisms, thereby reducing the waste of computational resources. By optimizing the network architecture to achieve multi-level information fusion, information bottleneck issues are avoided, which greatly improves the model's performance in complex orchard environments. Especially when facing challenges such as illumination variations, occlusion, and fine-grained fruit features, the model can still maintain high-precision detection, showing that the model can maintain accurate detection under illumination variation, occlusion, and fine-grained fruit appearance differences.

Beyond algorithmic optimization, this study further evaluates the edge deployment capability of the proposed WSS-YOLO model. To assess its practical applicability in situ agricultural scenarios, the model was deployed on an NVIDIA Jetson Orin Nano, as shown in Figure 11 and Table 5. During deployment, the input image resolution was set to 640×640 , which was consistent with the validation setting used in the accuracy evaluation. The reported speed of 23.0 FPS refers to model inference performance on the edge device and includes network forward propagation and post-processing, including non-maximum suppression. Image acquisition, data loading, visualization, and result saving were not included in the FPS calculation. No additional TensorRT acceleration, FP16 inference, or INT8 quantization was applied in this experiment. Under these deployment settings, WSS-YOLO maintained real-time inference capability on resource-constrained hardware,

suggesting its feasibility for continuous visual monitoring and intelligent agricultural detection tasks.

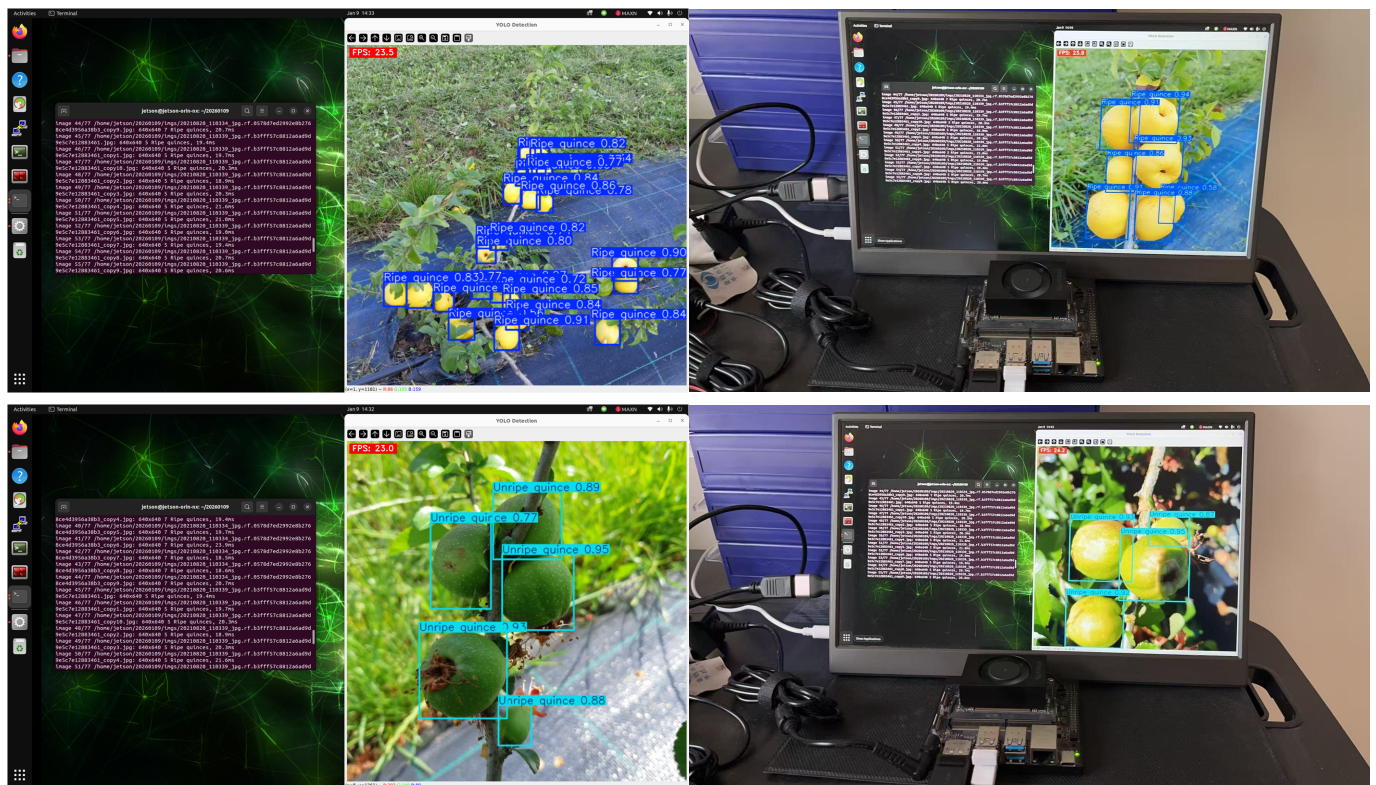


Figure 11. Real-time deployment of the model on NVIDIA Jetson Orin Nano.

Table 5. Device specifications of Jetson Orin Nano.

Item	Specification
Device	Jetson Orin Nano 8 GB
CPU	6-core Arm Cortex-A78AE v8.2 64-bit CPU, 1.5 MB L2 + 4 MB L3
GPU	NVIDIA Ampere architecture GPU with 1024 CUDA cores and 32 Tensor Cores
Memory	8 GB 128-bit LPDDR5, 102 GB/s memory bandwidth
Storage	128 GB SSD
Power consumption	7 W–15 W
Input size	640 × 640 px, consistent between training and deployment
Deployment optimization	No TensorRT, FP16, or INT8 acceleration was applied; native inference was used
Evaluation metrics	Model size, parameters, GFLOPs, FPS, inference latency, and power consumption

Furthermore, although the proposed WSS-YOLO model achieved promising detection performance, several limitations should be acknowledged. First, the experiments in this study were conducted on a single publicly available dataset collected using a smartphone camera. Although the dataset contains variations in shooting distance, illumination, occlusion, fruit overlap, and background complexity, the images are still derived from a limited orchard scenario. Therefore, the generalizability discussed in this study mainly refers to the environmental variations covered by the current dataset, rather than full transferability across different orchards, acquisition dates, seasons, or weather conditions. In addition, the training, validation, and test sets were constructed using a holdout split from the same dataset. Since all subsets originate from the same data source, similar background patterns or scene characteristics may exist across different subsets. In the current study, independent cross-orchard, cross-day, or cross-weather validation was not conducted. This

limitation may lead to a slightly optimistic estimation of the model's performance in unseen orchard environments.

The model may still face challenges under severe occlusion, dense fruit overlap, extreme illumination, or when fruit color is highly similar to the background. These factors may cause missed detections or inaccurate localization. Therefore, future work will focus on introducing multi-source datasets collected from different orchards, dates, weather conditions, and imaging devices. More rigorous cross-scene and cross-day validation will also be conducted to further evaluate the generalization ability of the proposed model in truly complex agricultural scenarios. In addition, more advanced data augmentation strategies and refined feature extraction methods will be explored to improve detection stability under extreme field conditions.

5. Conclusions

This study proposed WSS-YOLO, a lightweight fruit maturity detection model based on YOLOv11n for quince detection in complex orchard environments. The model integrates WaveletPool, a GSConv-based Slim-neck, and SimAM to improve feature preservation, reduce computational cost, and enhance discriminative responses to fruit regions. These components allow the model to better handle texture loss, background interference, and partial occlusion while maintaining a compact network structure.

Systematic experiments based on a multi-scenario quince maturity dataset showed that WSS-YOLO achieved 86.4% precision, 87.5% recall, and 93.4% mAP@0.5, exceeding the baseline YOLOv11n by 2.3%, 1.7%, and 2.5%, respectively. Heatmap visualization analysis further indicates that the proposed model improves localization accuracy of fruit targets and reduces background interference in natural environments.

Moreover, while achieving performance improvements, the model reduces computational costs. WSS-YOLO has only 2.23 M parameters, with floating-point operations (FLOPs) reduced to 4.1 G and a weight file size of only 4.7 MB, showing better overall performance compared with mainstream lightweight networks such as YOLOv8 and YOLOv5. This lightweight design enables the model to meet the requirements of real-time performance and high precision for agricultural harvesting robots and handheld mobile terminals, providing a feasible approach for non-destructive fruit detection in smart agriculture scenarios. Future work will focus on deploying this algorithm on embedded hardware platforms and further evaluating its long-term stability and generalization performance in actual harvesting operations.

Overall, the proposed WSS-YOLO demonstrates promising performance for lightweight fruit maturity detection in orchard environments. Future work will explore its application in more diverse orchard scenarios and broader agricultural environments to further evaluate its generalizability and practical potential.

Author Contributions: Conceptualization, X.W. and H.W.; methodology, X.W.; software, X.W.; validation, X.W., J.Z. and H.W.; formal analysis, X.W.; investigation, X.W.; resources, H.W.; data curation, X.W. and J.Z.; writing—original draft preparation, X.W.; writing—review and editing, X.W., J.Z. and H.W.; visualization, X.W.; supervision, H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Jilin Provincial Department of Education Research Project (Grant No. 2025199S1S301IX). The authors would like to thank the reviewers for their constructive comments and suggestions that helped improve the quality of this manuscript.

Data Availability Statement: The dataset used in this study can be freely downloaded from Zenodo at <https://zenodo.org/records/6402251> (accessed on 12 December 2025).

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

References

1. Gongal, A.; Amatya, S.; Karkee, M.; Zhang, Q.; Lewis, K. Sensors and systems for fruit detection and localization: A review. *Comput. Electron. Agric.* **2015**, *116*, 8–19. [\[CrossRef\]](#)
2. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [\[CrossRef\]](#)
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [\[CrossRef\]](#)
5. Lawal, M.O. Tomato detection based on modified YOLOv3 framework. *Sci. Rep.* **2021**, *11*, 1447. [\[CrossRef\]](#) [\[PubMed\]](#)
6. 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. [\[CrossRef\]](#)
7. Khanam, R.; Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
8. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [\[CrossRef\]](#)
9. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [\[CrossRef\]](#)
10. Hao, F.; Zhang, Z.; Ma, D.; Kong, H. GSBF-YOLO: A lightweight model for tomato ripeness detection in natural environments. *J. Real-Time Image Process.* **2025**, *22*, 47. [\[CrossRef\]](#)
11. Li, Z.; Hu, X.; Zhao, X.; Ye, H.; Chen, F.; Chen, X.; Li, X. Beyond obstacles: Feather-light YOLO11-LES for real-time ripeness detection of occluded strawberries in greenhouses. *J. Real-Time Image Process.* **2025**, *22*, 172. [\[CrossRef\]](#)
12. Chen, X.; Wang, G. PL-YOLO: A lightweight method for real-time detection of pomegranates. *J. Real-Time Image Process.* **2025**, *22*, 148. [\[CrossRef\]](#)
13. Ji, C.-L.; Yu, T.; Gao, P.; Wang, F.; Yuan, R.-Y. Yolo-tla: An efficient and lightweight small object detection model based on YOLOv5. *J. Real-Time Image Process.* **2024**, *21*, 141. [\[CrossRef\]](#)
14. Williams, T.; Li, R. Wavelet pooling for convolutional neural networks. In *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, 30 April–3 May 2018.
15. Liu, P.; Zhang, H.; Lian, W.; Zuo, W. Multi-level wavelet convolutional neural networks. *IEEE Access* **2019**, *7*, 74973–74985. [\[CrossRef\]](#)
16. de Souza Brito, A.; Vieira, M.B.; De Andrade, M.L.S.C.; Feitosa, R.Q.; Giraldo, G.A. Combining max-pooling and wavelet pooling strategies for semantic image segmentation. *Expert Syst. Appl.* **2021**, *183*, 115403. [\[CrossRef\]](#)
17. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*; ML Research Press: Norfolk, MA, USA, 2021; pp. 11863–11874. Available online: <https://proceedings.mlr.press/v139/yang21o.html> (accessed on 12 December 2025).
18. 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. [\[CrossRef\]](#)
19. 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. [\[CrossRef\]](#)
20. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *J. Real-Time Image Process.* **2024**, *21*, 62. [\[CrossRef\]](#)
21. Ding, P.; Li, T.; Qian, H.; Ma, L.; Chen, Z. A lightweight real-time object detection method for complex scenes based on YOLOv4. *J. Real-Time Image Process.* **2025**, *22*, 68. [\[CrossRef\]](#)
22. Chen, J.; Zhang, X.; Peng, X.; Xu, D.; Wu, D.; Xin, R. Shuffle-octave-yolo: A tradeoff object detection method for embedded devices. *J. Real-Time Image Process.* **2023**, *20*, 25. [\[CrossRef\]](#)
23. Kaufmane, E.; Sudars, K.; Namatēvs, I.; Kalniņa, I.; Judvaitis, J.; Balašs, R.; Strautiņa, S. QuinceSet: Dataset of annotated Japanese quince images for object detection. *Data Brief* **2022**, *42*, 108332. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Xu, Y.; Li, H.; Zhou, Y.; Zhai, Y.; Yang, Y.; Fu, D. GLL-YOLO: A Lightweight Network for Detecting the Maturity of Blueberry Fruits. *Agriculture* **2025**, *15*, 1877. [\[CrossRef\]](#)
25. Lin, X.; Liao, D.; Du, Z.; Wen, B.; Wu, Z.; Tu, X. SDA-YOLO: An object detection method for peach fruits in complex orchard environments. *Sensors* **2025**, *25*, 4457. [\[CrossRef\]](#) [\[PubMed\]](#)

26. Wang, J.; Shang, Y.; Zheng, X.; Zhou, P.; Li, S.; Wang, H. GreenFruitDetector: Lightweight green fruit detector in orchard environment. *PLoS ONE* **2024**, *19*, e0312164. [[CrossRef](#)] [[PubMed](#)]
27. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755. [[CrossRef](#)]

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