agriculture

MDPI

Article

A Feature Enhancement Framework for Joint Mango Fruit and Stem Detection in Complex Orchard Environments

Jiahuan Lu [1,2], Qihan Deng [1,2], Weiping Zheng [1,2], Binglong Cai [1,2], Shan Zeng [1,2] and Jiehao Li [1,2,*]

[1] College of Engineering, South China Agricultural University, Guangzhou 510642, China; jhlu@scau.edu.cn (J.L.); shanzeng@scau.edu.cn (S.Z.)
[2] National Key Laboratory of Agricultural Equipment Technology, South China Agricultural University, Guangzhou 510642, China
* Correspondence: jiehao.li@scau.edu.cn

Abstract

Reliable joint detection of mango fruits and stems is an essential upstream perception task for robotic harvesting, but remains challenging because stems are small, slender, frequently occluded, and visually degraded by illumination variation. This study proposes MangoNET, a YOLOv11n-based framework for joint mango fruit and stem detection in complex orchard environments. A P2 high-resolution detection head preserves fine spatial information for small targets, while SPPF-ELAN aggregates local and contextual features for partially visible objects. SENet recalibrates channel responses under illumination variation, and WIoU v3 regulates bounding-box samples with different localization qualities. A dataset containing 1782 original images of Tainong and Jinhuang mangoes was collected from two orchards and data augmentation was applied only to the training set, increasing its size from 1172 to 2886 images through rotation, contrast adjustment, and Gaussian noise addition. MangoNET achieved fruit and stem F1-scores of 0.920 and 0.916, respectively, with mAP50 and mAP50–95 values of 0.941 and 0.690. Compared with YOLOv11n, mAP50 and mAP50–95 increased by 1.6 and 2.9 percentage points, respectively, while stem recall increased from 0.877 to 0.906. Source-image-independent five-fold cross-validation yielded mean mAP50 and mAP50–95 values of 0.944 and 0.711, respectively. Pilot evaluations using images acquired by a UAV and an RGB-D camera in a geographically distinct orchard suggested that MangoNET could maintain detection performance in a different orchard environment. MangoNET supplies fruit and stem candidate regions for subsequent association, harvesting-point localization, and robotic manipulation.

Keywords: mango; stem detection; small-object detection; YOLOv11; agricultural robotics

Check for updates

Academic Editor: Francesco Marinello

Received: 30 July 2026
Revised: 27 August 2026
Accepted: 28 August 2026
Published: 31 August 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.

1. Introduction

Mango is one of the most widely cultivated and consumed tropical fruits because of its distinctive flavor, aroma, and nutritional value [1]. China is one of the world's major mango-producing countries, with a reported planting area of 5.619 million mu in 2021 [2]. In natural orchards, however, mango fruits are distributed irregularly within dense canopies and are frequently obscured by leaves, branches, or neighboring fruits [3]. Their visual appearance is also strongly affected by backlighting, low illumination, and locally overexposed backgrounds. These conditions are particularly challenging for mango stems, which are slender, visually inconspicuous, and often only partially visible. Machine vision provides rich environmental information at relatively low hardware cost and has therefore

become an important perception technology for automated agricultural operations [4]. For robotic mango harvesting, reliable joint detection of fruits and stems is an essential upstream task because it provides candidate regions for subsequent fruit–stem association, harvesting-point localization, three-dimensional positioning, and manipulator planning.

Deep-learning-based object detectors have substantially advanced fruit perception in complex agricultural environments. An improved YOLOv3 model combined with binocular stereo vision was developed for pineapple detection and three-dimensional localization under natural conditions [5]. A lightweight YOLOv4-tiny-based method was subsequently introduced for detecting Camellia oleifera fruits [6]. More recent studies have improved YOLOv8 for immature yellow-peach detection [7] and YOLOv4 for cherry detection in complex orchard scenes [8]. Beyond conventional fruit localization, a multitask YOLOv7-based network was developed to simultaneously perform cherry-tomato localization, ripeness grading, and cluster maturity assessment [9]. Other improvements have focused on blueberry detection using YOLOv8 [10], robust detection of Camellia oleifera fruits using YOLOv5s [11], lightweight pitaya detection under daytime and nighttime supplementary illumination [12], recognition of small and overlapping strawberry flowers and fruits [13], and real-time pineapple detection using YOLOv7-tiny [14]. Collectively, these studies demonstrate that architectural refinement, multiscale feature fusion, attention mechanisms, and lightweight design can improve fruit detection under complex field conditions. Nevertheless, most of these methods primarily focus on the fruit itself, whereas robotic harvesting of clustered or stem-connected fruits also requires reliable perception of harvesting-related structures.

Accordingly, increasing attention has been directed toward the joint perception of fruits, stems, and harvesting points. A lightweight YOLO-GP model was developed for grape and harvesting-point detection, achieving a harvesting-point localization error of less than 40 pixels [15]. A two-stage framework combining YOLOv3 and U-Net was used for lychee detection and stem segmentation under nighttime conditions [16]. For tomato harvesting, YOLOv5 was integrated with image-processing procedures to localize harvesting positions under multiple illumination conditions [17]. A lychee perception framework further combined fruit and stem detection, semantic stem segmentation, and skeleton extraction to determine harvesting points [18]. To better represent the elongated geometry of grape stems, oriented bounding boxes were incorporated into a YOLO11-based detector for harvesting-point localization and pose estimation [19]. Lightweight joint detection of banana clusters and stems was also achieved using an improved YOLOv4 framework [20], while a YOLOv8-pose-based method directly detected strawberries and their picking keypoints [21]. These studies employ different visual representations according to downstream requirements. Segmentation, oriented bounding boxes, skeletons, and keypoints can provide detailed geometric information for cutting-point or pose estimation, whereas horizontal bounding-box detection offers a relatively direct means of generating fruit and stem candidate regions. The present study focuses on this upstream candidate-detection stage rather than replacing subsequent fruit–stem association, stem segmentation, or harvesting-point localization.

Mango-specific perception methods have also progressed from fruit-only detection toward joint fruit–stem analysis. A multitask network based on YOLOv5s was developed to simultaneously detect mango fruits and stems, segment stem regions, and extract stem skeletons for harvesting-point localization [22]. Another framework combined an improved YOLOv8 detector with an instance-segmentation model and skeleton extraction to obtain mango harvesting points [23]. An improved YOLOv8-based joint detector was further deployed on an NVIDIA Jetson Orin Nano edge device, demonstrating the feasibility of lightweight mango fruit and stem detection [2]. These studies provide important

foundations for mango-harvesting perception. However, the reliable generation of fruit and stem candidate regions remains difficult when small stem size, partial occlusion, and illumination-induced feature degradation occur simultaneously. In particular, repeated downsampling may weaken the spatial information of slender stems, occlusion may remove critical local features, and uneven illumination may disturb the relative responses of feature channels. These coupled factors can result in missed stem detections, redundant bounding boxes, and confusion between stems, leaves, and branches. Therefore, a problem-oriented joint detector is required to improve the representation and localization of difficult mango fruit and stem targets under natural orchard conditions.

To address these challenges, this study develops MangoNET, a joint mango fruit and stem detection framework based on YOLOv11n. The main contributions are summarized as follows.

(1) A joint detection task is formulated for mango fruits and stems under natural orchard conditions, with particular attention to the coupled effects of small stem size, partial occlusion, and illumination variation. The resulting bounding boxes are intended to provide candidate regions for subsequent fruit–stem association and harvesting-point localization rather than directly representing complete stem geometry.
(2) MangoNET introduces a problem-oriented combination of network modifications. A P2 detection head preserves high-resolution spatial information for small stems and distant fruits, while an SPPF-ELAN module aggregates local and contextual features to improve the representation of partially visible targets. Channel recalibration is incorporated to alleviate feature-response imbalance under varying illumination, and WIoU v3 is employed to reduce the adverse influence of low-quality bounding-box samples during model optimization.
(3) A dedicated dataset containing two mango cultivars and diverse orchard conditions is constructed, and the proposed framework is evaluated through attention-module comparison, component ablation, five-fold cross-validation, comparison with representative object detectors, and pilot cross-site field experiments using image streams acquired by an unmanned aerial vehicle and an RGB-D camera. These evaluations examine the detection accuracy, stability, and application potential of MangoNET as an upstream perception component for robotic mango harvesting.

2. Materials and Methods

MangoNET was developed for joint detection of mango fruits and stems under natural orchard conditions. Figure 1 provides a system-level overview of the proposed framework and the pilot field-evaluation setup. The upper panel illustrates the input challenges, joint detection process, and detection outputs, whereas the lower panel presents the equipment configuration and data flow used in the field experiments.

2.1. Data Acquisition and Annotation

To evaluate joint mango fruit and stem detection under natural orchard conditions, an image dataset containing Tainong and Jinhuang mango cultivars was constructed. Images of Tainong mangoes were collected on 7 December 2024, in an orchard located in Zhongyang Village, Jiyang District, Sanya, Hainan Province, China (109°55′ E, 18°34′ N). Images of Jinhuang mangoes were collected on 8 December 2024, at Fuhua Farm in Tianya District, Sanya, Hainan Province, China (109°37′ E, 18°37′ N). The acquisition sites, cultivars, devices, image resolutions, and image quantities are summarized in Table 1.

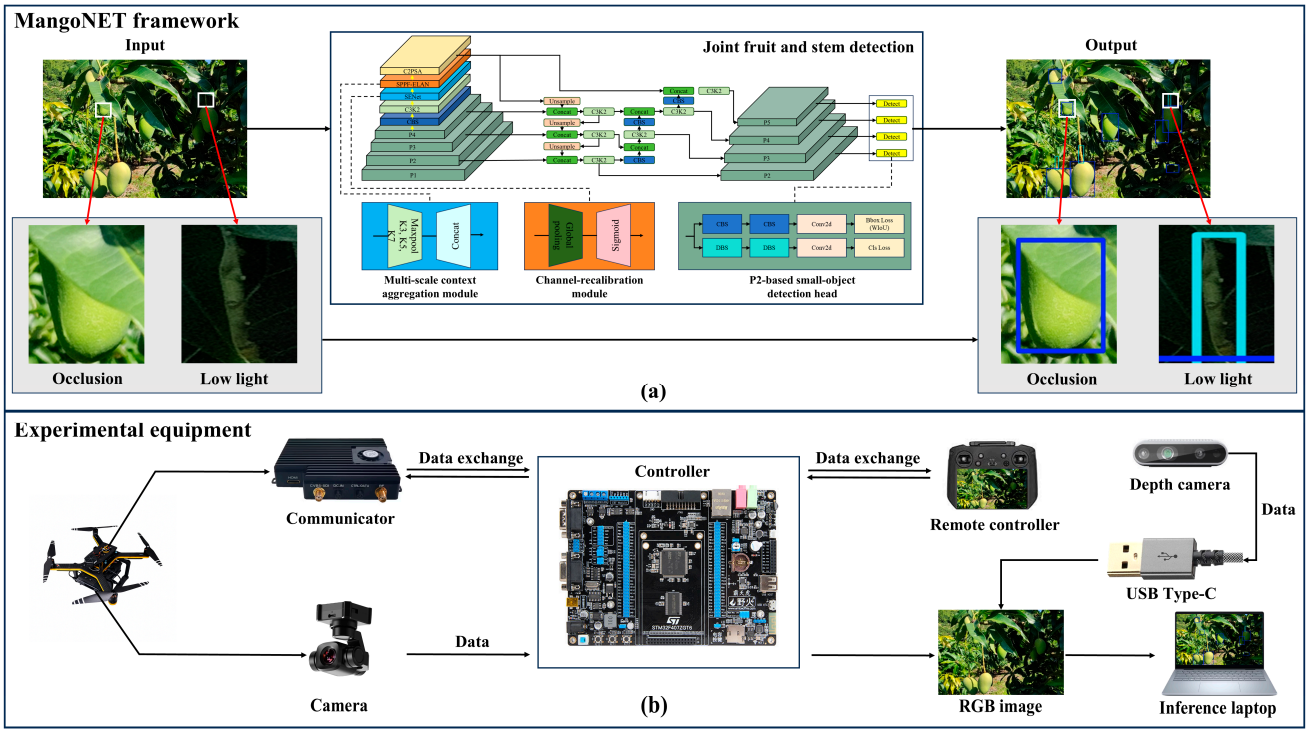


Figure 1. Overview of the proposed MangoNET framework and pilot field-evaluation setup: (**a**) joint mango fruit and stem detection framework; (**b**) equipment configuration and data flow.

Table 1. Details of mango image acquisition.

	Image Collection 1	Image Collection 2
Collection site	Zhongyang Village, Jiyang District, Sanya	Fuhua Farm in Tianya District, Sanya
Mango cultivar	Tainong	Jinhuang
Acquisition date	7 December 2024	8 December 2024
Weather	Sunny	Sunny
Acquisition devices	Xiaomi 14 and Redmi K40 Pro	Xiaomi 14 and Redmi K40 Pro
Image resolutions	3072 × 4096, 2464 × 3280, 4624 × 3472	3072 × 4096, 2464 × 3280, 4624 × 3472
Number of images corresponding to the above resolutions	644, 101, 146	528, 87, 276

Image acquisition was conducted between 08:00 and 17:30 on each collection day using Xiaomi 14 and Redmi K40 Pro smartphones (manufactured by Xiaomi Corporation, Beijing, China). Images were captured from top, frontal, and upward-facing viewpoints to cover the variable target appearances encountered during orchard observation and robotic harvesting. The collected scenes included normally illuminated, backlit, and low-light conditions, together with fruit–fruit overlap, leaf occlusion, stem occlusion, and branch interference. A total of 1782 original mango images were collected.

The images were manually annotated using LabelImg with two object categories: mango fruit and mango stem. Each fruit and stem instance was independently enclosed using a horizontal bounding box. For a physically connected fruit–stem pair, the two bounding boxes could naturally overlap near the attachment region. However, such spatial

overlap was not treated as an explicit one-to-one association label, and the present detector does not directly infer which stem belongs to a specific fruit in densely clustered scenes.

According to the number and spatial distribution of visible fruits, the collected images were further described as single-fruit or multi-fruit scenes. Single-fruit scenes contained one visible mango, whereas multi-fruit scenes contained clustered fruits or fruits distributed across multiple image regions. Both groups included representative illumination and occlusion conditions, as shown in Figure 2. Figure 3 further illustrates the fruit-bearing characteristics of mango trees and the principal visual challenges arising from dense foliage, overlapping fruits, slender stems, and branch interference.

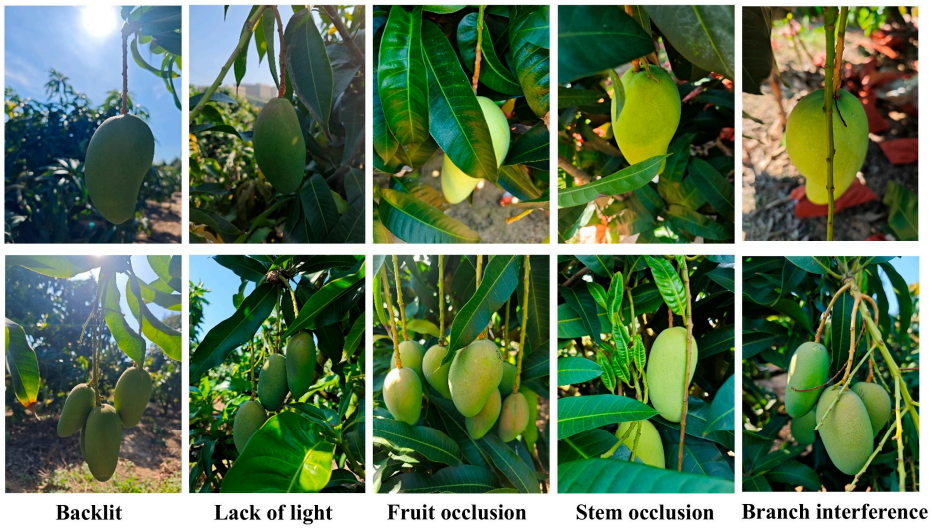


Figure 2. Representative mango fruit and stem scenes under natural orchard conditions.

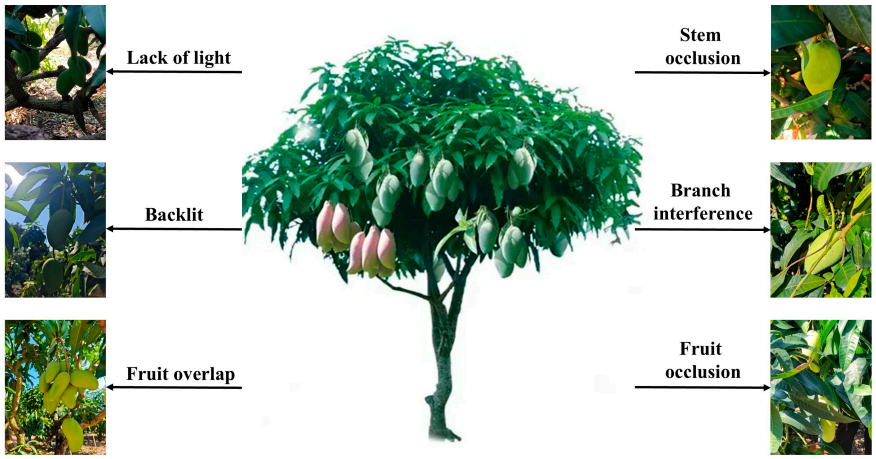


Figure 3. Schematic summary of the principal visual challenges in mango fruit and stem detection.

2.2. Data Augmentation and Dataset Partitioning

To increase the appearance diversity of mango fruits and stems under natural orchard conditions, the 1782 original images were first divided into training, validation, and test sets at a ratio of approximately 6.5:1.5:2. The exact composition of the training, validation, and test sets in terms of original images, cultivars, and orchard sources is provided in Table 2, while the corresponding acquisition conditions are detailed in Section 2.1 and Table 1. Data augmentation was then applied only to the training set using the OpenCV toolkit. The augmentation operations included image rotation, contrast adjustment, and Gaussian-noise addition. The bounding-box coordinates were transformed consistently

with the corresponding augmented images. After augmentation, the training set was expanded from 1172 to 2886 images, whereas the validation and test sets were retained as the original unaugmented images. Unless otherwise specified, all subsequent experiments were conducted using this predefined dataset partition. A separate robustness experiment involving motion blur was further conducted, as described in Section 3.7.

Table 2. Composition of original images in each dataset split by mango cultivar and orchard.

Dataset	Original Images	Tainong (Zhongyang Village)	Jinhuang (Fuhua Farm)
Training set	1172	596	576
Validation set	245	126	119
Test set	365	169	196
Total	1782	891	891

2.3. Baseline YOLOv11n Network

YOLOv11 is a one-stage object detector that performs feature extraction, multi-scale feature fusion, and bounding-box prediction within a unified architecture [24]. Its compact variants provide a favorable balance between detection accuracy and computational complexity. YOLOv11n was therefore selected as the baseline model in this study.

The baseline network extracts hierarchical features through its backbone and transfers the P3, P4, and P5 feature maps to the neck for bidirectional multi-scale fusion. The resulting feature maps are then delivered to three detection heads corresponding to different target scales. This architecture is effective for general object detection but presents several limitations for joint mango fruit and stem detection.

First, mango stems generally occupy smaller and narrower image regions than fruits. Their spatial details may therefore be weakened after repeated downsampling, particularly when stems are distant or partially visible. Second, when a target is occluded by leaves, branches, or neighboring fruits, local target information becomes incomplete, increasing the importance of contextual information. Third, backlighting and low illumination may alter feature responses and reduce the distinction between stems and the surrounding canopy. Finally, ambiguous target boundaries under these conditions can produce low-quality bounding-box samples during training. MangoNET was developed to alleviate these limitations through problem-oriented modifications to the YOLOv11n architecture.

2.4. Problem-Oriented Improvements to YOLOv11n

The detailed network architecture of the joint detector shown in the upper panel of Figure 1 is presented in Figure 4. Compared with the baseline YOLOv11n, MangoNET introduces four problem-oriented modifications while retaining the backbone–neck–head organization and one-stage detection paradigm:

(1) A channel-recalibration module was incorporated into the backbone to adaptively adjust channel responses under illumination-varying orchard scenes.
(2) An SPPF-ELAN module was introduced to strengthen multi-scale contextual aggregation for partially occluded fruits and stems.
(3) The original CIoU bounding-box regression loss was replaced with WIoU v3 to regulate the contribution of bounding-box samples with different localization qualities.
(4) A P2 detection head was added to preserve high-resolution spatial information for small stems and distant fruits.

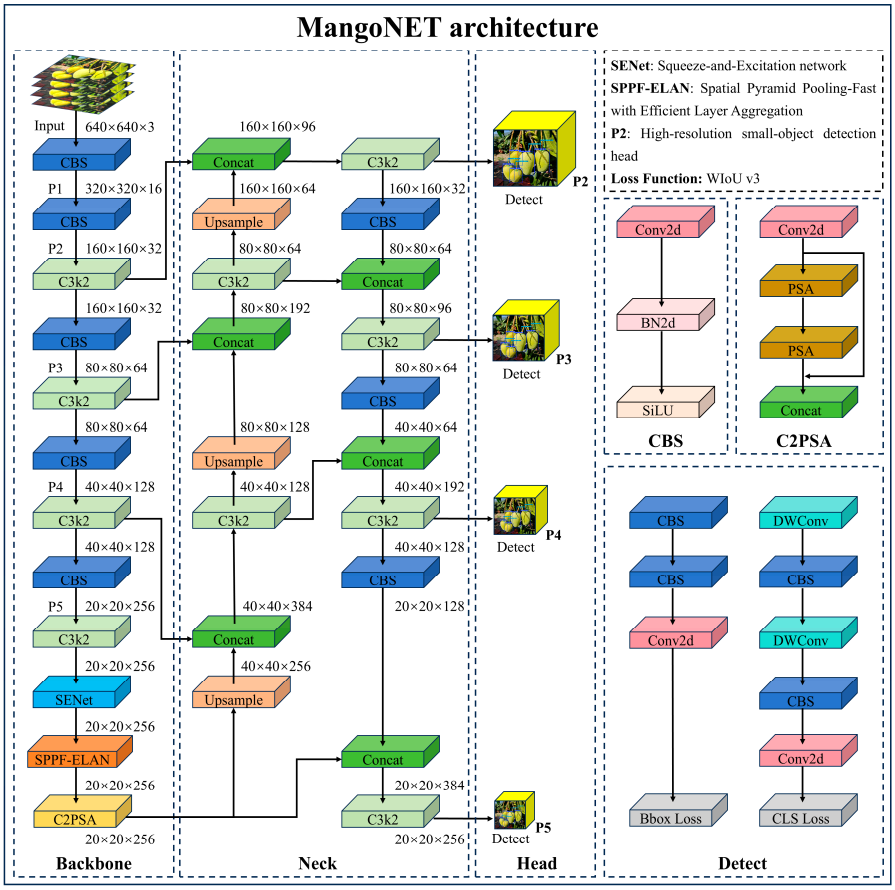


Figure 4. Detailed architecture of the joint mango fruit and stem detector shown in Figure 1.

These components address complementary difficulties rather than representing independent detection tasks. In particular, the channel-recalibration and SPPF-ELAN modules do not explicitly estimate illumination levels or occlusion ratios. Instead, they improve feature representation under conditions in which illumination variation and partial occlusion degrade the original feature responses.

2.4.1. Channel-Recalibration Module for Illumination-Varying Scenes

Illumination variation can substantially change the magnitude and distribution of feature responses extracted from orchard images. Under backlighting, strong background responses may weaken the relative prominence of fruit and stem features. Under low-light conditions, the responses associated with slender stem boundaries may become less distinguishable from the surrounding canopy. The baseline YOLOv11n network does not explicitly recalibrate the relative importance of individual feature channels according to the input scene.

To alleviate this limitation, an SENet module [25] was incorporated into the backbone as a channel-recalibration component. The module does not directly estimate scene illumination or perform image-level exposure correction. Instead, it learns channel-wise weights from global feature statistics and uses these weights to strengthen informative responses while suppressing less useful ones. Its structure is shown in Figure 5 and consists of squeeze, excitation, and scaling operations. The SENet module was inserted after the final C3k2 block of the backbone and before the SPPF-ELAN module.

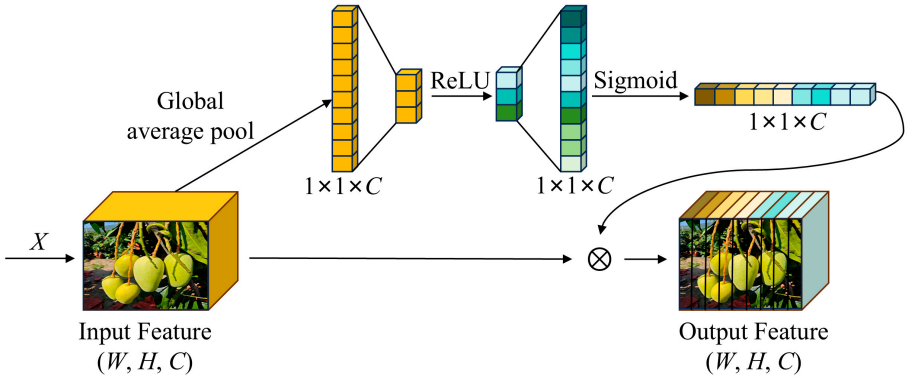


Figure 5. Structure of the SENet-based channel-recalibration module.

In the squeeze stage, an input feature map ($\boldsymbol{X} \in \boldsymbol{R}^{W\times H\times C}$) is compressed along its spatial dimensions using global average pooling. Each channel is represented by a scalar descriptor that summarizes its overall activation:

$$z_c = F_{\text{Squeeze}}(X_c) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} X_c(i,j) \tag{1}$$

where $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_c] \in \boldsymbol{R}^{W\times H\times C}$ denotes the input feature map; W, H, and C are its width, height, and number of channels, respectively; $\boldsymbol{X}_c \in \boldsymbol{R}^{W\times H}$ is the c-th channel; $\boldsymbol{X}_c(i, j)$ denotes the activation at spatial position (i, j) of the c-th channel; and z_c is the global descriptor obtained for that channel. Collecting the descriptors of all channels gives $\boldsymbol{Z} = [z_1, z_2, \ldots, z_C] \in \boldsymbol{R}^{C}$.

In the excitation stage, nonlinear dependencies among channels are learned using two fully connected layers. The first layer reduces the descriptor dimension to $C/16$, followed by a rectified linear unit activation. The second layer restores the dimension to C, and a sigmoid activation maps the resulting channel weights to the range from 0 to 1:

$$\boldsymbol{S} = F_{\text{Excitation}}(\boldsymbol{Z}, \boldsymbol{W}) = \sigma(\boldsymbol{W}_2\delta(\boldsymbol{W}_1\boldsymbol{Z})) \tag{2}$$

where $\boldsymbol{Z}$ is the channel descriptor obtained from the squeeze operation; $\boldsymbol{S} = [s_1, s_2, \ldots, s_C]^{\mathrm{T}}$ is the learned channel-weight vector; $\boldsymbol{W}_1 \in \boldsymbol{R}^{C/r\times C}$ and $\boldsymbol{W}_2 \in \boldsymbol{R}^{C\times C/r}$ are the learnable weight matrices of the dimensionality-reduction and dimensionality-restoration layers, respectively; r is the channel-reduction ratio and was set to 16 in this study; and $\delta(\cdot)$ and $\sigma(\cdot)$ denote the ReLU and sigmoid activation functions, respectively. Each element $s_c \in (0, 1)$ represents the learned importance of the corresponding feature channel.

In the scaling stage, the learned channel weights are multiplied by the original feature maps:

$$X_c^* = F_{\text{scale}}(X_c, s_c) = s_c \cdot X_c \tag{3}$$

where X_c^* denotes the recalibrated output of the c-th channel, X_c is the corresponding input channel, and s_c is its learned scalar weight. The multiplication between s_c and X_c is applied to all spatial positions of the channel. The complete output feature map is denoted by $\boldsymbol{X}^* = [X_1^*, X_2^*, \ldots, X_C^*]$.

The module therefore performs feature-level channel recalibration rather than explicit exposure compensation. Its expected role is to reduce illumination-induced imbalance among feature channels and preserve discriminative responses under backlit and low-light conditions. The practical effectiveness of the module is experimentally evaluated in Sections 3.2, 3.3 and 3.5.

2.4.2. Multi-Scale Context Aggregation for Partially Occluded Targets

Partial occlusion removes portions of the visible appearance of mango fruits and stems. Under such conditions, detection cannot rely exclusively on fine local textures and must also exploit larger contextual regions, such as the visible contour of a fruit, the spatial proximity between a stem and a fruit, and the surrounding fruit-bearing structure.

Spatial pyramid pooling aggregates contextual information over different effective receptive fields. The SPPF module simplifies the original SPP architecture [26] by using cascaded pooling operations to reduce computational redundancy. However, its sequential structure provides a relatively restricted feature-propagation pattern. The ELAN architecture [27], in contrast, employs parallel feature paths and hierarchical aggregation to improve gradient propagation and reuse intermediate features.

Based on these complementary characteristics, an SPPF-ELAN module was constructed to replace the original SPPF component in the backbone. As shown in Figure 6, the module combines parallel feature paths with cascaded spatial pooling. It is intended to enrich the representation of partially visible targets by integrating fine-scale local details with broader contextual information. The module does not explicitly determine whether a target is occluded or calculate an occlusion ratio. As illustrated in Figure 6a, the proposed SPPF-ELAN module consists of multiple cascaded SPPF blocks followed by an ELAN-style feature aggregation stage. The internal computation of each SPPF block, shown in Figure 6b, progressively enlarges the effective receptive field through successive max-pooling operations while preserving local feature information. The resulting hierarchical features, together with the output of the initial CBS layer, are then aggregated via channel-wise concatenation and fused by the final CBS layer to strengthen multi-scale feature representation. Accordingly, the overall feature extraction and aggregation process can be formulated as follows:

$$\boldsymbol{X}_i^{(l)} = P_{k_i}\left(\boldsymbol{X}_i^{(l-1)}\right) \tag{4}$$

where $\boldsymbol{X}_i^{(0)}$ denotes the feature map obtained after the input CBS layer of the i-th SPPF block, before any pooling operation, $\boldsymbol{X}_i^{(l)}$ ($l = 1,\ldots, L$) denotes the feature map obtained after the l-th max-pooling operation within the *i*-th SPPF block, P_{ki} (·) denotes a max-pooling operation with kernel size k_i, stride 1, and appropriate padding, and L is the number of successive max-pooling operations. In this work, $L = 3$.

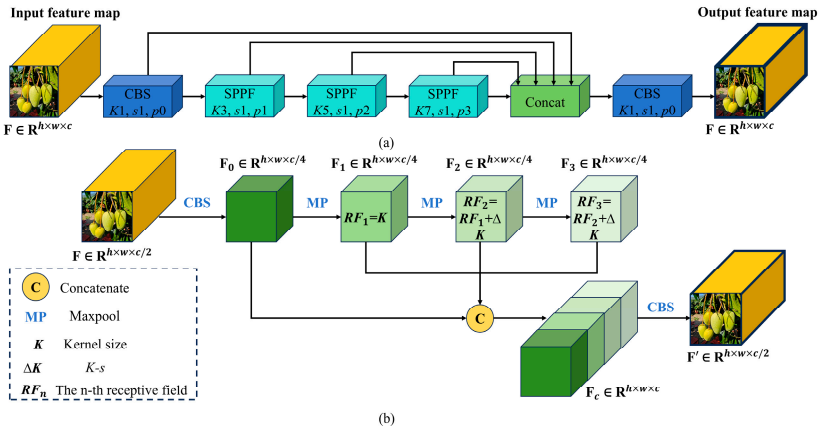


Figure 6. Structure of the multi-scale context aggregation module: (**a**) SPPF-ELAN module and (**b**) internal SPPF unit.

$$\boldsymbol{Y}_i = f_{1\times1}\left(\mathrm{Concat}\left(\boldsymbol{X}_i^{(0)}, \boldsymbol{X}_i^{(1)}, \ldots, \boldsymbol{X}_i^{(L)}\right)\right) \tag{5}$$

where Concat $(\cdot)$ denotes channel-wise concatenation, $f_{1\times1}(\cdot)$ represents a 1×1 convolution for channel fusion and channel reduction, and $\boldsymbol{Y}_i$ denotes the output feature map of the i-th SPPF block.

$$\boldsymbol{Z} = f_{\text{fusion}}(\text{Concat}(\boldsymbol{F}_0, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_N)) \tag{6}$$

where $\boldsymbol{F}_0$ denotes the output feature map of the initial CBS layer, $\boldsymbol{Y}_i$ denotes the output feature map of the i-th cascaded SPPF block, N is the total number of cascaded SPPF blocks, $f_{\text{fusion}}(\cdot)$ represents the final CBS layer for feature fusion, and $\boldsymbol{Z}$ is the output feature map of the proposed SPPF-ELAN module. In this work, $N = 3$, with $k_1 = 3$, $k_2 = 5$, and $k_3 = 7$.

The resulting representation contains both local and contextual information. When part of a fruit or stem is visually unavailable, the remaining visible features can be interpreted together with the surrounding scene context. The module is therefore expected to alleviate feature degradation caused by partial occlusion, although it does not reconstruct invisible target regions or explicitly model a fruit–stem association.

In Figure 6, CBS denotes a convolution-batch-normalization-SiLU block; **MP** denotes max pooling; $\boldsymbol{K}$, $\boldsymbol{s}$, and $\boldsymbol{p}$ denote the kernel size, stride, and padding, respectively; $\boldsymbol{RF}_n$ denotes the effective receptive field after the n-th pooling operation; and h, w, and c denote the height, width, and channel number of the feature map. $\Delta\boldsymbol{K}$ denotes the receptive-field increment introduced by one pooling operation.

2.4.3. Wise-IoU

The baseline YOLOv11n model employs Complete Intersection over Union loss, denoted as CIoU, for bounding-box regression [28]. CIoU jointly considers the overlap between the predicted and ground-truth boxes, the distance between their center points, and their aspect-ratio consistency. Its calculation is given in Equations (7)–(9):

$$L_{\text{CIoU}} = 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{\text{gt}})}{c^2} + \alpha v \tag{7}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{8}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan\frac{w}{h}\right)^2 \tag{9}$$

where $\boldsymbol{b} = (x, y)$ and $\boldsymbol{b}^{\text{gt}} = (x^{\text{gt}}, y^{\text{gt}})$ denote the center coordinates of the predicted and ground-truth bounding boxes, respectively; $\rho(\boldsymbol{b}, \boldsymbol{b}^{\text{gt}})$ denotes the Euclidean distance between their center points; c is the diagonal length of the smallest enclosing rectangle covering both boxes; IoU denotes the intersection-over-union between the predicted and ground-truth boxes; v measures their aspect-ratio inconsistency; and α is an adaptive weighting coefficient for the aspect-ratio term. The variables w and h denote the width and height of the predicted box, whereas w^{gt} and h^{gt} denote those of the ground-truth box.

CIoU provides a comprehensive geometric constraint for bounding-box regression. However, it does not explicitly allocate gradients according to the dynamic quality distribution of training samples. In the present task, partial occlusion, weak illumination, small stem size, and long imaging distance can generate samples with ambiguous visual boundaries and relatively low localization quality. Excessive optimization pressure from extremely low-quality samples may disturb the learning of ordinary, informative samples, whereas an excessive focus on already well-localized boxes may provide limited additional benefit.

To regulate the influence of samples with different localization qualities, WIoU [29] was adopted in place of CIoU. WIoU contains three versions, and WIoU v3 was used in this

study. WIoU v1 introduces a distance-based attention term into the IoU loss, as expressed in Equations (10)–(12).

$$L_{\mathrm{WIoUv1}} = R_{\mathrm{WIoU}} L_{\mathrm{IoU}} \tag{10}$$

$$R_{\mathrm{WIoU}} = \exp\left(\frac{(x - x_{\mathrm{gt}})^2 + (y - y_{\mathrm{gt}})^2}{W_{\mathrm{g}}^2 + H_{\mathrm{g}}^2}\right) \tag{11}$$

$$L_{\mathrm{IoU}} = 1 - IoU \tag{12}$$

where L_{WIoUvl} denotes the WIoU v1 regression loss; L_{IoU} is the basic IoU loss; and R_{WIoU} is the distance-attention term. The coordinates (x, y) and $(x_{\mathrm{gt}}, y_{\mathrm{gt}})$ denote the center coordinates of the predicted and ground-truth bounding boxes, respectively, while W_{g} and H_{g} denote the width and height of the minimum enclosing rectangle covering both the predicted and ground-truth bounding boxes. As the distance between the two box centers increases, R_{WIoU} assigns a larger regression weight.

WIoU v3 further introduces a dynamic non-monotonic focusing mechanism based on the relative quality of bounding-box samples. Its calculation is expressed in Equations (13)–(15).

$$L_{\mathrm{WIoUv3}} = \gamma L_{\mathrm{WIoUv1}} \tag{13}$$

$$\gamma = \frac{\beta}{\delta \alpha^{\beta - \delta}} \tag{14}$$

$$\beta = \frac{L_{\mathrm{IoU}}^{*}}{L_{\mathrm{IoU}}} \in [0, +\infty) \tag{15}$$

where L_{WIoUv3} denotes the WIoU v3 regression loss; γ is the dynamic non-monotonic focusing coefficient; and β is the outlier degree of the current bounding-box sample. In Equation (15), L_{IoU}^{*} denotes the moving average of the IoU loss calculated over training samples used to compute the focusing coefficient, with its gradient detached during the weighting calculation. The hyperparameters α and δ control the shape and peak position of the non-monotonic focusing function. A value of β close to 1 indicates that the sample quality is close to the average training level, whereas a substantially larger value indicates a relatively low-quality sample. The α and δ used in Equation (14) are WIoU v3 hyperparameters and are unrelated to the adaptive CIoU coefficient α in Equation (8) and the ReLU function $\delta(.)$ in Equation (2).

Unlike a monotonically increasing focusing strategy, the non-monotonic coefficient reduces the dominance of both very high-quality and extremely low-quality samples while assigning greater attention to ordinary samples that retain useful optimization information. In the mango dataset, this mechanism is expected to reduce the adverse influence of severely ambiguous boxes caused by occlusion or weak boundaries and to improve the stability of bounding-box regression. Its contribution is independently assessed in the ablation experiments.

2.4.4. P2-Based Small-Object Detection Head

For an input resolution of 640 × 640, the P2 feature map has a spatial resolution of 160 × 160, corresponding to an output stride of 4. The original YOLOv11n model transfers the P3, P4, and P5 feature maps from the backbone to the neck for multi-scale fusion. These feature maps are used to construct three detection outputs. The shallower P2 feature map is not directly employed for prediction.

Repeated downsampling increases semantic abstraction but reduces spatial resolution. This trade-off is particularly important for mango stems, which are generally narrower and occupy fewer pixels than the corresponding fruits. Once fine stem boundaries or attachment-region details are weakened during downsampling, they may be difficult to

recover solely through subsequent upsampling and feature fusion. Similar difficulties arise for distant fruits that occupy small image regions [30].

To preserve additional spatial information, the P2 feature map was introduced into the neck of MangoNET. The high-resolution P2 features were fused with the upsampled semantic features through concatenation and subsequent feature extraction. The resulting feature map, denoted as F2, was then connected to an additional detection head.

Consequently, MangoNET contains four detection heads rather than the three heads used in the original YOLOv11n architecture. The P2-based head focuses primarily on small stems and distant fruits, whereas the remaining heads retain their roles in detecting medium- and large-scale targets. Figure 7 illustrates the P2 feature-fusion process and the outputs generated by the four detection heads. The concatenation node highlighted by the dashed box represents the point at which the high-resolution P2 features are incorporated into the neck.

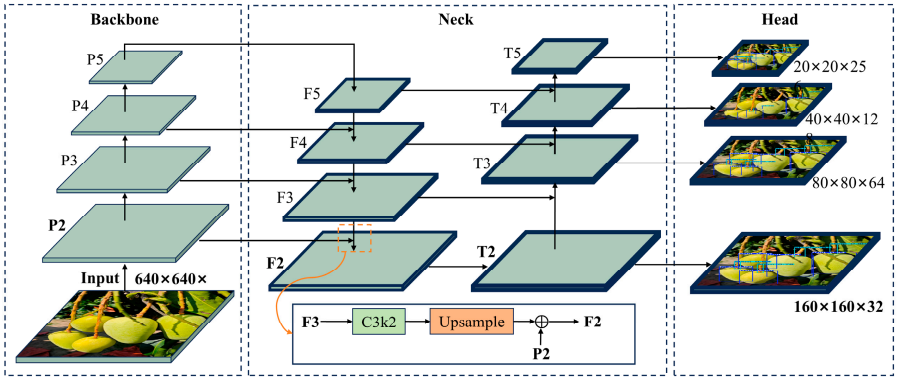


Figure 7. P2 feature-fusion process and the four detection outputs of MangoNET.

2.5. Evaluation Metrics

Precision, recall, F1-score, average precision, and mean average precision were used to evaluate fruit and stem detection. Precision measures the proportion of predicted positive instances that are correctly detected, whereas recall measures the proportion of ground-truth instances that are successfully detected. Their definitions are given in Equations (16) and (17):

$$P = \frac{TP}{FP + TP} \tag{16}$$

$$R = \frac{TP}{TP + FN} \tag{17}$$

where TP, FP, and FN denote the numbers of true-positive detections, false-positive detections, and false-negative instances, respectively. In this study, precision and recall are reported as dimensionless ratios ranging from 0 to 1.

The F1-score is the harmonic mean of precision and recall and provides a balanced evaluation when false detections and missed detections are both important:

$$F1 = \frac{2 \times P \times R}{P + R} \tag{18}$$

where $F1$ denotes the harmonic mean of precision and recall. A larger $F1$-score indicates a more balanced performance between false detections and missed detections.

Average precision summarizes the precision–recall relationship for one object category. Its continuous-form definition is given in Equation (19).

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{19}$$

where $P(R)$ denotes interpolated precision as a function of recall R. Although Equation (19) presents AP in continuous integral form, its practical value was calculated numerically from the discrete precision-recall curve generated over different confidence thresholds.

Mean average precision represents the mean AP over all object categories:

$$mAP = \sum_{i=1}^{n} \frac{AP_i}{n} \tag{20}$$

where AP_i denotes the average precision of the i-th object category and n is the total number of categories. In this study, n = 2, corresponding to the mango-fruit and mango-stem classes. Two IoU-based mAP metrics were reported. mAP50 denotes the mean AP evaluated at an IoU threshold of 0.50, whereas mAP50–95 denotes the mean AP averaged over IoU thresholds ranging from 0.50 to 0.95 at intervals of 0.05. Compared with mAP50, mAP50–95 imposes stricter requirements on bounding-box localization and provides a more comprehensive assessment of detection quality.

The computational complexity of the models was evaluated using GFLOPs, which represents the number of floating-point operations required for a single forward inference and can be calculated as:

$$GFLOPs = \frac{2}{10^9} \sum_{i=1}^{n} H_i W_i K_i^2 C_{i-1} C_i \tag{21}$$

where n denotes the number of convolutional layers; H_i and W_i represent the height and width of the output feature map of the i-th layer; K_i denotes the convolution kernel size; and C_{i-1} and C_i denote the numbers of input and output channels, respectively.

The real-time performance of the models was evaluated using FPS, which represents the number of image frames that can be processed per second and can be calculated as:

$$FPS = \frac{1000}{T_{\text{pre}} + T_{\text{inf}} + T_{\text{post}}} \tag{22}$$

where T_{pre}, T_{inf}, and T_{post} represent the average preprocessing time, inference time, and postprocessing time per image, respectively, measured in milliseconds. The factor 1000 converts milliseconds into seconds.

The model complexity was evaluated using Parameters (Params), which represents the total number of trainable parameters in the network and can be calculated as:

$$Params = \frac{\sum_{i=1}^{n} K_{\text{h}}^i K_{\text{w}}^i C_{\text{in}}^i C_{\text{out}}^i}{10^6} \tag{23}$$

where K_{h}^i and K_{w}^i denote the height and width of the convolution kernel in the i-th layer, respectively. C_{in}^i and C_{out}^i represent the input and output channels. n is the number of convolutional layers, and 10^6 converts the total parameters into millions (M).

3. Results

3.1. Experimental Setup

The experiments were implemented using PyTorch 2.4.1 within an Anaconda environment. The hardware and software configurations are summarized in Table 3 and the principal training hyperparameters are listed in Table 4. Unless otherwise specified, all attention-module comparisons, ablation studies, and model-comparison experiments were conducted using the predefined training, validation, and test subsets described in Section 2.2 and followed the same training and evaluation protocol.

Table 3. Hardware and software configuration used in the experiments.

Item	Configuration
Operating system	Windows
CPU	AMD Ryzen 5 7500F 6-Core Processor 3.70 GHz
GPU	NVIDIA GeForce RTX 4070
Integrated development environment	PyCharm 2024.1.2
Python version	Python 3.8.20
Deep-learning framework	PyTorch 2.4.1
CUDA version	CUDA 12.4

Table 4. Training hyperparameter settings.

Hyperparameter	Value
Epochs	400
Batch	16
Workers	8
AdamW learning rate	0.001
Momentum	0.9
Weight decay	0.0005
Input image size	640 × 640

3.2. Comparison and Selection of Attention Mechanisms

To identify an appropriate attention mechanism for MangoNET, ten representative modules were individually inserted at the same position in the YOLOv11n backbone and evaluated using the same dataset partition, training settings, and evaluation protocol. The evaluated modules included SENet, Coordinate Attention (CA) [31], Efficient Multi-Scale Attention (EMA) [32], Efficient Channel Attention (ECA) [33], Convolutional Block Attention Module (CBAM) [34], Simple Attention Module (SimAM) [35], Context Anchor Attention (CAA) [36], Global Attention Mechanism (GAM) [37], Bi-Level Routing Attention (BRA) [38] and Mixed local channel attention (MLCA) [39]. The results are summarized in Table 5.

Table 5. Comparison of attention mechanisms embedded in the YOLOv11n backbone.

Model	Fruit			Stem			mAP50	mAP50–95	Params (M)	Size (MB)	GFLOPs	FPS
	P	R	F1	P	R	F1						
YOLOv11n	0.949	0.881	0.914	0.908	0.877	0.892	0.925	0.661	2.58	5.5	6.3	132
+CA	0.945	0.880	0.911	0.915	0.871	0.892	0.926	0.665	2.59	5.5	6.3	125
+EMA	0.946	0.875	0.909	0.916	0.871	0.893	0.930	0.663	2.59	5.5	6.4	125
+ECA	0.948	0.876	0.911	0.910	0.862	0.885	0.928	0.658	2.58	5.5	6.3	137
+CBAM	0.948	0.875	0.910	0.912	0.868	0.889	0.925	0.657	2.65	5.6	6.4	133
+SimAM	0.950	0.880	0.914	0.915	0.873	0.894	0.930	0.659	2.58	5.5	6.3	132
+MLCA	0.957	0.869	0.911	0.915	0.862	0.888	0.925	0.661	2.58	5.5	6.4	133
+CAA	0.939	0.871	0.904	0.908	0.843	0.874	0.921	0.636	2.72	5.8	6.4	132
+GAM	0.916	0.865	0.890	0.884	0.819	0.850	0.901	0.592	4.22	8.8	7.6	128
+BRA	0.929	0.840	0.882	0.893	0.840	0.866	0.914	0.626	2.85	6.0	6.4	125
+SENet	0.957	0.879	0.916	0.918	0.862	0.889	0.930	0.665	2.59	5.5	6.3	133

Among the evaluated variants, the model incorporating SENet achieved competitive overall detection performance among the evaluated attention-enhanced models, with mAP50 and mAP50–95 values of 0.930 and 0.665, respectively. Its fruit and stem F1-scores reached 0.916 and 0.889, respectively, while maintaining comparable stem detection performance with the baseline model. Compared with the original YOLOv11n, SENet improved

mAP50 from 0.925 to 0.930 and mAP50–95 from 0.661 to 0.665 while maintaining comparable model complexity. Specifically, SENet introduced only a negligible increase in parameters, from 2.58 M to 2.59 M, while maintaining the same model size of 5.5 MB and computational cost of 6.3 GFLOPs. The inference speed remained comparable to the baseline model, reaching 133 FPS. Although several other attention modules achieved improvements in individual evaluation metrics, SENet provided the most balanced performance across fruit detection accuracy, overall detection quality, and computational efficiency. It was therefore selected as the channel-recalibration component of MangoNET.

3.3. Component-Wise Ablation Study

A component-wise ablation study was conducted to evaluate the individual and combined contributions of the SENet-based channel-recalibration module, SPPF-ELAN context aggregation module, P2 detection head, and WIoU v3 loss. The results of the eight configurations are presented in Table 6. Group 1 corresponds to the baseline YOLOv11n, Groups 2–5 independently evaluate the four modifications, Groups 6 and 7 examine their progressive integration, and the final configuration corresponds to the complete MangoNET.

The SENet-based channel-recalibration module increased mAP50 from 0.925 to 0.930 and mAP50–95 from 0.661 to 0.665. The SPPF-ELAN module produced the largest individual improvement in mAP50–95 among the single-component modifications, increasing mAP50–95 to 0.679, while the fruit and stem F1-scores increased to 0.912 and 0.894, respectively. This improvement was achieved with a moderate increase in model complexity, as the parameters increased from 2.58 M to 3.24 M and the computational cost increased from 6.3 to 6.8 GFLOPs, while the model size increased from 5.5 to 6.8 MB and inference speed decreased slightly from 132 to 130 FPS. The P2 detection head increased stem recall from 0.877 to 0.895 and stem F1-score from 0.892 to 0.911. It also increased mAP50 and mAP50–95 to 0.937 and 0.675, respectively. However, the introduction of the P2 branch resulted in a substantial increase in model complexity, with parameters increasing to 2.66 M, GFLOPs increasing from 6.3 to 10.2, and inference speed decreasing from 132 to 119 FPS. This improvement is consistent with the intended role of the high-resolution branch in preserving fine spatial information for slender stem targets. Replacing CIoU with WIoU v3 increased mAP50 to 0.938 and stem F1-score to 0.906.

The progressive integration results further demonstrate the complementarity of the introduced components. Combining SENet and SPPF-ELAN yielded an mAP50–95 of 0.679, while the further addition of the P2 detection head increased mAP50 and mAP50–95 to 0.938 and 0.684, respectively. The complete MangoNET achieved fruit and stem F1-scores of 0.920 and 0.916, together with mAP50 and mAP50–95 values of 0.941 and 0.690. Its parameter number, model size, computational cost, and inference speed were 3.32 M, 7.2 MB, 10.7 GFLOPs, and 106 FPS, respectively.

Compared with YOLOv11n, MangoNET increased fruit F1-score by 0.6 percentage points, stem recall by 2.9 percentage points, and stem F1-score by 2.4 percentage points. Its mAP50 and mAP50–95 increased by 1.6 and 2.9 percentage points, respectively. The larger gain in mAP50–95 compared with mAP50 suggests that the proposed modifications contributed not only to target retrieval but also to bounding-box localization under stricter IoU thresholds. These accuracy improvements were accompanied by an increase in parameters from 2.58 M to 3.32 M, model size from 5.5 to 7.2 MB, and computational cost from 6.3 to 10.7 GFLOPs, together with a decrease in inference speed from 132 to 106 FPS. This accuracy–efficiency trade-off is discussed further in Section 4.

Table 6. Component-wise ablation results of MangoNET.

Group	YOLOv11n	SE Net	SPPF ELAN	P2	WIoU	Fruit			Stem			mAP50	mAP50–95	Params (M)	Size (MB)	GFLOPs	FPS
						P	R	F1	P	R	F1						
1	√	-	-	-	-	0.949	0.881	0.914	0.908	0.877	0.892	0.925	0.661	2.58	5.5	6.3	132
2	√	√	-	-	-	0.957	0.879	0.916	0.918	0.862	0.889	0.930	0.665	2.59	5.5	6.3	133
3	√	-	√	-	-	0.958	0.871	0.912	0.915	0.873	0.894	0.932	0.679	3.24	6.8	6.8	130
4	√	-	-	√	-	0.951	0.878	0.913	0.928	0.895	0.911	0.937	0.675	2.66	5.8	10.2	119
5	√	-	-	-	√	0.953	0.876	0.913	0.936	0.878	0.906	0.938	0.660	2.58	5.5	6.3	132
6	√	√	√	-	-	0.945	0.890	0.917	0.914	0.894	0.904	0.933	0.679	3.25	6.9	6.8	127
7	√	√	√	√	-	0.956	0.883	0.918	0.926	0.906	0.916	0.938	0.684	3.32	7.2	10.7	106
MangoNET	√	√	√	√	√	0.960	0.883	0.920	0.927	0.906	0.916	0.941	0.690	3.32	7.2	10.7	106

3.4. Training Hyperparameter Sensitivity and Convergence Analysis

To evaluate the rationality of the main training hyperparameters, a one-factor-at-a-time sensitivity analysis was conducted for the learning rate, batch size, input image size, and number of training epochs. The adopted configuration used a learning rate of 0.001, a batch size of 16, an input resolution of 640 × 640, and 400 epochs, while all other training settings were kept fixed when varying a single parameter. The results are summarized in Table 7.

Table 7. Sensitivity analysis of the main training hyperparameters.

Hyperparameter	Value	Fruit			Stem			mAP50	mAP50–95	Training Time (h)
		P	R	F1	P	R	F1			
Learning rate	0.0005	0.943	0.886	0.914	0.916	0.919	0.917	0.937	0.677	6
Batch	8	0.961	0.877	0.917	0.934	0.900	0.917	0.944	0.681	8
Input image size	512 × 512	0.959	0.881	0.918	0.927	0.889	0.908	0.939	0.664	5.6
Epochs	300	0.958	0.871	0.912	0.929	0.896	0.912	0.939	0.676	5
	350	0.957	0.885	0.920	0.922	0.900	0.911	0.941	0.680	5.6
Baseline	-	0.960	0.883	0.920	0.927	0.906	0.916	0.941	0.690	6

Note: The baseline configuration used a learning rate of 0.001, batch size 16, input size 640 × 640, and 400 epochs. All other training settings were fixed when varying a single hyperparameter.

Reducing the learning rate from 0.001 to 0.0005 slightly improved stem recall and F1-score, but decreased mAP50–95 from 0.690 to 0.677. Reducing the batch size from 16 to 8 increased mAP50 from 0.941 to 0.944, but decreased mAP50–95 from 0.690 to 0.681 and increased the training time from 6 to 8 h. The effect of input resolution was more pronounced: reducing the image size from 640 × 640 to 512 × 512 decreased fruit and stem F1-scores from 0.920 and 0.916 to 0.918 and 0.908, respectively, while mAP50–95 dropped from 0.690 to 0.664. These results indicate that the adopted learning rate and batch size provide a favorable balance between detection performance and training efficiency, while the 640 × 640 input resolution better preserves fine spatial information for small and slender stem targets.

The number of training epochs was further examined by comparing 300, 350, and 400 epochs. The corresponding mAP50–95 values were 0.676, 0.680, and 0.690, with training times of 5.0, 5.6, and 6.0 h, respectively. The continued improvement at 400 epochs indicates that shorter training schedules did not achieve the same overall detection performance, while the additional training cost remained limited.

Overall, the adopted configuration of a learning rate of 0.001, a batch size of 16, an input image size of 640 × 640, and 400 epochs achieved the highest mAP50–95 of 0.690 among the evaluated settings. The sensitivity analysis therefore provides quantitative support for the selected training hyperparameters rather than relying solely on empirical or default settings.

3.5. Module-Level Qualitative Analysis Under Illumination and Occlusion Conditions

Figure 8 presents qualitative examples to illustrate the effects of the channel-recalibration and SPPF-ELAN modules under several challenging detection conditions. These examples are intended to provide an intuitive visualization of the observed detection behaviors rather than standalone evidence of module effectiveness, and should therefore be interpreted together with the quantitative ablation results in Table 6.

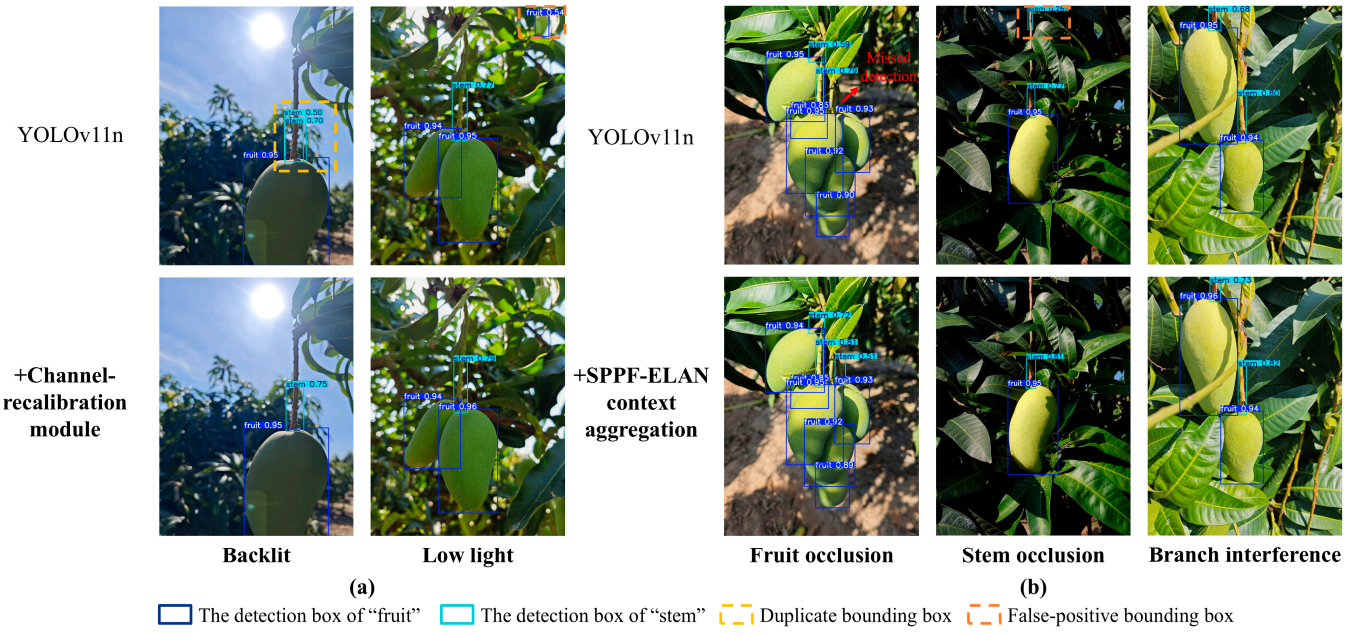


Figure 8. Module-level qualitative comparisons under representative illumination and occlusion conditions: (**a**) YOLOv11n versus YOLOv11n with SENet-based channel recalibration; (**b**) YOLOv11n versus YOLOv11n with SPPF-ELAN context aggregation.

As shown in Figure 8a, the baseline YOLOv11n generated duplicate stem boxes in the backlit scene and failed to provide a complete stem response in the low-light scene. After incorporating the SENet-based channel-recalibration module, the predicted stem regions were more spatially consistent and the duplicate response was suppressed. This qualitative observation is consistent with the improved stem recall and F1-score reported in Table 5.

Figure 8b compares the baseline with the model incorporating SPPF-ELAN. In the fruit-occlusion example, the modified model recovered a stem that was missed by the baseline. Under stem occlusion and branch interference, the SPPF-ELAN variant produced more complete fruit and stem predictions with fewer false-positive or duplicate boxes. These examples qualitatively illustrate that combining local target cues with broader contextual information can alleviate some detection failures caused by partial occlusion. However, they should not be interpreted as evidence that the module can explicitly estimate the occlusion ratio or reconstruct invisible stem regions.

3.6. Full-Model Qualitative Comparison and Feature-Attribution Analysis

Figure 9 compares the complete MangoNET with YOLOv11n five selected challenging orchard conditions. Under backlighting and low illumination, MangoNET reduced duplicate stem predictions and recovered stem regions missed by the baseline. In fruit- and stem-occlusion scenes, the proposed model produced more complete detections of partially visible targets. Under branch interference, MangoNET reduced confusion between branches and stems while maintaining fruit localization.

These examples are consistent with the quantitative improvements in stem recall, stem F1-score, and mAP50–95 reported in Table 6. Nevertheless, Figure 9 represents a qualitative comparison of selected cases rather than an independent statistical evaluation of illumination or occlusion robustness.

Grad-CAM++ [40] was further used to examine the spatial distribution of feature attribution for YOLOv11n and MangoNET. As shown in Figure 10, in these selected examples, the activation regions of MangoNET were generally more concentrated around visible fruit and stem structures in the selected backlit, low-light, occluded, and branch-interference

examples. In particular, MangoNET retained stronger activation around slender stem regions in the low-light and stem-occlusion scenes. These visualizations provide qualitative indications that, in the examined cases, the modified network places greater emphasis on task-relevant regions. However, Grad-CAM++ reflects feature attribution rather than causal reasoning and therefore cannot independently establish the mechanism responsible for the performance improvement.

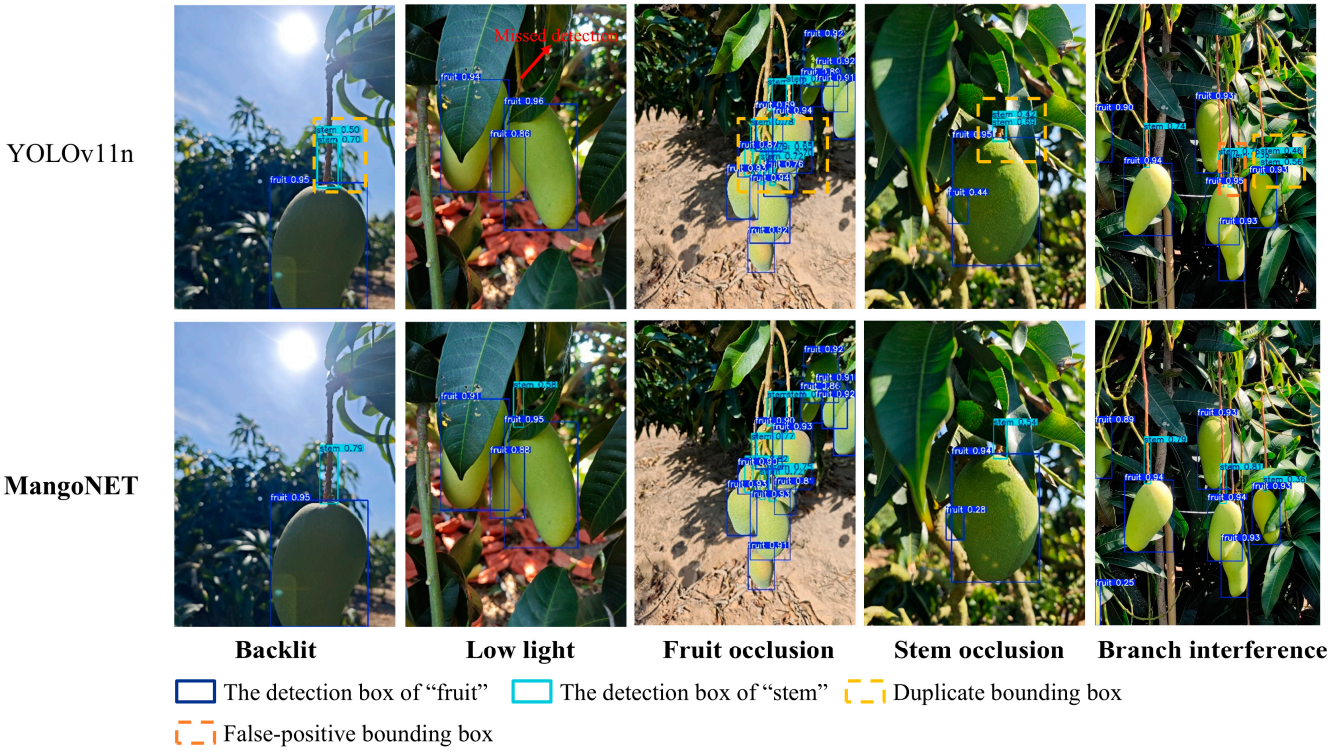


Figure 9. Qualitative comparison between YOLOv11n and MangoNET under representative illumination and occlusion conditions.

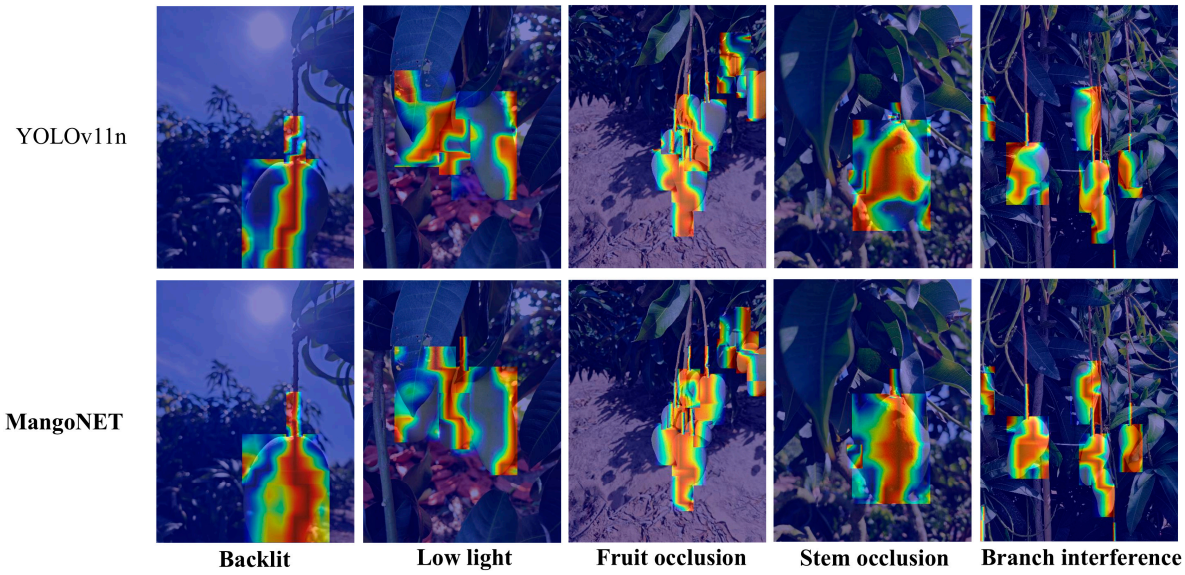


Figure 10. Grad-CAM++ feature-attribution maps of YOLOv11n and MangoNET under representative orchard conditions.

3.7. Robustness Evaluation Under Motion Blur

To evaluate the robustness of MangoNET to motion blur, an additional controlled experiment was conducted. Motion-blurred training samples were generated only from the unaugmented original images contained in the predefined training set and were added to the original training data, while the validation set remained unchanged. For testing, only the unaugmented original images contained in the predefined test set were used. Based on the same test images, mild, moderate, and severe motion blur conditions were generated using equivalent blur lengths of 3, 5, and 7 pixels, respectively, at the model input resolution of 640 pixels. The original annotations were retained because motion blur changes image appearance without altering object locations. This design ensured that the test images under different blur conditions contained identical scenes and annotations and differed only in blur severity.

As shown in Table 8, the original MangoNET maintained relatively stable detection performance under mild and moderate motion blur, indicating that the model itself has a certain degree of robustness to moderate image degradation. The introduction of motion blur augmentation resulted in only limited differences under no or mild motion blur, whereas its advantage became more evident as the blur severity increased, particularly for stem detection. Under severe motion blur, the stem mAP50 increased from 0.885 to 0.923, while mAP50–95 increased from 0.474 to 0.496. Stem recall and F1-score also increased from 0.823 to 0.884 and from 0.856 to 0.901, respectively. For fruit detection, mAP50 increased from 0.894 to 0.906 and mAP50–95 increased from 0.775 to 0.782. These results indicate that MangoNET can maintain satisfactory detection performance under mild and moderate motion blur, while targeted motion blur augmentation further improves its robustness when more severe image degradation occurs.

Table 8. Detection performance under different motion blur levels.

Condition	Model	Fruit					Stem				
		P	R	F1	mAP50	mAP50–95	P	R	F1	mAP50	mAP50–95
No motion blur	MangoNET	0.960	0.883	0.920	0.942	0.831	0.927	0.906	0.916	0.939	0.549
	+MBA	0.961	0.884	0.921	0.945	0.833	0.930	0.906	0.918	0.936	0.530
Mild motion blur	MangoNET	0.962	0.873	0.915	0.935	0.828	0.922	0.907	0.914	0.940	0.554
	+MBA	0.964	0.877	0.918	0.940	0.826	0.927	0.900	0.913	0.932	0.529
Moderate motion blur	MangoNET	0.953	0.846	0.896	0.918	0.808	0.896	0.873	0.884	0.910	0.520
	+MBA	0.961	0.858	0.907	0.924	0.808	0.920	0.884	0.902	0.923	0.518
Severe motion blur	MangoNET	0.944	0.805	0.869	0.894	0.775	0.891	0.823	0.856	0.885	0.474
	+MBA	0.940	0.841	0.888	0.906	0.782	0.918	0.884	0.901	0.923	0.496

Note: MBA denotes motion blur augmentation applied during model training.

The qualitative results in Figure 11 show a similar trend. Under mild and moderate motion blur, most fruits and stems can still be reliably detected by the original MangoNET. As the blur severity increases, however, lower-confidence detections become more evident in several thin stem regions. In comparison, the model trained with motion blur augmentation maintains more stable detection responses, particularly under severe motion blur.

Overall, MangoNET demonstrates inherent robustness to mild and moderate motion blur, while severe motion blur has a greater adverse effect on thin stem detection. Introducing motion-blurred samples during training can effectively alleviate this degradation and further improve the robustness of MangoNET under strongly motion-blurred imaging conditions.



Figure 11. Qualitative comparison of MangoNET with and without motion blur augmentation under different motion blur conditions.

3.8. Sensitivity to Dataset Partitioning

Five-fold cross-validation was conducted to examine the sensitivity of MangoNET to different data partitions. The 1782 original images were first divided into five mutually exclusive subsets at the source-image level. In each fold, four subsets were used for training and the remaining subset was used for validation. Data augmentation was applied only to the training portion after the fold split, while the validation subset remained unaugmented. Precision, recall, F1-score, mAP50, mAP50–95 and FPS were calculated for each fold, and the results are presented in Table 9.

Table 9. Source-image-independent five-fold cross-validation results of MangoNET.

Fold	Fruit			Stem			mAP50	mAP50–95	FPS
	P	R	F1	P	R	F1			
1	0.947	0.897	0.921	0.924	0.911	0.917	0.938	0.712	105
2	0.962	0.906	0.933	0.942	0.921	0.931	0.952	0.710	105
3	0.967	0.899	0.932	0.929	0.910	0.919	0.940	0.718	106
4	0.955	0.890	0.921	0.933	0.910	0.921	0.950	0.701	107
5	0.949	0.893	0.920	0.929	0.913	0.921	0.941	0.715	106
Average	0.956	0.897	0.925	0.931	0.913	0.922	0.944	0.711	105.8
Standard deviation	0.008	0.006	0.007	0.007	0.004	0.005	0.006	0.006	0.837

Across the five folds, MangoNET achieved average mAP50 and mAP50–95 values of 0.944 and 0.711, with corresponding standard deviations of 0.006 and 0.006, respectively. The standard deviations of fruit precision, recall, and F1-score were 0.008, 0.006, and 0.007, respectively, while those of stem precision, recall, and F1-score were 0.007, 0.004, and 0.005. The average inference speed was 105.8 FPS, with a standard deviation of 0.837 FPS, showing only minor variation across the five folds.

The relatively small variations in detection performance and inference speed across the five source-image-independent folds indicate that MangoNET maintained stable performance and computational efficiency under different data partitions. In addition, the fold-averaged results in Table 9 should not be directly compared with the predefined test-split results in Tables 6 and 10.

Table 10. Comparison with representative object detectors on the predefined test subset.

Model	Fruit			Stem			mAP50	mAP50–95	Params (M)	Size (MB)	GFLOPs	FPS
	P	R	F1	P	R	F1						
RT-DETR-l	0.743	0.761	0.752	0.711	0.752	0.731	0.797	0.504	31.99	66.3	103.4	56
Faster-RCNN	0.680	0.843	0.753	0.416	0.694	0.520	0.688	-	28.29	108.1	115.7	23
SSD	0.937	0.715	0.811	0.778	0.309	0.442	0.694	-	26.30	91.1	62.8	85
YOLOv8n	0.944	0.876	0.909	0.915	0.853	0.883	0.931	0.663	3.01	6.3	8.1	130
YOLOv11n	0.949	0.881	0.914	0.908	0.877	0.892	0.925	0.661	2.58	5.5	6.3	132
MangoNET	0.960	0.883	0.920	0.927	0.906	0.916	0.941	0.690	3.32	7.2	10.7	106

3.9. Comparison with Representative Object Detectors

MangoNET was compared with representative transformer-based, two-stage, conventional one-stage, and lightweight YOLO detectors. All models were trained and evaluated using the same predefined dataset partition and evaluation protocol. Table 10 reports the results on the fixed test subset rather than the five-fold averages presented in Table 9. In addition to detection accuracy and model size, the number of parameters (Params), computational complexity (GFLOPs) and inference speed (FPS) were included to provide a more comprehensive evaluation of model performance and efficiency.

MangoNET achieved mAP50 and mAP50–95 values of 0.941 and 0.690, respectively. Compared with YOLOv11n, mAP50 increased by 1.6 percentage points and mAP50–95 increased by 2.9 percentage points. For fruit detection, precision, recall, and F1-score increased by 1.1, 0.2, and 0.6 percentage points, respectively. For stem detection, the corresponding improvements were 1.9, 2.9, and 2.4 percentage points. The larger improvement in stem-related metrics, particularly recall, indicates that MangoNET was more effective in reducing missed detections of the challenging stem class.

MangoNET also achieved a higher mAP50 than RT-DETR-l [41], Faster-RCNN [42], SSD [43] and YOLOv8n [44]. However, these architectures differ substantially in model scale and design. The comparison should therefore be interpreted as a representative cross-architecture evaluation rather than a strictly complexity-matched state-of-the-art benchmark. Regarding model complexity, MangoNET contained 3.32 M parameters and occupied 7.2 MB, which were slightly higher than YOLOv11n with 2.58 M parameters and 5.5 MB, but remained substantially smaller than RT-DETR-l, Faster-RCNN, and SSD. MangoNET required 10.7 GFLOPs and achieved 106 FPS, compared with 6.3 GFLOPs and 132 FPS for YOLOv11n. Although its computational cost was higher than those of YOLOv11n and YOLOv8n, it remained substantially lower than those of RT-DETR-l, Faster-RCNN, and SSD, with an FPS higher than that of these three models.

Representative detection outputs are shown in Figure 12. RT-DETR-l produced duplicate or low-confidence stem predictions in several challenging scenes. Faster-RCNN and SSD exhibited more frequent false-positive fruit or stem detections, particularly under low illumination, fruit occlusion, and branch interference. YOLOv8n performed more consistently but still produced duplicate or missed stem detections in selected difficult cases. MangoNET generated more complete fruit and stem predictions in the displayed examples, consistent with its higher stem recall and improved mAP50–95 performance.

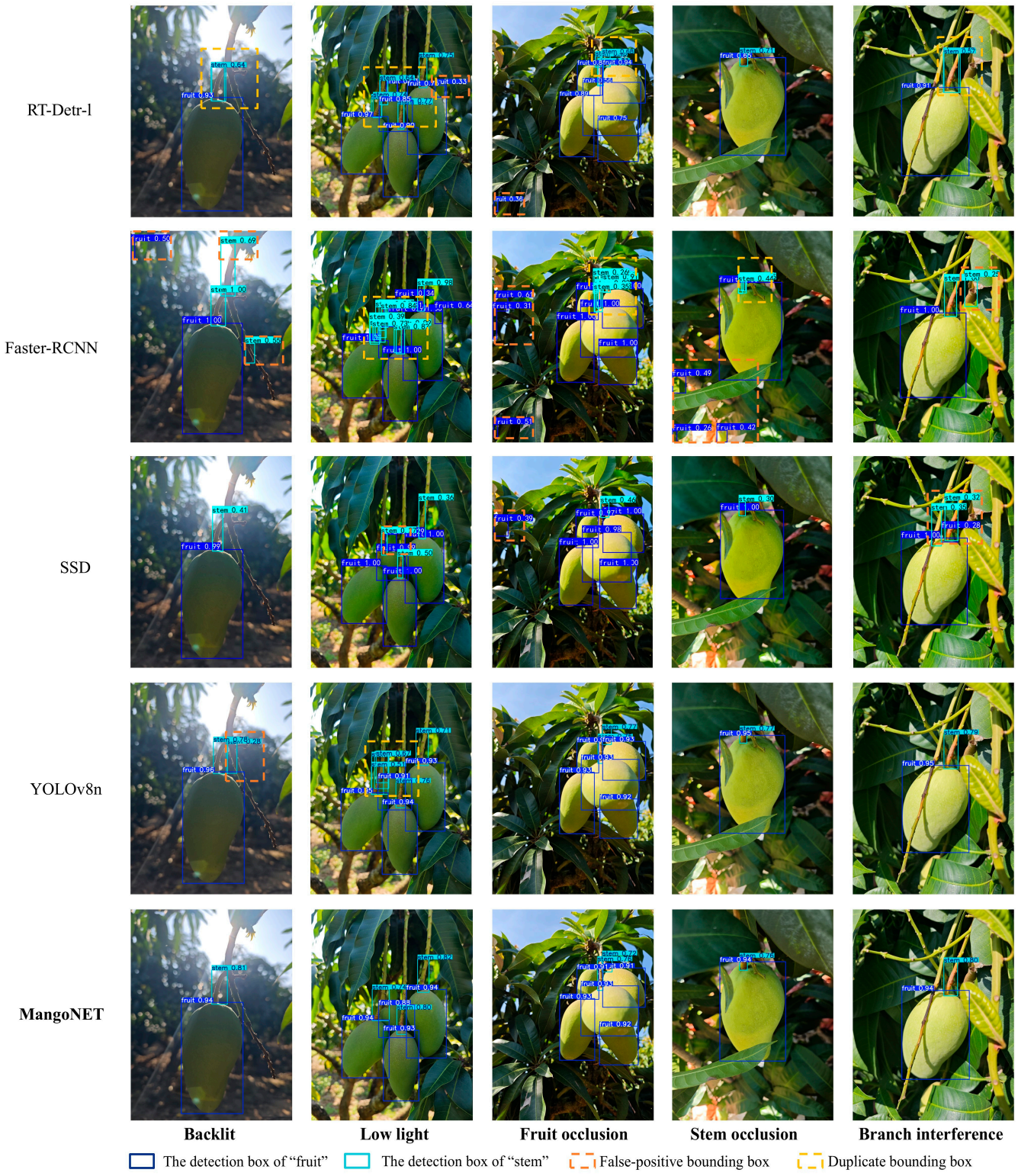


Figure 12. Representative detection outputs of different models under backlighting, low illumination, fruit occlusion, stem occlusion, and branch interference.

3.10. Pilot Cross-Site Field Evaluation

To examine the preliminary potential for cross-site applicability, pilot field experiments were conducted at Nantai Ecological Farm in Wuguishan, Zhongshan, Guangdong Province, China (113°37′ E, 22°44′ N). This orchard was geographically distinct from the two training-data collection sites in Sanya, Hainan Province. The evaluation used a DJI Mavic 3 Pro

UAV (manufactured by DJI, Shenzhen, China), an Intel RealSense D435i RGB-D camera (manufactured by Intel Corporation, Santa Clara, CA, USA), and a laptop equipped with an NVIDIA GeForce GTX 1650 Ti GPU and an AMD Ryzen 5 4600H processor. The practical deployment process of MangoNET using UAV and RGB-D camera is demonstrated in Video S1.

In the UAV experiment, two mango trees were selected as evaluation targets. The UAV moved horizontally at an approximately constant distance of 3 m from the canopy, as illustrated in Figure 13. Images were acquired by the UAV camera and processed using the computing device. The visible-fruit detection rate was calculated as the number of manually annotated visible fruits correctly detected in the selected UAV images divided by the total number of manually annotated visible fruits. As is shown in Figure 14, MangoNET achieved a visible-fruit detection rate of 84.4%, compared with 73.3% for YOLOv11n. The result indicates that MangoNET detected more target instances than YOLOv11n in the selected cross-site scenes of this pilot evaluation. Because the evaluation involved only two trees, it should be interpreted as a pilot demonstration rather than a statistically comprehensive orchard-scale assessment.

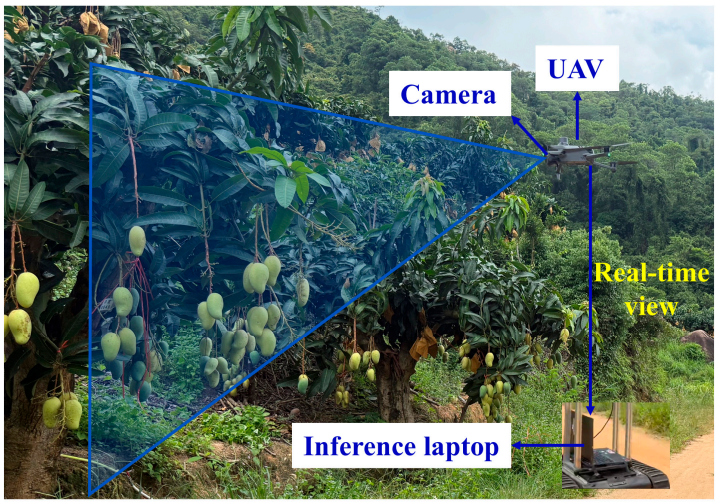


Figure 13. UAV image-acquisition geometry and computing setup in the cross-site orchard.



(a) YOLOv11n (b) **MangoNET**

Figure 14. Comparison of YOLOv11n and MangoNET using a representative UAV-acquired orchard image.

In the second experiment, RGB image streams were acquired using the Intel RealSense D435i camera while an operator manually moved the sensor around the fruit-bearing branches to simulate viewpoint changes during robotic manipulation. Model inference

was performed on the laptop rather than on the camera itself. Six representative frames were selected from the continuous RGB-D video sequences for qualitative visualization in Figure 15 and do not represent the entire RGB-D experiment. In these selected examples, YOLOv11n missed several overlapping or heavily occluded fruits and stems and occasionally merged adjacent fruits into one bounding box. MangoNET reduced these failure cases in the selected frames and generated more complete predictions under viewpoint variation, occlusion, and tied-stem interference.

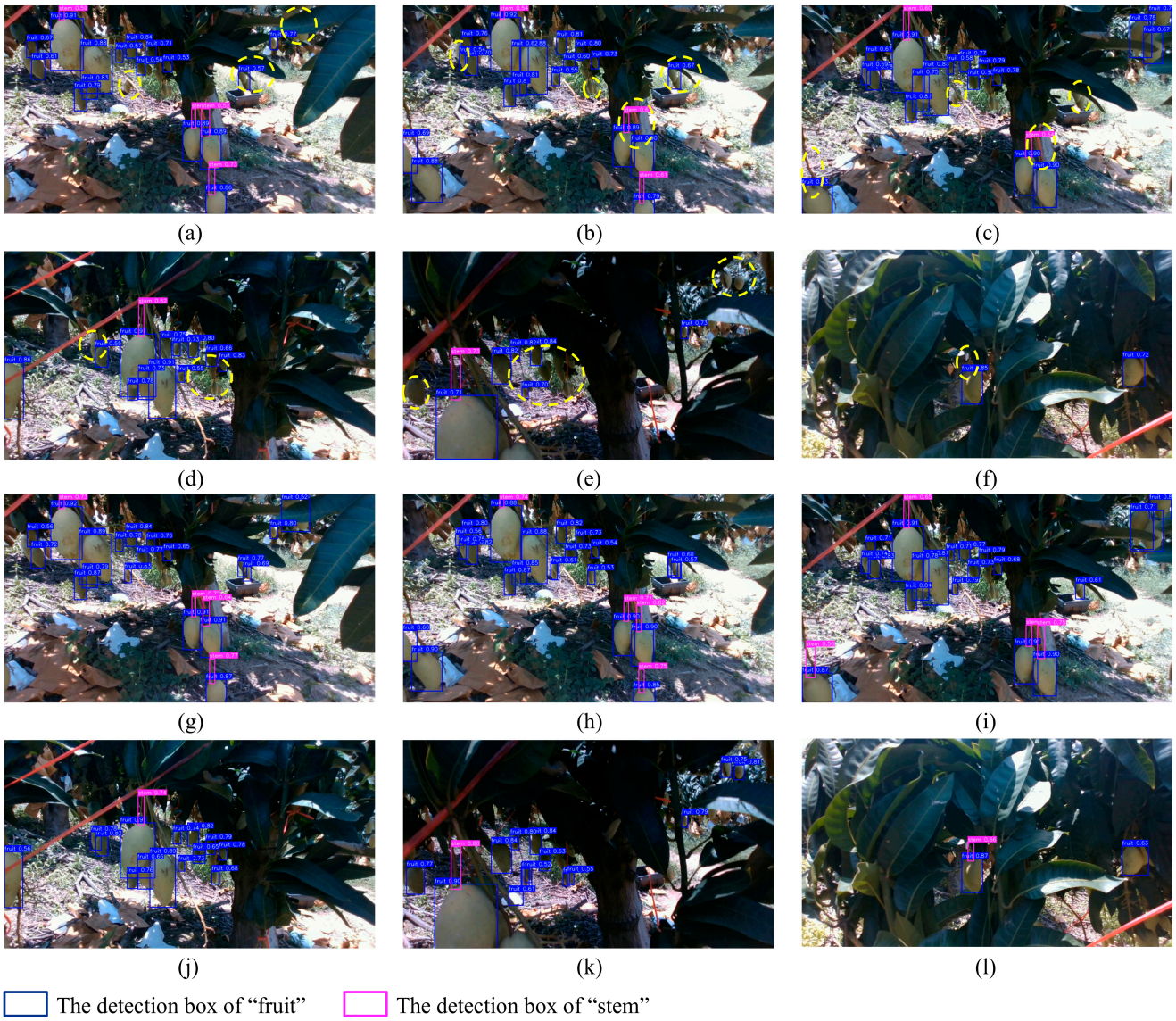


Figure 15. Representative RGB frames acquired using the Intel RealSense D435i: (**a**–**f**) YOLOv11n; (**g**–**l**) MangoNET. Yellow dashed circles indicate missed detections.

To further assess the computational feasibility of the models in practical applications, their inference speeds were evaluated on the same laptop used in the field experiments. YOLOv11n achieved 62 FPS, corresponding to an average end-to-end processing latency of 16.1 ms per frame, whereas MangoNET achieved 48 FPS with an average latency of 20.8 ms per frame. Although MangoNET introduced additional computational overhead, it maintained real-time processing capability on the experimental computing platform.

Although MangoNET was trained using images collected in Hainan Province, the pilot experiments in Zhongshan demonstrate the feasibility of applying the detector to images acquired using different imaging platforms in a geographically distinct orchard. However,

these experiments were designed as exploratory field demonstrations rather than statistically powered evaluations of cross-site generalization. Therefore, no conclusion regarding broad geographical generalization is drawn from the current results. Larger-scale quantitative evaluations involving more trees, complete video sequences, and multiple orchards are required to systematically assess the cross-site generalization capability of MangoNET.

4. Discussion

MangoNET achieved higher overall detection and localization performance than YOLOv11n, with the most pronounced improvements observed for the stem class. Stem recall increased from 0.877 to 0.906, whereas stem F1-score increased from 0.892 to 0.916. In comparison, the improvements for fruit detection were smaller because the baseline already achieved relatively high fruit precision and F1-score. These results indicate that the introduced modifications mainly benefited the more difficult slender and partially visible stem targets.

The difference between the gains in mAP50 and mAP50–95 provides further insight into the contribution of MangoNET. Compared with YOLOv11n, mAP50 and mAP50–95 increased by 1.6 and 2.9 percentage points, respectively. The larger gain under stricter IoU thresholds suggests that the proposed modifications improved bounding-box localization quality in addition to reducing missed detections. Among the individual components, SPPF-ELAN produced the largest single improvement in mAP50–95 among the individual modifications, while the P2 detection head yielded a pronounced increase in stem recall and F1-score. SENet provided a consistent but relatively modest improvement in detection performance, whereas WIoU v3 further enhanced localization accuracy when integrated with the other components.

These improvements were accompanied by increased computational cost. Compared with the baseline YOLOv11n, MangoNET increased the number of parameters from 2.58 M to 3.32 M, model size from 5.5 to 7.2 MB, and computational complexity from 6.3 to 10.7 GFLOPs, while inference speed decreased from 132 to 106 FPS. Among the introduced components, the P2 detection head accounted for most of the additional computational demand. Specifically, introducing P2 alone increased computational complexity from 6.3 to 10.2 GFLOPs and reduced inference speed from 132 to 119 FPS. This increase is mainly attributed to the high-resolution feature maps used by the P2 branch, which preserve fine spatial details but require additional computation. Despite this computational penalty, the P2 head increased mAP50 from 0.925 to 0.937 and mAP50–95 from 0.661 to 0.675, while improving stem recall from 0.877 to 0.895 and stem F1-score from 0.892 to 0.911. These results demonstrate that the additional computational cost of the P2 branch is justified by its ability to enhance the detection of slender and partially occluded stem targets. Therefore, MangoNET represents an accuracy–efficiency trade-off rather than a purely lightweight optimization.

The pilot field experiments further demonstrate the practical feasibility of applying MangoNET to images acquired using different imaging platforms in a geographically distinct orchard. Nevertheless, the evaluation was limited to two trees and selected frames from the camera sequences. These experiments therefore demonstrate application potential rather than statistically validated cross-regional generalization. Compared with mango-harvesting perception methods that integrate object detection with stem segmentation and skeleton extraction [22,23], MangoNET focuses on the preceding candidate-generation stage and does not directly recover stem centerlines or harvesting points. This narrower task formulation enables a unified bounding-box detector but provides less geometric information for manipulation. In addition, although a previous mango fruit–stem detector was evaluated on an NVIDIA Jetson Orin Nano [2], MangoNET was evaluated for inference

speed on the same laptop used in the field experiments and maintained real-time processing capability on this experimental computing platform. However, its deployment efficiency on dedicated edge-computing hardware has not yet been experimentally validated. These differences clarify that MangoNET complements, rather than replaces, segmentation-, keypoint-, and edge-deployment-oriented solutions.

The present framework also has a task-level limitation. MangoNET outputs horizontal bounding boxes but does not explicitly establish one-to-one fruit–stem associations. Although such boxes are suitable for candidate-region generation, they cannot fully represent the elongated geometry and spatial orientation of mango stems. A complete robotic harvesting system would therefore require subsequent association, local segmentation or keypoint estimation, depth-based three-dimensional localization, and manipulator trajectory planning.

In addition, the dataset was collected from two orchards in Sanya under sunny daytime conditions and contains only Tainong and Jinhuang cultivars. Its geographical, varietal, weather, and nighttime diversity remains limited. Future work should therefore expand the dataset across regions and seasons, quantify performance according to illumination and occlusion severity, evaluate fruit–stem association accuracy, and investigate lightweight deployment on edge-computing platforms.

5. Conclusions

This study developed MangoNET for joint mango fruit and stem detection in complex orchard environments. By integrating SENet-based channel recalibration, SPPF-ELAN multi-scale context aggregation, a P2 high-resolution detection head, and WIoU v3 into YOLOv11n, MangoNET achieved fruit and stem F1-scores of 0.920 and 0.916, respectively. Its mAP50 and mAP50–95 reached 0.941 and 0.690, corresponding to improvements of 1.6 and 2.9 percentage points over the baseline. Stem recall increased from 0.877 to 0.906, indicating that the proposed modifications primarily reduced missed detections of small, slender, and partially visible stems. Source-image-independent five-fold cross-validation further demonstrated stable detection performance across different data partitions, while pilot evaluations using images acquired by a UAV and an RGB-D camera in a geographically distinct orchard suggested that MangoNET retained detection capability under an unseen orchard condition. These results support MangoNET as an upstream perception component for generating fruit and stem candidate regions in robotic harvesting systems. Future work will focus on explicit fruit–stem association, harvesting-point localization, three-dimensional pose estimation, and closed-loop robotic manipulation.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/agriculture16171888/s1, Video S1: Overview of the proposed MangoNET framework, including its key modules and architectural enhancements, and detection demonstrations using images acquired by a UAV and an RGB-D camera.

Author Contributions: Conceptualization, J.L. (Jiahuan Lu); formal analysis, Q.D.; funding acquisition, J.L. (Jiahuan Lu); investigation, Q.D., W.Z. and B.C.; methodology, Q.D. and J.L. (Jiahuan Lu); resources, J.L. (Jiahuan Lu) and J.L. (Jiehao Li); supervision, J.L. (Jiahuan Lu), S.Z. and J.L. (Jiehao Li); validation, Q.D.; writing—original draft, Q.D.; writing—review and editing, J.L. (Jiahuan Lu), S.Z. and J.L. (Jiehao Li). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Guizhou Provincial Key Technology Research ad Development Program under Grant 2024100, Guangxi Science and Technology Program (GK JF2504240002), Youth S&T Talent Support Programme of Guangdong Provincial Association for Science and Technology (SKXRC2026749), Guangdong Science and Technology Programme (2025B02020100367), Guang-

dong High-Level Talents Special Support Program (2024TQ08Z107), and Guangzhou Science and Technology Plan (2025A04J3660).

Data Availability Statement: The mango image dataset collected in this study, including the original images and corresponding bounding-box annotation files, is publicly available through Mendeley Data at https://doi.org/10.17632/yvx4s4ckj3.2. The trained model weights, source code, and detailed experimental results are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

References

1. Liu, B.; Xin, Q.; Zhang, M.; Chen, J.; Lu, Q.; Zhou, X.; Li, X.; Zhang, W.; Feng, W.; Pei, H.; et al. Research Progress on Mango Post-Harvest Ripening Physiology and the Regulatory Technologies. *Foods* **2023**, *12*, 173. [CrossRef] [PubMed]
2. Gu, Z.; He, D.; Huang, J.; Chen, J.; Wu, X.; Huang, B.; Dong, T.; Yang, Q.; Li, H. Simultaneous detection of fruits and fruiting stems in mango using improved YOLOv8 model deployed by edge device. *Comput. Electron. Agric.* **2024**, *227*, 109512. [CrossRef]
3. Lu, J.; Tu, Z.; Qian, Z.; Cai, B.; Deng, Q.; Yang, Y.; Li, J. A Multi-Model Fusion Framework for Robust Mango Detection in Complex Orchard Environments. *Agriculture* **2026**, *16*, 1770. [CrossRef]
4. Zhang, J.; Kang, N.; Qu, Q.; Zhou, L.; Zhang, H. Automatic fruit picking technology: A comprehensive review of research advances. *Artif. Intell. Rev.* **2024**, *57*, 54. [CrossRef]
5. Liu, T.-H.; Nie, X.-N.; Wu, J.-M.; Zhang, D.; Liu, W.; Cheng, Y.-F.; Zheng, Y.; Qiu, J.; Qi, L. Pineapple (Ananas comosus) fruit detection and localization in natural environment based on binocular stereo vision and improved YOLOv3 model. *Precis. Agric.* **2023**, *24*, 139–160. [CrossRef]
6. Tang, Y.; Zhou, H.; Wang, H.; Zhang, Y. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Syst. Appl.* **2023**, *211*, 118573. [CrossRef]
7. Xu, D.; Xiong, H.; Liao, Y.; Wang, H.; Yuan, Z.; Yin, H. EMA-YOLO: A Novel Target-Detection Algorithm for Immature Yellow Peach Based on YOLOv8. *Sensors* **2024**, *24*, 3783. [CrossRef] [PubMed]
8. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2023**, *35*, 13895–13906. [CrossRef]
9. Chen, W.; Liu, M.; Zhao, C.; Li, X.; Wang, Y. MTD-YOLO: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Comput. Electron. Agric.* **2024**, *216*, 108533. [CrossRef]
10. Gai, R.; Liu, Y.; Xu, G. TL-YOLOv8: A Blueberry Fruit Detection Algorithm Based on Improved YOLOv8 and Transfer Learning. *IEEE Access* **2024**, *12*, 86378–86390. [CrossRef]
11. Zhu, A.; Zhang, R.; Zhang, L.; Yi, T.; Wang, L.; Zhang, D.; Chen, L. YOLOv5s-CEDB: A robust and efficiency Camellia oleifera fruit detection algorithm in complex natural scenes. *Comput. Electron. Agric.* **2024**, *221*, 108984. [CrossRef]
12. Li, H.; Gu, Z.; He, D.; Wang, X.; Huang, J.; Mo, Y.; Li, P.; Huang, Z.; Wu, F. A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments. *Comput. Electron. Agric.* **2024**, *220*, 108914. [CrossRef]
13. Bai, Y.; Yu, J.; Yang, S.; Ning, J. An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosyst. Eng.* **2024**, *237*, 1–12. [CrossRef]
14. Li, J.; Li, C.; Zeng, S.; Luo, X.; Chen, C.P.; Yang, C. A lightweight pineapple detection network based on YOLOv7-tiny for agricultural robot system. *Comput. Electron. Agric.* **2025**, *231*, 109944. [CrossRef]
15. Zhao, R.; Zhu, Y.; Li, Y. An end-to-end lightweight model for grape and picking point simultaneous detection. *Biosyst. Eng.* **2022**, *223*, 174–188. [CrossRef]
16. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A visual detection method for nighttime litchi fruits and fruiting stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [CrossRef]
17. Miao, Z.; Yu, X.; Li, N.; Zhang, Z.; He, C.; Li, Z.; Deng, C.; Sun, T. Efficient tomato harvesting robot based on image processing and deep learning. *Precis. Agric.* **2023**, *24*, 254–287. [CrossRef]
18. Qi, X.; Dong, J.; Lan, Y.; Zhu, H. Article Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet. *Remote Sens.* **2022**, *14*, 2004. [CrossRef]
19. Li, P.; Chen, J.; Chen, Q.; Huang, L.; Jiang, Z.; Hua, W.; Li, Y. Detection and picking point localization of grape bunches and stems based on oriented bounding box. *Comput. Electron. Agric.* **2025**, *233*, 110168. [CrossRef]
20. Fu, L.; Yang, Z.; Wu, F.; Zou, X.; Lin, J.; Cao, Y.; Duan, J. YOLO-Banana: A Lightweight Neural Network for Rapid Detection of Banana Bunches and Stalks in the Natural Environment. *Agronomy* **2022**, *12*, 391. [CrossRef]
21. Ma, Z.; Dong, N.; Gu, J.; Cheng, H.; Meng, Z.; Du, X. STRAW-YOLO: A detection method for strawberry fruits targets and key points. *Comput. Electron. Agric.* **2025**, *230*, 109853. [CrossRef]

22. Zhang, B.; Xia, Y.; Wang, R.; Wang, Y.; Yin, C.; Fu, M.; Fu, W. Recognition of mango and location of picking point on stem based on a multi-task CNN model named YOLOMS. *Precis. Agric.* **2024**, *25*, 1454–1476. [CrossRef]
23. Li, H.; Huang, J.; Gu, Z.; He, D.; Huang, J.; Wang, C. Positioning of mango picking point using an improved YOLOv8 architecture with object detection and instance segmentation. *Biosyst. Eng.* **2024**, *247*, 202–220. [CrossRef]
24. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. 2024. Available online: http://arxiv.org/abs/2410.17725 (accessed on 8 July 2026).
25. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. 2019. Available online: http://arxiv.org/abs/1709.01507 (accessed on 3 July 2026).
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
27. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing Network Design Strategies Through Gradient Path Analysis. 2022. Available online: http://arxiv.org/abs/2211.04800 (accessed on 15 August 2026).
28. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. 2019. Available online: http://arxiv.org/abs/1911.08287 (accessed on 7 August 2026).
29. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. 2023. Available online: http://arxiv.org/abs/2301.10051 (accessed on 24 August 2026).
30. Li, R.; Shen, Y. YOLOSR-IST: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and YOLO. *Signal Process.* **2023**, *208*, 108962. [CrossRef]
31. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
32. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
33. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11534–11542.
34. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
35. 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, Online, 18–24 July 2021; pp. 11863–11874.
36. Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; Yao, Y. Poly kernel inception network for remote sensing detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 27706–27716.
37. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561.
38. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R.W.H. Biformer: Vision transformer with bi-level routing attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 10323–10333.
39. Wan, D.; Lu, R.; Shen, S.; Xu, T.; Lang, X.; Ren, Z. Mixed local channel attention for object detection. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106442. [CrossRef]
40. Chattopadhyay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018. [CrossRef]
41. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
42. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
44. Jocher, G.; Nishimura, K.; Mineeva, T.; Vilariño, R. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 11 August 2026).

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.