mathematics

MDPI

Article

EdgeFormer-YOLO: A Lightweight Multi-Attention Framework for Real-Time Red-Fruit Detection in Complex Orchard Environments

Zhiyuan Xu [1,†], Tianjun Luo [1,†], Yinyi Lai [1,2,†], Yuheng Liu [3] and Wenbin Kang [2,4,*]

[1] Department of Mechanical Engineering, Hohai University, Nanjing 211100, China
[2] Department of Mechanical Engineering, City University of Hong Kong, Hong Kong 999077, China
[3] Department of Data and Systems Engineering, University of Hong Kong, Hong Kong 999077, China
[4] Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK
* Correspondence: wenbin.kang@cityu.edu.hk
† These authors contributed equally to this work.

Abstract

Accurate and efficient detection of red fruits in complex orchard environments is crucial for the autonomous operation of agricultural harvesting robots. However, existing methods still face challenges such as high false negative rates, poor localization accuracy, and difficulties in edge deployment in real-world scenarios involving occlusion, strong light reflection, and drastic scale changes. To address these issues, this paper proposes a lightweight multi-attention detection framework, EdgeFormer-YOLO. While maintaining the efficiency of the YOLO series' single-stage detection architecture, it introduces a multi-head self-attention mechanism (MHSA) to enhance the global modeling capability for occluded fruits and employs a hierarchical feature fusion strategy to improve multi-scale detection robustness. To further adapt to the quantitative deployment requirements of edge devices, the model introduces the arsinh activation function, improving numerical stability and convergence speed while maintaining a non-zero gradient. On the red fruit dataset, EdgeFormer-YOLO achieves 95.7% mAP@0.5, a 2.2 percentage point improvement over the YOLOv8n baseline, while maintaining 90.0% precision and 92.5% recall. Furthermore, on the edge GPU, the model achieves an inference speed of 148.78 FPS with a size of 6.35 MB, 3.21 M parameters, and a computational overhead of 4.18 GFLOPs, outperforming some existing mainstream lightweight YOLO variants in both speed and mAP@50. Experimental results demonstrate that EdgeFormer-YOLO possesses comprehensive advantages in real-time performance, robustness, and deployment feasibility in complex orchard environments, providing a viable technical path for agricultural robot vision systems.

Keywords: EdgeFormer-YOLO; red fruit detection; agricultural picking robot; multi-head self-attention; real-time

MSC: 68T07

Academic Editor: Nicolae Herisanu

Received: 29 October 2025
Revised: 22 November 2025
Accepted: 24 November 2025
Published: 26 November 2025

Citation: Xu, Z.; Luo, T.; Lai, Y.; Liu, Y.; Kang, W. EdgeFormer-YOLO: A Lightweight Multi-Attention Framework for Real-Time Red-Fruit Detection in Complex Orchard Environments. *Mathematics* **2025**, *13*, 3790. https://doi.org/10.3390/math13233790

1. Introduction

Red fruits, such as tomatoes, strawberries, cherries, and red apples, are widely cultivated across global agricultural systems and hold significant economic value, serving as staple cash crops for numerous farms and agricultural enterprises. But their harvest window seldom exceeds five days [1]. Manual picking provides only 0.2–0.3 t per worker per day and suffers from large inter-individual variability in ripeness judgment, driving up

post-harvest waste and grading costs. Autonomous harvesting robots are therefore transitioning from a technological vision to an industry imperative [2]. However, in natural environments, tomato fruits often grow in dense clusters, are severely obscured, and experience dramatic changes in illumination, posing a significant challenge to robotic vision systems. Recent surveys on harvesting robot vision highlight orchard bottlenecks—occlusion, scale variance, illumination shifts, and fruit–foliage color similarity—making accurate, real-time fruit detection both a prerequisite and a persistent bottleneck for efficient, low-cost robotic picking [3]. Therefore, building a robust and efficient red fruit detection system is of great significance for promoting the practical application of agricultural robots [4].

Traditional image detection methods [5] usually rely on artificially designed features, such as color, shape, and texture, combined with machine learning classifiers for target recognition. For example, some studies separate red fruits from the background using color segmentation algorithms and then classify them based on shape features such as area, perimeter, and circularity [6]. Such methods perform well in scenes with simple backgrounds and obvious contrast between the fruit and the background, and have the advantages of low computational complexity and easy deployment. However, these methods place high demands on image quality and background conditions, making them susceptible to factors such as light changes, shadows, and noise. Furthermore, in complex greenhouse environments, fruits are often blocked by leaves, light is uneven, and the color of red fruits of different maturity levels varies significantly, making traditional features difficult to generalize. In addition, classification methods usually take the entire image as a unit and cannot provide spatial location information of the fruit, making it difficult to meet the high precision requirements of the robot vision system for the integration of detection and positioning [7].

In recent years, the Transformer structure has demonstrated strong modeling capabilities in visual tasks; especially its self-attention mechanism can effectively capture long-range dependencies and alleviate the interference caused by occlusion and complex background. Vision Transformer and its variants achieve better accuracy than Convolutional Neural Networks (CNNs) in fruit detection tasks by dividing images into patches and modeling global context. Detection Transformer uses global self-attention to eliminate anchor boxes and non-maximum suppression (NMS), directly modeling the long-range dependencies between fruits and complex backgrounds, thereby improving robustness under occlusion and overlap [8]. However, Transformer-based methods are sensitive to data scale and computation, and the complexity of self-attention grows quadratically with image resolution. It is not friendly to edge devices, and there is still a problem of insufficient positioning accuracy in dense small target detection [9]. Therefore, despite its advantages in feature expression, it still faces challenges in agricultural robot scenarios with high real-time requirements.

In this case, the You Only Look Once (YOLO) series of methods has become the mainstream choice in agricultural vision systems with its end-to-end detection framework and extremely high inference speed [10]. While maintaining real-time performance, YOLOv5 significantly improves detection accuracy and robustness in complex environments by introducing the CSP structure, PANet neck, and Anchor-Free mechanism. For the red fruit detection task, YOLO can simultaneously perform classification and localization in a single forward pass, outputting the fruit's bounding box and confidence score, which directly supports robot grasping path planning. Furthermore, YOLO supports lightweight deployments, such as YOLOv5n and YOLOv8n, which enable real-time inference at over 30 FPS on edge devices like NVIDIA Jetson, meeting the real-time demands of field operations [11]. Despite these advances, several important research gaps remain in red fruit detection for practical orchard harvesting [12]. At first, existing YOLO-based detectors are

still vulnerable when strong occlusion, specular highlights, and large-scale variance occur simultaneously in dense canopies, which can lead to missed detections of ripe fruits. For example, single-layer feature fusion in prior lightweight detectors, such as Faster-YOLO-AP [13], struggles with extreme specular reflections and simultaneous large vs. small fruit instances. In addition, there is a lack of lightweight detector designs that are explicitly tailored to meet the strict latency and computational constraints of edge devices mounted on harvesting platforms, while maintaining sufficient robustness for complex orchard scenes [14].

Therefore, further optimizing the architecture to balance model lightweight and efficiency while further improving accuracy and making it truly suitable for automated harvesting by agricultural robots has become a high-interest topic of research. Figure 1 shows the procedure from the orchard image set to the final detection results, which intuitively demonstrates the processing flow. This paper proposes a red fruit detection framework for edge deployment—EdgeFormer-YOLO—whose core contributions are as follows:

(1) We embed a multi-head self-attention (MHSA) mechanism into the YOLO backbone network, overcoming the limitations of convolutional local receptive fields and effectively modeling long-range contextual relationships between fruits, significantly mitigating the missed detection problem caused by occlusion.
(2) By designing a hierarchical feature fusion strategy, the model integrates high-order semantics and fine-grained texture information across four scales, effectively improving the joint detection capability for small and large objects and enhancing robustness in scenes with strong light reflection and drastic scale changes.
(3) We introduce the arsinh activation function to replace the traditional SiLU function. Its non-zero derivative across the entire real domain effectively alleviates the gradient vanishing problem during edge quantization, improving the model's numerical stability and convergence efficiency in low-precision deployment environments.

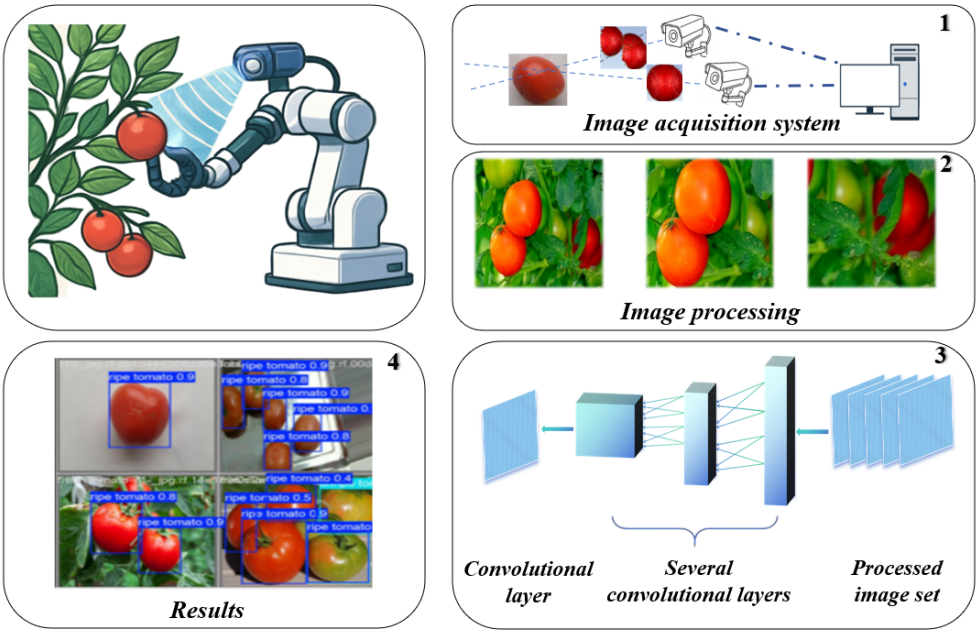

Figure 1. End-to-end pipeline of the agricultural picking robot vision system processing flow: (**1**) field image acquisition, (**2**) offline data augmentation, (**3**) hierarchical convolution & feature extraction, (**4**) final red fruit bounding boxes with maturity labels.

On the red fruit dataset, EdgeFormer-YOLO, with a model size of 6.35 MB, 3.21 M parameters, and 4.18 GFLOPs of computation, achieves an inference speed of 148.78 FPS and a detection accuracy of 95.7% mAP@0.5 on the edge GPU. It establishes new state-of-the-art

(SOTA) efficiency on the mAP@50-speed Pareto frontier, achieving a performance breakthrough while maintaining a nanoscale lightweight structure, which provides a feasible solution for the real-time deployment of agricultural robots in real orchard environments.

2. Related Work

Early picking robots used mechanical contact sensors to locate red fruits—when the sensor touched a fruit, it triggered the mechanical arm to grab. However, this method had a high fruit damage rate and could not distinguish maturity. With the advancement of machine learning technologies, several studies have begun exploring their application in red fruit recognition [15]. These approaches typically combine image feature extraction with classification, training machine learning models to achieve automated red fruit recognition. For example, algorithms such as support vector machines (SVM) and Random Forests have been used to classify feature vectors of red fruits [16]. Kumari and Singh used RGB-based color, shape, and texture features combined with an SVM classifier to grade guava fruits by quality and maturity automatically [17]. Pereira et al. used digital imaging and Random Forests to predict the ripeness of papaya fruits. The advantage of these methods lies in their ability to automatically learn feature representations of red fruits, thereby improving recognition accuracy and robustness [18]. However, machine learning–based methods still face certain limitations. The feature extraction process still requires manual design, making it difficult to fully exploit global image information. In addition, the generalization capability and real-time performance of such models remain to be improved.

In recent years, deep learning technology has made significant progress in classification [19], recognition [20], segmentation [21], and object detection [22]. CNNs, as powerful deep learning models, can automatically learn image feature representations, thereby enhancing recognition accuracy and efficiency. Several studies have attempted to apply CNNs to red fruit recognition [23]. This approach can fully utilize global image information and automatically learn the feature representations of red fruits, thus improving recognition accuracy and robustness. For instance, Fast R-CNN, by sharing convolutional features and employing RoI pooling, significantly reduces redundant computation for candidate regions, enabling end-to-end training while maintaining detection accuracy, and is particularly suitable for fine-grained classification of high-resolution fruit images [24]. However, Fast R-CNN still relies on external region proposal generators, which slows down overall inference and makes it unsuitable for real-time fruit harvesting. Moreover, such two-stage architectures exhibit insufficient sensitivity to small targets heavily occluded by branches and leaves, and suffer severe accuracy degradation under nighttime illumination scenarios.

In contrast, the one-stage object detection algorithm, YOLO, achieves a better balance between accuracy and real-time performance. YOLO [25], first introduced by Redmon et al. in 2016, treats object detection as a regression problem, using a single forward pass of a neural network to obtain object bounding boxes and class information, and has since evolved into its 12th generation. However, YOLO still lags behind two-stage methods in detecting small or densely packed objects. To address this, YOLO9000 [26] adopted concepts from Faster R-CNN, incorporating anchor mechanisms and multi-scale training, and using Darknet19 as its backbone, which significantly improved small object detection while further enhancing model speed. For instance, Lu et al. added a Convolutional Block Attention Module, focusing solely on the target canopy to improve detection accuracy [27]. Fang et al. combined YOLOv5s with cascaded dilated convolutions, spatial pyramid pooling, and Ghost Convolution networks to develop a deep learning–based method for online identification of red bayberry fruits [28]. After several iterations, YOLOv8 [29] introduced deformable convolutions, a decoupled head structure, and distribution focal loss, modified the C2f architecture, and offered five versions of different sizes: Nano (n), Small (s),

Medium (m), Large (l), and Extra Large (x). Wu et al. proposed SGW-YOLOv8n for apple detection/segmentation, reporting mAP75.9%, at 44.37 FPS, but the model still needs validation under different lighting conditions [30]. Wang et al. employed the YOLOv8+ model in combination with image processing methods for strawberry detection and ripeness classification [31]. Li et al. built YOLOv8s-Longan for a fruit-picking UAV, achieving an 18.1 MB model, 45–50 FPS, and 87.5% accuracy in orchards, but dense overlaps and occlusions still caused misses during fast flight [14]. YOLOv10 focuses on performing object detection through NMS-free training [32]. YOLOv12 integrates self-attention mechanisms into the model, making it more robust in complex scenarios while enabling fast and efficient computation [33]. Wang et al. introduced Green Fruit Detector and achieved 94.5%/84.4%/85.9% accuracy on pear/guava/green apple, but fruit–foliage color similarity and occlusion remain inherently challenging in complex orchards [34].

3. Materials and Methods

3.1. Dataset

The red fruit dataset used in this study contains 2607 images. Among them, 2101 images form the training set contains 5696 tomato instances (4062 ripe and 1634 unripe), a mildly imbalanced ratio of about 2.5:1. To mitigate this issue, we resampled the minority class during offline data augmentation and combined it with more geometric augmentation methods, such as rotation, mirroring, random cropping, achieving a near 1:1 class ratio. The remaining images were assigned to the validation set, containing 465 tomato instances, including 356 ripe tomatoes and 109 unripe tomatoes. Each image is accompanied by a corresponding annotation file in text format, which contains the label information for one or more target objects. Figure 2 illustrates tomato images in the dataset under various challenging conditions, such as strong light occlusion, overlapping branches and leaves, and large variations in fruit size. In these images, blue bounding boxes are used to annotate unripe red fruits, while green bounding boxes are used to annotate ripe red fruits. The images in this dataset come from tomato farms, greenhouse growing facilities, and user uploads, which the Roboflow platform aggregates to enhance diversity. The dataset is designed to simulate the complexity of red fruit target detection in real-world environments, providing diverse testing scenarios for the development and evaluation of target detection algorithms in harvesting robots. It covers challenging real-world conditions such as strong illumination, occlusions by branches and leaves, and large-scale variations, providing diverse scenarios for evaluating harvesting-oriented detectors. Although the dataset's heterogeneity, while undocumented, represents real-world orchard variability that any deployable robot vision system must handle and it will not affect the core conclusions of this work.

3.2. Model

In this study, we propose an advanced, efficient, and robust object detection framework tailored for red fruit recognition. The framework predicts bounding boxes and class probabilities directly from input images through a single forward pass, thereby significantly reducing computational overhead. Compared with two-stage detectors, it achieves a much faster inference speed. Figure 3 illustrates the overall pipeline. As shown in Figure 3, our pipeline takes a 640 $\times$ 640 image as input, extracts and fuses features through CBS and C2f blocks, integrates MHSA for long-range context, and performs up-sampling and multi-scale fusion before predicting bounding boxes and classes in a single pass.

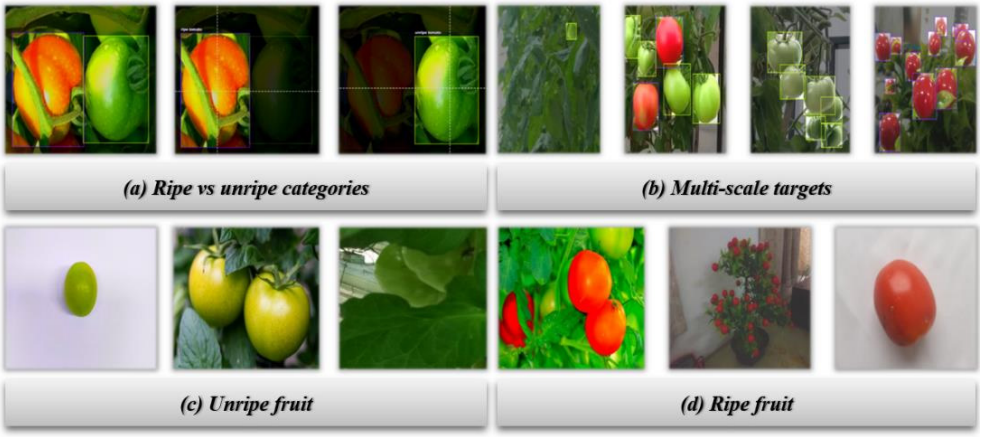

Figure 2. Examples from our red-fruit detection dataset in a complex environment. (**a**) ripe vs. unripe category split; (**b**) multi-scale fruits in one frame; (**c**) dense unripe fruit cases with Occlusion; (**d**) fully ripe cases with lighting changes and specular reflections.

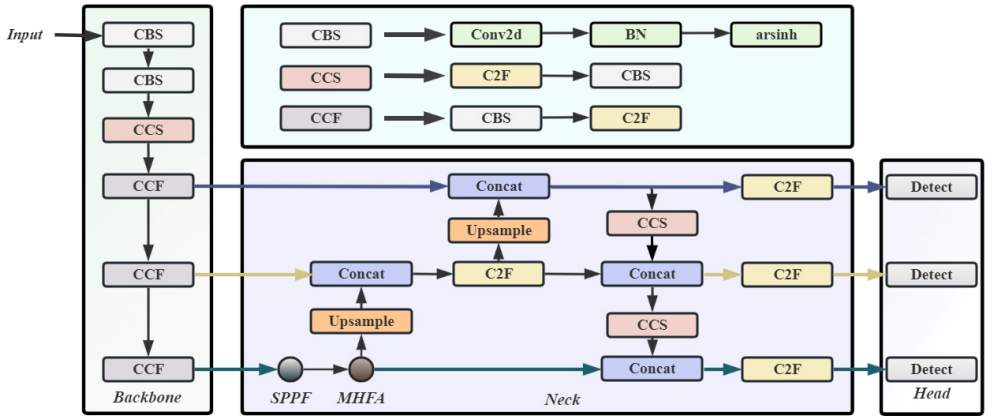

Figure 3. The overall pipeline of EdgeFormer-YOLO.

3.2.1. Feature Pyramid Network

In FPN, high-level semantic features are propagated to lower layers, enriching the semantic content of low-level features. Meanwhile, lateral connections are employed to add feature maps from the bottom-up and top-down pathways, ensuring that feature maps at each level retain both high-level semantics and low-level details, thereby enabling FPN to generate high-quality feature maps at multiple scales and enhancing object detection performance. Many SOTA detection frameworks, such as Faster R-CNN [35] and Mask R-CNN [36], have incorporated FPN, achieving notable improvements in detection accuracy under complex scenarios.

3.2.2. Multi-Head Self-Attention

To address the occlusion-induced miss detection of fruits in automated harvesting systems, we integrate MHSA into the feature extraction backbone. Convolutional operators, constrained by local receptive fields, fail to model long-range dependencies between visible and shaded fruits. MHSA overcomes this limitation by computing global pairwise affinities in a single layer, thereby explicitly encoding the contextual relationship between occluded and non-occluded fruits. The mechanism decomposes the feature map into h independent subspaces; each sub-space applies its own linear projections to generate Query, Key, and Value representations, enabling every head to specialize in distinct spatial or semantic regions. The concatenated outputs aggregate complementary information, yielding a comprehensive feature representation that is robust to partial occlusion. As shown in

Figure 4, input features are linearly projected into Q/K/V, passed through scaled dot product attention, score weighted aggregated, concatenated, and linearly transformed to yield occlusion robust global context that is injected back into the backbone. Consequently, the proposed architecture can significantly improve classification accuracy under heavy occlusion and variable illumination compared with purely convolutional counterparts.

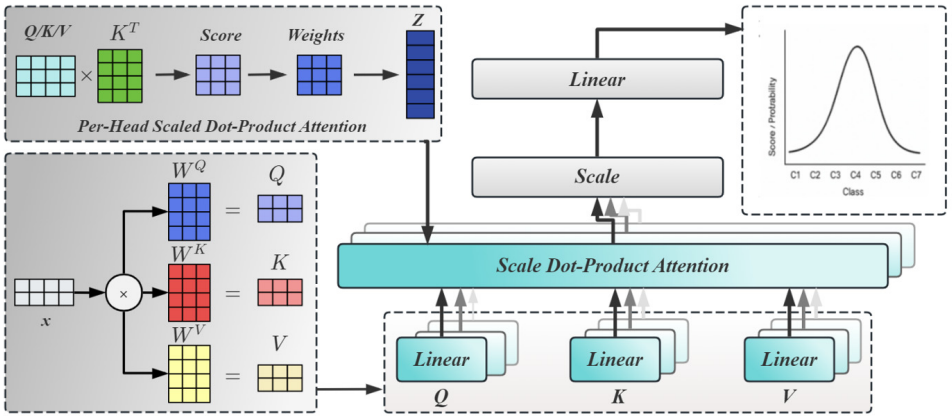

Figure 4. Detailed pipeline of the embedded MHSA module.

In EdgeFormer-YOLO, we insert a single global MHSA block after the SPPF module in the neck. This design allows the SPPF module to first aggregate multi-scale local responses, while the subsequent MHSA layer focuses on modeling long-range relationships between partially occluded and visible fruits on the compressed representation. In addition, we deliberately avoid inserting MHSA at multiple scales or stacking additional attention blocks throughout the neck, because such configurations would significantly increase latency and memory consumption on edge devices and conflict with the lightweight design goal of EdgeFormer-YOLO.

3.2.3. Hierarchical Feature Fusion

Our model adopts a hierarchical feature fusion strategy rather than relying solely on the features from the final layer. By integrating features from multiple layers, the model can simultaneously capture high-level semantic information and low-level fine-grained details. This hierarchical fusion improves the model's effectiveness in detecting red fruits of varying sizes and in diverse environments. The fused features are fed into the detection head, ensuring that the model has access to comprehensive, multi-scale information for accurate detection. This approach enhances the model's ability to handle diverse and complex scenarios, thereby improving overall detection performance.

3.2.4. Activation Function

We replaced the original SiLU activation in the CBS structure with the arsinh activation, which is known as inverse hyperbolic sine activation. Figure 5 shows arsinh, SiLU, and ReLU together with their first derivatives over the range $[-100, 100]$. As shown in Figure 5, compared with SiLU, it offers an everywhere-positive, monotonically decreasing first derivative that is strictly bounded by 1, yielding gentler gradient variation in both positive and negative regimes and thus superior numeric stability in very deep nets. Furthermore, compared to the sigmoid function, arsinh avoids gradient vanishing when the input is excessively small or large. Its symmetric inverse hyperbolic form further provides exact zero-mean outputs without learnable scalars, reducing systematic bias and parameter count. In addition, near the origin, the function behaves similarly to the Tanh function. As the absolute value of the input increases, the function flattens and approaches the shape

of a logarithmic function. Its derivative is continuous everywhere, so that the function is smooth, which helps alleviate the vanishing gradient problem. Its formula is as follows:

$$\text{arsinh}(x) = \ln(x + \sqrt{x^2 + 1}) \tag{1}$$

where x is the input.

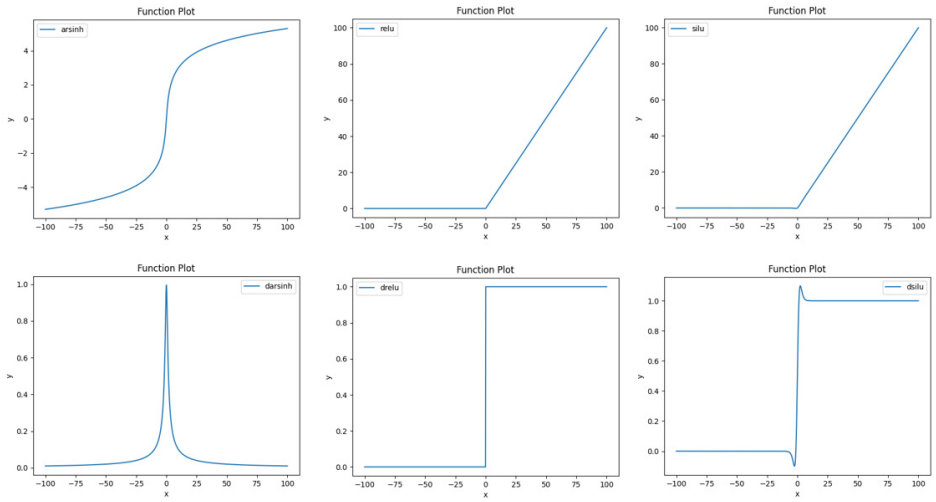

Figure 5. A comparison of activation functions arsinh, ReLU, SiLU, and their derivatives.

3.3. Evaluation Metrics

In this study, we employed average precision (AP), mean average precision (mAP), and the precision–recall (PR) curve as evaluation metrics.

3.3.1. Average Precision

AP represents the AP for a single class and can be categorized according to target size into small objects (smaller than 32 × 32 pixels), medium objects (between 32 × 32 and 96 × 96 pixels), and large objects (larger than 96 × 96 pixels). The equation is as follows:

$$\text{AP} = \sum_{i=1}^{N} P_i * \Delta R_i \tag{2}$$

where N is the number of recall points, P_i is the precision at the i^{th} point, and ΔR_i is the recall increment at the i^{th} step.

3.3.2. mAP

mAP is the mean of the AP values across all classes and is used as a comprehensive metric for multi-class detection tasks. It is further divided into mAP@0.5 (mAP50) and mAP@0.5:0.95 (mAP50–95). The former refers to the AP computed with an Intersection over Union (IoU) threshold of 0.5, which evaluates the model's detection capability under lenient overlap conditions. The latter calculates the AP over multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05, reflecting the model's overall robustness in predicting object locations. The formula is as follows:

$$\text{mAP} = \frac{1}{c}\sum_{i=1}^{c} \text{AP}_i \tag{3}$$

where c is the number of classes, and AP_i is the AP of class i.

3.3.3. Precision–Recall Curve

The PR curve is obtained by ranking the predicted probabilities for positive instances in descending order, adjusting the positive/negative threshold from the highest to the lowest probability, and recording precision and recall at each threshold. The calculation formulas for the PR curve are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

where TP refers to correctly detected bounding boxes, determined by whether the IoU between the predicted bounding box and the ground truth is greater than or equal to a predefined threshold. False Positive (FP) refers to incorrectly detected bounding boxes, determined when the IoU between the predicted bounding box and the ground truth is less than the predefined threshold. False Negative (FN) refers to missed detections, meaning the number of ground truth bounding boxes that were not detected by the model.

4. Results

4.1. Experimental Setup

To enhance the recognition generalization capability of the model and improve its practicality in real robotic harvesting environments, data augmentation was applied to the training set, including operations such as mirroring, rotation, cropping, scaling, and saturation adjustment. Each sample was also resized to 640 × 640 pixels. These augmentation techniques not only improve the model's generalization capability but also expand the dataset size and enhance its overall quality, thereby further improving the model's overall performance. As illustrated in Figure 6, the augmentation operations, crop, mirror, saturation shift, and rotation, are applied on-the-fly during training to simulate illumination and viewpoint variations encountered by robotic harvesters, which jointly expand sample diversity and improve model generalization in complex orchard scenes.

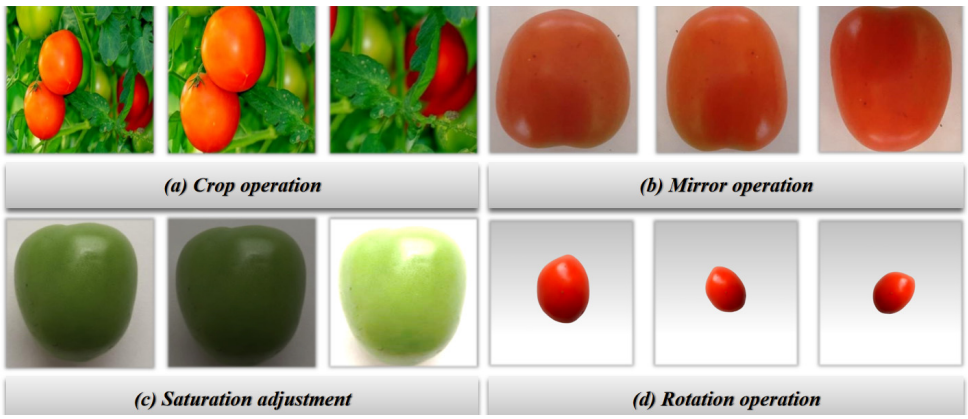

Figure 6. Offline augmentation strategies employed for the red fruit dataset. (**a**) random cropping, (**b**) horizontal mirroring, (**c**) saturation jittering, and (**d**) rotation.

In addition, the CPU selected for this experiment is an Intel (R) Core (TM) i7-12650HF with a base frequency of 2.3 GHz and equipped with 32 GB of RAM. The GPU used is an NVIDIA GeForce RTX 4090 with 24 GB of dedicated memory. The operating system is Windows 10 (x64) version. The neural network framework adopted is Torch 2.6.0, which supports PyTorch 2.6.0 and CUDA 12.4, making it suitable for training and inference of deep learning models. The programming language used is Python 3.9. During model

training, the number of epochs was set to 100, the batch size to 32, and the initial learning rate to 0.01, with the Adam optimizer employed. The complete experimental platform is summarized in Table 1, ensuring reproducibility of the reported results.

Table 1. Software and hardware configuration used for training and evaluation.

Configuration	Details
CPU	Intel (R) Core (TM) i7-12650HFCPU@2.3 GHz
CPU Memory	32 GB
GPU	NVIDIA GeForce RTX 4090
GPU Memory	24 GB
Operating System	Windows10 (x64)
Neural Network Framework	torch2.6.0 + cu124
Development Language	Python 3.9

4.2. Experimental Results

To comprehensively evaluate the performance of the proposed EdgeFormer-YOLO framework for red fruit detection in complex orchard environments, this paper conducted systematic comparative experiments against the current mainstream lightweight YOLO series models, including YOLOv3-Tiny, YOLOv5n, YOLOv6n, YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n, and YOLOv12n. All models were trained and tested using a unified dataset partitioning, training strategy, and hardware configuration to ensure fair and reproducible experimental results.

The experimental results, shown in Table 2, illustrate that our proposed EdgeFormer-YOLO achieved excellent performance across multiple key performance metrics. Specifically, the model achieved 95.7% mAP@0.5, an improvement of 2.2 percentage points over the baseline YOLOv8n model, ranking first among all compared SOTA models. These results clearly demonstrate that by introducing the MHSA and hierarchical feature fusion, the model achieves enhanced robustness and discriminative capabilities in complex scenarios such as occlusion, illumination variations, and scale changes. In terms of precision and recall, EdgeFormer-YOLO achieved balanced performance of 90.0% and 92.5%, respectively, demonstrating its strong object recall while maintaining a low false detection rate. Although some models, such as YOLOv10n and YOLOv11n, slightly outperform our proposed method in Precision, they all fall short of our proposed method in terms of mAP@0.5, indicating EdgeFormer-YOLO's overall superiority in detection performance. Furthermore, under the more stringent mAP@0.5–0.95 metric, EdgeFormer-YOLO achieved an AP of 80.7%, demonstrating its stable localization accuracy across different IoU thresholds. Although this metric is slightly lower than YOLOv9t, considering the economic benefits of orchard environment applications, false positives are more harmful than missed positives. EdgeFormer-YOLO still has greater practicality and potential for widespread adoption in edge device deployment scenarios.

Table 3 summarizes the measured FPS of EdgeFormer-YOLO and mainstream lightweight YOLO variants on an edge GPU (RTX 4090, FP16) in terms of FLOPs, parameter count, model size, and performance. It is shown that EdgeFormer-YOLO achieves 148.78 FPS on the edge GPU, RTX 4090. It still holds a net advantage of 34.48 FPS over the YOLOv12n (114.30 FPS), the latest YOLO model, while also leading by 1.8 percents in mAP@0.5, achieving the highest mAP@0.5 and high speed. Its FLOPs, 4.18 GB, and parameter count, 3.21 MB, maintain its nano-level lightweight design. The model size is only 6.35 MB, allowing it to be loaded into Jetson Orin's L2 cache in one go, eliminating the need for frequent VRAM refreshes and further reducing latency jitter. In short, EdgeFormer-YOLO not only boasts the fastest inference speed but also raises the detection limit, providing an extreme performance margin for real-time harvesting.

Table 2. Benchmark comparative results on the red fruit test set with the SOTA lightweight object detection models.

Model	Precision	Recall	mAP@0.5	mAP@0.5–0.95
YOLOv3-Tiny(u) [37]	0.907	0.937	0.933	0.805
YOLOv5nu [38]	0.913	0.936	0.944	0.819
YOLOv6n [39]	0.924	0.905	0.940	0.803
YOLOv8n [29]	0.894	0.933	0.935	0.798
YOLOv9t [40]	0.897	0.947	0.950	0.827
YOLOv10n [32]	0.932	0.895	0.938	0.823
YOLOv11n [41]	0.930	0.929	0.943	0.819
YOLOv12n [33]	0.914	0.936	0.939	0.826
Ours	0.900	0.925	0.957	0.807

Table 3. Comparison of computational efficiency and performance with the SOTA lightweight object detection models.

Model	FLOPs (G)	Parameters (M)	FPS	Model Size
YOLOv3-Tiny(u) [37]	9.56	12.17	550.71	23.31
YOLOv5nu [38]	3.92	2.65	214.79	5.31
YOLOv8n [29]	4.43	3.16	232.56	6.25
YOLOv9t [40]	4.24	2.13	70.14	4.74
YOLOv10n [32]	4.37	2.78	142.95	5.59
YOLOv11n [41]	3.31	2.62	175.05	5.35
YOLOv12n [33]	3.33	2.60	114.30	5.34
EdgeFormer-YOLO	4.18	3.21	148.78	6.35

Figure 7 also shows the representative detection visualization results of the model in an actual tomato ripeness detection task. It can be seen that EdgeFormer-YOLO accurately identifies and locates ripe and unripe fruit under varying lighting, occlusion, and fruit density conditions. Its bounding box regression is precise, with minimal false positives and missed detections, further validating its practical application value in agricultural robot vision systems.

4.3. Ablation Study

To systematically evaluate the contribution of each key module in EdgeFormer-YOLO to overall detection performance, this paper designed and conducted a series of ablation studies. This experiment sequentially removed core components of the model, including the MHSA and the arsinh activation function, and conducted comparative analysis using the same dataset and training configuration. The experimental results are shown in Table 4. All metrics were measured on the validation set to ensure comparability and reproducibility.

Table 4. Ablation experiment on the red fruit test set.

Model	Precision	Recall	mAP@0.5	mAP@0.5–0.95
w/o arsinh and MHSA	0.894	0.933	0.935	0.798
w/o MHSA	0.914	0.913	0.944	0.803
w/o arsinh	0.919	0.922	0.942	0.808
Ours	0.900	0.925	0.957	0.807

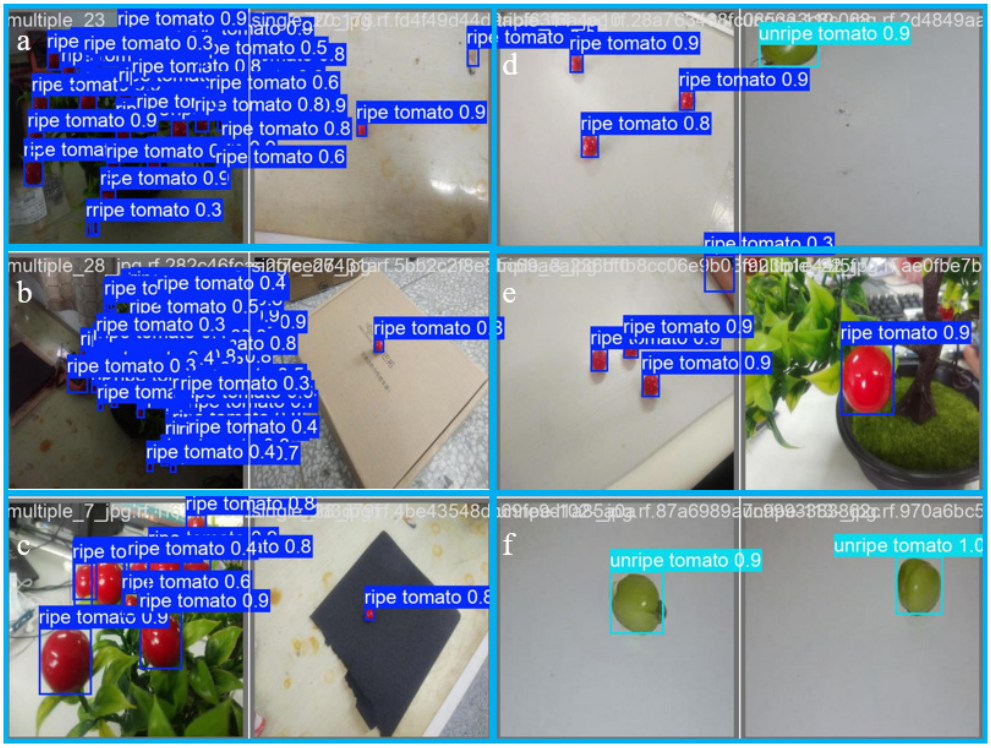

Figure 7. Qualitative results and visualization of the tomato maturity detection experiment under challenging orchard conditions. (**a**) Multiple fruit overlap vs. single fruit detection under strong light reflection (**b**) Multiple fruit overlap vs. single fruit detection (**c**) Multiple fruit overlap and foliage occlusion vs. single fruit detection (**d**) Different fruit identification at small scale vs. large scale (**e**) Same fruit identification at small scale vs. large scale, maintaining high confidence prediction even after edge quantization (**f**) Single fruit detection after preprocessing vs. before preprocessing.

In addition, Table 5 shows the computational performance metrics after removing MHSA and arsinh, which presents an ablation study dissecting the computational impact of core design choices. Removing arsinh unlocks a 45.2% FPS surge (215.73 vs. 148.78) without altering FLOPs or parameters, exposing its substantial inference overhead despite being a parameter-free operation. Stripping MHSA yields a modest 4.6% FPS gain (155.67 vs. 148.78) but incurs a 4.6% FLOPs penalty (4.43 G vs. 4.18 G), revealing MHSA as a parameter-efficient mechanism that compresses computational cost via optimized global context aggregation. Critically, both ablations sacrifice detection accuracy (per validation mAP). The full EdgeFormer-YOLO model therefore embraces these components as a strategic trade-off: arsinh ensures quantization stability while MHSA maximizes accuracy per FLOP, accepting a controlled speed reduction for superior end-to-end performance and deployment robustness.

Table 5. Ablation experiment of computational efficiency and performance.

Model	FLOPs (G)	Parameters (M)	FPS	Model Size
w/o arsinh	4.18	3.21	215.73	6.35
w/o MHSA	4.43	3.16	155.67	6.25
EdgeFormer-YOLO	4.18	3.21	148.78	6.35

As Table 4 shows, after removing the MHSA module, the model's mAP@0.5 performance dropped from 95.7% to 94.4%, a decrease of 1.3 percentage points. Simultaneously, recall dropped from 92.5% to 91.3%. This demonstrates that MHSA plays a key role in improving the model's ability to detect occluded fruit. Because fruit in orchards is often

partially obscured by branches and leaves, traditional convolutional operations struggle to model long-range spatial dependencies. However, MHSA effectively captures the contextual associations between obscured and visible fruit through a global self-attention mechanism, significantly reducing the missed detection rate. Additionally, after removing the arsinh activation function and reverting to the original SiLU function, the model's mAP@0.5 dropped to 94.2%, while Precision and Recall dropped to 91.9% and 92.2%, respectively. This demonstrates the positive impact of the arsinh function on improving model training stability and convergence efficiency. Its non-zero derivative effectively mitigates the vanishing gradient problem across the entire real domain, particularly during edge quantization deployment, maintaining the continuity and numerical stability of feature representation, thereby improving the model's robustness in practical applications. Furthermore, when both the MHSA and arsinh modules are removed simultaneously, the model's performance degrades most significantly, with mAP@0.5 dropping to 93.5%, comparable to the original YOLOv8n baseline model. This result further validates the synergistic effect between the various modules in the proposed method. Specifically, MHSA provides stronger global modeling capabilities, while the arsinh function ensures the stability of deep networks during training and deployment. The combination of these two ensures a better balance between accuracy and efficiency.

In summary, the ablation results fully validate the effectiveness and necessity of the key modules in EdgeFormer-YOLO. MHSA and the hierarchical feature fusion play a key role in improving the model's adaptability to complex environments, while the arsinh activation function provides important guarantees for training stability and robustness in edge deployments.

5. Discussion

This paper conducts an in-depth analysis of EdgeFormer-YOLO's detection performance in complex orchard scenes, focusing on the mechanisms by which its key design contributes to improved accuracy, its advantages over existing methods, and its feasibility and potential limitations in practical agricultural robot deployment.

Above all, from a model architecture perspective, EdgeFormer-YOLO effectively alleviates the problem of the limited local receptive field of traditional CNNs under occlusion conditions by introducing MHSA. Experimental results show that the introduction of MHSA significantly improves the model's detection capability when fruit is partially occluded by branches and leaves, with a mAP@0.5 metric improvement of 1.3 percentage points compared to a baseline model without MHSA. This result validates the necessity of global context modeling in agricultural vision tasks, particularly in environments with dense fruit and complex backgrounds. MHSA helps the model establish long-range spatial dependencies, thereby enhancing the recognition of occluded objects. We also considered windowed or local variants. However, on edge devices, the additional block and masking operations introduced by windowed attention disrupt the CUDA core-friendly one-dimensional continuous memory access pattern, leading to non-linear latency increases. Given the severely unfavorable cost–benefit ratio, we abandoned this approach. More complex local designs, such as Shifted Windows, are theoretically feasible but would significantly increase deployment engineering complexity. In addition, Table 5's ablation experiments also show that removing MHSA only results in a 4.6% FPS improvement (155.67 vs. 148.78), conversely proving that its computational burden is reasonable and indispensable. Additionally, the introduction of a hierarchical feature fusion mechanism mitigates response bias caused by strong light reflections and background interference. Notably, the combined use of this mechanism with MHSA improves detection accuracy

without significantly increasing the number of model parameters. In this way, the design meets the lightweight model requirements of edge devices.

In terms of activation functions, this paper adopts arsinh instead of the traditional SiLU function to mitigate the vanishing gradient problem during edge quantization and improve the model's numerical stability under low-precision deployment conditions. Ablation experiments show that the introduction of arsinh improves mAP@0.5 by 1.5 percentage points and slightly accelerates model convergence. This result demonstrates that the choice of activation function not only affects the efficiency of gradient propagation during model training but also, to a certain extent, its robustness when deployed on edge devices. Compared to other non-linear activation functions, arsinh has nonzero derivatives throughout the real domain, which helps maintain continuity and stability in feature representation during quantization. Compared to other methods that improve accuracy by stacking convolutional layers or increasing network width, EdgeFormer-YOLO achieves more efficient feature extraction and representation while maintaining a lightweight structure. Moreover, the convergence behavior is visualized in Figure 8, where both training losses and validation metrics plateau within 80 epochs, confirming the stability conferred by the arsinh activation and MHSA module. We also provide the training curves of advanced models for comparison in the appendix, as shown in Figures A1–A8, to further demonstrate the superiority of our proposed model. Moreover, Figure 9 presents a quantitative comparison of the confusion matrix between EdgeFormer-YOLO and seven latest lightweight YOLO systems on the tomato ripeness classification task. EdgeFormer-YOLO achieved the highest classification accuracy and the fewest misclassifications in both the "ripe tomato" and "immature tomato" categories. This demonstrates its high robustness and reliability in determining fruit ripeness in complex orchard environments. However, it should be noted that in orchard target detection tasks, we often encounter situations where one instance corresponds to multiple bounding boxes and missed detections occur. Therefore, the number of confusion matrices varies among different models.

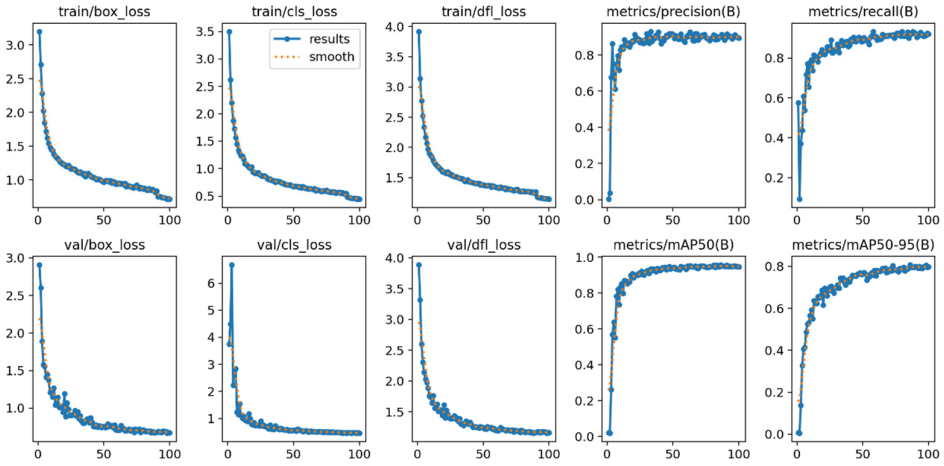

Figure 8. Training curves of EdgeFormer-YOLO.

As shown in Tables 2 and 3, EdgeFormer-YOLO achieves 95.7% mAP@0.5 and 148.78 FPS with a nano-level resource budget of 4.18 GFLOPs, 3.21 M parameters, and 6.35 MB. While YOLOv12n outperforms our model in Precision and Recall, its detection efficiency is too low, which runs at only 76.8% the speed of our model, making it unsuitable for orchard detection. In addition, YOLOv10n and YOLOv6n slightly outperform our model in precision; they suffer from relatively low recall, resulting in higher false negative rates in scenarios with dense fruit and severe occlusion. In contrast, EdgeFormer-YOLO performs better on the mAP@0.5 metric, indicating stronger overall detection capabilities across

different crossover ratio thresholds, particularly in orchard environments with varying fruit sizes and complex lighting conditions, demonstrating greater robustness. Furthermore, from the perspective of agricultural robot vision systems, FN often poses a greater systemic risk than FP. False positives can be corrected through subsequent robotic arm verification or multiple recognitions, while false negatives may result in unharvested fruit, directly impacting system integrity and operational efficiency. Therefore, in the trade-off between precision and recall, improving recall and overall detection capabilities is more practical than simply pursuing high precision. EdgeFormer-YOLO achieves the highest detection accuracy and high frame rate in this trade-off, while maintaining nano-level computational and memory overhead, providing an immediately deployable resource-efficiency optimal solution for real-time harvesting robots.

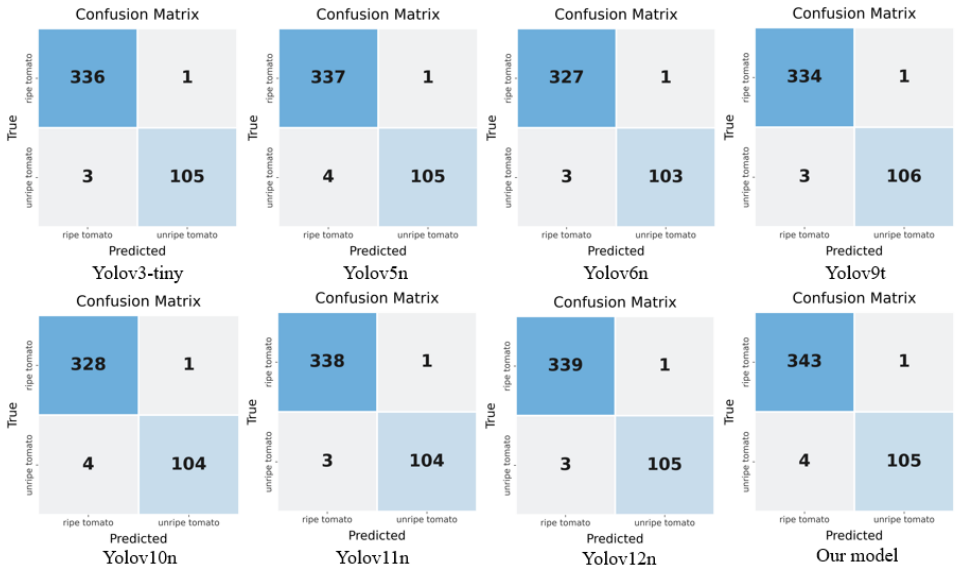

Figure 9. Comparison of confusion matrices of various models on the classification task included in tomato maturity detection.

Furthermore, from the FPS dimension in Table 3, EdgeFormer-YOLO achieves 148.78 FPS, setting a new benchmark for balancing real-time inference and performance. In a typical robotic arm control closed loop at 30 Hz, single frame inference requires only about 6.7 ms, leaving a time margin of about 26.6 ms for image acquisition, path planning, and servo drive, easily supporting continuous high-speed operations. It is worth noting that this speed is achieved while maintaining competitive model efficiency with 4.18 GFLOPs and 3.21 M parameters. MHSA's global modeling and arsinh's quantization friendliness jointly compress redundant calculations, making balanced design no longer a burden on speed. While YOLOv3-Tiny and earlier models achieve higher FPS, EdgeFormer-YOLO's advantage lies in its superior accuracy-FPS synergy among recent lightweight architectures as shown in Table 3. Therefore, its FPS advantage means that in real orchard scenarios, it can improve harvesting speed, shorten the single-vehicle operation cycle, and leave sufficient computing power margin for subsequent multi-machine collaborative harvesting.

Although EdgeFormer-YOLO achieves promising detection performance overall, it may still face challenges in some extreme orchard scenarios. Under strong back-lighting or specular highlights, ripe fruits can be visually suppressed by the surrounding canopy, making them harder to distinguish from the background. When fruits are heavily occluded by dense leaves and branches, only very small visible regions remain, which can be confused with background clutter or assigned low confidence. In addition, very small and distant fruits, or fruits whose color is very similar to the surrounding canopy, are more likely to be mis-localized or missed. These potential limitations are common in practical

orchard detection tasks and also suggest directions for further improvement. The dataset used in this study also has inherent limitations that may affect the generalization of the detector. Although it contains 2607 annotated images, Roboflow does not provide per-image acquisition metadata, precluding a precise source-ratio analysis. The model may not fully capture the appearance variations caused by different cultivars, pruning systems, camera viewpoints, weather patterns or seasonal stages. Future work will consider introducing multi-source data fusion and transfer learning strategies to improve the model's adaptability to diverse agricultural ecosystems. Furthermore, field sunlight angle, cloud cover, and greenhouse supplemental lighting can introduce color temperature drift of more than 2000 K within seconds. This drift causes drastic changes in the color features of the same fruit between adjacent frames. As shown in Figure 7, the false detection rate of the single frame model increases when strong light reflection and shadow coexist. Introducing the ConvLSTM temporal modeling module and using a sliding window to fuse spatiotemporal context can utilize short temporal consistency to smooth out illumination perturbations. In addition, for edge hardware computing power and memory hard limitations, although the current model of 4.18 GFLOPs/6.35 MB meets the Jetson Orin L2 cache, it is still redundant on the lower-power Nano 4 GB or RK3588 NPU. In the future, we can adopt a compression strategy of structured pruning combined with offline knowledge distillation for practical deployment.

In summary, EdgeFormer-YOLO significantly improves the accuracy and robustness of red fruit detection in complex orchard environments while maintaining a lightweight structure by integrating a global attention mechanism, feature pyramid network, hierarchical feature fusion mechanism, and a novel activation function. This research provides a viable visual solution for agricultural robots to achieve efficient and autonomous harvesting in natural environments and lays the foundation for its subsequent application in edge smart agricultural equipment.

6. Conclusions

This paper proposes EdgeFormer-YOLO, a lightweight multi-attention detection framework for real-time red fruit detection in complex orchard environments. While maintaining the efficiency of the YOLO family of single-stage detection architectures, this model effectively improves detection accuracy and robustness under complex conditions such as occlusion, strong lighting, and scale variations by using MHSA, feature pyramid network, hierarchical feature fusion mechanism, and arsinh activation function. Experimental results show that EdgeFormer-YOLO achieves 95.7% mAP@0.5 on the red fruit dataset, a 2.2 percentage point improvement over the YOLOv8n baseline model. It also achieves a good balance between precision and recall, reaching 90.0% and 92.5%, respectively. In addition, with a 30.2% relative speedup over YOLOv12n (148.78 vs. 114.30 FPS) and a 1.8-point mAP@0.5 lead, EdgeFormer-YOLO establishes new state-of-the-art efficiency on the accuracy-speed Pareto frontier. Ablation experiments further validate the effectiveness of each key module, particularly the significant contribution of MHSA in alleviating missed detections caused by occlusion and the arsinh function in improving training stability and adaptability to edge deployments. In summary, EdgeFormer-YOLO achieves a good trade-off between accuracy, efficiency, and deployment feasibility, providing a reliable visual perception solution for agricultural robots to achieve efficient and autonomous fruit picking in natural orchard environments.

Our future work will focus on the following directions: (1) exploring deployment optimization strategies for the model on lower-computing edge devices, such as quantitative pruning and knowledge distillation; (2) introducing temporal information modeling to improve the model's continuous detection capabilities in dynamic video

streams; and (3) extending the model to fruit detection tasks across multiple varieties and growth stages, enhancing its generalization and practical value in different agricultural ecological environments.

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

Funding: This research was funded by the start-up grant and teaching start-up grant from City University of Hong Kong, and grant numbers are 9610735 and 6000926, respectively.

Data Availability Statement: The data that support the findings of this study are openly available. Open Data Portal at https://universe.roboflow.com/yassero/toamate/dataset/12 (accessed on 1 July 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

MHSA	Multi-head Self-attention
CNN	Convolutional Neural Network
NMS	Non-maximum Suppression
YOLO	You Only Look Once
SVM	Support Vector Machines
SOTA	State-of-the-art
AP	Average Precision
mAP	Mean Average Precision
PR	Precision–Recall

Appendix A

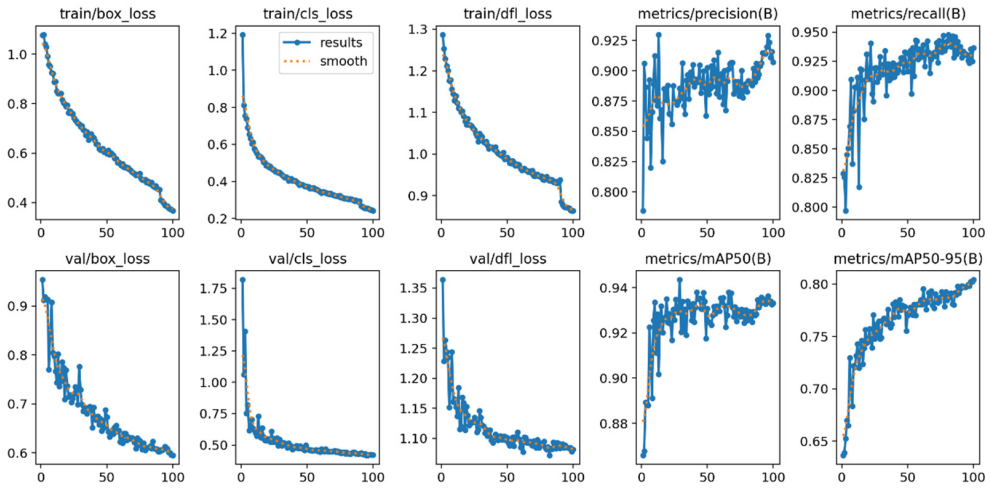

Figure A1. Training curves of YOLOv3-Tiny(u).

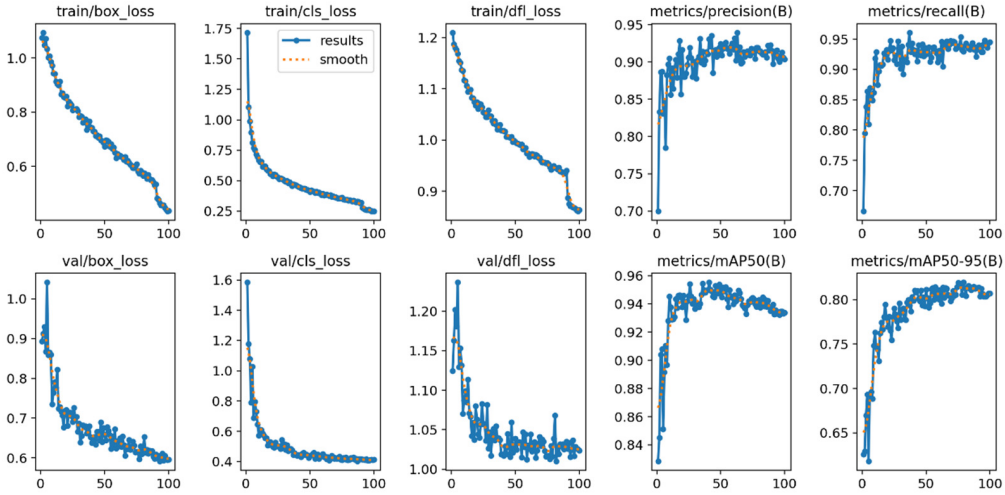

Figure A2. Training curves of YOLOv5nu.

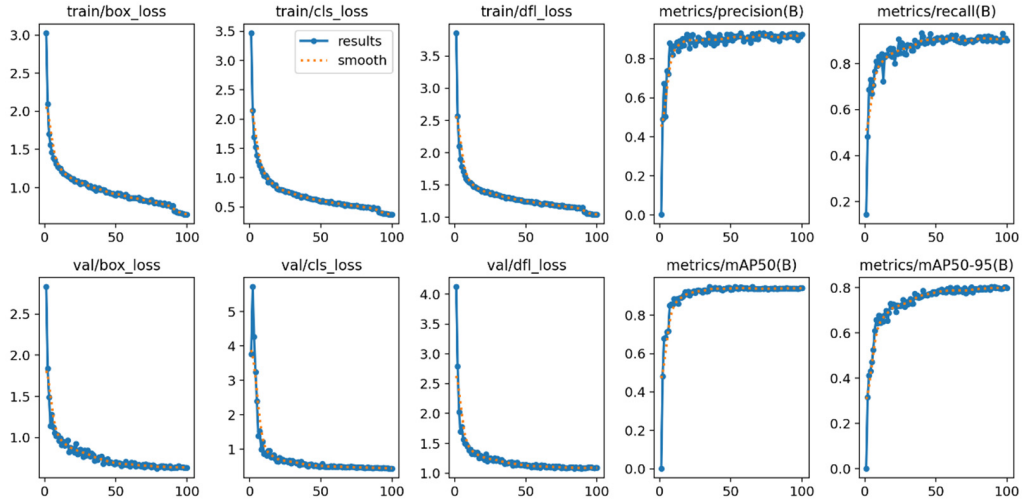

Figure A3. Training curves of YOLOv6n.

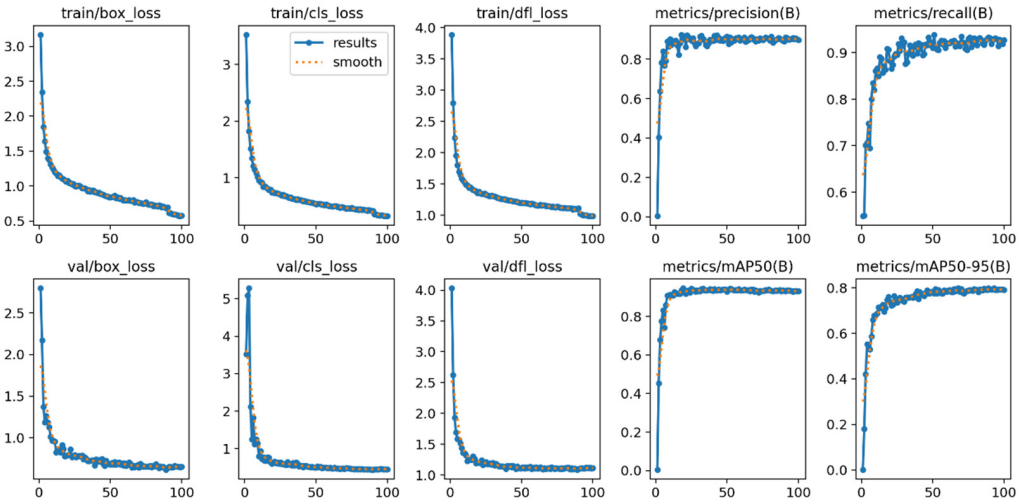

Figure A4. Training curves of YOLOv8n.

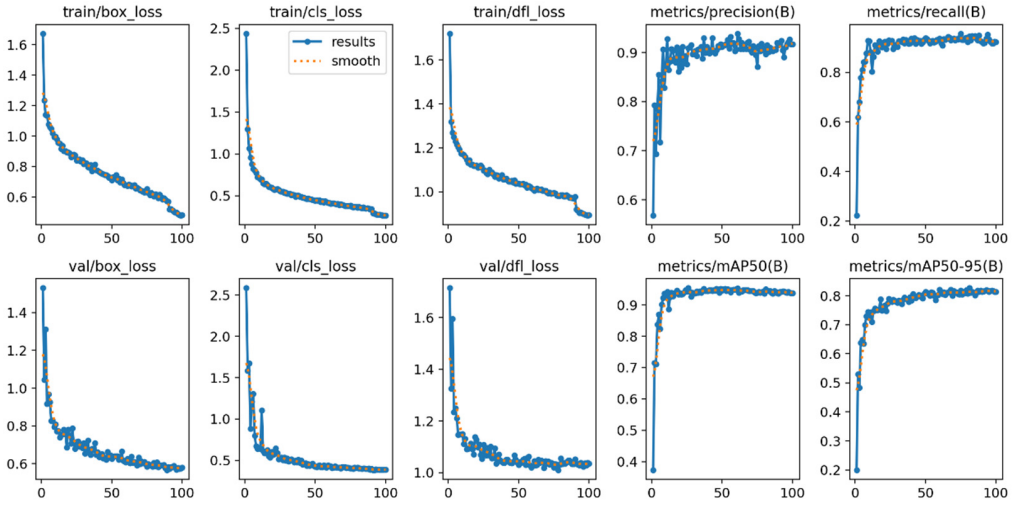

Figure A5. Training curves of YOLOv9t.

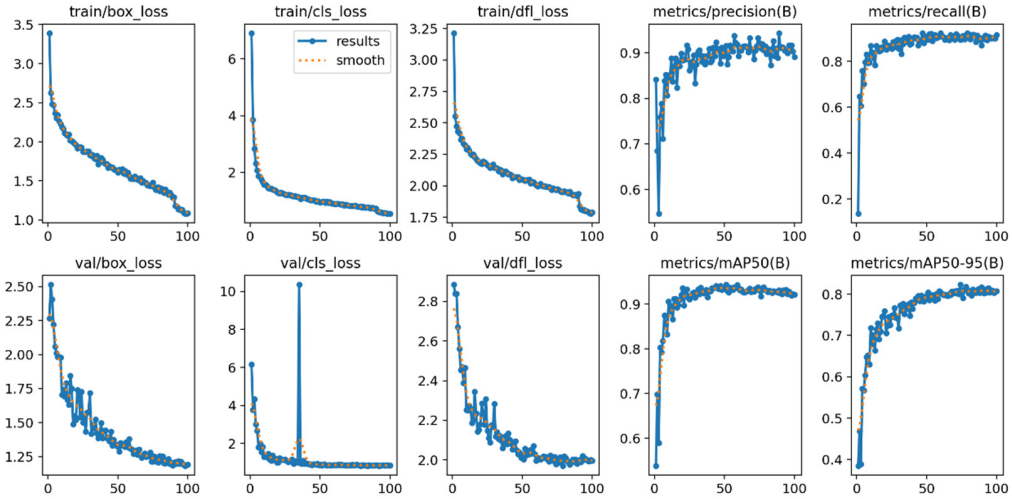

Figure A6. Training curves of YOLOv10n.

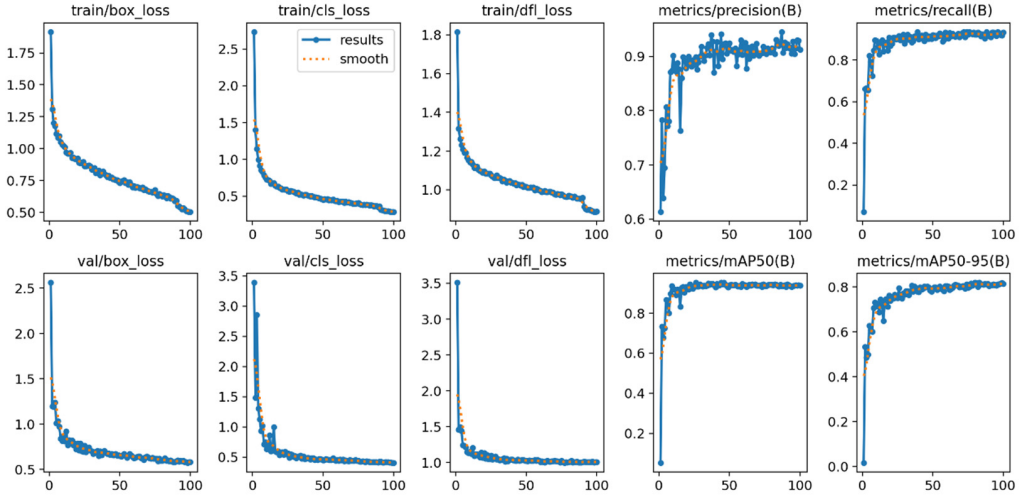

Figure A7. Training curves of YOLOv11n.

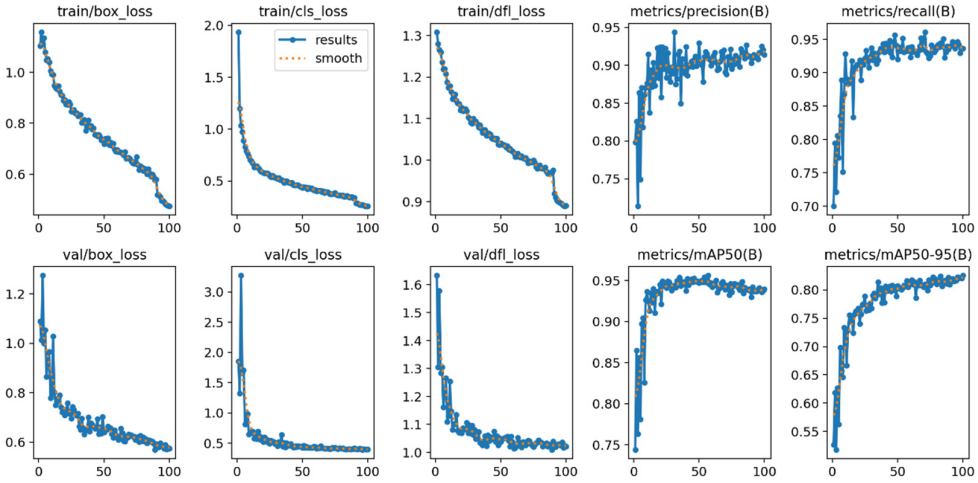

Figure A8. Training curves of YOLOv12n.

References

1. Lin, G.; Zhu, L.; Li, J.; Zou, X.; Tang, Y. Collision-Free Path Planning for a Guava-Harvesting Robot Based on Recurrent Deep Reinforcement Learning. *Comput. Electron. Agric.* **2021**, *188*, 106350. [CrossRef]
2. Rajendran, V.; Debnath, B.; Mghames, S.; Mandil, W.; Parsa, S.; Parsons, S.; Ghalamzan-E, A. Towards Autonomous Selective Harvesting: A Review of Robot Perception, Robot Design, Motion Planning and Control. *J. Field Robot.* **2024**, *41*, 2247–2279. [CrossRef]
3. Montoya-Cavero, L.-E.; Díaz De León Torres, R.; Gómez-Espinosa, A.; Escobedo Cabello, J.A. Vision Systems for Harvesting Robots: Produce Detection and Localization. *Comput. Electron. Agric.* **2022**, *192*, 106562. [CrossRef]
4. Yin, H.; Sun, Q.; Ren, X.; Guo, J.; Yang, Y.; Wei, Y.; Huang, B.; Chai, X.; Zhong, M. Development, Integration, and Field Evaluation of an Autonomous Citrus-harvesting Robot. *J. Field Robot.* **2023**, *40*, 1363–1387. [CrossRef]
5. Wang, W.; Li, C.; Xi, Y.; Gu, J.; Zhang, X.; Zhou, M.; Peng, Y. Research Progress and Development Trend of Visual Detection Methods for Selective Fruit Harvesting Robots. *Agronomy* **2025**, *15*, 1926. [CrossRef]
6. Riehle, D.; Reiser, D.; Griepentrog, H.W. Robust Index-Based Semantic Plant/Background Segmentation for RGB- Images. *Comput. Electron. Agric.* **2020**, *169*, 105201. [CrossRef]
7. Huang, Y.; Xu, S.; Chen, H.; Li, G.; Dong, H.; Yu, J.; Zhang, X.; Chen, R. A Review of Visual Perception Technology for Intelligent Fruit Harvesting Robots. *Front. Plant Sci.* **2025**, *16*, 1646871. [CrossRef]
8. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision–ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; Volume 12346, pp. 213–229, ISBN 978-3-030-58451-1.
9. Li, Y.; Miao, N.; Ma, L.; Shuang, F.; Huang, X. Transformer for Object Detection: Review and Benchmark. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107021. [CrossRef]
10. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
11. Vijayakumar, A.; Vairavasundaram, S. YOLO-Based Object Detection Models: A Review and Its Applications. *Multimed. Tools Appl.* **2024**, *83*, 83535–83574. [CrossRef]
12. Zhu, F.; Zhang, W.; Wang, S.; Jiang, B.; Feng, X.; Zhao, Q. Apple-Harvesting Robot Based on the YOLOv5-RACF Model. *Biomimetics* **2024**, *9*, 495. [CrossRef]
13. Liu, Z.; Rasika, D.; Abeyrathna, R.M.; Mulya Sampurno, R.; Massaki Nakaguchi, V.; Ahamed, T. Faster-YOLO-AP: A Lightweight Apple Detection Algorithm Based on Improved YOLOv8 with a New Efficient PDWConv in Orchard. *Comput. Electron. Agric.* **2024**, *223*, 109118. [CrossRef]
14. Li, J.; Wu, K.; Zhang, M.; Chen, H.; Lin, H.; Mai, Y.; Shi, L. YOLOv8s-Longan: A Lightweight Detection Method for the Longan Fruit-Picking UAV. *Front. Plant Sci.* **2025**, *15*, 1518294. [CrossRef] [PubMed]
15. De-la-Torre, M.; Zatarain, O.; Avila-George, H.; Muñoz, M.; Oblitas, J.; Lozada, R.; Mejía, J.; Castro, W. Multivariate Analysis and Machine Learning for Ripeness Classification of Cape Gooseberry Fruits. *Processes* **2019**, *7*, 928. [CrossRef]
16. Taner, A.; Mengstu, M.T.; Selvi, K.Ç.; Duran, H.; Kabaş, Ö.; Gür, İ.; Karaköse, T.; Gheorghiță, N.-E. Multiclass Apple Varieties Classification Using Machine Learning with Histogram of Oriented Gradient and Color Moments. *Appl. Sci.* **2023**, *13*, 7682. [CrossRef]

17. Kumari, A.; Singh, J. Designing of Guava Quality Classification Model Based on ANOVA and Machine Learning. *Sci. Rep.* **2025**, *15*, 33920. [CrossRef]
18. Santos Pereira, L.F.; Barbon, S.; Valous, N.A.; Barbin, D.F. Predicting the Ripening of Papaya Fruit with Digital Imaging and Random Forests. *Comput. Electron. Agric.* **2018**, *145*, 76–82. [CrossRef]
19. Lai, Y.; Cao, A.; Gao, Y.; Shang, J.; Li, Z. Advancing Efficient Brain Tumor Multi-Class Classification: New Insights from the Vision Mamba Model in Transfer Learning. *Int. J. Imaging Syst. Technol.* **2025**, *35*, e70177. [CrossRef]
20. Qin, H.; Xu, T.; Tang, Y.; Xu, F.; Li, J. OSFormer: One-Step Transformer for Infrared Video Small Object Detection. *IEEE Trans. Image Process.* **2025**, *34*, 5725–5736. [CrossRef]
21. Shang, J.; Lai, Y. An Optimal U-Net++ Segmentation Method for Dataset BUSI. In Proceedings of the 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 29–31 March 2024; pp. 1015–1019.
22. Amjoud, A.B.; Amrouch, M. Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access* **2023**, *11*, 35479–35516. [CrossRef]
23. Stasenko, N.; Shadrin, D.; Katrutsa, A.; Somov, A. Dynamic Mode Decomposition and Deep Learning for Postharvest Decay Prediction in Apples. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2518411. [CrossRef]
24. Halstead, M.; McCool, C.; Denman, S.; Perez, T.; Fookes, C. Fruit Quantity and Ripeness Estimation Using a Robotic Vision System. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2995–3002. [CrossRef]
25. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
26. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
27. Lu, S.; Chen, W.; Zhang, X.; Karkee, M. Canopy-Attention-YOLOv4-Based Immature/Mature Apple Fruit Detection on Dense-Foliage Tree Architectures for Early Crop Load Estimation. *Comput. Electron. Agric.* **2022**, *193*, 106696. [CrossRef]
28. Yang, C.; Liu, J.; He, J. A Lightweight Waxberry Fruit Detection Model Based on YOLOv5. *IET Image Process.* **2024**, *18*, 1796–1808. [CrossRef]
29. Varghese, R.; Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; pp. 1–6.
30. Wu, T.; Miao, Z.; Huang, W.; Han, W.; Guo, Z.; Li, T. SGW-YOLOv8n: An Improved YOLOv8n-Based Model for Apple Detection and Segmentation in Complex Orchard Environments. *Agriculture* **2024**, *14*, 1958. [CrossRef]
31. Wang, C.; Wang, H.; Han, Q.; Zhang, Z.; Kong, D.; Zou, X. Strawberry Detection and Ripeness Classification Using YOLOv8+ Model and Image Processing Method. *Agriculture* **2024**, *14*, 751. [CrossRef]
32. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
33. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**, arXiv:2502.12524.
34. 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]
35. 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]
36. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
37. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
38. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Zeng, Y.; Wong, C.; Montes, D.; et al. *Ultralytics/Yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*; Zenodo: Geneva, Switzerland, 2022. [CrossRef]
39. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
40. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision–ECCV 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2025; Volume 15089, pp. 1–21, ISBN 978-3-031-72750-4.
41. Viso.AI; Boesch, G. Yolov11: A New Iteration of "You Only Look Once". 2024. Available online: https://viso.ai/computer-vision/yolov11/ (accessed on 21 October 2024).