Method for lightweight litchi fruit detection in complex environments via improved YOLO11n-MAS

Jing Ling[1*], Yaozhong Mei[2], Zheng Zuo[3], Shengkang Pan[2]
(1. *School of Artificial Intelligence, Guangzhou Maritime University, Guangzhou 510725, China*;
2. *School of Low-altitude Equipment and Intelligent Control, Guangzhou Maritime University, Guangzhou 510725, China*;
3. *School of Electronics and Electrical Engineering, Ningxia University, Yinchuan 750021, China*)

Abstract: Accurate and real-time detection of litchi fruit is crucial for the development of automated harvesting systems and the advancement of precision agriculture. This study addresses the challenge of litchi fruit detection in complex orchard environments characterized by branch occlusion, fruit overlap, and variable lighting conditions. A lightweight detection model, YOLO11n-MAS, was proposed to achieve efficient and robust fruit identification. A self-constructed Zengcheng litchi dataset was first established to provide high-quality data for training and validation, thereby improving the model's detection performance. The YOLO11n-MAS model was built on the YOLO11n architecture, incorporating a novel C3k2-EMBC module and a Bi-Level Routing Attention (BRA) mechanism to reconstruct the backbone network, utilizing a parameter-efficient Slimneck to optimize the neck, and introducing a specialized Detect_LSDECD module into the detection head to balance performance and efficiency. Experimental results show that the YOLO11n-MAS model achieves a mean average precision (mAP50) of 0.916, outperforming other advanced lightweight models including YOLOv8n and the baseline YOLO11n. Additionally, the model's parameter size is only 2.34 M, and its GFLOPs is 5.5, representing reductions of 9.4% and 12.7%, respectively, compared to YOLO11n, significantly lowering computational costs while maintaining high detection performance. The results demonstrate the effectiveness of the C3k2-EMBC, BRA, Slimneck, and Detect_LSDECD modules. The development of the YOLO11n-MAS model can provide an efficient and lightweight solution for litchi fruit detection in resource-constrained environments, offering strong technical support for intelligent harvesting systems.
Keywords: litchi, lightweight network, YOLO11n, complex environment, fruit detection
DOI: 10.25165/j.ijabe.20261904.10519

Citation: Ling J, Mei Y Z, Zuo Z, Pan S K. Method for lightweight litchi fruit detection in complex environments via improved YOLO11n-MAS. Int J Agric & Biol Eng, 2026; 19(4): 227–234.

1 Introduction

Litchi is a significant economic crop in Southern China. However, the litchi industry faces challenges due to labor-intensive harvesting and rising workforce costs. The integration of agricultural robotics for automated harvesting is essential for modern orchard management, and precise fruit detection is a critical prerequisite for this process[1,2]. Machine learning technologies have been widely adopted across various agricultural tasks[3].

The integration of machine vision is pivotal for modern orchard management, particularly for robotic harvesting. In object detection, algorithms are generally categorized into two-stage and single-stage detectors. Early research often utilized two-stage algorithms or high-precision classification networks. For instance, Habaragamuwa et al.[4] employed deep convolutional neural networks (DCNN) for strawberry classification, while Liu et al.[5] improved Faster R-CNN for high precision. However, two-stage models typically suffer from high computational latency (e.g., 0.54 s per image), rendering them unsuitable for the real-time requirements of agricultural robotics. While earlier lightweight models like LeNet have been explored, their architectures are often limited in efficiency when processing large-scale unstructured data.

To circumvent speed bottlenecks, single-stage detectors, exemplified by the YOLO series, have become the leading choice for real-time applications[6]. Recent studies have aimed to balance robustness and efficiency by integrating attention mechanisms and lightweight architectures. For example, Li et al.[7] combined YOLOv4-tiny with the CBAM mechanism for pepper detection. Gao et al.[8] proposed YOLO-FL for an intelligent flower-thinning robot, achieving a mean average precision of 79.97%. Similarly, Chen et al.[9] developed a lightweight method for tomato leaf disease recognition based on an improved YOLO11s. These advancements demonstrate the potential of lightweight models for edge deployment[10].

Despite these successes, detecting litchi fruits in unstructured orchard environments remains a significant challenge. Litchi fruits are characterized by dense clusters, color similarity to foliage, severe overlap, and minute target features[11]. Standard lightweight YOLO variants often lack sufficient feature extraction capabilities for such complex scenarios, leading to missed detections, while high-precision models are too computationally heavy for edge devices. Therefore, it is essential to develop a model that can capture critical features under occlusion conditions while

Received date: 2026-02-14 **Accepted date:** 2026-06-02
Biographies: Yaozhong Mei, Undergraduate student, research interests: intelligent detection and intelligent information processing, Email: yao3550344713@163.com; **Zheng Zuo**, Graduate student, research interests: intelligent detection and intelligent information processing, Email: zuozheng713@163.com; **Shengkang Pan**, Undergraduate student, research interests: electronic information engineering, Email: 2078585486@qq.com.
***Corresponding author: Jing Ling**, Associate professor, research interest: Intelligent Detection and Intelligent Information Processing. School of Artificial Intelligence, Guangzhou Maritime University, Guangzhou 510725, China. Tel: +86-15209508892, Email: lingjing0519@163.com.

minimizing computational overhead.

To address the aforementioned challenges, the primary objective of this study is to develop a robust, lightweight, and real-time litchi detection method suitable for edge deployment on intelligent harvesting robots. Based on the YOLO11 object detection framework, this research proposes an improved model named YOLO11n-MAS to balance detection precision and computational efficiency. Specifically, this study aims to achieve the following research objectives:

1. Dataset construction for complex environments: To build and augment a comprehensive litchi dataset captured under diverse natural conditions (e.g., variable lighting, foliage occlusion, and dense fruit overlapping), thereby providing a highly realistic data foundation for model training and evaluation.

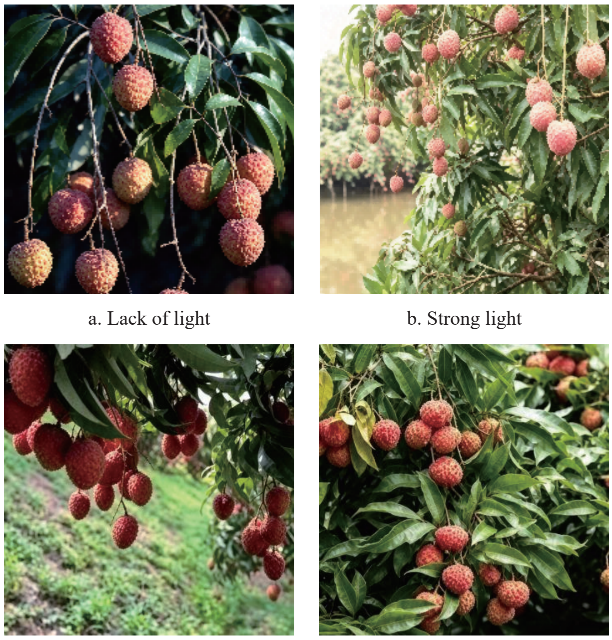
2. Algorithmic optimization for accuracy and efficiency: To systematically optimize the network architecture of the baseline model—spanning the backbone, neck, and detection head. The objective is to enhance the model's feature extraction capability under severe occlusion and refine the localization accuracy for small and dense targets, while strictly maintaining a compact model size and low computational latency.

3. Practical validation and system development: To rigorously evaluate the robustness of the proposed model against current state-of-the-art lightweight algorithms, and to develop a scenario-specific interactive demonstration system. This aims to bridge the gap between theoretical model design and practical application, verifying the feasibility of the algorithm for real-world automated harvesting tasks.

2 Materials and methods

2.1 Dataset acquisition and construction

The experimental dataset was constructed using data collected from standardized litchi orchards in Zengcheng District, Guangzhou, China. Image acquisition was performed during the peak maturity period from June to July 2024 using a Canon M100 camera. To ensure the robustness of the detection model in unstructured environments, images were captured under diverse conditions, including varying lighting angles (front, back, and scattered light) and complex spatial distributions (foliage occlusion, fruit overlap, and dense clustering). Additionally, high-quality images from open-source platforms were integrated to expand data diversity. After rigorous screening and cleaning to remove blurry, overexposed, and underexposed low-quality samples, a total of 3939 raw images were obtained. Figure 1 illustrates representative samples of the collected dataset.



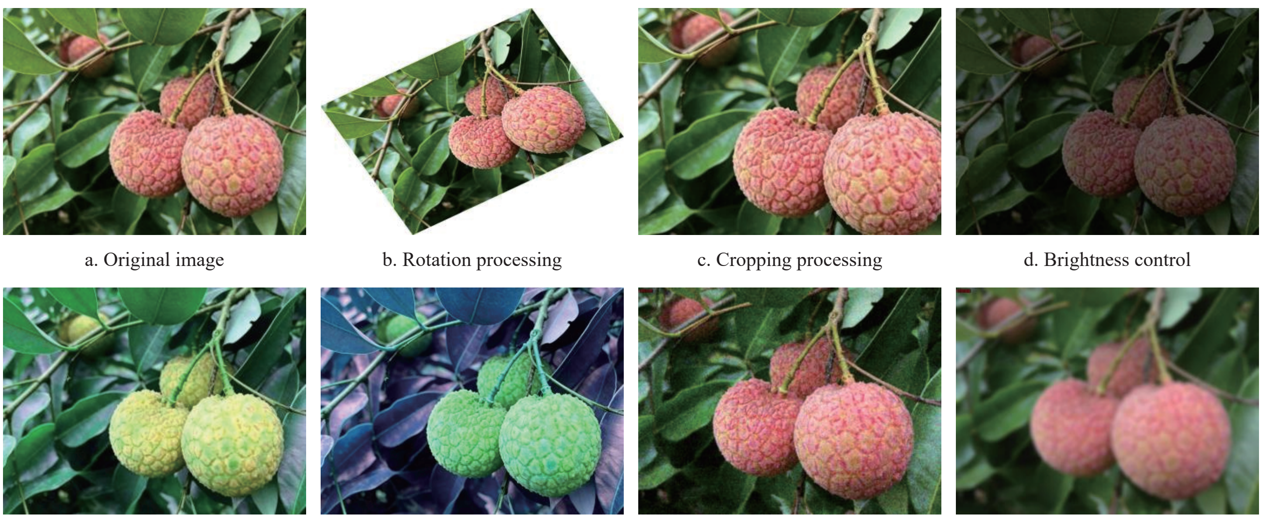
a. Lack of light b. Strong light

c. Branches and foliage occlusion d. Fruit occlusion

Figure 1 Samples of the images of litchi fruit

2.2 Data augmentation

To prevent model overfitting and enhance generalization capabilities, offline data augmentation techniques were applied to the raw dataset. Operations included random rotation, horizontal/vertical flipping, brightness adjustment, and noise addition (Gaussian blur, Poisson noise)[12]. These methods simulated environmental interferences such as camera shake and varying weather conditions, thereby improving the model's detection accuracy and adaptability to complex orchard environments. After augmentation, the dataset was expanded to 5437 images. The dataset was randomly partitioned into training, validation, and test sets in a ratio of 7:2:1, containing 3806, 1087, and 544 images, respectively. Figure 2 illustrates the visual effects of selected data augmentation methods.



a. Original image b. Rotation processing c. Cropping processing d. Brightness control

e. Hue shift (Hue=30) f. Hue shift (Hue=60) g. Additive poisson noise h. Gaussian blur

Figure 2 Data examples of different augmentation methods

2.3 Improvement strategies for YOLO11n

To effectively enhance the detection performance of litchi fruits within complex natural environments, this study adopts YOLO11n as the baseline architecture. YOLO11 represents the latest generation of real-time object detection architectures, achieving significant optimizations in training strategies, inference speeds, and model complexity compared to the YOLOv8 series[13]. However, standard lightweight YOLO variants often lack sufficient feature extraction capabilities for dense and occluded litchi clusters.

The architectural overview of the proposed YOLO11n-MAS is illustrated in Figure 3. The network incorporates the following primary enhancements:

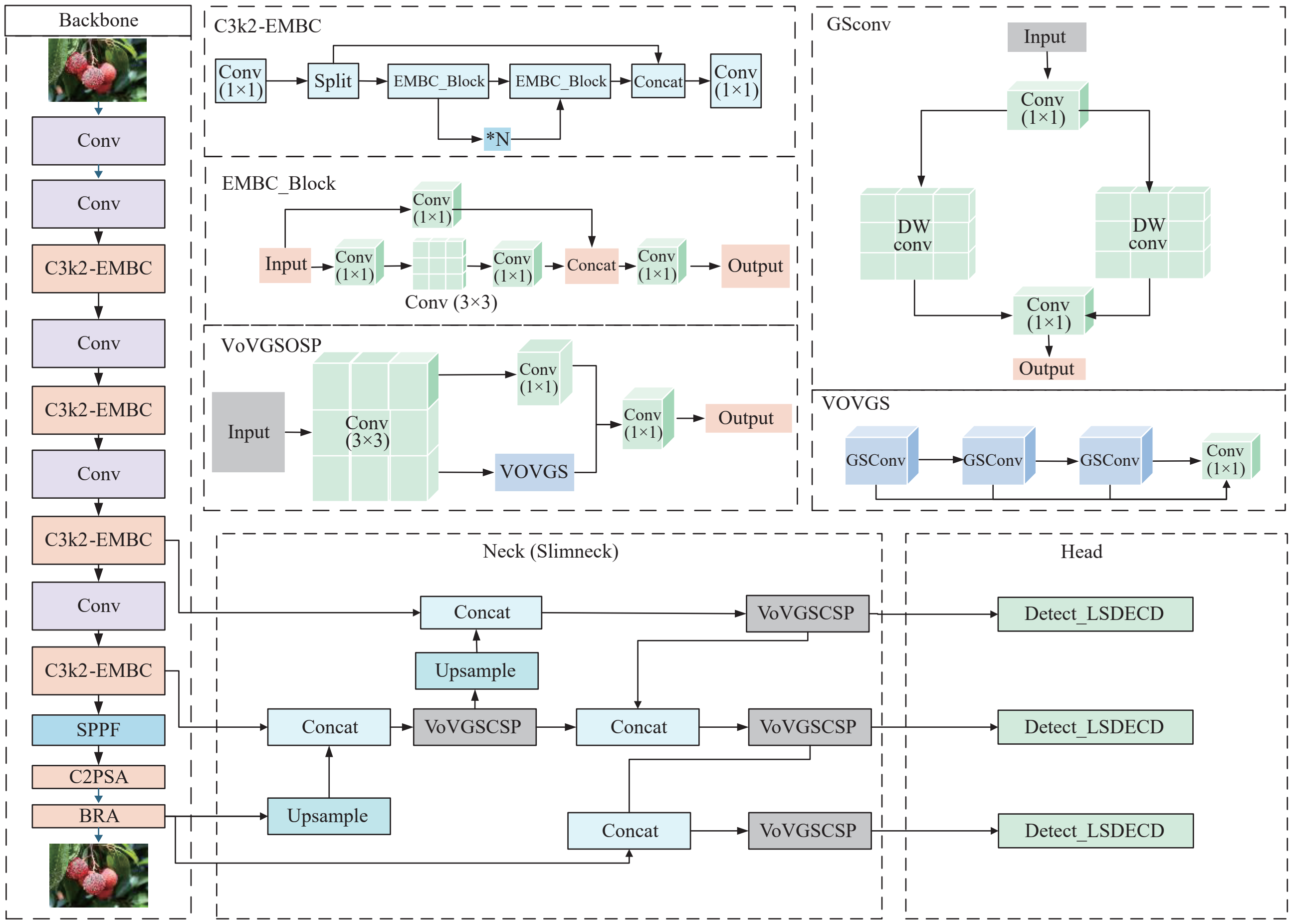


Figure 3 Architecture diagram of YOLO11n-MAS model

1) C3k2-EMBC module: This module is integrated to bolster the capture of key litchi features under conditions of complex lighting and occlusion.

2) Slimneck architecture: A Slimneck network is constructed to mitigate parameter redundancy and computational load while ensuring robust feature fusion stability.

3) Detect_LSDECD head: A lightweight detection head, Detect_LSDECD, is designed to refine the localization accuracy and discernibility of dense and overlapping targets.

4) Bi-Level Routing Attention (BRA) mechanism: A lightweight BRA mechanism is embedded at the terminal stage of the backbone, enabling efficient global context modeling and long-range dependency capture with minimal computational overhead.

2.3.1 Backbone reconstruction using C3k2-EMBC and BRA

1) C3k2-EMBC multi-scale feature hierarchical reinforcement mechanism

The standard C3k2 module in the YOLO11n backbone relies on full-channel convolution, which often leads to feature oversmoothing and missed detections when processing small[14] or overlapping targets. To mitigate this deficiency, the C3k2-EMBC module was designed to replace the original C3k2 unit. The core component, the EMBC_Block, employs an asymmetric channel-splitting strategy inspired by small-object detection optimization[15]. It constructs a residual branch utilizing Depthwise Separable Convolution (DWConv)[16,17] to capture high-frequency details such as surface textures and contours, effectively preventing feature over-smoothing[18]. Meanwhile, the main branch extracts contextual semantic information. This multi-branch design enhances the model's feature representation capabilities while maintaining low computational overhead.

The EMBC_Block processes the input feature map $X\in\mathbb{R}^{C_{in}\times H\times W}$ through five convolutional layers. Initially, an asymmetric channel splitting strategy ($\lambda=0.25$) divides the input into a main branch ($X_{\rm main}$) and a residual branch ($X_{\rm res}$), as shown in Equation (1). The main branch extracts contextual semantics via a 1×1 convolution (Equation (2)). The residual branch, designed for fine-grained feature capture, employs a bottleneck structure consisting of a 1×1 compression, a 3×3 Depthwise Separable Convolution ($DWConv_{3\times3}$), and a 1×1 restoration (Equation (3)). Finally, the outputs are concatenated and fused through a 1×1 convolution to generate the final output X' (Equations (4) and (5)).

$$\begin{cases} X_{\rm main}=(X,1-\lambda)\in R^{(1-\lambda)C_{\rm in}\times H\times W} \\ X_{\rm res}=(X,\lambda)\in R^{\lambda C_{\rm in}\times H\times W} \end{cases} \tag{1}$$

$$X_{\rm main\_proc}=\sigma(X_{\rm main}*W_1)\in R^{C_{\rm in}/2\times H\times W} \tag{2}$$

where, σ denotes the activation function (ReLU), and * represents the convolution operation.

$$\begin{cases} X_{\mathrm{res1}} = \sigma(X_{\mathrm{res}} * W_2) \in R^{(C_{\mathrm{in}}/4)\times H\times W} \\ X_{\mathrm{res2}} = \sigma(X_{\mathrm{res1}} * W_3^{\mathrm{dw}}) \in R^{(C_{\mathrm{in}}/2)\times H\times W} \\ X_{\mathrm{res\_proc}} = \sigma(X_{\mathrm{res2}} * W_4) \in R^{(C_{\mathrm{in}}/2)\times H\times W} \end{cases} \quad (3)$$

$$X_{\mathrm{concat}} = C_{\mathrm{oncat}}(X_{\mathrm{main\_proc}}, X_{\mathrm{res\_proc}}) \in R^{C_{\mathrm{in}}\times H\times W} \quad (4)$$

$$X' = \sigma(X_{\mathrm{concat}} * W_5) \in R^{C\times H\times W} \quad (5)$$

Furthermore, the lightweight nature of DWConv significantly reduces computational overhead while enhancing detail extraction performance. Through this enhanced residual connection design, the C3k2-EMBC module provides subsequent network layers with high-quality feature maps rich in detail, improving the model's robustness in environments characterized by branch occlusion and lighting fluctuations.

2) Bi-Level Routing Attention mechanism

To address the limitation of the YOLO11n backbone in modeling long-range dependencies—critical for detecting dispersed or occluded litchis—a lightweight Bi-Level Routing Attention (BRA) mechanism was embedded at the terminal stage of the backbone. This module employs a two-step sparse attention strategy: it first partitions feature maps into regional blocks and selectively routes relevant regions, then performs fine-grained attention within the selected blocks. This design enables efficient global context modeling with significantly reduced computational overhead compared to standard global attention. By capturing long-range spatial relationships, the BRA module enhances the model's semantic understanding of orchard layouts, thereby improving detection robustness under challenging conditions.

2.3.2 Lightweight neck construction with Slimneck

Although YOLO11n utilizes a bidirectional FPN-PAN architecture[19,20] (shown in Figure 4), it exhibits significant computational redundancy and fine-grained feature degradation when applied to complex orchard environments[21]. To resolve these bottlenecks, Slimneck was introduced, a lightweight reconstruction of the neck network. This architecture integrates GSConv for efficient downsampling. By synthesizing standard convolution with low-cost depthwise operations, GSConv minimizes computational load while preserving spatial expressiveness[22,23]. Furthermore, the VoVGSCSP module is adopted for robust feature fusion. It employs a split-processing strategy: a residual branch preserves high-frequency details (e.g., litchi textures) via minimal processing to prevent feature erosion, while a backbone branch reinforces semantic extraction through cascaded GSConvs. The synergistic fusion of these branches yields a representation rich in both semantics and detail, effectively balancing high detection accuracy with the efficiency required for deployment on edge devices.

2.3.3 Integrating Detect_LSDECD into the detection head

The native YOLO11n decoupled head relies on independent convolutions for each scale, sacrificing model lightness for accuracy. Moreover, its dependence on Batch Normalization (BN) induces instability during single-image inference—a common scenario in real-time monitoring. To overcome these limitations, we propose Detect_LSDECD, a lightweight shared detail-enhanced detection head. Drawing on Detail Enhanced Convolution (DEConv)[24,25], this architecture integrates prior information to bolster representation, utilizing re-parameterization to eliminate additional inference costs.

Crucially, Group Normalization (GN) replaces Batch Normalization (BN) to decouple normalization performance from batch size, which improves inference stability on edge devices for single-image inference scenarios. By enforcing parameter sharing across detection scales, the design drastically reduces redundancy, achieving a synergy of robustness, accuracy, and efficiency (shown in Figure 5).

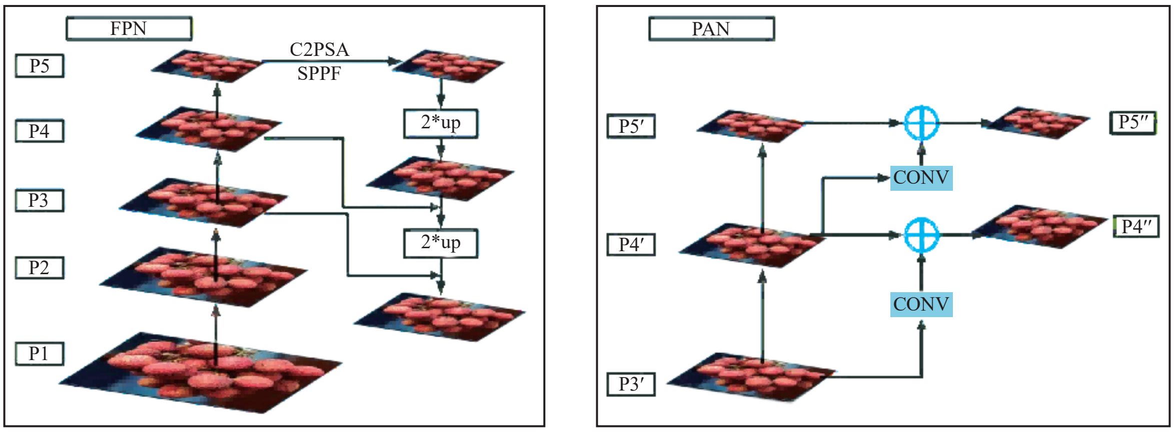


Figure 4 Bidirectional Feature Fusion Pyramid (FPN+PAN)

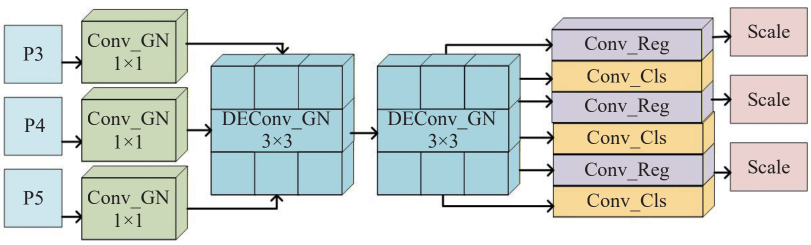


Figure 5 Architecture diagram of Detect_LSDECD detection head

Functionally, Detect_LSDECD operates on feature layers P3-P5, excluding high-resolution P1-P2 layers to mitigate noise and computational overhead. Feature maps first pass through scale-specific 1×1 Conv_GN modules for dimension alignment. Subsequently, they traverse two cascaded, scale-shared DEConv_ GN modules, effectively extracting granular details while minimizing parameter count. The output is then bifurcated into independent 1×1 convolutional branches for localization (Conv_Reg) and classification (Conv_Cls), supplemented by a scale layer to refine multi-scale adaptability.

2.4 Experimental implementation

All experiments were conducted on a workstation running a 64-bit Windows 11 operating system, equipped with an NVIDIA GeForce RTX 4060 GPU (12 GB VRAM) and 16 GB of RAM. The deep learning environment was established using the PyTorch 2.8 framework with Python 3.8 and CUDA 12.7 acceleration.

For model training, the input image resolution was standardized to 640×640 pixels. The optimization process utilized Stochastic Gradient Descent (SGD) with a batch size of 16. A cosine annealing strategy was employed to schedule the learning rate, which decayed from an initial value of 0.01 to a final value of 0.001. The model was trained for a total of 300 epochs to ensure convergence. To quantitatively evaluate detection performance, particularly for small litchi targets, standard metrics including Precision (P), Recall (R), and Mean Average Precision (mAP50) were adopted[26]. Furthermore, computational complexity and efficiency were assessed based on parameter count (Params), Floating Point Operations (FLOPs), and model weight size.

$$\text{Precision}(P) = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{6}$$

$$\text{Recall}(R) = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{7}$$

$$F_1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{8}$$

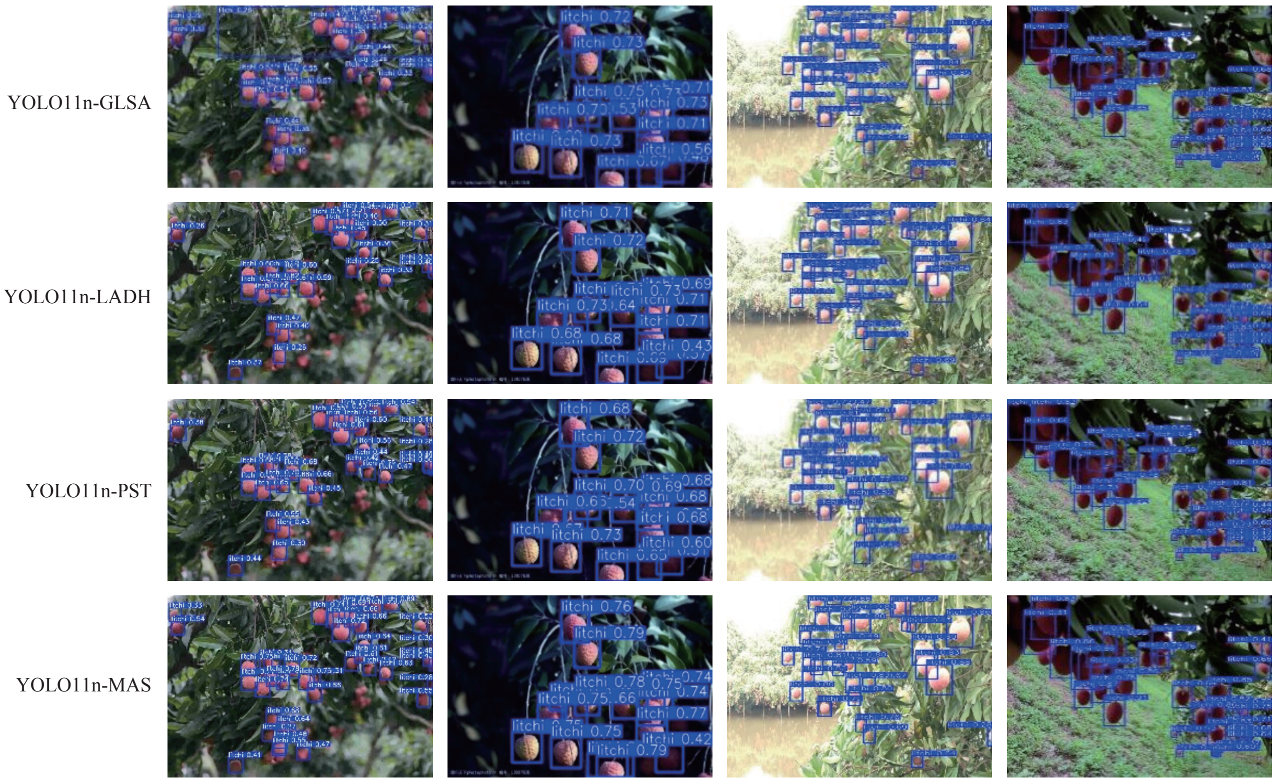
$$\text{mAP} = \frac{1}{c}\sum_{i=1}^{c}\text{AP}_i \tag{9}$$

where, TP denotes the count of correctly detected fruits, FP represents generated bounding boxes with incorrect positioning, and FN refers to fruit instances that were missed. AP evaluates the precision of the detection head.

3 Results and analysis

3.1 Analysis of detection performance across different scenarios

To rigorously assess the robustness of the proposed YOLO11n-MAS model, its detection performance was benchmarked against several competing methods, YOLO11n-GLSA, YOLO11n-LADH, and YOLO11n-PST, under diverse and challenging environmental conditions. For this purpose, three specialized test subsets (strong light, leaf occlusion, and fruit occlusion) were curated from the original dataset for quantitative evaluation, and representative low-light cases were additionally included for qualitative visualization in Figure 6. The evaluation was conducted on three specialized test subsets: strong light (56 images containing 243 fruit instances), leaf occlusion (71 images containing 488 fruit instances), and dense fruit-on-fruit occlusion (49 images containing 532 fruit instances).

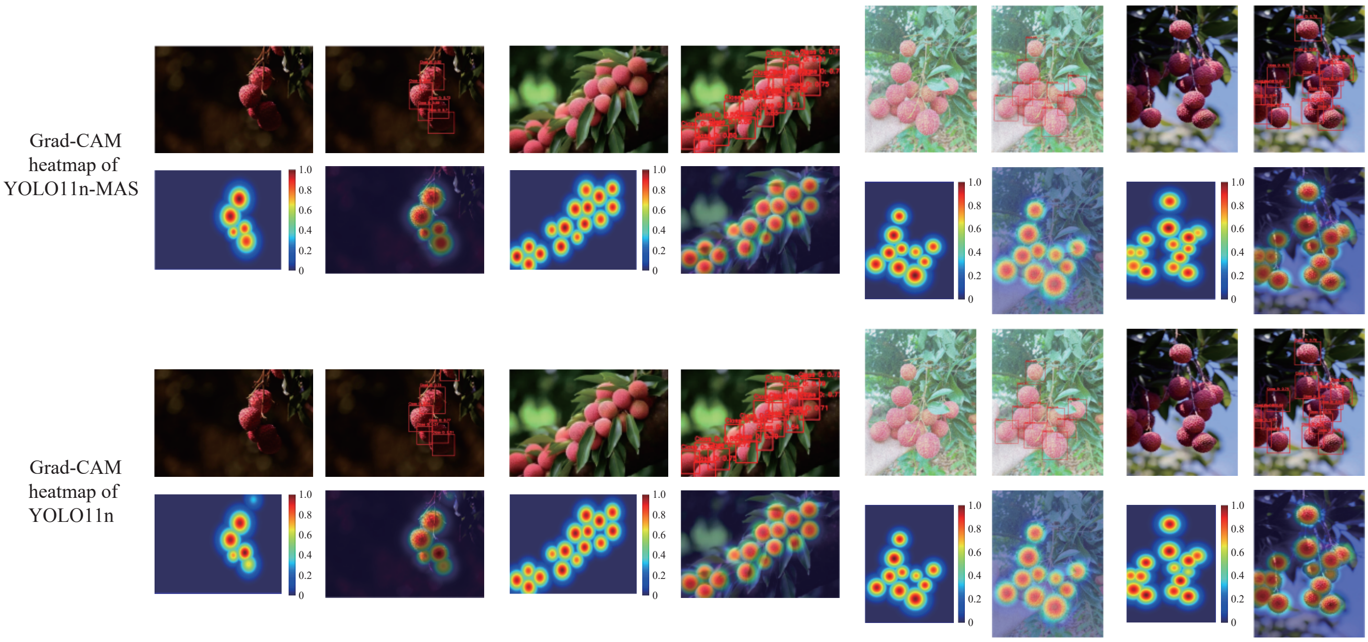


a. Branches and foliage occlusion b. Low light c. Strong light d. Fruit occlusion

Figure 6 Qualitative visualization of detection results for litchi fruits

3.2 Visualization experiments in complex environments

To elucidate the interpretability of YOLO11n-MAS, Grad-CAM[27] was employed to generate Grad-CAM activation heatmaps. Figure 7 compares the proposed model against the baseline YOLO11n across four scenarios: low light, strong light, branches and foliage occlusion, and fruit occlusion. Under challenging lighting conditions, the baseline model suffered from diffuse activation and noise interference, whereas YOLO11n-MAS concentrated on litchi fruit textures, effectively reducing missed detections under low-light conditions and false positives under strong-light conditions. Regarding occlusions, the proposed model accurately focused on visible regions and boundaries, mitigating leaf mis-activation and resolving the chaotic patterns observed in the baseline during fruit-on-fruit overlaps.



a. Low light b. Strong light c. Branches and foliage occlusion d. Fruit occlusion

Figure 7 Grad-CAM visualization for litchi fruit detection

Quantitative evaluation using the Average Attention IoU indicator shows that our model achieves an average attention alignment of 0.850, which is 0.900 percentage points higher than the baseline (0.841), as listed in Table 1. Consistent improvements are observed across all four complex scenarios, demonstrating that the improved model better focuses on visible fruit areas and boundaries.

Table 1 Quantitative evaluation of Grad-CAM activation based on average attention IOU

Scenario	Low light	Strong light	Leaf occlusion	Fruit occlusion	Average
YOLO11n Average Attention IoU	0.806	0.844	0.855	0.862	0.841
YOLO11n-MAS Average Attention IoU	0.811	0.858	0.857	0.874	0.850
Improvement	0.5%	1.4%	0.2%	1.2%	0.9%

3.3 Ablation studies

To quantitatively deconstruct the effectiveness of our proposed architectural enhancements, a comprehensive ablation study was conducted on the test set. This study systematically evaluated the individual and synergistic contributions of the C3k2-EMBC module, the Slimneck structure, the Detect_LSDECD detection head, and the BRA module. The results, summarized in Table 2, validate the efficacy of our integrated design.

Initially, replacing the baseline neck with the proposed Slimneck increased the mAP50 to 0.901, while the parameter count was reduced to 2.57 M. This indicates that Slimneck can improve detection performance while maintaining the model's lightweight characteristics. When only the Detect_LSDECD module was introduced into the baseline model, the mAP50 remained at 0.885, while the parameters decreased from 2.58 M to 2.26 M (about a 12.4% reduction). This suggests that Detect_LSDECD mainly contributes to model lightweighting when used alone, rather than directly improving accuracy. After further integrating the C3k2-EMBC module, the mAP50 increased to 0.912. The lack of direct accuracy improvement when Detect_LSDECD is used alone is attributed to its design as a lightweight detection head. It only optimizes computational and parameter efficiency by replacing redundant convolutions and pruning non-critical channels, without introducing additional feature enhancement mechanisms to improve the model's feature representation capability. In addition, combining Slimneck with the Detect_LSDECD head achieved an mAP50 of 0.904 and reduced the parameters to 2.25 M, showing the best lightweighting effect among the tested variants.

Table 2 Results of the ablation test

YOLO11n baseline	LSDECD	C3k2-EMBC	Slimneck	BRA	mAP50	Precision	Recall	Para-meters	Model Size
✓	×	×	×	×	0.885	0.842	0.818	2.582	5.2
✓	×	✓	×	×	0.885	0.841	0.822	2.875	5.8
✓	×	×	✓	×	0.901	0.862	0.831	2.569	5.2
✓	✓	×	×	×	0.885	0.853	0.821	2.260	4.9
✓	✓	✓	×	×	0.898	0.855	0.823	2.553	5.6
✓	✓	×	✓	×	0.904	0.847	0.832	2.247	5.0
✓	×	✓	✓	×	0.912	0.875	0.839	2.662	5.4
✓	✓	✓	✓	×	0.917	0.874	0.843	2.358	5.4
✓	✓	✓	✓	✓	0.916	0.872	0.858	2.339	5.2

The final YOLO11n-MAS model achieved an mAP50 of 0.916 with a precision of 87.2%. Although the variant without BRA reached an mAP50 of 0.917, adding BRA increased recall from 0.843 to 0.858. This indicates that the model can detect more complete targets under occlusion and dense-cluster conditions, which improves robustness in complex orchard environments. Overall, these results show that the proposed modules improve detection performance and robustness while reducing model complexity, achieving a good balance among accuracy, robustness, and lightweight design.

3.4 Comparative experiments with mainstream lightweight object detection models

To benchmark the performance of the proposed YOLO11n-MAS model, a comparative study was conducted against nine prominent lightweight models, including YOLOv5n and YOLOv8n. All models were evaluated under identical conditions using the same litchi dataset to ensure a fair comparison. The quantitative results are summarized in Table 3, and a comprehensive performance overview is visualized in the radar chart in Figure 8.

Table 3 Comparative experiments of mainstream lightweight object detection models

Models	mAP50	Precision	Recall	GFLOPS	Parameters
YOLOv5n	0.862	0.823	0.810	5.8	2.189
YOLOv8n	0.881	0.837	0.821	6.8	2.685
YOLO11n	0.885	0.842	0.818	6.3	2.582
YOLOv10n	0.871	0.816	0.819	6.5	2.265
Hyper-YOLO	0.901	0.822	0.838	9.5	3.621
YOLO-SDI	0.884	0.858	0.836	6.7	2.628
YOLO-GLSA	0.876	0.851	0.835	8.6	3.738
YOLOv6n	0.877	0.814	0.826	11.4	4.155
YOLO-LADH	0.912	0.861	0.843	6.3	2.582
YOLO11n-MAS	0.916	0.872	0.858	5.5	2.340

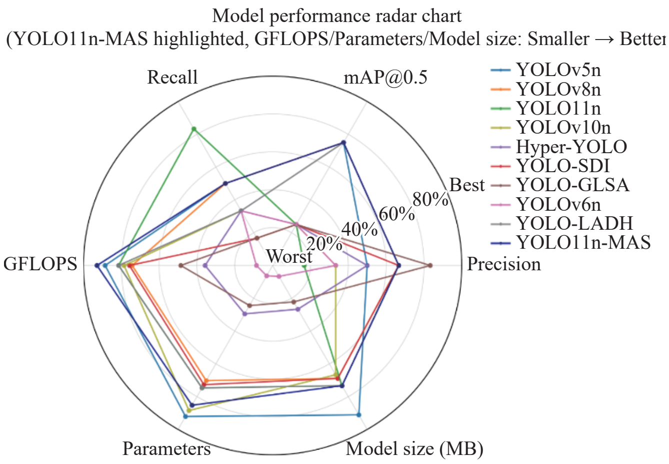


Figure 8 Performance comparison radar chart of different mainstream lightweight object detection models

In terms of detection accuracy, YOLO11n-MAS achieved an mAP50 of 0.916, a precision of 0.872, and a recall of 0.858. These results surpass those of several established models, such as YOLOv5n, and are competitive among current lightweight detectors, as detailed in Table 3. In the radar chart (Figure 8), the vertices corresponding to these accuracy metrics are positioned toward the outer edge, indicating robust detection capability.

Regarding resource efficiency, YOLO11n-MAS demonstrates notable advantages. It requires only 5.5 GFLOPs of computational cost, contains 2.34 M parameters, and has a model size of 5.2 MB. Compared to YOLOv8n, it achieves reductions of 19.1% in computational load and 13.0% in parameter count. Relative to the baseline YOLO11n, the reductions are 12.7% in GFLOPs and 9.4% in parameters. The radar chart further reflects these efficiencies, with the points representing GFLOPs and parameters extending outward.

Overall, YOLO11n-MAS is a highly efficient litchi fruit detection model, outperforming other models in key metrics while maintaining a lightweight structure. This balance highlights its potential as a suitable solution for deployment on resource-constrained edge devices.

3.5 Development of interactive demonstration software and visual model verification

To validate the practical deployment performance of YOLO11n-MAS on edge mobile devices, the proposed model and the baseline YOLO11n were ported to an Android smartphone, and a dedicated detection demonstration application was developed. The trained PyTorch weight files were first exported to the ONNX format and subsequently converted to the NCNN format optimized for mobile platforms. The trained PyTorch model was further converted and optimized for Android edge deployment using the NCNN framework. The desktop inference weight size optimized for PyTorch is 5.2 MB, and after NCNN framework-specific optimizations including constant folding, redundant layer pruning, and lossless weight compression, the final mobile deployment weight size is reduced to 4.68 MB with negligible accuracy loss.

A total of 480 litchi images randomly sampled from the 544-image test set using a fixed random seed (seed=42) were used to evaluate the performance of both models within the Android application. Statistical analysis confirmed that this sample is highly representative of the full test set, with less than 1% difference in the proportion of images across the four scenarios and only a 0.2 difference in the average number of fruits per image. As presented in Table 4, YOLO11n-MAS achieves a mean average precision (mAP50) of 91.6% and a precision of 87.2%, significantly outperforming the baseline YOLO11n (88.5% mAP50 and 84.2% precision). Moreover, the missed detection rate is reduced from 8.7% to 4.6%, demonstrating the effectiveness of the proposed lightweight improvements in complex orchard environments.

Table 4 Edge deployment performance of YOLO11n and YOLO11n-MAS on Android

Models	mAP50/%	Precision/%	Missed rate/%
YOLO11n	88.5	84.2	8.7
YOLO11n-MAS	91.6	87.2	4.6

The functional interface and detection results of the application are illustrated in Figure 9. Under challenging scenarios such as leaf occlusion, fruit overlap, low light, and strong light, YOLO11n-MAS consistently delivers stable and precise detection. This tool provides intuitive visual validation of the model's robustness in real-world environments and offers a feasible technical solution for the edge deployment of intelligent litchi harvesting systems.

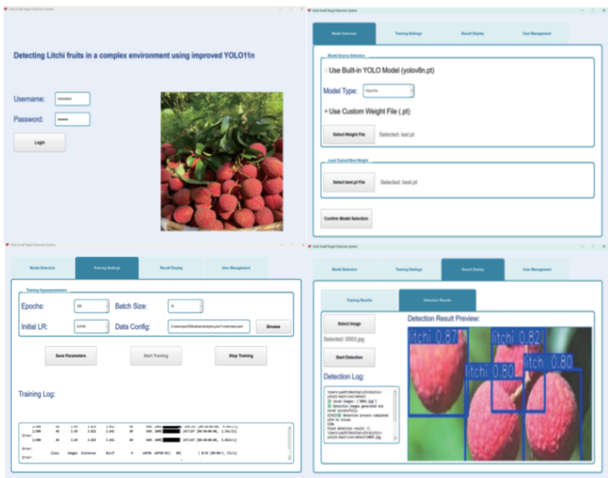


Figure 9 Interactive demonstration platform for litchi object detection

4 Conclusions

To address the challenges of missed and false detections of litchi fruit in complex natural environments, this study proposes the YOLO11n-MAS network, a model that balances inference speed with precision. The main conclusions are as follows:

1) Compared with the baseline YOLO11n, the proposed YOLO11n-MAS model improves the mean average precision (mAP50) by 3.1 percentage points, while simultaneously reducing the parameter count by 9.4% and the computational complexity (GFLOPs) by 12.7%.

2) Compared to other mainstream lightweight networks such as YOLOv8n, YOLO11n-MAS demonstrates superior efficiency and accuracy. With a desktop inference weight size of only 5.2 MB (further compressed to 4.68 MB for NCNN-based Android edge deployment), the YOLO11n-MAS model achieves a precision of 87.2%, a recall of 0.858, a mean average precision (mAP50) of 91.6%, and an F1-Score of 0.865.

3) In targeted tests simulating real-world complex scenarios, YOLO11n-MAS demonstrated excellent environmental adaptability. Visual analysis confirmed the model's ability to overcome interference from strong light, insufficient lighting, and foliage/fruit occlusion, validating its robustness and practical potential for in-field deployment.

The improved YOLO11n-MAS algorithm proposed in this study can provide an effective and efficient solution for the real-time detection of fruit targets that are obscured or in challenging lighting. This work offers algorithmic support for intelligent agriculture applications such as early yield estimation, pest and disease monitoring, and automated harvesting. While the model is highly effective for litchi, its application to other crops may require further fine-tuning to ensure optimal accuracy and stability.

Acknowledgements

This work was financially supported by the 2025 Guangdong Provincial Key Discipline Research Enhancement Project (Grant No. 2025ZDJS129), in part by the 2025 Guangdong Provincial Education Science Planning Project (Grant No. 2025GXJK0504), and in part by the Guangzhou Education Bureau Graduate Student Scientific Research Project (Grant No. 2024312026).

[References]

[1] Jiang N H, Liu W, Xiao Z D, Xiang X, Zhong Y. Metabolomics unveiled the accumulation characteristics of taste compounds during the development and maturation of litchi fruit. Foods, 2025; 14(1): 144.

[2] Handel M J. Labor shortages: What is the problem. Intereconomics, 2024; 59(3): 136–142.

[3] Ling J, Heldman D R. Integration of machine learning technologies in food flavor research: Current applications, challenges, and future perspectives. Int J Agric & Biol Eng, 2025; 18(3): 1–11.

[4] Habaragamuwa H, Ogawa Y, Suzuki T, Shiigi T, Ono M, Kondo N. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. Engineering in Agriculture, Environment and Food, 2018; 11(3): 127–138.

[5] Liu Y. An improved faster R-CNN for object detection. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou: IEEE, 2018; pp.119–123. doi: 10.1109/ISCID.2018.10128.

[6] Alif M A R, Hussain M. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. arXiv preprint, 2024; arXiv: 2406.10139. doi: 10.48550/arXiv.2406.10139.

[7] Li X, Pan J D, Xie F P, Zeng J P, Li Q, Huang X J, et al. Fast and accurate green pepper detection in complex backgrounds via an improved YOLOv4-tiny model. Computers and Electronics in Agriculture, 2021; 191: 106503.

[8] Gao A, Du Y H, Li Y Q, Song Y P, Ren L L. Apple flower phenotype detection method based on YOLO-FL and application of intelligent flower thinning robot. Int J Agric & Biol Eng, 2025; 18(3): 236–246.

[9] Chen D X, Liu L, Zhao L, Shi Y, Wang F, Meng D, et al. Method for lightweight tomato leaf disease recognition based on improved YOLO11s. Int J Agric & Biol Eng, 2025; 18(5): 298–305.

[10] Xu Y K, Khan T M, Song Y, Meijering E. Edge deep learning in computer vision and medical diagnostics: A comprehensive survey. Artificial Intelligence Review, 2025; 58(3): 93.

[11] He Z L, Xiong J T, Lin R, Zou X J, Tang L Y, Yang Z G, et al. A method of green litchi recognition in natural environment based on improved LDA classifier. Computers and Electronics in Agriculture, 2017; 140: 159–167.

[12] Chen M, Jin C Q, Ni Y L, Yang T X, Xu J S. A dataset of the quality of soybean harvested by mechanization for deep-learning-based monitoring and analysis. Data in Brief, 2024; 52: 109833.

[13] Sapkota R, Karkee M. Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. arXiv preprint arXiv: 2410.19869, 2024.

[14] Liu C G, Gao G S, Huang Z Y, Hu Z H, Liu Q J, Wang Y H. YOLC: You only look clusters for tiny object detection in aerial images. IEEE Transactions on Intelligent Transportation Systems, 2024; 25(10): 13863–13875.

[15] Kotthapalli M, Ravipati D, Bhatia R. YOLOv1 to YOLO11: A comprehensive survey of real-time object detection innovations and challenges. arXiv preprint, 2025; arXiv: 2508.02067.

[16] He A, Li X B, Wu X M, Su C Y, Chen J, Xu S, et al. ALSS-YOLO: An adaptive lightweight channel split and shuffling network for TIR wildlife detection in UAV imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2024; 17: 12345–12356.

[17] Howard A G, Zhu M L, Chen B, Kalenichenko D, Wang W J, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint, 2017; arXiv: 1704.04861v1. doi: 10.48550/arXiv.1704.04861.

[18] Wu X, Chen Z, Wang W, Jadbabaie A. A non-asymptotic analysis of oversmoothing in graph neural networks. In: Proc. 40th Int. Conf. on Machine Learning, Honolulu, HI: PMLR, 2023; pp.37228–37244. doi: 10.5555/3618408.3618416.

[19] Lin T Y, Dollár P, Girshick R, He K M, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, 2017; pp.936–944. doi:10.1109/CVPR.2017.106.

[20] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint, 2020; arXiv: 2004.10934. doi: 10.48550/arXiv.2004.10934.

[21] Gu Z N, He D Q, Huang J D, Chen J Q, Wu X H, Huang B C, et al. Simultaneous detection of fruits and fruiting stems in mango using improved YOLOv8 model deployed by edge device. Computers and Electronics in Agriculture, 2024; 227: 109512.

[22] Li H L, Li J, Wei H B, Liu Z, Zhan Z F, Ren Q L. Slim-neck by GSConv: A lightweight-design for real-time detector architectures. Journal of Real-Time Image Processing, 2024; 21(3): 62.

[23] Huang D Q, Tu Y T, Zhang Z H, Ye Z K. A lightweight vehicle detection method fusing GSConv and coordinate attention mechanism. Sensors, 2024; 24(8): 2394.

[24] Chen Z X, He Z W, Lu Z M. DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. IEEE Transactions on Image Processing, 2024; 33: 1002–1015.

[25] Wen G H, Li M, Luo Y H, Shi C S, Tan Y F. The improved YOLOv8 algorithm based on EMSPConv and SPE-head modules. Multimed Tools Applications, 2024; 83(21): 61007–61023.

[26] Badgujar C M, Poulose A, Gan H. Agricultural object detection with You Only Look Once (YOLO) algorithm: A bibliometric and systematic literature review. arXiv preprint, 2024; arXiv: 2401.10379. doi:10.48550/arXiv.2401.10379.

[27] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE, 2017; pp.618–626. doi:10.1109/ICCV.2017.74.