Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **11**, Number **1**, February **2026**

An Improved Lightweight Pear Leaf Disease Detection Algorithm Based on YOLO11

Hui-Qun Hong[1]

[1]Zhicheng College, Fuzhou University, Fuzhou 350002, China
02125017@fdzcxy.edu.cn

Jeng-Shyang Pan[2,3,*]

[2]School of Artificial Intelligence/School of Future Technology,
Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]Department of Information Management,
Chaoyang University of Technology, Taichung 41349, Taiwan
100112@nuist.edu.cn

Nian He[1]

[1]Zhicheng College, Fuzhou University, Fuzhou 350002, China
02121003@fdzcxy.edu.cn

Feng-Hua Huang[4,5]

[4]Fujian Key Laboratory of Spatial Information Perception and Intelligent Processing,
Yango University, Fuzhou 350015, China
[5]Fujian University Engineering Research Center of Spatial Data Mining and Application,
Yango University, Fuzhou 350015, China
fhhuang@ygu.edu.cn

Jian-Bin Zhai[6]

[6]School of Information Engineering,
Yango University, Fuzhou 350015, China
jianbin.zhai@ygu.edu.cn

[*]Corresponding author: Jeng-Shyang Pan

Received September 14, 2025, revised November 27, 2025, accepted January 18, 2026.

Abstract. *To address the challenges faced by existing leaf disease detection models under resource-constrained scenarios, including the difficulty in balancing high accuracy with low complexity, poor compatibility for edge deployment, and insufficient detection robustness in complex real-world environments, this study proposes an improved lightweight pear leaf disease detection algorithm based on YOLO11, named YOLO11-Ghost. Firstly, GhostBottleneckStack modules are employed to replace the original C3k2 modules in both the backbone and neck networks. By leveraging feature redundancy reuse, linear transformation, and depthwise separable convolution, this modification reduces model complexity and computational overhead, while preserving leaf texture and morphological features, thereby enhancing detection efficiency. Secondly, to optimize edge deployment performance, the HardSwish activation function which characterized by smooth transition and bounded output, is adopted to replace the SiLU activation function. This adjustment not only simplifies the model's computational graph structure but also improves TensorRT INT8 quantization accuracy, ultimately boosting the model's adaptability to edge devices. Finally, the model is trained and validated on the DiaMOS Plant dataset, a multi-category pear leaf disease dataset collected from real pear tree growing scenarios. Experimental results indicate that, compared with YOLO11s, YOLO11-Ghost achieves a 68.39% reduction in model parameters, a 78.97% decrease in computational complexity (FLOPS), and a 67.59% shrinkage in model size, exhibiting significant lightweight advantages. In terms of detection performance, Recall is increased by 2.66%, whereas Precision and mAP50 decrease by 6.97% and 3.1%, respectively. Notably, the improved algorithm has two current limitations: first, despite the increased Recall, accuracy loss occurs as core metrics (Precision and mAP50) decline to a certain extent; second, the DiaMOS Plant dataset suffers from class imbalance, which may compromise detection performance for rare pear leaf disease types. From the perspective of real needs in resource-constrained scenarios, the deployment benefits of the model's lightweight design, such as reduced storage usage and faster inference on edge devices, far outweigh the impact of accuracy loss, making the current performance loss acceptable. Future work will focus on compensating for the accuracy gap through methods like feature enhancement, providing directions for the further application and optimization of this algorithm in practical agricultural settings.*

Keywords: YOLO11, Ghost Convolution, HardSwish, Edge Deployment

1. **Introduction.** Pears are highly popular fruits. Scientific identification of pear leaf diseases can assist in formulating differentiated prevention and control strategies, reducing the use of chemical pesticides, and increasing pear yields. Traditional manual inspection is inefficient [1] and highly subjective. Early machine learning algorithms based on color and texture features struggle to achieve accurate classification when faced with leaf overlap, sudden changes in illumination, and subtle feature variations in the early stages of diseases [2]. Most convolutional neural networks (CNNs) often suffer from insufficient detection accuracy or excessively large model parameters; moreover, in complex orchard scenarios, dense branch and leaf occlusion and uneven distribution of illumination conditions directly affect detection reliability [3, 4]. With the widespread application of artificial intelligence (AI) technology [5–9], it is particularly important to research AI algorithms that feature high accuracy, strong generalization ability, and deployability on edge devices [10].

1.1. **Current status of crop disease detection methods.** Existing deep learning methods have been widely applied in crop disease detection. Li et al. [11] proposed a lightweight model named Apnet, which achieved an accuracy of 87.1% and a recall rate of 75.6% on a self-built dataset. However, the model's detection accuracy decreases under conditions of multi-leaf occlusion and drastic changes in illumination. Tian et al. [12] combined image enhancement technology with the CycleGAN generative adversarial network and applied it to the detection of apple anthracnose, achieving favorable results. In the

detection of citrus diseases and pests. Luo et al. [13] introduced a near-feature pyramid network structure into the neck network, which improved detection accuracy and computational efficiency, but the model lacked robustness in natural orchard environments. To enhance the detection performance for small targets. Qiu et al. [14] proposed an adaptive spatial feature fusion module that uses dynamic weights to allocate the contribution of feature maps at different scales. Nevertheless, no optimization was made for complex agricultural backgrounds, resulting in a relatively high false detection rate of disease spots. Hu et al. [15] proposed a lightweight instance segmentation model called BHI-YOLO, which combines Mosaic data enhancement for feature fusion and achieved certain results, but it showed insufficient sensitivity to small-target diseases.

Most of the aforementioned methods either rely on complex network architectures to improve performance or adopt lightweight networks. These complex models have excessively high FLOPs and a large number of parameters, which are not conducive to deployment on resource-constrained mobile devices. The adopted lightweight networks methods are mostly limited to training with single-background data from laboratory environments.

1.2. **Development of YOLO series algorithms.** Object detection, as one of the core tasks in computer vision, has seen milestone advancements with the development of the YOLO series algorithms [16, 17]. Introduced in 2015, YOLOv1 [18] pioneered the one-stage detection paradigm by employing a single convolutional neural network to simultaneously predict bounding boxes and class probabilities, significantly simplifying the detection pipeline. With continuous algorithmic evolution, YOLOv4 [19] in 2020 explicitly structured the architecture into three functional components—backbone, neck, and head—and introduced the Bag of Freebies and Bag of Specials strategies, achieving breakthrough improvements in both speed and accuracy. In 2024, YOLOv9 [20] addressed the issue of information loss in deep networks by proposing the Programmable Gradient Information (PGI) mechanism and a lightweight Generalized Efficient Layer Aggregation Network (GELAN), effectively mitigating gradient information decay. Released in the same year, YOLO11 further integrated advanced feature extraction techniques, enhancing detail capture capability while maintaining a lightweight design, which significantly boosted the model's accuracy and real-time performance across various vision tasks.

1.3. **Main work of this paper.** To address the challenges of model complexity and edge deployment, this paper proposes YOLO11-Ghost, a lightweight algorithm improved upon YOLO11 for real-time pear leaf disease detection. The core innovations and contributions are outlined as follows:

(1) Lightweight design of network architecture. The GhostBottleneckStack lightweight module is introduced into the backbone and neck networks of YOLO11 to replace the original C3k2 module, integrating lightweight strategies such as depthwise separable convolution, Ghost convolution, and group convolution. Under the premise of retaining key fine-grained features of pear leaf diseases, this design achieves a significant reduction in the computational complexity of the model, effectively resolving the contradiction between high-precision detection and lightweight deployment.

(2) Quantization adaptation for edge deployment. To solve the issue of computational graph redundancy and significant INT8 quantization accuracy loss caused by the decomposition of the default SiLU activation function during ONNX export, it is replaced with the HardSwish function. This optimization simplifies the graph structure and enhances quantization stability, better adapting to the low-power, high-real-time requirements of edge devices in orchard environments.

(3) Robustness validation with real-world data. Systematic experiments are conducted on the DiaMOS Plant dataset, which is collected from real pear orchard environments.

Ablation and comparative studies demonstrate that YOLO11-Ghost exhibits superior robustness under challenging conditions like complex illumination and leaf occlusion compared to models trained on lab-only data, providing a promotable solution for practical agricultural applications.

2. **Related work.**

2.1. **Lightweight detection models and development status.** Lightweight object detection models have made significant progress in balancing accuracy and efficiency. EfficientDet [21] utilizes the efficient EfficientNet as its backbone for feature extraction and adopts a novel weighted feature pyramid network, BiFPN, to achieve simple and fast multi-scale feature fusion. It uniformly optimizes network depth, width, and resolution through a compound scaling method, and its lightweight version performs excellently on mobile devices. Variants of the MobileNet-YOLO series, such as MobileNetV3-YOLO [22, 23], combine depthwise separable convolution with the YOLO detection head, significantly reducing computational complexity while maintaining detection performance. Knowledge distillation methods [24] transfer knowledge from large-scale models to lightweight models through a teacher-student network framework, further improving the performance of small models. However, these methods face common challenges in complex agricultural scenarios: EfficientDet-Lite has limited adaptability to dense branch and leaf occlusion; MobileNet-YOLO variants lack detection stability under sudden illumination changes; knowledge distillation methods require complex training strategies and have a strong dependence on the quality of teacher models. Most existing methods have not been specifically optimized for the quantization compatibility of edge device deployment, and their practicality in real orchard environments needs to be improved.

2.2. **Object detection algorithm: YOLO11.** In its backbone network, YOLO11 [17] updates the C2f module in YOLOv8 to the C3k2 block. It improves feature extraction efficiency through a parallel convolution design and flexible parameter configuration, and achieves multi-scale feature fusion through Spatial Pyramid Pooling-Fast (SPPF). Additionally, a Convolutional Block with Parallel Spatial Attention (C2PSA) is added after the SPPF module. By leveraging multi-scale convolutions and channel weighting, this module enhances the model's ability to focus on complex occluders and key regions. The network structure diagram of YOLO11 is shown in Figure 1.

The C3k2 module evolves from the C2f module, as shown in Figure 2. In the C3k2 block, when the hyperparameter c3k is set to True, the bottleneck block is replaced with C3k; when the hyperparameter c3k is set to False, the bottleneck block becomes a regular Bottleneck, and C3k2 is converted into C2f. The C2f module performs feature extraction through two convolutional layers and multiple Bottleneck blocks: the first convolutional layer is a 1×1 convolution, which is used to reduce the number of channels, the second is also a 1×1 convolution, which is used to restore the number of output channels; between them are multiple Bottleneck blocks, which are responsible for feature extraction.

C2PSA extends C2f through the introduction of Position-Sensitive Attention (PSA). It leverages the multi-head attention mechanism and feed-forward neural network (FFN) to enhance feature extraction capability, and can optionally add a residual structure to optimize gradient propagation and network training. The use of FFN enables mapping input features to a higher-dimensional space, capturing complex non-linear relationships of input features and allowing the model to learn richer feature representations.

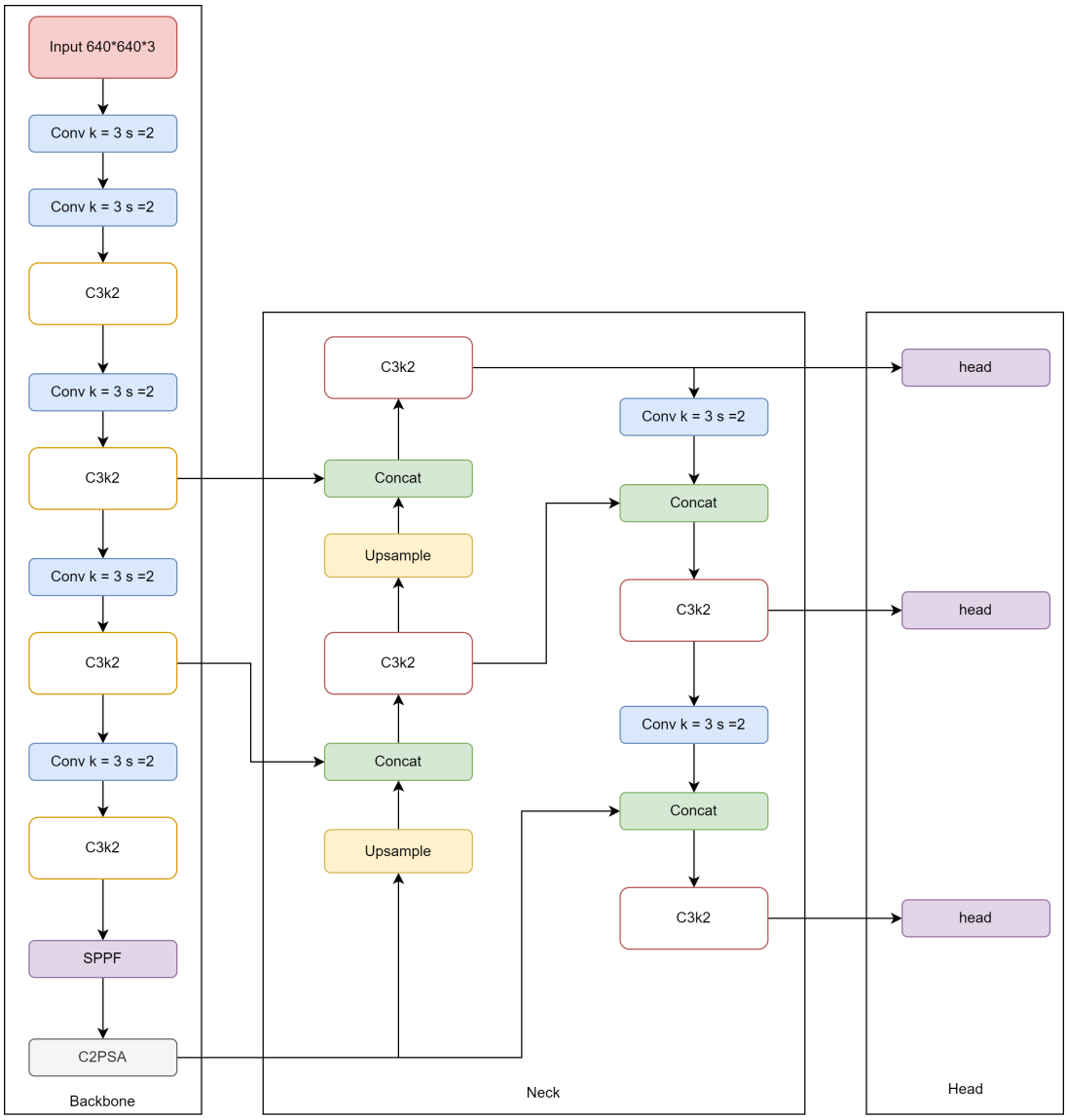


Figure 1. Network structure diagram of YOLO11

2.3. **Limitations of YOLO11 in agricultural applications.** Although YOLO11 improves feature extraction efficiency through its C3k2 and C2PSA modules, it still exhibits the following limitations in practical pear leaf disease detection scenarios:

(1) High computational complexity: The C3k2 module relies on dense convolutional operations, including 1×1 channel reduction convolution, multiple sets of 3×3 standard convolutions, and 1×1 channel expansion convolution, resulting in a large number of parameters and high computational load per module. This leads to significant inference latency on computationally constrained agricultural edge devices.

(2) Insufficient deployment compatibility: The default SiLU activation function decomposes into a Mul(Sigmoid(x), x) operator combination when exporting to ONNX format, increasing the number of computational graph nodes. This causes a substantial drop in mAP50 after TensorRT INT8 quantization, significantly compromising disease detection accuracy.

2.4. **Design of the improved YOLO11-Ghost model.** To address the issues of parameter redundancy and insufficient deployment compatibility in the original YOLO11

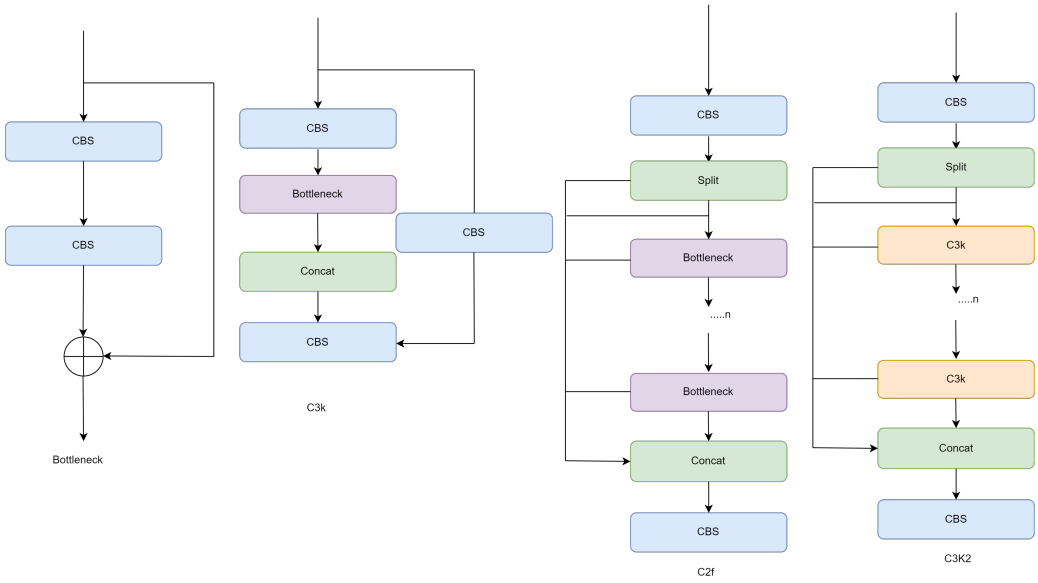


Figure 2. Network structure of C3k2

model for pear leaf disease recognition, this study proposes the YOLO11-Ghost model. Through lightweight convolution reconstruction and activation function optimization, the model significantly enhances operational efficiency on resource-constrained edge devices while maintaining accuracy.

As shown in Figure 3, YOLO11-Ghost retains the overall architecture of the original model but introduces two key improvements:

(1) Replacing the C3k2 modules in both the backbone and neck networks with the newly designed GhostBottleneckStack modules (detailed structure in Figure 4) to reduce computational complexity;

(2) Uniformly replacing the SiLU activation function with the more deployment-friendly HardSwish function. The detection head remains unchanged, preserving the three-scale multi-scale target detection capability.

2.4.1. *Replacement of C3k2 with GhostBottleneckStack.* To tackle the high computational density of the C3k2 modules, a lightweight solution based on the Ghost module is adopted. As illustrated in Figure 5, the Ghost module [25] first extracts partial features using a small number of convolutional kernels, then combines them with inexpensive linear operations such as depthwise separable convolution to generate rich feature representations. By stacking Ghost modules to form GhostBottleneck (Figure 4), traditional convolution is replaced with a combination of "sparse convolution" and "linear transformation," significantly reducing computational resource demands while maintaining performance. The module further employs a grouped convolution strategy (Figure 6) to optimize the depthwise convolution operation, effectively improving operational efficiency in resource-constrained scenarios such as mobile devices [22].

2.4.2. *Replacement of the SiLU activation function with HardSwish.* To resolve the issues of computational graph complexity and quantization accuracy drop associated with the SiLU activation function during deployment, this study adopts HardSwish as a replacement. HardSwish is a piecewise linear approximation of the Swish function, which is defined as follows:

$$\text{Swish}(x) = x \cdot \sigma(\beta x) \tag{1}$$

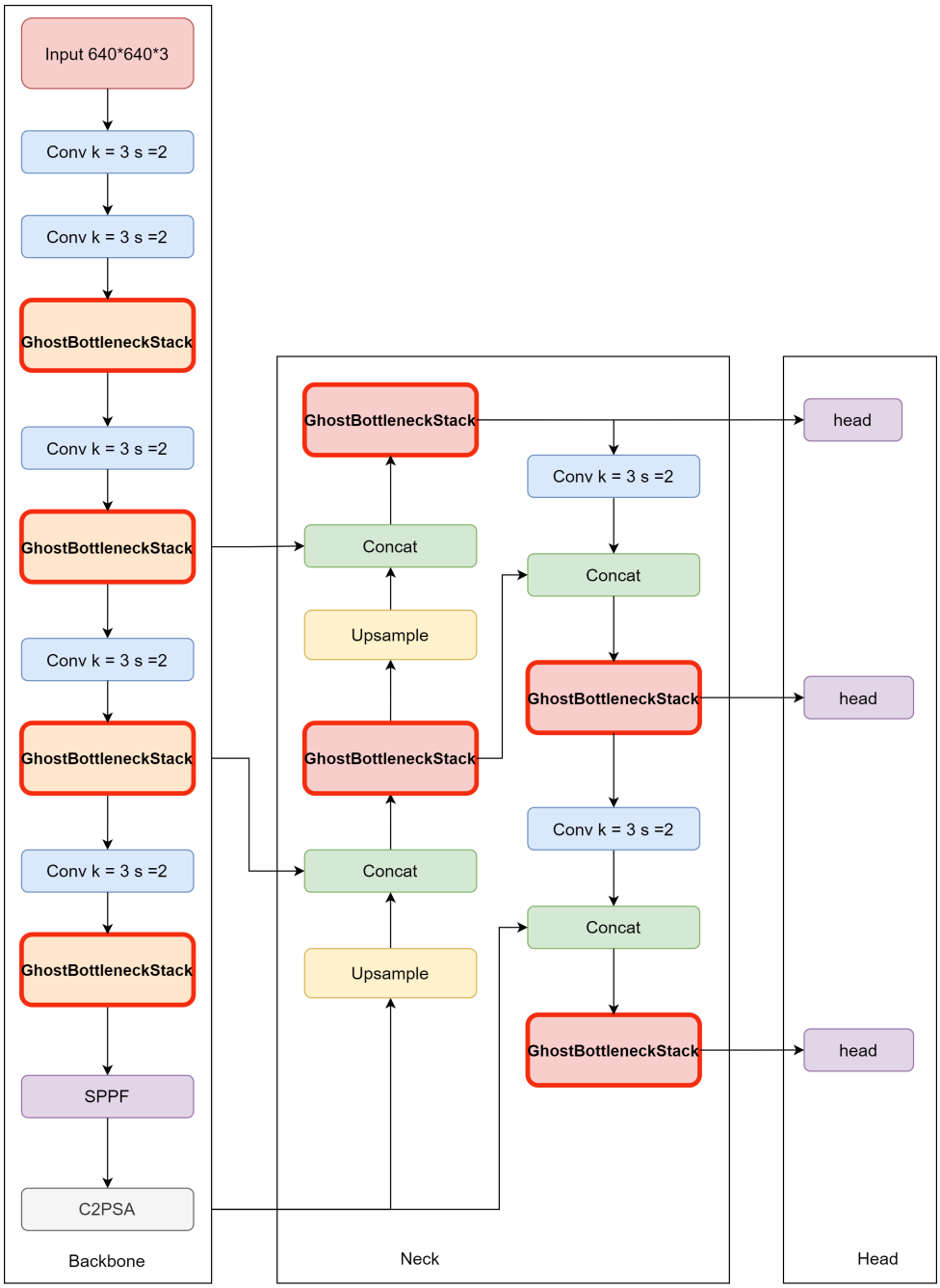


Figure 3. Architecture diagram of the YOLO11-Ghost model

The Swish function can effectively mitigate the gradient vanishing problem; however, its inclusion of the Sigmoid operation leads to computational complexity. HardSwish simplifies this through a piecewise linear function, expressed as:

$$\text{HardSwish}(x) = x \cdot \text{HardSigmoid}(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6} \tag{2}$$

This design significantly improves computational efficiency while preserving the advantages of the Swish function's non-monotonicity and smoothness. HardSwish supports operator fusion in mainstream inference engines and exhibits excellent quantization compatibility, making the model better suited for edge computing deployment scenarios.

3. **Experimental results and analysis.**

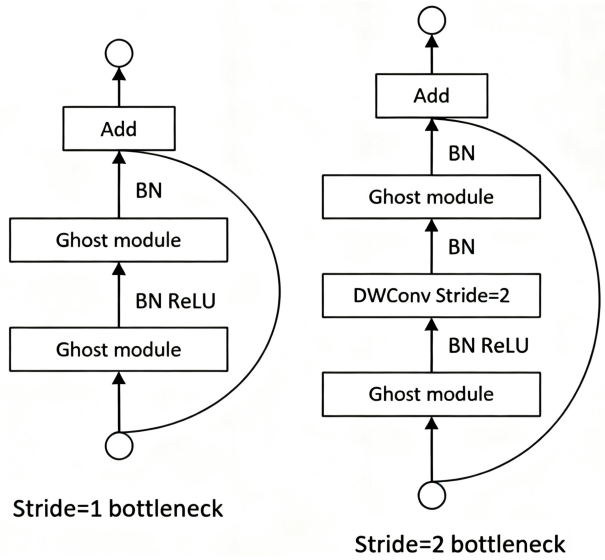


Figure 4. GhostBottleneckStack module

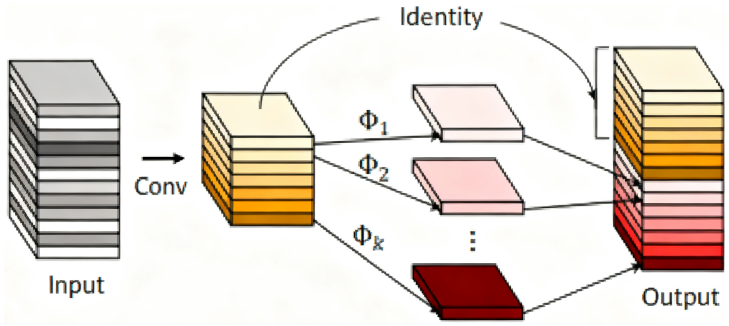


Figure 5. Ghost module

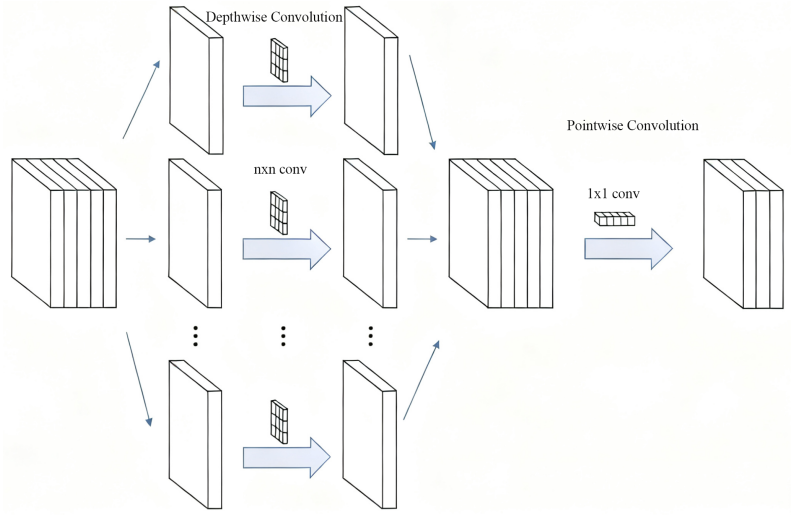
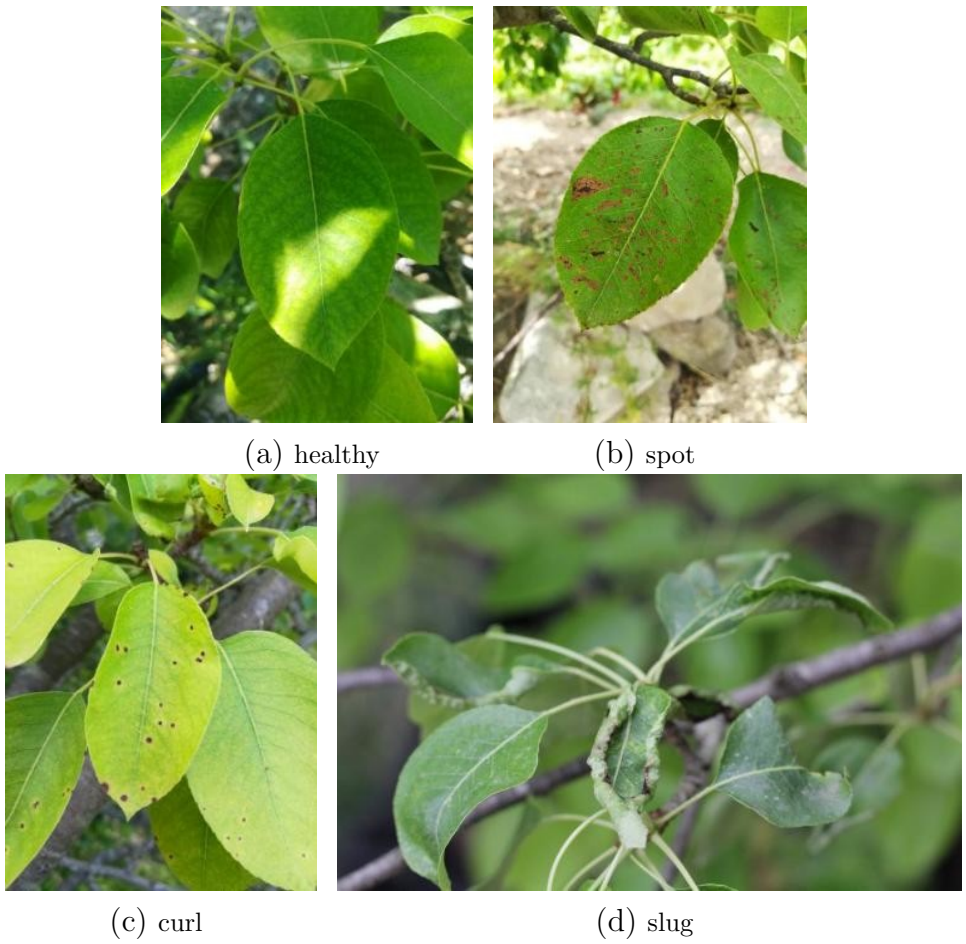


Figure 6. Depthwise separable convolution module

3.1. **Dataset.** This study used the DiaMOS Plant dataset [16], which was constructed for pear leaf disease detection tasks and consists of images captured in real-world scenarios. The image data of this dataset was collected from actual orchards on Sardinia, Italy, under varying illumination conditions, shooting angles, and background environments. It can

comprehensively and realistically capture the disease characteristics of pear leaves in their natural growth state, providing a data foundation that aligns with practical application scenarios for model training.

The dataset contains 3,505 images, including 3,006 leaf images covering 4 typical symptom categories: healthy (43 images), spot (884 images), curl (54 images), and slug (2,025 images). For the convenience of the research, only these four target categories (healthy, spot, curl, and slug) were detected in this study. Representative samples of the DiaMOS Plant dataset are shown in Figure 7. Pear fruits in the dataset were not included in this experiment. The dataset was divided into training, validation, and test sets at an approximate ratio of 7:1.5:1.5, and the detailed division is presented in Table 1 below.



(a) healthy (b) spot

(c) curl (d) slug

Figure 7. Representative Samples of the DiaMOS Plant Dataset

Table 1. Dataset partitioning table

Symptom Type	Training Set	Validation Set	Test Set	Total Number
healthy	30	6	7	43
spot	619	133	132	884
curl	38	8	8	54
slug	1418	304	303	2025
total	2105	451	450	3006

The dataset exhibits a significant class imbalance issue, with the number of healthy and curl samples being substantially lower than that of slug samples. This may lead

to model bias toward the majority class, loss function imbalance, and evaluation metric distortion. To address this, minority classes (healthy and curl) were expanded to approximately 200 samples each through data augmentation, and GAN-based oversampling. The medium-sized class (spot) was slightly augmented to 800 samples, while the majority class (slug) was mildly undersampled to 1,200 samples. The training set ratio was adjusted to healthy:curl:spot:slug $\approx$ 1:1:4:6, reducing inter-class sample disparity while preserving the original class importance order. To maintain evaluation consistency, the quantities of the validation and test sets remain unadjusted. Preserving the original distribution of the validation and test sets helps reflect real-world scenarios.

3.2. **Training.** Table 2 summarizes the experimental training environment, while Table 3 details the key hyperparameters. The model was trained for 300 epochs with a batch size of 32 and an initial learning rate of 0.001. To mitigate overfitting risks associated with the small-scale dataset, a comprehensive strategy combining strong regularization and extensive data augmentation was implemented. Regularization techniques included weight decay, DropBlock, and label smoothing. Data augmentation encompassed color space adjustments (hue, saturation, lightness), random flipping, mosaic stitching, and image mixing. These measures collectively enhanced the model's robustness to varying lighting conditions, leaf orientations, and complex backgrounds.

Table 2. Experimental conditions

Experimental Environment	Details
Programming language	Python 3.10
Operating system	Ubuntu 22.04 LTS
Deep learning framework	2.0.1+cu117
CPU	13th Gen Intel(R) Core(TM) i7-13700
GPU	NVIDIA GeForce RTX 4090 D

Table 3. Key parameters of experimental training

Training Parameters	Details
Epochs	300
Batch size	32
Image size(pixels)	640*640
Pre-training weights	None

3.3. **Model evaluation metrics.** In this study, bounding box loss, classification loss, distributed focal loss, precision, and recall are used to conduct a comprehensive and effective evaluation of the model during the experiment. The loss function of YOLO is designed to optimize both classification and localization tasks, and usually includes bounding box regression loss, classification loss, distributed focal loss, etc. Bounding box regression loss is the part used to optimize the difference between the predicted bounding box and the ground truth bounding box. Classification loss is the part used to optimize the accuracy of the model's prediction of target categories, ensuring that the model correctly identifies which category an object in the image belongs to. Distributed focal loss mainly solves the problems of class imbalance in object detection and improves the model's performance in handling small targets and hard samples. Precision measures the proportion of samples predicted as positive by the model that are actually positive. A high precision means that most of the samples predicted as positive by the model are

correct, reducing false positives. Recall measures the proportion of all actually positive samples that are correctly predicted by the model. A high recall means that the model can capture most of the actually positive samples, reducing false negatives.

3.4. **Model training.** Figure 8 illustrates the variation trends of bounding box loss, classification loss, distributed focal loss, precision, and recall of the model across different training epochs. It provides curves depicting changes in loss functions and evaluation metrics during the training process of the improved model, offering critical evidence for assessing model convergence, generalization capability, and overfitting risks.

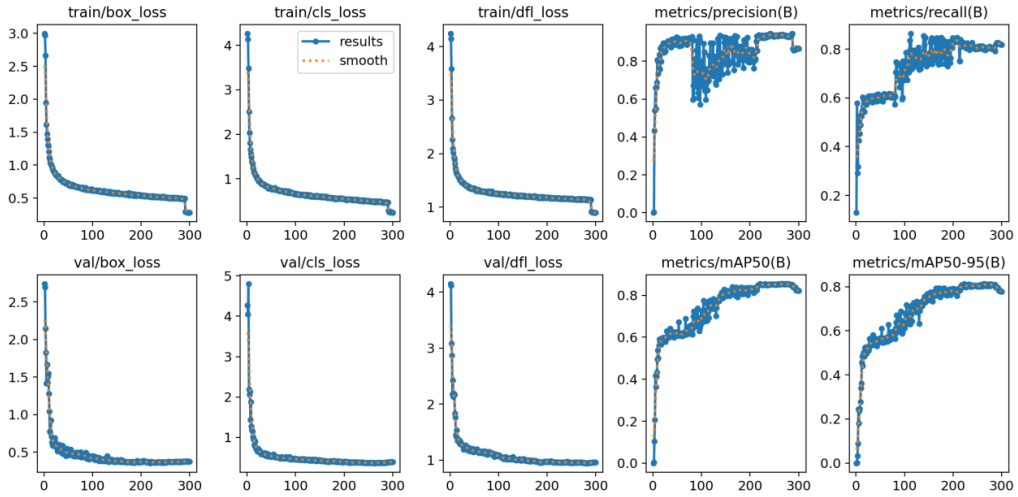


Figure 8. Training results of YOLO11-Ghost

3.4.1. *Analysis of loss function convergence.* The bounding box regression loss (box_loss), classification loss (cls_loss), and distributed focal loss (dfl_loss) on both the training and validation sets show synchronous declining trends, gradually stabilizing after 150 training epochs. The close alignment between the validation and training loss curves without significant divergence indicates no severe overfitting during training. This outcome is closely associated with the strong regularization strategy and early stopping mechanism employed.

3.4.2. *Stability evaluation of performance metrics.* The Precision, Recall and mean Average Precision (mAP50, mAP50-95) curves exhibit steady increases with training epochs and eventually saturate in the later stages, further confirming the effectiveness and stability of the model's learning process. The final validation mAP50 reaches approximately 0.822, demonstrating the model's strong object detection capability.

Although the single training curve indicates favorable convergence, three independent repeated experiments were conducted to verify robustness. The performance metrics are reported based on mean and standard deviation in Table 4.

3.4.3. *Control of overfitting risks.* By monitoring the plateau in validation metrics, the absence of fluctuating increases in all loss functions during the later training stages confirms that the regularization strategy and data augmentation methods effectively enhanced model generalization. The convergence pattern of the distributed focal loss (dfl_loss) suggests that feature learning for minority classes still requires enhancement. Overall, the

Table 4. Performance statistics of the improved model (YOLO11-Ghost) from three independent runs

Experiment	Precision	Recall	F1-Score	mAP50	mAP50-95
Run 1	86.52%	81.45%	85.74%	81.95%	77.61%
Run 2	87.03%	82.16%	86.24%	82.37%	78.02%
Run 3	86.41%	81.73%	85.97%	82.08%	77.79%
Mean±Std	86.65%±0.26%	81.78%±0.29%	85.98%±0.21%	82.13%±0.18%	77.81%±0.18%

training curves indicate that the improved model achieves stable convergence and satisfactory generalization on the given dataset, with well-controlled overfitting risks. Subsequent statistical results from multiple experiments further validate the reliability and consistency of the model's performance.

To validate the stability of the model performance, three independent experiments were conducted. As shown in Table 4, the improved model demonstrates high consistency across all key metrics: the standard deviation of mAP50 is only 0.18%, and that of Recall is 0.29%. This indicates that the model is robust and relatively insensitive to parameter initialization and training randomness. Furthermore, the model maintains its lightweight characteristics, making it suitable for deployment on edge devices.

3.5. **Comparison experiment.**

3.5.1. *Comparative analysis.* To systematically evaluate the effectiveness of the improved model YOLO11-Ghost, this section compares it with several mainstream lightweight models: EfficientDet-D1, MobileNetV3-YOLOv5s, YOLOv5nu, YOLOv8n, YOLO11n, and YOLO11s.The comparative analysis reveals the following key findings:

(1) YOLO-Ghost exhibits significant lightweighting advantages. As shown in Table 5, compared to YOLO11s, YOLO11-Ghost achieves a substantial reduction in both parameters and FLOPs, demonstrating the effectiveness of its lightweight design. Furthermore, compared to YOLO11n, YOLO11-Ghost significantly reduces FLOPs by 30.8% while increasing the parameter count by only 14.1%, highlighting its core advantage in higher computational efficiency.

Table 5. Comprehensive performance comparison on the DiaMOS plant dataset

Model	Precision	Recall	F1-Score	mAP50	mAP50-95	Params (M)	FLOPs (B)
EfficientDet-D1	89.5%	74.3%	81.2%	80.5%	76.1%	6.6	6.1
MobileNetV3-YOLOv5s	88.2%	75.8%	81.6%	81.0%	76.8%	5.4	12.2
YOLOv5nu	90.1%	77.5%	83.3	82.8%	78.5%	2.6	7.7
YOLOv8n	91.8%	78.6%	84.7%	84.2%	79.9%	3.2	8.7
YOLO11n	92.82%	76.56%	85%	83.85%	79.88%	2.62	6.5
YOLO11s	93.73%	79.22%	86.15%	85.3%	81.28%	9.46	21.4
YOLO11-Ghost	86.76%	81.88%	86%	82.2%	77.83%	2.99	4.5

(2) YOLO-Ghost achieves a superior performance trade-off. Compared to YOLO11n, YOLO11-Ghost experiences a decrease in Precision (-6.06%) and mAP50-95 (-2.05%). However, this represents an intentional design trade-off. The optimization objective of YOLO11-Ghost is not solely to pursue the highest accuracy, but rather to greatly enhance the model's Recall and efficiency under controllable accuracy loss. YOLO11-Ghost achieves the highest Recall among all models at 81.88%, which is an increase of 5.32% over YOLO11n.

(3) Comparison with YOLOv5nu/YOLOv8n. YOLO11-Ghost exhibits a clear computational efficiency advantage, with FLOPs 41.6% and 48.3% lower than YOLOv5nu and YOLOv8n, respectively. Although its mAP50-95 is slightly lower than that of YOLOv8n, its advantage in Recall is significant. This reflects the differentiated path chosen by the YOLO11-Ghost model in balancing accuracy, recall, and efficiency.

(4) Comparison with EfficientDet-D1. Compared to the classic lightweight architecture EfficientDet-D1, YOLO11-Ghost achieves higher mAP50-95 and Recall with fewer parameters and FLOPs, demonstrating the advanced nature of its architecture.

3.5.2. *Mechanism analysis of performance trade-offs.* (1) Mechanism of accuracy decline. The decrease in the Precision and mAP of YOLO11-Ghost primarily stems from the application of the GhostBottleneckStack module. While this module reduces computational load through operations like depthwise separable convolutions, this process sacrifices some high-frequency detailed features. This leads to a decrease in classification confidence and localization for some hard samples, consequently causing the drop in accuracy metrics. This is a typical manifestation of the "efficiency for feature representation" trade-off in lightweight design.

(2) Mechanism of recall improvement. The significant improvement in Recall benefits from the synergistic effect of the HardSwish activation function and the lightweight design. HardSwish offers better non-linear characteristics in low-precision regions, enabling more effective activation of weak features. Combined with the reduced model complexity from GhostBottleneck, the model's sensitivity to features of blurred and small-sized targets in the background is enhanced, thereby reducing missed detections and improving Recall.

3.5.3. *Scenario applicability.* The characteristics of "high recall, relatively high efficiency, and acceptable accuracy" make YOLO11-Ghost particularly suitable for application scenarios where tolerance for missed detections is extremely low, such as crop disease identification. Conversely, for scenarios demanding high precision, such as certain industrial quality inspections, this model is not the optimal choice.

3.6. **Ablation experiments.** To further verify the effectiveness of each improvement in the proposed modified model regarding computational resource requirements, ablation experiments were conducted using the test set. Following the principle of controlling variables, we incorporated each module into the neural network one by one; after which, the model's parameters, FLOPS, detection time per image, and model size were measured to evaluate the effectiveness of the lightweight design. The results of the ablation study are summarized in Table 6.

As shown in Table 6, compared with those of YOLO11s, the model parameters decreased by 68.39%, FLOPS were reduced by 78.97%, and the model size decreased by 67.59%—indicating that the modified model is significantly more lightweight. After the GhostBottleneckStack module and HardSwish were integrated into YOLO11s, the model parameters increased by 14% while its FLOPS were reduced by 30.8%. Despite the slight increase in parameter count, the computational complexity was significantly reduced. The improved model is thus more suitable for edge devices or real-time inference scenarios, providing different optimization directions for accuracy-efficiency trade-offs in various scenarios.

As shown in Table 5, compared with YOLO11s and YOLO11n, the improved model exhibited a decrease in precision but an increase in recall. Since GhostBottleneck significantly reduces computational load through lightweight design, this design inevitably loses some detailed features—resulting in slightly lower precision than that of YOLO11n. As an efficient activation function, HardSwish can enhance the nonlinear expression of features

and is compatible with hardware acceleration, leading to more stable feature extraction on edge devices.

Table 6. Ablation experiments

Model	Precision	Recall	F1-Score	mAP50	mAP50-95	Params (M)	FLOPs (B)
YOLO11s	93.73%	79.22%	86.15%	85.3%	81.28%	9.46	21.4
YOLO11s +GhostBottleneckStack	86.51%	77.84%	84.31%	82.93%	78.96%	2.69	5.5
YOLO11s+HardSwish	87.42%	77.23%	85.47%	84.21%	80.36%	2.62	6.2
YOLO11s +GhostBottleneckStack +HardSwish	86.76%	81.88%	86%	82.2%	77.83%	2.99	4.5

4. **Conclusion and future work.**

4.1. **Conclusion.** This paper proposes YOLO11-Ghost, a pear leaf disease detection algorithm based on an improved YOLO11 model. The core improvements of the model include:

Replacing the original C3k2 module with the GhostBottleneckStack module. This leverages the synergistic effect of a lightweight network stacking structure designed based on the Ghost module concept and depthwise separable convolution, reducing computational overhead while preserving leaf texture and morphological features, thereby enhancing the ability to capture characteristics of early-stage pear tree diseases. Globally adopting the HardSwish activation function to address the operator decomposition issue of the SiLU activation function during ONNX export and to improve TensorRT INT8 quantization accuracy.

Experiments on a pear leaf disease dataset collected from complex orchard scenarios demonstrate that, compared to YOLO11s, the proposed model reduces the number of parameters by 68.39%, computational FLOPs by 78.97%, and model size by 67.59%, exhibiting significant lightweight characteristics. Simultaneously, the Recall rate increased by 2.66%, indicating an improvement in reducing missed detections of disease targets.

However, the lightweight improvements also come with a trade-off in accuracy. The model's Precision decreased by 6.97%, and the mean Average Precision (mAP50) decreased by 3.1%. This is primarily because the Ghost module generates "ghost" feature maps through linear operations, whose representational capacity has an inherent upper limit compared to the full computation of standard convolutions. While drastically reducing model complexity, this design inevitably leads to a decreased ability to learn some subtle and challenging features, consequently increasing the likelihood of false positives and affecting overall localization and classification accuracy.

Nevertheless, the YOLO11-Ghost model provides an efficient solution for pear leaf disease detection in resource-constrained scenarios. Its high recall rate makes it particularly suitable for preliminary screening tasks in large-scale orchard inspections. It can rapidly process massive images generated by monitoring equipment, effectively locating suspected disease areas even under conditions of overlapping leaves or during the early stages of disease, thus providing reliable data support for subsequent precision pesticide application.

4.2. **Limitations and future work.** This study still has several limitations that need further exploration in future work:

(1) Detection performance for minority classes and complex diseases needs improvement. The model's detection performance is relatively lower for disease categories with fewer samples or atypical features, such as "leaf curl disease." This indicates insufficient generalization capability of the model on class-imbalanced datasets. Future work will focus on strategies like introducing Focal Loss to address the class imbalance issue and collecting more samples of rare diseases to optimize the data distribution.

(2) Trade-off between model accuracy and speed. The current model emphasizes extreme lightweight design, resulting in significant accuracy loss. The next step involves exploring more refined lightweight designs, such as employing differentiated lightweight strategies at different network layers or introducing lightweight attention mechanisms, striving to recover accuracy loss as much as possible while maintaining high efficiency.

(3) Generalization and potential biases of the dataset. Although the dataset used in this study comes from real complex orchard scenarios, potential hidden biases may still exist regarding lighting conditions, capture devices, pear tree varieties, etc. This might affect the model's generalization ability to other orchards, different varieties, or broader environmental conditions. Future plans include collaborating with orchards in different regions to build a larger-scale, more diverse dataset and conducting rigorous evaluation of the model's cross-domain generalization performance.

(4) Feasibility of extending to other crops. The proposed model is specifically designed for pear leaf disease detection. While its lightweight architecture offers valuable insights for disease detection in other crops, direct application might not guarantee satisfactory performance. Future work will validate the transferability of this method and adapt it to the disease characteristics of different crops, aiming to develop a universal lightweight framework for crop disease detection.

In conclusion, this research provides technical support for implementing smart agriculture and lays a preliminary technical foundation for the intelligent upgrade of pest and disease control. Addressing the aforementioned limitations will be the focus of subsequent research efforts.

Acknowledgment. This research was supported by the Natural Science Foundation of Fujian Province in China (Grant No.2019J01088). This research was supported by the 2024 Undergraduate Higher Education Teaching Research Project of Fujian Provincial Department of Education (Project No. FBJY20240280).

REFERENCES

[1] T. Wu, Z. Miao, W. Huang, W. Han, Z. Guo, and T. Li, "Sgw-yolov8n: An improved yolov8n-based model for apple detection and segmentation in complex orchard environments," *Agriculture*, vol. 14, no. 11, p. 1958, 2024.

[2] D. Wang and D. He, "Channel pruned yolo v5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning," *Biosystems Engineering*, vol. 210, no. 6, pp. 271–281, 2021.

[3] Y. Hui, C. Zhao, Z. Song, and X. Zhao, "Context-enhanced yolov5s for accurate apple detection in complex natural environments," *Measurement Science and Technology*, vol. 36, no. 1, p. 015432, 2024.

[4] Y. Xu and L. Zuodong, "Fruit fast tracking and recognition of apple picking robot based on improved yolov5," *IET Image Processing*, vol. 18, no. 12, pp. 3179–3191, 2024.

[5] T.-Y. Wu, C. Yu, H. Li, S. Kumari, and L. Y. Por, "Tegoa-cnn: An improved gannet optimization algorithm for cnn hyperparameter optimization in remote sensing sence classification," *Remote Sensing*, vol. 17, no. 17, p. 3087, 2025.

[6] B. Jiang, H.-B. Li, Y.-L. Xu, K. Ding, and X.-D. Zhao, "Tfanet: Vehicle sound-based classification utilizing time-frequency hybrid attention," *Journal of Network Intelligence*, vol. 10, no. 1, pp. 112–129, 2025.

[7] Z.-B. Huang, X.-Y. Chen, and E. Huang, "Deep learning model-based emotion recognition for visual communication in human-computer interaction," *Journal of Network Intelligence*, vol. 10, no. 1, pp. 502–515, 2025.

[8] R. P. A. Pramesti, M. R. Supriyadi, M. R. Alfin, J. G. Pinem, A. H. K. M. Subekti, U. Chasanah, G. M. Putra, M. Wibowo, D. H. Budiarti, J. Muliadi *et al.*, "Microscopic image classification of fungal colonies using vgg pre-trained model," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 16, no. 2, pp. 541–553, 2025.

[9] M. A. Assaad, M. I. Saleh, and R. Mahrousseh, "A novel framework for accurate brain tumor detection in mri scans using cnn, mlp, and knn techniques," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 16, no. 2, pp. 590–602, 2025.

[10] H. Mo and L. Wei, "Lightweight citrus leaf disease detection model based on arms and cross-domain dynamic attention," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 7, p. 102133, 2024.

[11] M. Li, Z. Tao, W. Yan, S. Lin, K. Feng, Z. Zhang, and Y. Jing, "Apnet: Lightweight network for apricot tree disease and pest detection in real-world complex backgrounds," *Plant Methods*, vol. 21, no. 1, p. 4, 2025.

[12] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense," *Journal of Sensors*, vol. 2019, no. 1, p. 7630926, 2019.

[13] D. Luo, Y. Xue, X. Deng, B. Yang, H. Chen, and Z. Mo, "Citrus diseases and pests detection model based on self-attention yolov8," *IEEE Access*, vol. 11, pp. 139 872–139 881, 2023.

[14] M. Qiu, L. Huang, and B.-H. Tang, "Asff-yolov5: Multielement detection method for road traffic in uav images based on multiscale feature fusion," *Remote Sensing*, vol. 14, no. 14, p. 3498, 2022.

[15] H. Hu, M. Chen, L. Huang, and C. Guo, "Bhi-yolo: A lightweight instance segmentation model for strawberry diseases," *Applied Sciences*, vol. 14, no. 21, p. 9819, 2024.

[16] G. Fenu and F. M. Malloci, "Diamos plant: A dataset for diagnosis and monitoring plant disease," *Agronomy*, vol. 11, no. 11, p. 2107, 2021.

[17] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[20] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision*. Springer, 2024, pp. 1–21.

[21] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[23] S. Ennaama, H. Silkan, A. Bentajer, and A. Tahiri, "Enhanced real-time object detection using yolov7 and mobilenetv3," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19 181–19 187, 2025.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[25] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.