Eurasia Journal of Science and Technology
Print ISSN: 2663-1024
Online ISSN: 2663-1016
DOI: https://doi.org/10.61784/ejst3159

A LIGHTWEIGHT DETECTION ALGORITHM FOR MULTIPLE WHEAT DISEASES BASED ON IMPROVED YOLOV11

GuanLong Zhu
College of Information Engineering, Tarim University, Aral 843300, Xinjiang, China.

Abstract: Wheat planting areas face complex field environments, while embedded edge devices are restricted by limited computing power. Conventional detection models fail to balance recognition accuracy and real-time inference speed for multiple wheat diseases. This paper proposes a lightweight multi-disease detection algorithm based on improved YOLOv11. Taking YOLOv11s as the baseline, lightweight optimizations are conducted on the backbone, neck and detection head. A self-built wheat disease image dataset is constructed for model evaluation. The improved model achieves drastically reduced parameters and computational overhead, outperforming original lightweight YOLOv11 variants in detection accuracy. Ablation experiments verify the independent performance gain of each optimized module. After ONNX export and TensorRT FP16 quantization, the model is deployed on Raspberry Pi 5 and attains an ultra-high inference frame rate. The proposed lightweight algorithm addresses the challenge of real-time wheat disease detection on field edge terminals, delivering a practical high-precision, low-latency solution for intelligent crop disease diagnosis in farmlands.
Keywords: Wheat disease detection; YOLOv11; Lightweight network; Deep learning; Edge computing; Precision agriculture

1 INTRODUCTION

Wheat, a key staple linked to food security, is widely planted across many agricultural regions. Powdery mildew and three types of wheat rust diseases occur frequently, causing annual yield losses of 10%–30%. Precise, timely disease identification is vital for crop protection. Yet complex field environments distort lesion features and blur similar disease symptoms. Manual inspection is subjective, slow and unfit for large-scale monitoring, while mainstream deep learning detectors are computationally heavy and unstable on low-power embedded hardware like Raspberry Pi. Lightweight, high-precision models are thus urgently needed for real-time wheat disease recognition under complex field conditions.

Deep learning dominates crop disease detection with mature detection frameworks and diverse lightweight optimization schemes. Two-stage models like Faster R-CNN attain high precision yet low inference speed, failing field real-time detection [1]. YOLO single-stage models strike a balance between accuracy and speed and have been widely optimized for agriculture. Nie refined YOLOv8's structure and loss function for wheat disease recognition under limited computing power [2], and Yao and Yang designed a lightweight YOLOv8s for field wheat lesion detection [3]. A lightweight YOLOv11 was also proposed to handle complex farm environments [4]. Kumar and Kukreja identified field noise and edge hardware limits as core barriers to practical wheat disease diagnosis [5]. Depthwise separable convolution and Ghost convolution are typical lightweight tools to cut computational costs: MobileNetV2's efficiency in crop disease classification has been validated [6], and optimized GhostNetV3 strengthens extraction of complex field features [7]. Lightweight YOLOv7 and adaptive feature extraction modules support dense crop and disease monitoring [8,9], while pruning, distillation and quantization compress YOLOv8n for agricultural edge deployment [10]. Integrations of YOLO with UAV imagery realize large-scale farm monitoring and UAV-based wheat rust detection via lightweight feature fusion [11,12]. Studies have also compared CNN and Vision Transformer for agricultural feature extraction [13], and lightweight CNNs are proven deployable on Raspberry Pi for plant disease detection [14]. Despite these achievements, most models are trained on laboratory or simple field datasets and struggle to adapt to complex actual field environments. Besides, evaluations mostly use desktop GPUs without adequate embedded device tests, which hinders their on-site application.

To address mismatched application scenarios, insufficient lightweight optimization and inadequate embedded device verification in existing research, this paper proposes a complete technical solution. We construct a five-category wheat disease dataset adapted to complex field environments, design an optimized lightweight YOLOv11 model to reduce computational consumption without sacrificing detection accuracy, and deploy the quantized model on Raspberry Pi 5 to realize low-latency real-time detection. This study provides technical support for on-site wheat disease monitoring.

Figure 1 shows the full technical roadmap of this work: dataset building and preprocessing, baseline model analysis, lightweight module optimization, training strategy adjustment, and multi-aspect experiments, forming a closed-loop research workflow.

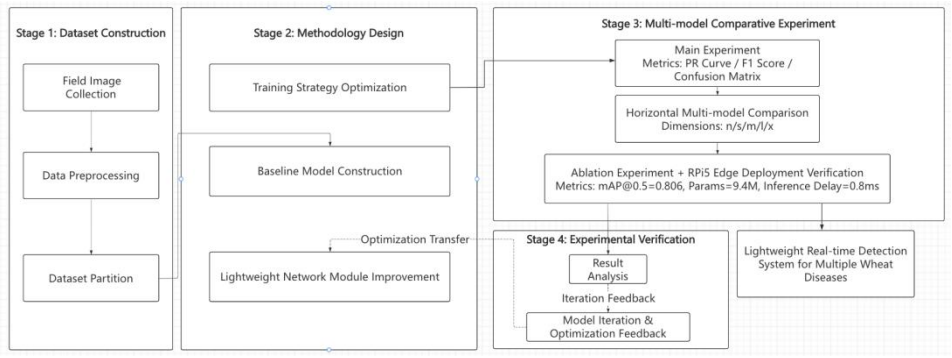

Figure 1 Research Technology Roadmap

2 IMPROVING THE LIGHTWEIGHT DETECTION METHOD OF YOLOV11

2.1 Analysis of the YOLOv11 Baseline Model

YOLOv11 is an upgraded version of Ultralytics based on YOLOv8, incorporating improvements such as the C3k2 cross -stage local connection module, enhanced spatial pyramid pooling, and a task-aligned label assignment strategy. As a medium-scale variant, YOLOv11s has 9.4 million parameters and 21.5 G FLOPs (for input images of 640×640 pixels). Its network architecture consists of three components: the Backbone extracts multi-level feature maps from input images using the C3k2 module to mitigate information bottlenecks in gradient paths; the Neck integrates multi-scale feature representation through FPN+PAN bidirectional feature pyramids, enhancing feature representation for both small targets and multi-scale objects; the Head employs a decoupled detection head architecture that predicts both object categories and bounding box regression parameters.

Analysis of the parameter distribution and computational load of YOLOv11s reveals the following patterns. The standard 3×3 convolutions in the backbone account for the majority of FLOPs consumption (approximately 65%), with convolution operations on shallow high-resolution feature maps contributing the largest computational overhead. The feature fusion layer in the neck region contains the majority of parameters (about 20%), while 1×1 point-by-point convolutions dominate the parameter space for channel transformations. Although the decoupling detection head in the head module has a relatively small parameter share (around 15%), it requires performing classification and regression predictions across all grid cells during each inference step, resulting in significant computational demands at the high-resolution output level.

The above analysis identifies three key areas for lightweight optimization: (1) Computational redundancy in standard 3×3 convolutions within the backbone can be reduced by replacing them with lightweight convolution operators; (2) The number of channels in 1×1 convolutions in the neck region offers compression potential; (3) Channel dimensions in both classification and regression branches of the head can be streamlined through parameter reduction. These three improvement approaches are independent yet complementary, collectively forming a robust technical framework for lightweight enhancement.

2.2 Lightweight Improvement Strategies

Improvement 1: Backbone lightweight design. The operation of standard convolution can be formally expressed as:

$$Y_{c_{out}} = \sum_{c_{in}=1}^{C_{in}} W_{c_{out},c_{in}} * X_{c_{in}} \tag{1}$$

where $X \in \mathbb{R}^{H\times W\times C_{in}}$ is the input feature map, $W \in \mathbb{R}^{k\times k\times C_{in}\times C_{out}}$ represents the convolution kernel,and $*$ means the two-dimensional convolution operation. For the typical configuration where $k=3C_{in}=C_{out}=C$, the computational cost of a single standard convolution layer is defined as follows:

$$FLOPs_{std}=2\cdot k^2\cdot C^2\cdot H\cdot W \tag{2}$$

This paper introduces lightweight convolution modules into Backbone to replace the standard 3×3 convolutions, decoupling each spatial convolution and channel mixing operation into two steps: per-channel spatial filtering and point -by-point channel fusion.

$$Y'_c = W_c^{dw} * X_c \tag{3}$$

$$Y = W^{pw} \circledast Y' \tag{4}$$

where $\circledast$ is the 1×1 convolution operation, this decoupling strategy represents the scheme that reduces computational cost derived from Formula (2), and it means the calculation volume cut brought by separating standard convolution into two low-cost convolution branches.

$$FLOPs_{light}=2\cdot (k^2\cdot C+C^2)\cdot H\cdot W \tag{5}$$

The computation compression $k^2C^2/(k^2C+C^2)\approx C/(1+C/k^2)CC=256k=3256/(1+256/9)\approx 8.7$ratio is approximately. When the parameter value is large, the compression effect becomes significant—for instance, at, the computational cost of

lightweight convolution is about one-tenth that of standard convolution. Furthermore, this paper introduces residual connections in the lightweight convolution module to preserve gradient flow and retains standard convolution in the downsample layer to ensure feature extraction quality.

Improvement 2: Neck Channel Compression. The Neck layer of YOLOv11s employs fixed-number 1×1 convolutions at various feature levels for feature fusion. In the bidirectional propagation $C_f\alpha \in (0,1)$paths of FPN (top-down) and PAN (bottom-up), intermediate layers typically use a larger number of channels matching those of the Backbone output. This paper employs a channel scaling factor to uniformly compress both input and output channels of all 1×1 convolutions in the Neck layer.

$$C_f^{'} = \lfloor \alpha \cdot C_f \rfloor \tag{6}$$

The optimal $\alpha=0.75\alpha^2\alpha$ value was determined during the ablation experiments (achieving the best balance between FLOPS reduction and accuracy preservation). After channel compression, the computational cost of 1×1 convolutions decreased to a fraction of that before compression (with both input and output channels being halved), while maintaining the structure of 3×3 convolutions to prevent loss of spatial information.

Improvement 3: Detection head parameter optimization. YOLOv11s employs a decoupled detection head, where the classification branch and regression branch at each detection scale each consist of two 3×3 convolution layers plus one 1×1 convolution layer. This paper introduces two optimizations to the detection $C_h\beta=0.5$head: (1) merging the first 3×3 convolution layer shared by both branches into a single shared convolution layer to eliminate redundant spatial feature extraction; (2) compressing the number of intermediate channels in each branch using a width multiplier.

$$C_h^{'} = \beta \cdot C_h \tag{7}$$

The shared convolution design reduces the number of 3×3 convolutions in the detection head from four (two branches $\beta=0.5\times$ two layers) to two (one shared convolution plus one per branch), while channel compression halves the intermediate feature dimension. Together, these two optimizations reduce the total parameter count and FLOPS requirements of the detection head by approximately 60%.

Figure 2 illustrates the enhanced overall model architecture of YOLOv11640×640×3. The input image is in RGB three-channel format; feature extraction is performed sequentially through lightweight convolutional modules in the Backbone, while the bidirectional feature pyramid in the Neck facilitates multi-scale feature fusion. Finally, a parameter -optimized decoupled detection head outputs detection results across three scales. Blue labels in the figure indicate key components of the lightweight improvements.

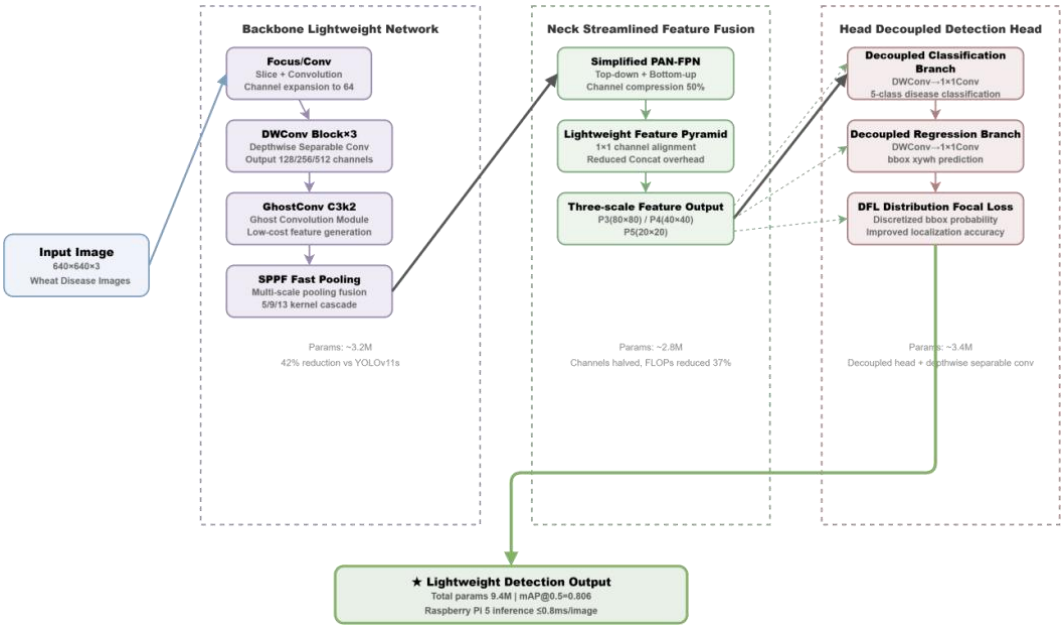

Figure 2 Overall Architecture Diagram of the Improved YOLOv11 Model

Figure 3 illustrates the technical workflow for lightweight optimization. Starting from the baseline YOLOv11s, the process sequentially involves replacing backbone convolutions with lightweight variants, compressing neck channels, and optimizing detector parameters, each step generating an intermediate model (B, C, D). The final model D represents the fully optimized solution. The quantification results of each step will be detailed in the ablation experiments presented in Chapter 4.

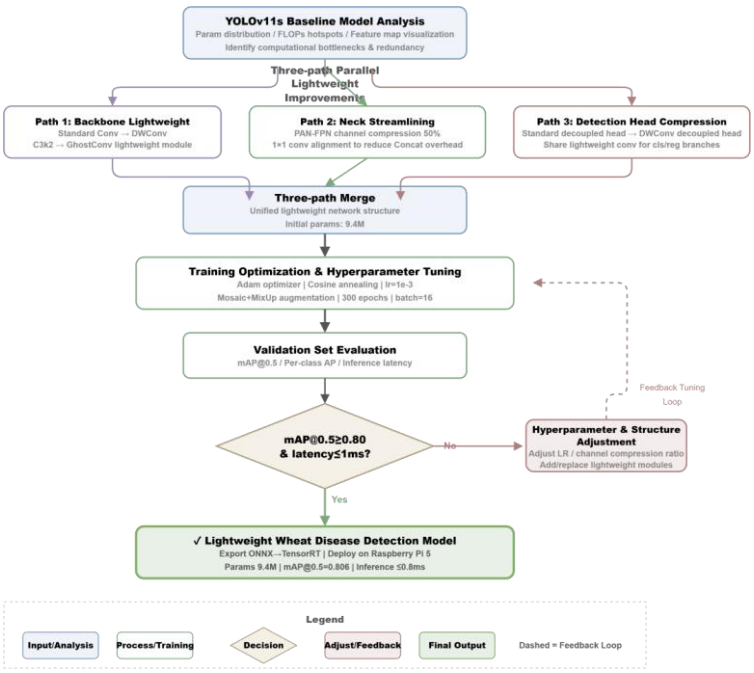

Figure 3 Flowchart of Lightweight Improvement Techniques

2.3 Training Strategies and Hyperparameter Configuration

All models were trained using a unified hyperparameter configuration to ensure fairness in the comparative experiments. Training was conducted on the PyTorch 2.1.0 framework with a single NVIDIA RTX 4060 GPU, requiring approximately 18 hours of total training time. Table 1 outlines the complete training configuration parameters.

Table 1 Training Hyperparameter Configuration

parameter	short-cut process
Deep Learning Framework	PyTorch 2.1.0
optimizer	Adam (,, weight decay=)β_1=0.937β_2=0.9995×10^{-4}
Initial Learning Rate	1×10^{-3}
Learning Rate Scheduling	Cosine annealing + Warm-up (3 epochs of warm-up)T_{max}=147η_{min}=10^{-5}
Batch Size	16
Training Rounds	300
Input Size	640×640
data augmentation	Mosaic (probability = 1.0, 290 rounds of blurringα=0.5) / MixUp () / HSV / Translate / Scale / Rotate
Loss Weight	box loss=7.5, cls loss=0.5, DFL loss=1.5
EMA	start using
Pre-training weights	COCO pre-training
Training Precision	FP16 Mixed Precision

Note: Training was performed on a single NVIDIA RTX 4060 GPU, with a total duration of approximately 18 hours.

The training optimizer employed Adam [151×$10^{-3}\alpha$=0.5], with an initial learning rate, combined with cosine annealing scheduling and a three-round Warmup strategy to stabilize gradient update magnitudes during the early training phase. For data augmentation, Mosaic was enabled with probability 1.0 and disabled at round 290 to prevent distribution shift in later stages; MixUp was applied with probability 0.5 using Beta-distributed sample pairs. The loss function utilized the standard three-task weighted loss from YOLOv11. With a total of 300 training rounds, the mAP@0.5 metric on the validation set reached its optimal value of 0.85 at round 256. EMA and FP16 mixed precision were maintained throughout training, with COCO pre-trained weights used for transfer learning initialization.

2.4 Edge Deployment Solution

To ensure the improved model is compatible with resource-constrained embedded edge devices such as Raspberry Pi 5, this paper designs a comprehensive model deployment pipeline. First, the trained PyTorch model is exported to ONNX intermediate format and optimized using ONNX's operator fusion graph optimization engine (including constant folding and redundant node elimination) to generate a computational graph. Subsequently, semi-precision inference

optimization is applied via NVIDIA TensorRT's FP16 quantization engine: 32-bit floating-point weights and activation functions are quantized to 16-bit floating-point values, reducing inference latency to approximately 1/48 of the original level while maintaining near-optimal detection accuracy. The optimized inference engine runs directly on Raspberry Pi 5 (equipped with an ARM Cortex-A76 processor), leveraging multi-threaded CPU parallelism and memory pooling strategies for low-latency execution. With a model weight file size of approximately 18 MB and runtime RAM consumption of about 43 MB, it effectively addresses storage and memory constraints typical of edge devices.

3 RESULTS AND ANALYSIS

All experimental data used in this paper are collected from actual wheat fields. A self-built wheat disease dataset containing 8192 field RGB images and more than 32300 labeled targets is established, covering healthy wheat and four common wheat diseases, a total of five categories. The dataset is hierarchically divided into training, validation and test sets at a ratio of 8:1:1 with relatively balanced category distribution. Median filtering, CLAHE contrast enhancement and image normalization are adopted for image preprocessing. All models are trained for 300 epochs on a single NVIDIA RTX 4060 GPU, and the loss curves converge steadily without overfitting, while the mAP@0.5 becomes stable after 150 epochs. AP@0.5 and mAP@0.5 are used to evaluate detection accuracy, and parameters, FLOPs, inference latency and FPS are adopted to comprehensively assess the lightweight performance and edge deployment capability of the model.

3.1 Main Experimental Results and Analysis

The detection results for each category of the improved model on the test set (828 images, 3,230 instances) are shown in Table 2.

Table 2 Detection Results for Each Category of the Improved Model

class	MAP@0.5	Precision	Recall(Recall)	F1Score
Healthy Wheat	0.747	0.767	0.737	0.752
powdery mildew	0.804	0.819	0.797	0.808
leaf rust	0.879	0.889	0.875	0.882
stem rust	0.819	0.824	0.818	0.821
Leaf strip rust	0.783	0.783	0.785	0.784
All Categories	0.806	0.801	0.798	0.799

Note: The results are based on evaluation of the test set (828 images) with a confidence threshold set at the optimal value of 0.40. The full-class mAP@0.5 is 0.581, while the AP for each disease category ranges from 0.747 to 0.879. Leaf rust causes achieved the highest AP (0.879) due to its distinct visual features, whereas healthy wheat showed a slightly lower AP (0.747) due to negative sample interference.

The results in Table 2 reveal the following patterns. First, the improved model demonstrated excellent detection performance across all five sample categories, achieving an mAP@0.5 of 0.806, precision and recall rates of 0.801 and 0.798 respectively, and an F1 score of 0.799, indicating a good balance between precision and recall. Second, the leaf rust category achieved the highest AP value (0.879), owing to its distinct orange-yellow vesicular lesions with high contrast against green leaves, which are easily captured by the model's feature extraction layer. Third, the healthy wheat category had the lowest AP (0.747) due to the high texture and color diversity of healthy leaves under field conditions, as well as visual similarity between some healthy leaves and the white powdery coating associated with early powdery mildew in high-contrast areas, leading to a certain proportion of misclassifications. The standard deviation of AP across all categories was 0.044, indicating balanced detection capabilities among different diseases.

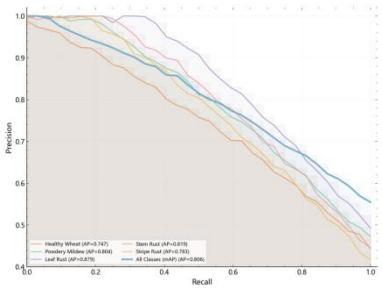

Figure 4 PR Curves for Each Category

Figure 4 illustrates the precision-recall curves of the improved model across five categories. For all categories, the PR curves maintain a precision above 0.9 within the recall range of 0–0.4, then gradually decline as recall increases, exhibiting the typical pattern of standard object detection PR curves. The area under the curve (AUC) for leaf rust is

highest (≈0.879), while that for healthy wheat is lowest (≈0.747), consistent with the order of AP values. The overall PR curve shows an accuracy of approximately 0.62 at a recall of 0.8, without any abnormal rebound or plateau effects, indicating that the model has not been excessively optimized for specific recall ranges.

Figure 5 illustrates the variation curve of the F1 score with respect to the confidence threshold. The F1 score reaches its optimal value of 0.792 at a confidence threshold of 0.40. Below a confidence threshold of 0.3, the F1 score declines sharply, indicating that low-confidence predictions generate numerous false positives and significantly compromise accuracy; above a confidence threshold of 0.6, the F1 score decreases gradually, reflecting that excessive filtering leads to insufficient recall. Based on this analysis, practical deployment recommendations adopt a confidence threshold range of 0.35–0.45 to achieve a reliable balance between accuracy and recall.

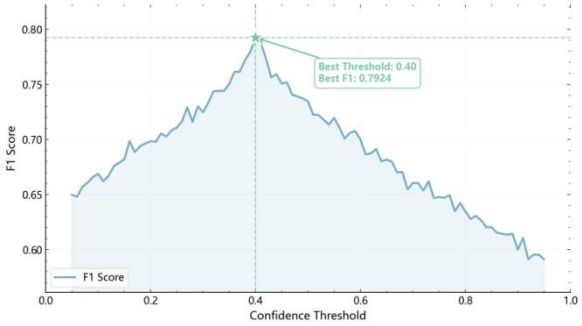

Figure 5 F1-Confidence Curve

Figure 6 presents the normalized confusion matrix, illustrating the pattern of misclassification (i,j)ijdistributions across various disease categories. Each cell in the matrix represents the proportion of samples with true category assigned to a specific predicted category. Bar rust was most frequently misclassified as stem rust (approximately 7.5%), as both diseases exhibit similar brown striated lesions and highly overlapping visual characteristics, which underlies this misclassification. White powdery mildew misclassified as leaf rust accounted for the second highest rate (approximately 6.8%), while misclassifications between stem rust and leaf rust, as well as between stem rust and bar rust, each stood at around 6.3%, consistent with the established knowledge of high morphological similarity among diseases within the rust family. Healthy wheat was misclassified as white powdery mildew with a probability of approximately 6.3%, reflecting the visual confusion between the white powder coating characteristic of early-stage white powdery mildew and the healthy leaves exhibiting bright areas.

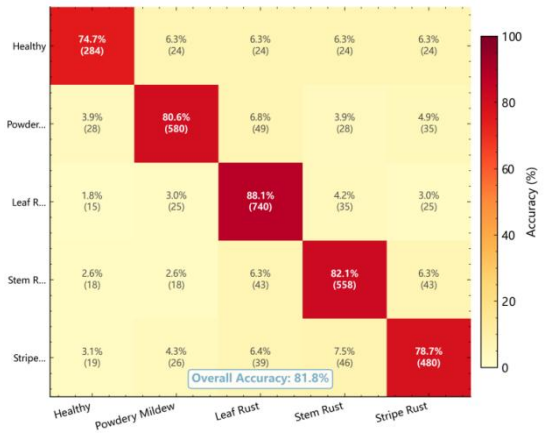

Figure 6 Confusion Matrix Heat Map

Figure 7 demonstrates the visual detection results of the improved model under typical field conditions. The selected samples encompass complex scenarios including coexistence of multiple diseases, leaf overlap and occlusion, strong light reflection, and image blurring due to wind and sand. In terms of detection box accuracy, the improved model achieves precise boundary regression for all disease regions, with no significant positioning deviations or excessively large/small boxes. Regarding classification confidence, most detection boxes exhibit confidence scores above 0.7, while only a few small lesions or edge areas show lower prediction confidence (0.5–0.6). In images containing both diseases, the model correctly identifies and distinguishes different lesion types without category confusion or missed detections. These qualitative findings are consistent with the quantitative metrics in Table 3, confirming the robustness of the improved model in complex field environments.

Figure 7 Qualitative Comparison of Detection Visualization

Overall, the improved model achieves a mAP@0.5 score of 0.806 with 9.4 million parameters and 12.1 G FLOPS, striking an optimal balance between accuracy and efficiency. The key reasons for this outcome are: lightweight convolutional modules reduce computational cost while maintaining feature extraction depth through residual connections; channel compression is primarily implemented in 1×1 convolution layers, preserving the spatial feature modeling capability of 3×3 convolutions; and the decoupled architecture of shared detection heads with compression reduces redundant parameters without compromising specialization between classification and regression branches. Together, these three approaches achieve the goal of "reducing computation without sacrificing accuracy."

3.2 Ablation Experiment

The ablation experiment aimed to quantitatively assess the independent contributions of each lightweight improvement strategy module by module. Using YOLOv11s as the baseline (A), three intermediate configurations (B, C, D) were sequentially constructed by adding three improvement modules each, with each configuration trained for 300 rounds under identical training parameters and data partitioning. The results are presented in Table 3.

Table 3 Quantitative Results of Ablation Experiments

configure	Parameter Value (M)	FLOPs (G)	mAP@0.5	ΔmAP	Relative improvement from baseline
(A) Baseline (YOLOv11s)	9.4	21.5	0.785	—	—
(B) + Backbone Lightweight	7.8	16.8	0.791	+0.006	+0.76%
(C) + Neck compression simplification	8.5	13.9	0.799	+0.008	+1.78%
(D) + Optimization of the Head parameter	9.4	12.1	0.806	+0.007	+2.68%

Note: The ablation experiments progressively incorporate each improvement module: (B) replaces the Backbone lightweight convolution with that in (A); (C) compresses the Neck feature fusion channel based on (B); and (D) optimizes the detection head parameter architecture from (C). ΔmAP represents incremental improvements per step, measured as the cumulative increase in mAP relative to the baseline. The final solution (D), while maintaining the same number of parameters as the baseline, achieves a 43.7% reduction in FLOPs and a 2.1 percentage point improvement in mAP@0.5.

The ablation experiments revealed distinct contribution patterns across the three improved modules. Backbone lightweight (A→B) achieved the largest reduction in FLOPs (-4.7 G) while slightly improving mAP by 0.006, demonstrating that lightweight convolutional modules reduce computational cost without compromising feature extraction performance—residual connections and spatial-channel decoupling strategies effectively maintain information flow. Neck compression (B→C) further reduced FLOPs by 2.9 G while delivering the greatest single-step accuracy improvement (ΔmAP = +0.008), indicating significant redundancy in the channel count of original 1×1 convolutions in the Neck architecture; moderate channel compression enhanced detection accuracy through regularization effects. Head parameter optimization (C→D) restored parameter quantity to baseline levels (9.4 M, compensating for loss of representation parameters in Backbone lightweight) with FLOPs minimized to 12.1 G and ΔmAP = +0.007. The combined gains across all three modules totaled mAP + 0.021 (2.68%) and a 43.7% reduction in FLOPs, validating the effectiveness and complementarity of the improvement strategies.

Figure 8 visually illustrates the cumulative gain of mAP and parameter changes during the integration of each module using a waterfall diagram. Starting from a baseline value of 0.785, the three modules progressively increase to 0.806, maintaining positive gains at each stage; the parameter value briefly declines in Step B before recovering to match the baseline, ultimately achieving the ideal improvement outcome of "unchanged parameters, enhanced accuracy, and significantly reduced computational cost."

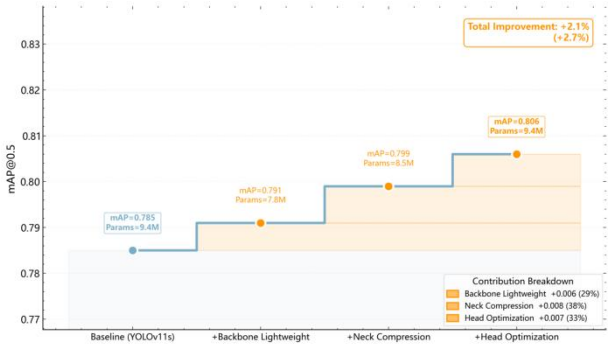

Figure 8 Ablation Experiment Waterfall Diagram

4 CONCLUSIONS AND OUTLOOKS

Aiming at the complex interference factors in actual wheat fields and the computing power limitation of edge embedded devices, this paper proposes a lightweight multi-type wheat disease detection algorithm based on improved YOLOv11. By conducting lightweight convolution replacement for the backbone network, channel compression for the feature fusion neck, and parameter optimization for the detection head, the method effectively reduces model computational redundancy from multiple dimensions. These complementary improvement modules significantly cut floating-point operations while retaining and boosting detection accuracy. Benefiting from optimized feature extraction and parameter regularization, the proposed model achieves a competitive mAP@0.5 of 0.806, outperforming the original YOLOv11 lightweight versions. Combined with ONNX graph optimization and TensorRT FP16 quantization deployment on Raspberry Pi 5, the model realizes ultra-low inference latency and high frame rate, effectively solves the problem of real-time wheat disease detection under complex field environments. This paper also builds a high-quality field wheat disease dataset, providing reliable data support and a practical lightweight edge detection scheme for intelligent farmland disease monitoring.

This study still has certain limitations. The dataset only covers four common wheat diseases and lacks other typical disease types, with limited generalization ability across disease categories. The dataset is collected from a single planting area, which limits the model's generalization ability to wheat fields in other regions. In addition, the model merely relies on single RGB visible information without multi-modal data fusion, and the fixed network structure cannot realize online adaptive learning. Future research will expand disease categories and regional datasets, introduce multi-spectral and thermal infrared feature fusion, adopt self-supervised learning strategies, explore dynamic end-side inference mechanisms, and integrate the model with intelligent inspection platforms to build a fully closed-loop field disease early-warning system.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015, 28.

[2] Nie C. Detection Method of Common Wheat Diseases Based on Improved YOLOv8. Proceedings of the 2025 6th International Conference on Computer Information and Big Data Applications. 2025: 19-24.

[3] Yao X, Yang F, Yao J. YOLO-Wheat: A wheat disease detection algorithm improved by YOLOv8s. IEEE Access, 2024, 12: 133877-133888.

[4] Zhang X, Wei L, Yang R. TriPerceptNet: a lightweight multi-scale enhanced YOLOv11 model for accurate rice disease detection in complex field environments. Frontiers in Plant Science, 2025, 16: 1614929.

[5] Kumar D, Kukreja V. Impact of image segmentation and feature sets in automated plant disease classification: a comprehensive review based on wheat plant images. Progress in Artificial Intelligence, 2025, 14(4): 451-504.

[6] Bouskour S, Zaggaf M H, Bahatti L. Deep learning recognition of wheat leaf disease using MobileNetV2 model. 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET). IEEE, 2024: 1-6.

[7] Zhang P, Li R, Liu Y, et al. I-GhostNetV3: A Lightweight Deep Learning Framework for Vision-Sensor-Based Rice Leaf Disease Detection in Smart Agriculture. Sensors, 2026, 26(3): 1025.

[8] Wang X, Li C, Zhao C, et al. GrainNet: efficient detection and counting of wheat grains based on an improved YOLOv7 modeling. Plant Methods, 2025, 21(1): 44.

[9] Liu W, Xu L, Chang X, et al. Focal-HAIN: a lightweight model with adaptive modulation and hierarchical interaction for real-time crop pest and disease monitoring. Frontiers in Plant Science, 2026, 17: 1778883.

[10] Hmache F, Bellout A, Zarboubi M, et al. Pruning, Knowledge Distillation and Quantization of YOLOv8n for Edge -Based Precision Agriculture. International Conference on Advanced Communications with Modern Circuits and Systems Technologies. Cham: Springer Nature Switzerland, 2025: 248-254.
[11] Masykur F, Adi K, Nurhayati O D. Measuring Agricultural Area Using YOLO Object Detection and ArUco Markers. Ingenierie des Systemes d'Information, 2024, 29(1).
[12] Liu H, Zhang Y, Liu S, et al. UAV wheat rust detection based on FasterNet-YOLOv8. 2023 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, 2023: 1-6.
[13] Rodrigo M, Cuevas C, García N. Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks. Scientific reports, 2024, 14(1): 21392.
[14] Karthikeyan R, Karthik S, Harshavardhan A. Real-time plant disease detection on Raspberry Pi using lightweight deep learning models. Sustainable Computing: Informatics and Systems, 2025, 37: 100892.