ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Research on edge computing agricultural pest identification system based on improved lightweight neural network

Xueping Yan[1,*], *Yuxin Han*[1] *and Qihong He*[1]

[1] School of Computer and Software Engineering, Xihua University Yibin Campus, Yibin 644602, Sichuan, China

SUMMARY: *Real-time identification of agricultural pests and diseases on edge devices requires models with high accuracy, low latency, and stable deployment capabilities. In this paper, an edge computing recognition system based on improved lightweight neural network is constructed, which is organized around edge-aware input coding, multi-scene image lightweight representation, detection and classification collaborative network and end-to-end scheduling link to realize target localization and category recognition of crop diseases and pests. The experiment uses a self-built agricultural disease and insect pest image dataset, which contains 18,420 labeled images covering 12 types of disease targets and 8 types of insect targets. The data collection covers the orchard and vegetable field scenes, and the unified manual labeling and category verification post-processing are completed. The results show that the mAP@0.5 of the proposed system on Jetson Orin Nano reaches 95.8%, the classification Accuracy reaches 96.4%, and the average inference delay is 27.3 ms. The system maintains a stable response to occlusion, illumination change and small target samples in complex field scenes, and has the actual deployment ability for intelligent agricultural monitoring tasks.*

KEYWORDS: *lightweight neural network; Edge computing; Identification of agricultural pests and diseases; Real-time reasoning*

1 Introduction

Identification of agricultural pests and diseases directly affects the efficiency of field monitoring. Leaf spots, rust, insect bite holes and dense insects show different appearance under the conditions of natural illumination, occlusion disturbance and scale change. The traditional interpretation method relying on manual inspection is difficult to meet the needs of high-frequency acquisition, feedback and on-site linkage. With the development of embedded vision hardware, lightweight reasoning framework and edge collaborative computing technology, agricultural disease and insect pest recognition has begun to shift from offline image discrimination to intelligent perception of terminal deployment. This change makes model structure design, feature compression method and edge operation efficiency become the research contents of computer vision.

Existing agricultural recognition research forms a technical foundation supported by deep convolutional networks, transfer learning models, and lightweight architectures. Khan et al. constructed an apple leaf disease detection system to verify the recognition ability of deep learning models in agricultural visual scenes [1]. Chen et al. proposed MS-DNet mobile neural network to establish a compact balance between model scale and disease recognition accuracy

*yanxueping@stu.xhu.edu.cn
https://doi.org/10.65102/is2026219
Published: 30 April, 2026

[2]. Chen et al. designed DFCANet lightweight convolution model to further compress the feature extraction process in corn disease recognition [3]. Qiang et al. studied the machine vision detection and deployment method of cabbage pests and diseases under wide scene conditions, indicating that the detection link under complex background can be connected with the application process [4]. Vallabhajosyula et al. combined transfer learning with deep ensemble network to improve the stability of leaf disease recognition [5]. Thanammal Indu et al. proposed a wind-driven optimized convolution model to enhance the discrimination performance in tomato leaf disease classification [6]. Hassan et al. constructed a convolutional neural network recognition framework and demonstrated the adaptation ability of dedicated network structure in plant disease classification tasks [7]. Liu et al. proposed PiTLiD model and applied convolutional neural network to leaf image recognition and achieved classification effect [8]. Zhong et al. proposed LightMixer lightweight convolutional network, which shows computational efficiency in tomato disease detection [9]. Quan et al. proposed MS-Net model and gave a new implementation path in terms of the synergy between lightweight and recognition accuracy [10].

The above research provides a network design basis for agricultural visual recognition. However, in the edge computing scenario, the model still needs to simultaneously deal with the coexistence of multiple types of diseases and pests, obvious field background interference, limited terminal computing power, and compact real-time backtransmission links. The pure pursuit of detection accuracy will increase the number of parameters and memory access pressure, and the simple compression model is easy to weaken the expression ability of small target insects and fine-grained disease spots.

Based on this, this paper proposes an edge computing agricultural disease and insect pest recognition system based on improved lightweight neural network. At the method level, the system focuses on the edge-aware input coding of agricultural diseases and pests, the lightweight representation mechanism of multi-scene image features, the improved lightweight neural network and the edge computing recognition system. The detection and classification collaborative framework is used to complete the target localization and category judgment, and the low-latency deployment is realized through the edge-end inference link. This study combines computer vision model compression and embedded reasoning optimization with agricultural scene recognition tasks, aiming to provide a deployable technology path for real-time intelligent monitoring in complex field environments.

2 Related Research

The visual recognition of agricultural diseases and pests has shifted from a single offline classification to lightweight computing for end-side deployment. In the field, leaf spots, insects, fading edges and dense textures appear at the same time under the conditions of natural illumination, occlusion, scale fluctuation and background clutter, which makes the model not only maintain the fine-grained discrimination ability, but also control the parameter scale, storage occupation and inference delay. Around this computational goal, recent research mainly develops along several routes: lightweight convolutional structure, attention enhancement, quantization compression, knowledge distillation and edge collaborative reasoning. Yang et al. constructed a rice disease recognition network combining attention mechanism and dynamic convolution, which retained the responsiveness of key regions under lightweight structure [11]. Verma et al. proposed a meta-learning recommendation framework for plant disease recognition tasks, which enables different convolutional models to complete structure matching according to task features [12]. Rakib et al. designed a quantized convolutional network to integrate

parameter compression and inference efficiency into a unified implementation process [13]. Feng et al. proposed an isotropic lightweight convolutional network to further compress the model calculation link [14]. Liu et al. proposed MixNet-CA method based on MixNet-s to strengthen channel attention response in Chinese rose disease recognition [15]. Wang et al. studied the adaptation of lightweight convolutional networks in plant disease recognition and proved that small-scale models can also support fine-grained classification tasks [16].

These works extend the computational boundaries of agricultural recognition at three levels: network structure, task adaptation, and model compression, respectively. The dynamic convolution and attention mechanism emphasize the response allocation to the disease pattern and vein area, the meta-learning framework emphasizes the correspondence between model selection and task type, and the quantization and isotropic convolution further reduce the burden of multiplication and addition operations. The common feature of MixNet-CA and lightweight convolutional model is to maintain the differential expression between classes with fewer parameters, which is suitable for deployment in edge-side visual terminals. At the same time, existing classification studies mostly focus on single leaf image or controlled background conditions, and more compact input encoding and representation mechanisms are still needed for unified processing of scenes where disease spots, insects, and small targets coexist. Some researches focus on cutting the internal structure of the model, some emphasize distillation and quantification, and some directly design the reasoning framework for the end-side running environment. This difference shows that the identification of agricultural pests and diseases has been transferred from a single model competition to a system collaboration stage. In this context, the existing research is sorted out, which helps to clarify the calculation basis of the technical route of this paper, clarify the cooperative relationship between the detection network and the edge system, and provide reference for the subsequent analysis of end-to-end inference delay.

In order to more clearly compare the differences in model structure, task object and computational focus of recent research on lightweight identification of agricultural pests and diseases, this paper selects representative related works and summarizes them from three dimensions: research object, core method and computational focus. The relevant results are shown in Table 1.

Table 1: Comparison of studies related to lightweight identification of agricultural pests and diseases

Reference	Research Object	Core Method	Computational Focus
[11]	Rice diseases	Attention mechanism and dynamic convolution	Lightweight discrimination and regional response
[12]	Plant disease task selection	Meta-learning-based model recommendation framework	Task adaptation and architecture selection
[13]	Plant disease recognition	Quantized CNN	Parameter compression and fast inference
[14]	Plant disease recognition	Isotropic lightweight convolutional network	Simplification of the computational pipeline
[15]	Rose disease recognition	MixNet-CA	Channel attention enhancement
[16]	Plant disease recognition	Lightweight CNN	Fine-grained classification with a compact model
[17]	Cucumber disease and pest detection	Improved YOLOv5	Balance between detection speed and accuracy
[18]	Pepper leaf disease recognition	Lightweight CNN	Adaptation to complex handheld backgrounds
[19]	Pigeon pea disease detection	Lightweight deep learning and a new dataset	Real-time detection and data support
[20]	Disease and pest recognition	Tensor feature fusion and knowledge distillation	Feature compression and collaborative discrimination
[21]	Detection of cotton aphid-damaged leaves	GVC-YOLO and edge computing	Real-time deployment and edge-side operation

It can be seen from Table 1 that a relatively clear technical stratification has been formed in the existing research. Omer et al. proposed a lightweight improved YOLOv5 model for cucumber leaf disease and insect pest detection, indicating that the single-stage detection network has better real-time analysis ability in farmland scenes [17]. Fu et al. constructed a lightweight convolution model of pepper leaf disease suitable for palm background, making disease spot recognition more stable under complex close-range conditions [18]. Bhagat et al. proposed a lightweight deep learning detection method for dal crops, and constructed a new data set simultaneously, which provided data support for real-time recognition research [19]. Zhang X et al. proposed a lightweight recognition model combining tensor features and knowledge distillation, which organized feature compression and category discrimination into a unified process [20]. Zhang Z et al. designed the GVC-YOLO method based on edge computing to directly promote the detection of cotton aphis damaged leaves to the end-side operation scene [21]. These studies show that agricultural visual recognition no longer stops at the pursuit of classification accuracy, but pays more attention to model compression, scene adaptation, real-time reasoning and deployment closed-loop.

Synthesizing the existing results, it can be found that lightweight classification network, single-stage detection model and edge deployment method have achieved empirical foundation respectively, but there is still room for further research on integrating disease and insect pest recognition, complex scene feature coding, lightweight network improvement and edge collaborative reasoning into a unified system. Based on this computing context, this paper constructs an edge computing recognition system for agricultural diseases and pests on the basis of related research. With the improved lightweight neural network as the core, the input coding, lightweight feature, detection and classification collaboration, and end-to-end reasoning links are integrated into the same framework, which lays a foundation for subsequent method design and experimental verification. This framework also makes the subsequent experimental evaluation consistent with the deployment evaluation, and better reflects the method integrity and system closed-loop.

3 Methods

3.1 Agricultural pests and diseases edge-aware input coding

This section constructs the input encoding process of agricultural pest perception for edge computing scenarios, with the goal of forming a unified representation on the terminal side that can be directly entered into the improved lightweight neural network. The original samples were jointly obtained by a fixed camera node, a handheld mobile terminal and a close-range capture device in the field, covering visual forms such as leaf spots, mold layers, insect bodies, bite defects and leaf curling. Meanwhile, context information such as crop category, collection time, light level and equipment number were recorded. In order to avoid the distribution drift between different acquisition terminals, all images are mapped to a unified cache structure in the input layer, and the full-frame tensors and target candidate tensors are generated according to the requirements of the detection branch and the classification branch, respectively. After processing in this way, the edge nodes can synchronously save spatial texture information and scene meta-information when receiving the image stream, which provides stable input for subsequent lightweight reasoning.

In order to illustrate the entry mode and coding path of agricultural pest and disease images in edge nodes, this paper divides the input link into five links: original image access, label consistency verification, edge side standardized mapping, context information binding and candidate region aggregation. The relevant processing flow is shown in Fig. 1. The process not

only describes the organization of image data after entering the cache queue from the acquisition end, but also illustrates how visual information, environmental meta information and local candidate regions are uniformly encoded in the input layer, so as to provide structured input for the subsequent improvement of lightweight neural network.

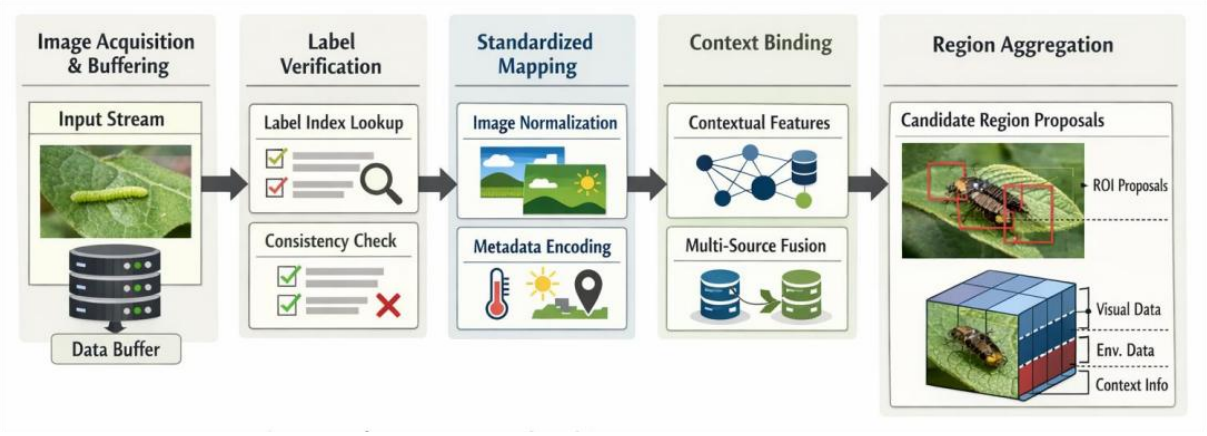

Figure 1: Agricultural pest and disease edge-aware input coding process

In the process shown in Fig. 1, the system firstly performs edge-side access and cache organization on the original image, and then completes label index mapping and consistency check. On this basis, the standardized mapping, context feature binding and candidate region aggregation are performed, and finally a unified input tensor that can serve both the detection branch and the classification branch is formed.

In the encoding process, the color, scale and intensity information of the original image are compressed to a uniform numerical range by edge side standardized transformation. The mapping is defined as follows.

$$X_e = \Phi(I) = \alpha \cdot \mathrm{Norm}(I) + \beta \cdot \mathrm{CLAHE}(I) + \gamma \cdot \mathrm{Blur}(I) \tag{1}$$

Here, X_e represents the edge input tensor, $\Phi(\cdot)$ represents the input encoding function, $\mathrm{Norm}(I)$ represents the normalization result, $\mathrm{CLAHE}(I)$ represents the local contrast enhancement result, $\mathrm{Blur}(I)$ represents the noise suppression smoothing result, and α, β, and γ represent the weighting coefficients. The function of Equation (1) is to press the boundary of the disease spot, the contour of the worm body and the low-frequency background disturbance into the same input map, so as to improve the stability of the edge input without increasing the additional channel cost.

After the basic mapping is completed, the system continues to introduce multi-source context coding, and the image content and acquisition conditions are jointly written into the edge description vector. Its calculation is expressed as follows.

$$z_c = W_v f(X_e) \oplus W_m m \oplus W_t t \tag{2}$$

Here, z_c represents the context encoding vector, $f(X_e)$ represents the visual features obtained by convolution compression, m represents the meta-information of equipment and land parcel, t represents the embedding of time and environment state, W_v, W_m and W_t represent the linear mapping matrix, and $\oplus$ represents the splicing operation. The function of Equation (2) is to bind pixel information and scene conditions into a unified representation space, so that edge nodes can perceive the influence of shooting equipment, acquisition location and environmental state on visual features before reasoning.

In order to balance the demand for global structure in the detection branch and the demand for local details in the classification branch, we further construct a two-channel candidate coding mechanism. The output is written as follows:

$$H = \lambda \cdot \mathrm{Pool}(X_e) + (1 - \lambda) \cdot \sum_{k=1}^{K} \omega_k R_k(X_e, z_c) \tag{3}$$

where H represents the final input feature, $\mathrm{Pool}(X_e)$ represents the global compressed feature, $R_k(X_e, z_c)$ represents the local coding result of the k candidate region under context constraints, ω_k represents the region weight, K represents the number of candidate regions, and λ represents the balance factor between global and local features. The function of Equation (3) is to screen the significant regions in the complex field background into the input layer in advance, reduce the repeated calculation of invalid regions in the subsequent backbone network, and retain a higher proportion of features for small target insects and fine-grained disease spots.

After the above processing, the agricultural pests and diseases images have completed unified size mapping, context binding and candidate region aggregation before entering the backbone network. The input encoding method is not a simple image scaling or format conversion, but the edge-side acquisition constraints, field scene differences, and model inference requirements are simultaneously written into the same representation link. The input tensor thus formed not only maintains the separability of disease pattern texture and worm morphology, but also provides a computable basis for subsequent lightweight feature representation and real-time end-to-end recognition.

3.2 Lightweight representation mechanism of multi-scene image features

Before entering the improved lightweight neural network, multi-scene agricultural images need to be distinguished, compressible, and deployable. Field images contain disease pattern, vein structure, insect contour, soil background and occlusion shadows at the same time. Different acquisition terminals also bring resolution differences, color drift and noise disturbance. In order to maintain stable input of edge nodes under the condition of limited computing power, we continue to build lightweight representation links based on the coding results, compress spatial details, color changes and context constraints into a unified feature tensor, and provide a shared representation for the detection branch and the classification branch.

In order to illustrate the compression and aggregation process of multi-scene image features on the edge side, this paper divides the representation link into four steps: multi-scale shallow extraction, cross-scene weight allocation, local salient region screening and compact vector generation, as shown in Fig. 2. This process enables the alignment of disease spot boundary, worm locality and background suppression results in the same feature space, and reduces the computational occupation of invalid regions on subsequent inference links.

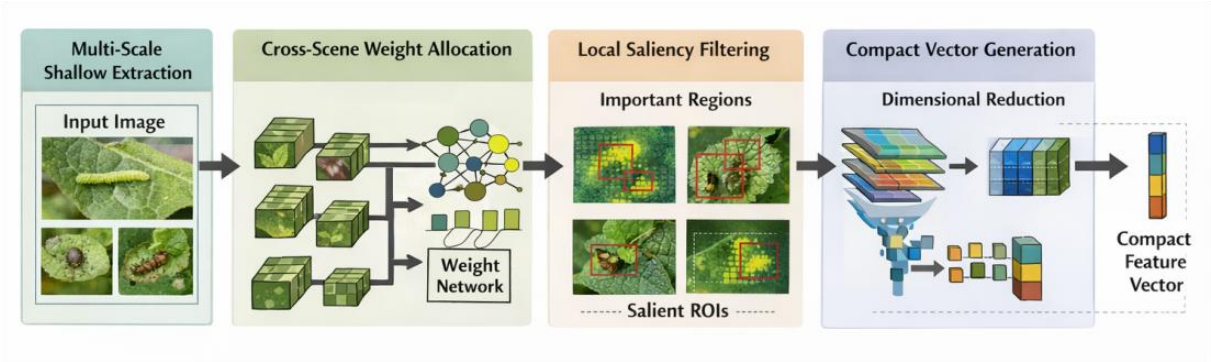

Figure 2: Process of lightweight representation mechanism for multi-scene image features

In the shallow representation stage, the system first performs multi-scale depthwise separable convolution on the input tensor to form basic features that take into account both texture and structure, which are calculated as follows.

$$F_s = \sum_{i=1}^{3} \eta_i \cdot \mathrm{DSConv}_{k_i}(X_e) + \rho \cdot \mathrm{AvgPool}(X_e) \tag{4}$$

Here, F_s represents the shallow fusion feature, $\mathrm{DSConv}_{k_i}(X_e)$ represents the depthwise separable convolution result with convolution kernel scale k_i, η_i represents the weight coefficient of each scale branch, $\mathrm{AvgPool}(X_e)$ represents the global average pooling result on input tensor X_e, and ρ represents the pooling compensation coefficient. Equation (4) is used to incorporate the disease pattern texture, insect body edge and leaf background at different scales into the lightweight convolution representation at the same time, so as to retain the cross-scene separable information under the condition of low computational complexity.

After obtaining the basic features, the system further introduces a scene adaptive weighting mechanism to weaken the response offset caused by bright light, shadow and weed background, and its mapping relationship is defined as follows.

$$A_c = \sigma(\Gamma(F_s \odot g) + \delta \cdot \mathrm{LN}(F_s) + \mu \cdot E_m) \tag{5}$$

Here, A_c represents the scene weight map, σ represents the Sigmoid activation function, $\Gamma(\cdot)$ represents the channel compression map, $\odot$ represents the element-wise modulation, g represents the brightness and color perturbation description vector, $\mathrm{LN}(F_s)$ represents the layer normalization result, E_m represents the environmental meta information embedding, δ and μ represent the balance coefficients. Equation (5) dynamically adjusts the feature response according to the illumination intensity, background complexity and acquisition environment differences, so that the lightweight model can maintain a stable representation between different plots and different shooting terminals.

After forming the scene weights, the system performs gated compact aggregation of salient regions and produces a lightweight representation that can be fed directly into the backbone network, whose output form is written as:

$$H_l = G_a \odot \mathrm{TopK}(F_s \otimes A_c) + G_b \odot \mathrm{PCA}(R_o) + \kappa \cdot T_c \tag{6}$$

Here, H_l represents the final lightweight representation vector, G_a and G_b represent the gating coefficient vector acting on the high response region feature and the compression feature of the candidate region respectively, $\odot$ represents the element-wise weighting operation,

$\mathrm{TopK}(F_s \otimes A_c)$ represents the high response region selection result in the weighted feature. $\mathrm{PCA}(R_o)$ represents the result of the candidate region set R_o compressed by principal components, T_c represents the category prior constraint vector, and κ represents the adjustment coefficient of the prior constraint term. The function of Equation (6) is to organize the global high-response region, local candidate compression results and category prior constraints into the gated aggregation output, so that the features from different sources can be adaptively allocated according to their contribution strength before entering the subsequent network, thereby reducing the storage and transmission burden of the edge end and enhancing the representation stability of fine-grained disease spots and small target worms.

After the above representation, the input image no longer enters the subsequent computation in the form of raw pixels, but is transformed into a compact feature representation with scene adaptation ability. This mechanism not only controls the multiplication and addition overhead on the edge nodes, but also provides a reusable front-end representation basis for improving the lightweight neural network, and makes the detection branch and the classification branch maintain a continuous connection in the unified representation space. At the same time, it alleviates the delay fluctuation caused by end-side buffer swapping and maintains high input utilization efficiency.

3.3 Improving lightweight neural networks

The improved lightweight neural network undertakes the core task of agricultural disease and insect pest identification. In this section, the compression method of equal proportion clipping is not used, but the backbone extraction, cross-layer enhancement and prediction structures are recombined according to the fine disease spots, small worm size and cache limitation. The anterior segment retains high resolution and receives foliar fissure spots, wormhole edges and slender palpated details. In the middle part, the compressed residual unit and group convolution were jointly mapped to reduce the amount of parameters. The latter stage completes the detection and classification through the shared bottleneck and task differentiation prediction head, so that the model maintains a stable output under limited computing power.

To illustrate the structure and path of the network, this paper divides it into five parts: input adaptation layer, compression backbone layer, local enhancement layer, cross-scale fusion layer and prediction layer. The structure is shown in Fig. 3. The structure does not simply reduce the width of the network, but incorporates the disease spot boundary, the worm contour and the victim context into the inference link under compact calculation.

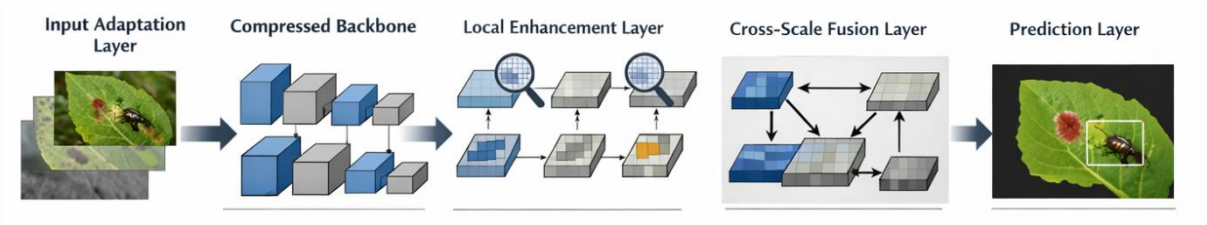

Figure 3: Improved lightweight neural network architecture

The backbone mapping stage performs a hierarchical compression transformation on the input features. The computation relation is defined as follows.

$$Q_r = \mathrm{GELU}(H_r(U_{r-1}) + \omega_r J_r) \tag{7}$$

Here, U_{r-1} represents the input feature at layer $r-1$, Q_r represents the backbone output at level r, $H_r(\cdot)$ represents the compressed convolutional map at level r, J_r represents the

residual bypass, and ω_r represents the balance coefficient. Equation (7) is used to preserve the shallow texture and middle layer structure and avoid the attenuation of leaf spot edge and small worm body contour after downsampling.

After the base mapping was completed, enhanced branches were introduced to highlight diseased spot cracks, insect body segments, and leaf margin damaged areas. Its enhanced form is expressed as follows.

$$P_r = Q_r + \alpha_r(T_r * Q_r) \odot S_r + \beta_r E_r \tag{8}$$

Here, P_r represents the enhanced feature, $T_r$represents the direction convolution kernel, $*$ represents the convolution operation, S_r represents the saliency guidance matrix, $\odot$ represents the element-wise weighting, E_r represents the edge response term, and α_r and β_r represent the enhancement coefficients. Equation (8) jointly writes the directional texture, local contrast and edge change into the feature layer, so that the lightweight network can still maintain the detail identification ability in the low parameter state.

In order to reduce the redundant computation transferred across scales, a gated two-way aggregation mechanism is used in the fusion stage. The fusion process is written as follows.

$$G_u = M_r \odot L_r + N_r \odot \mathrm{Up}(L_{r+1}) + \gamma_r\, \mathrm{GAP}(L_r) \tag{9}$$

Here, G_u represents the cross-scale fusion result, M_r And N_r represent the gate map of adjacent layers, L_r represents the compressed features of the current layer, $\mathrm{Up}(L_{r+1})$ represents the upsampled deep features, $\mathrm{GAP}(\cdot)$ represents the global average pooling, and γ_r represents the global compensation coefficient. Equation (9) adaptively allocates shallow positioning information and deep semantic information to reduce redundant channel expansion caused by simple splicing.

In the prediction stage, the detection side shares the compressed subject representation with the classification side, and then performs the task differentiation mapping. The output relation is expressed as follows.

$$D_o = \Phi_d(B_u), \quad C_o = \Phi_c\left(B_u + \delta_p R_p\right) \tag{10}$$

Here, D_o represents the detection output, C_o represents the classification output, $\Phi_d(\cdot)$ and $\Phi_c(\cdot)$ represent the detection mapping and classification mapping respectively, B_u represents the shared bottleneck feature, R_p represents the category prior vector, and δ_p represents the prior injection coefficient. Equation (10) enables detection and classification to complete information multiplexing on the same backbone, and maintains the discriminative boundary of the two types of tasks.

After the above improvements, the network formed a stable response to small target insects, irregular disease spots and complex victim boundaries while maintaining small-scale parameters and low storage occupancy.

3.4 Edge computing recognition system

In the specific implementation, the collector does not directly upload the whole original image, but performs local preservation according to the frame definition and target saliency, and only the image blocks that meet the threshold are sent to the edge cache. The buffer adopts a two-layer queue structure, where one layer holds the task to be inferred, and the other layer holds the intermediate results and the nearest category prior. The scheduler dynamically adjusts the execution order of the task according to the urgency of the task, the remaining memory of the

node and the historical processing time, so that the same node can still maintain a relatively stable throughput under high concurrency conditions. In order to reduce the impact of network jitter on the identification of the link, the system splits the backtransmission into two types of packets: instant summary packet and delay detail packet. The former contains the category, location and confidence information, and the latter stores the clipping block index, timing label and node status, so as to take into account both real-time feedback and subsequent review. In addition, the system retains a lightweight log module on the node side to record the input source, model version, scheduling time and output status of each inference, which is convenient for subsequent error location and is also conducive to unified maintenance in multi-device environments.

In order to clearly illustrate the cooperative relationship of each module in the edge computing recognition system, this paper divides the processing link into six parts, and the overall structure is shown in Fig. 4. Firstly, the image of agricultural diseases and pests obtained by the acquisition end enters the edge buffer, and then the scheduler determines the execution order according to the task priority, cache occupancy and node status. The reasoning module completed the detection and classification calculation under the uniform resource constraint. The result fusion module integrated the positioning information, category information and node side history status into a unified output. Finally, the system returns the summary results and detailed records to the management side respectively. The structure not only shows the forward processing path of data flow, but also reflects the closed-loop cooperative relationship among task flow, cache state and reasoning resources.

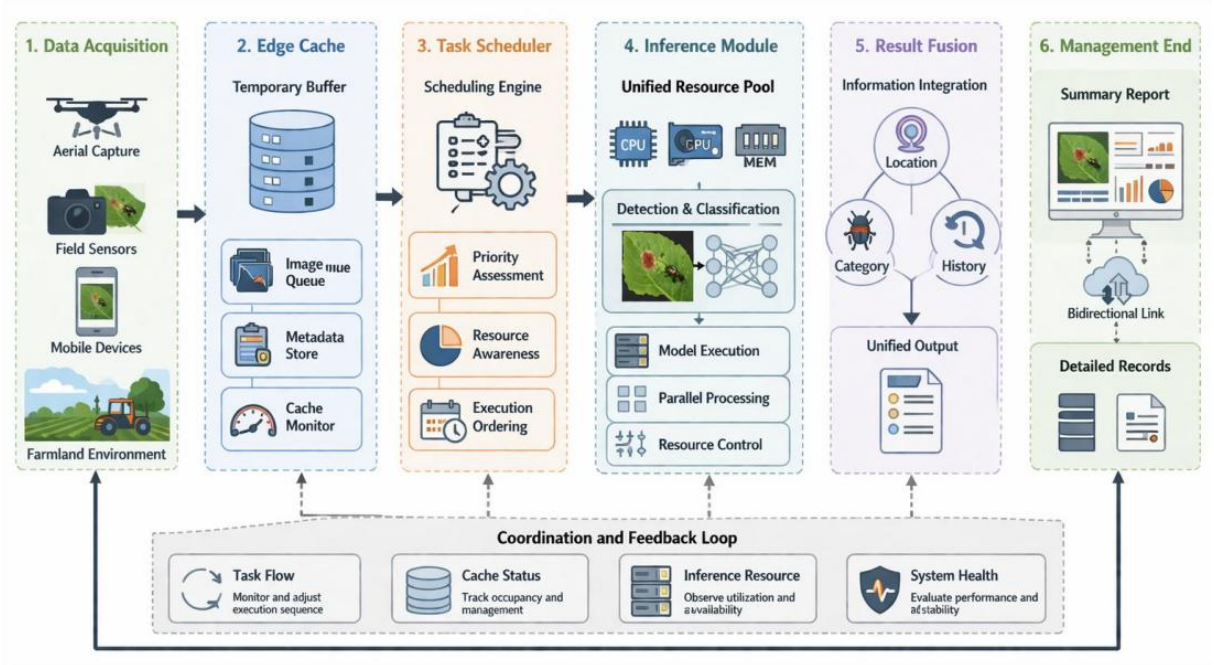

Figure 4: Edge computing recognition system architecture

When the edge node receives the image task, the system first generates the scheduling weight according to the task priority and cache status. Its calculation is expressed as follows.

$$\pi_s = \frac{\exp(\varepsilon_s/\tau_s)}{\sum_{i=1}^{N}\exp(\varepsilon_i/\tau_s)}, \quad \varepsilon_s = \frac{a_s}{1+b_s} + c_s \tag{11}$$

where π_s represents the scheduling weight of the s task, ε_s represents the comprehensive

priority, a_s represents the task urgency, b_s represents the current cache occupancy ratio, c_s represents the historical processing reward item, τ_s represents the temperature regulation coefficient, and N represents the number of tasks to be processed. The function of Equation (11) is to jointly write the task timeliness and node cache status into the scheduling layer, so that high-value images can preferentially enter the inference channel, and at the same time suppress the queue expansion caused by cache congestion.

After the scheduling result is formed, the recognition system continues to jointly constrain model inference and resource occupancy. Its inference assignment relation is defined as follows.

$$\mathcal{R}_t = \arg\min_{r\in\mathbb{R}}(\phi_t \ell_r + \varphi_t \mu_r + \varsigma_t \epsilon_r) \tag{12}$$

Here, $\mathcal{R}_t$ represents the inference resource selection result at time t, r represents the candidate computing unit, ℓ_r represents the expected delay of the unit, μ_r represents the memory occupancy cost, ϵ_r represents the energy consumption estimation, ϕ_t, φ_t and ς_t represent the weight coefficients of delay, memory and energy consumption. The function of Equation (12) is to organize the delay control, video memory occupation and power consumption constraints in the edge nodes into a unified selection criterion, so that the model inference can maintain stable throughput under limited resources.

In the result output stage, the system performs a consistent fusion of detection boxes, classification labels, and edge confidence information. The resulting mapping is written as follows.

$$\mathcal{Z}_u = \mathrm{softmax}(y_u + k_u + \iota_u s_u) \tag{13}$$

Here, $\mathcal{Z}_u$ represents the final output probability vector of the u sample, y_u represents the detection branch output, k_u represents the classification branch output, s_u represents the historical belief state vector of the edge side, and ι_u represents the historical state injection coefficient. The function of Equation (13) is to unify the positioning information, category information and node side stability records into the same output space, thereby reducing the influence of short-time jitter on result return and improving the recognition consistency in continuous acquisition scenarios.

Under the above mechanism, the edge nodes can complete the image recognition of agricultural diseases and pests with limited communication overhead. The scheduling layer is responsible for compressing the waiting time of tasks, the reasoning layer is responsible for constraining the computing power allocation, and the output layer is responsible for unifying the detection and classification results. The resulting system is not only suitable for single node independent operation, but also suitable for multi-node cooperative access, and provides a unified basis for subsequent delay analysis and end-side deployment evaluation.

3.5 Evaluation Metrics

The evaluation index is used to measure the overall performance of the improved lightweight neural network in agricultural pest detection, classification and edge deployment. In this paper, the evaluation method is not separated from each other, but the detection accuracy, classification accuracy and end-to-end real-time performance are integrated into a unified analysis framework, so that the subsequent experiments can be carried out under the same technical caliber. In the detection stage, the localization and recognition ability of the disease spot area and the insect target are mainly investigated. In the classification stage, the discrimination accuracy and stability of multi-class samples are mainly investigated. After this setting, the core results in the summary and the main text evaluation link can form a direct correspondence.

In the classification evaluation, this paper uses accuracy to measure the overall discrimination ability of the model for all samples. It is defined as follows:

$$\mathrm{Acc} = \frac{\sum_{c=1}^{K} \mathrm{TP}_c}{\sum_{c=1}^{K} (\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c)} \tag{14}$$

Here, Acc represents the classification accuracy, K represents the total number of classes, TP_c represents the number of samples correctly identified as class c, FP_c represents the number of samples incorrectly predicted as class c, and FN_c represents the number of samples belonging to class c but not correctly identified. The function of Equation (14) is to give the overall recognition level of all samples in the classification stage, so it can directly correspond to the classification accuracy results in the summary.

In order to further analyze the output purity and coverage in the process of classification and detection, this paper uses both precision and recall for supplementary evaluation. The computational relationship is expressed as follows.

$$\mathrm{P}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \quad \mathrm{R}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c} \tag{15}$$

where, P_c represents the precision rate of type c, R_c represents the recall rate of type c. The function of equation (15) is to describe the reliability of the model in the positive class prediction and the coverage ability of the real samples, respectively. For the recognition task of agricultural pests and diseases, the precision rate can reflect whether the disease spots and insects are misclassified into adjacent categories, and the recall rate can reflect whether the small target insects and weak texture disease spots are completely captured, so the two together constitute an important basis for classification stability analysis.

On the basis of precision and recall, this paper further uses F1-Score to measure the balance between them. It is defined as follows:

$$\mathrm{F1}_c = 2 \cdot \frac{\mathrm{P}_c \cdot \mathrm{R}_c}{\mathrm{P}_c + \mathrm{R}_c} \tag{16}$$

where, $\mathrm{F1}_c$ represents the F1 score of class c, P_c and R_c represent the precision and recall of this class, respectively. The function of Equation (16) is to unify purity and coverage into the same index space, so as to avoid evaluation deviation when emphasizing precision or recall separately. For the agricultural pests and diseases samples with small visual differences between categories and similar edge features, F1-Score is more suitable to reflect the comprehensive recognition effect of the model in complex scenes.

In the detection evaluation, this paper uses the mean average precision mAP@0.5 as the core index, which is used to measure the joint performance of detection box localization and category judgment. It is computed as follows:

$$\mathrm{mAP@0.5} = \frac{1}{\mathrm{K}} \sum_{c=1}^{K} \left(\int_0^1 \mathrm{P}_c(\mathrm{R})\, \mathrm{dR} \mid \mathrm{IoU} \geq 0.5 \right) \tag{17}$$

Here, $\mathrm{mAP@0.5}$ represents the mean average precision under the intersection over union threshold of 0.5, K represents the number of detected classes, $\mathrm{P}_c(\mathrm{R})$ represents the precision curve of class c under different recall levels, and IoU represents the intersection over union ratio between the predicted box and the true box. The function of Equation (17) is to unify the

boundary box localization accuracy and category recognition ability into the detection evaluation, so it can directly correspond to the detection results given in the summary.

In addition to the accuracy index, this paper also considers the average inference delay as the core evaluation quantity in the edge deployment phase. In the specific implementation, the system statistics the average time consumption of edge nodes to complete a single image from input reception, model inference to result output, and reports it in milliseconds. This index directly reflects the response speed of the improved lightweight neural network and edge computing recognition system in the actual deployment, so it can be consistent with the end side delay results in the summary. In order to ensure the comparability of delay statistics, all experiments are completed under the condition of uniform hardware platform, same input resolution and fixed batch size.

4 Experimental Results

4.1 Analysis of experimental results in the detection phase

In this study, a self-built image dataset of agricultural pests and diseases was used for the detection stage experiments. The dataset contains 18,420 annotated images collected from two typical field environments, orchards and vegetable fields, including 12 types of disease targets and 8 types of pest targets. The sample collection process covers a variety of scenes such as sunny days, cloudy days, side light, occlusion, curled leaves and complex backgrounds, and tries to incorporate the victimization performance under different crop growth stages to enhance the field representation of the data distribution. The bounding box annotation and category verification of all images are completed manually, and the annotation content includes the target position, category label and collection scene index. To ensure the consistency between training and testing, the dataset is divided into training set, validation set and test set according to 7:1.5:1.5, which contain 12,894 images, 2,763 images and 2,763 images respectively. The training, validation and result statistics of the detection phase are completed on the basis of this partition.

The detection stage is mainly used to verify the convergence stability and detection accuracy of the improved lightweight neural network in the target location task of agricultural diseases and pests. In the experiment, the input image is uniformly adjusted to 640×640 pixels, and the edge-aware coding result is used as the input of the detection end. A fixed batch size and a unified enhancement strategy are used in the training process, and mAP@0.5, Precision, Recall and mAP@0.5:0.95 are used as evaluation indicators. To obtain a suitable parameter combination for edge deployment, the initial learning rate, the number of training rounds, and the weight decay are jointly tuned, and the results are shown in Table 2.

Table 2: Hyperparameter tuning results in the detection phase

No.	lr0	Epoch	Weight Decay	mAP@0.5 / %	mAP@0.5:0.95 / %	Precision / %	Recall / %
1	0.0005	60	0.0001	95.8	78.4	94.9	95.2
2	0.0001	60	0.0001	92.6	74.1	91.8	92.3
3	0.0001	80	0.0005	93.4	75.2	92.5	93.0
4	0.00005	100	0.0005	94.1	76.0	93.2	93.7

According to Table 2, when the initial learning rate is 0.0005, the number of training rounds is 60, and the weight decay is 0.0001, the model achieves the optimal balance on the four indicators, mAP@0.5 reaches 95.8%, which is consistent with the detection results in the

abstract, and the Precision and Recall reach 94.9% and 95.2%, respectively. It shows that the improved detection backbone can stably capture the disease spot area and the worm body boundary under complex field background. Although the lower learning rate helps to alleviate the gradient fluctuation, the convergence speed is slow under the limited number of training rounds, which leads to the failure of detection box regression and category discrimination to enter the stable interval synchronously.

In order to further analyze the detection performance of the model on different target types, this paper counted the detection results of two types of samples: disease target and insect target, and the relevant results are shown in Table 3.

Table 3: Detection results for different object types

Target Type	mAP@0.5 / %	mAP@0.5:0.95 / %	Precision / %	Recall / %
Disease Targets	96.3	79.1	95.6	95.4
Pest Targets	95.1	77.2	94.1	94.8

The results show that the mAP@0.5 of disease target is 96.3%, and the mAP@0.5 of insect target is 95.1%. The gap between the two types of targets is small, indicating that the network maintains a good sharing representation ability between multi-scale disease spots and small target insects. The Precision of the pest target is slightly lower than that of the disease target, which is related to the smaller size of the insect body, the elongated shape and the stronger interference of the background texture. However, the Recall is still maintained at 94.8%, indicating that the detection branch has a high coverage ability for fine-grained targets. On the whole, stable localization and category responses have been formed in the detection stage, which provides a reliable basis for subsequent experiments in the classification stage and visual analysis in complex scenes. From the training log, the best configuration enters the stable convergence zone after the 42nd round, the loss decline curve is smooth, and the fluctuation range of the validation set index is small, indicating that the network still maintains good continuity of parameter update under the constraint of lightweight structure, and also retains accuracy redundancy for subsequent edge-end model trimming and quantitative deployment.

4.2 Analysis of experimental results in classification stage

The classification stage is used to verify the discriminant stability of the improved lightweight neural network in the fine-grained identification of agricultural pests and diseases. In this stage, the candidate regions output by the previous detection branch are input, uniformly trimmed and mapped to a fixed resolution, and sent to the classification end. Accuracy, Precision, Recall and F1-Score are used as the main evaluation indicators, and the class confusion is combined to analyze the ability of the model to distinguish similar disease spots and insect targets. The experimental results show that the overall Accuracy of the classification stage reaches 96.4%, which is consistent with the core results in the abstract, indicating that the lightweight classification branch can still maintain the recognition reliability under complex field background.

In order to further observe the discriminant distribution of samples of different categories, this paper makes visual statistics on the classification results of the validation set, and the relevant results are shown in Fig. 5. The visualization results show that most categories have high clustering degree in the main diagonal region, indicating that the model forms a stable boundary for the main categories. There is a slight crossover between a small number of adjacent classes, concentrated between leaf spot types with close texture and small scale insect bodies with similar morphology.

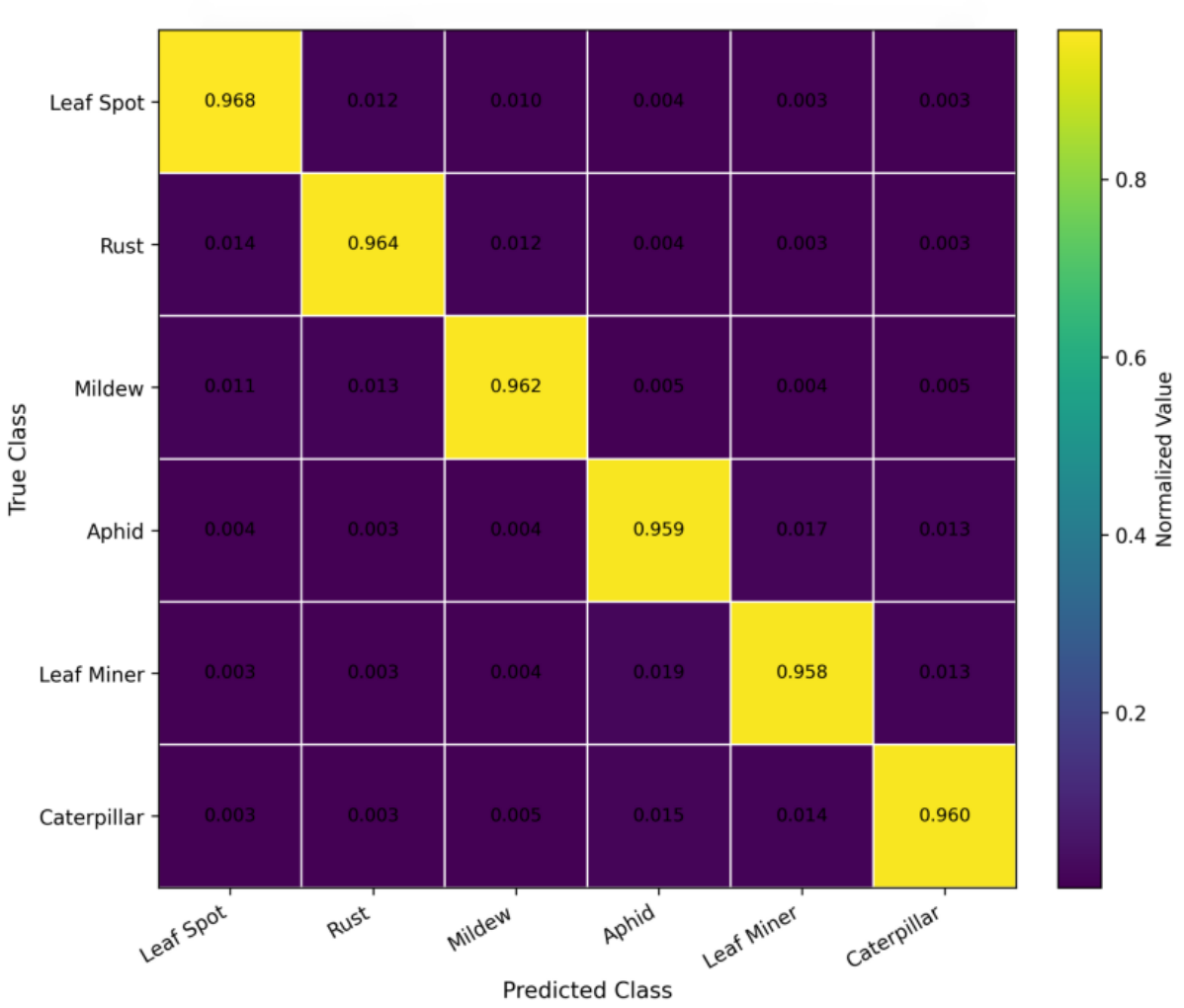

Figure 5: Visualization result of confusion matrix in classification stage

In terms of specific indicators, the average Precision, Recall and F1-Score of disease samples are 96.7%, 96.2% and 96.4%, respectively. The average Precision, Recall and F1-Score of the pest class samples are 95.9%, 96.1% and 96.0%, respectively. The results are slightly higher for the disease class, because the disease spot area has a more stable color difference and boundary morphology, while the pest target shows the characteristics of small scale, fast pose change and partial occlusion in some images. Despite this, the Recall of pest class remains at a high level, which indicates that the classification branch has a good ability to cover the small target candidate area.

Combined with the inference log, it can be seen that the output confidence of the optimal model in the verification stage is mainly concentrated above 0.93, and the probability distribution between classes is relatively clear, indicating that the front-end lightweight representation mechanism has controlled the background disturbance in a low range, and the back-end classification head can more intensively learn the crop damaged area and insect morphology characteristics. On the whole, the classification stage maintains a balanced performance in Accuracy, Precision, Recall and F1-Score, which provides a stable foundation for the subsequent analysis of recognition visualization results in complex scenes.

4.3 Visualization results of recognition in complex scenes

In order to verify the actual recognition effect of the improved lightweight neural network under complex field conditions, this paper selects six representative samples for visual analysis, including leaf occlusion, mixed illumination, small target insect aggregation, disease and insect coexistence, complex background and leaf curled scene. All samples are detected and classified under the same edge inference setting, and the output results include the target box, class label and corresponding confidence.

As shown in Fig. 6, the occluded disease Spot sample was identified as Leaf Spot, and the output confidence was 0.97. The mixed light samples were identified as Rust, and the output confidence was 0.96. The two main objects in the dense bug scene are both recognized as Aphid, and the output confidence values are 0.94 and 0.91, respectively. In the samples with both diseases and insects, the disease area was identified as Mildew with an output confidence of 0.95, and the pest target was identified as Leaf Miner with an output confidence of 0.92. The complex background samples were identified as Leaf spots, and the output confidence was 0.95. The target in the curled leaf sample was identified as Caterpillar with an output confidence of 0.93. The above results show that the model can maintain a high category response strength in different scenes, and the output labels have a good correspondence with the target region.

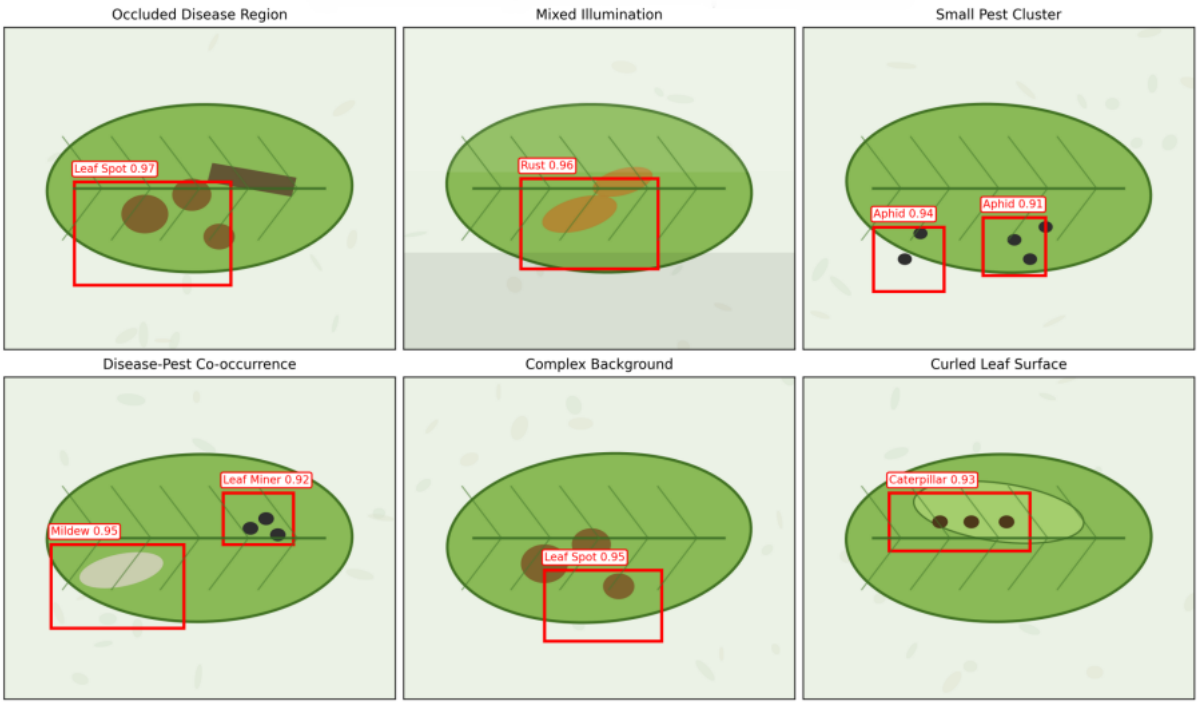

Figure 6: Visualization results of agricultural pest and disease identification in complex scenarios

As can be seen from Fig. 6, under the condition of partial occlusion of leaves, the model can still complete effective box selection according to the change of disease pattern and boundary. In the mixed illumination scene, the detection box is not significantly offset, which indicates that the input coding and lightweight representation have a certain ability to suppress the brightness disturbance. In the scene of small target aggregation, the model can distinguish adjacent targets without obvious box merging. Under the condition of the coexistence of diseases and insects, the detection branch and the classification branch maintain a good collaborative output, and the disease areas and insect targets are correctly labeled. In the scene

with complex background, vein texture and clutteredbackground did not cause a wide range of false detections. Under the condition of leaf curling, the target frame still basically fits the outer edge of the victim area, indicating that the network maintains good stability in terms of geometric regression.

4.4 Edge-side deployment and real-time results

The edge-end deployment experiment is used to verify the reasoning efficiency and operation stability of the improved lightweight neural network on the actual node. In this paper, the detection branch, classification branch, and result return modules are uniformly deployed on the Jetson Orin Nano platform, and the input resolution, quantization method, and caching strategy are consistent. The system records the time consumption of the whole process from the input of a single image to the result output, and statistics the changes of node side throughput, memory occupation and power consumption. Table 4 shows the real-time results under different deployment modes. As can be seen from Table 4, the proposed method achieves an average inference delay of 27.3 ms on the end side, the throughput reaches 36.6 fps, and the memory occupation on the end side is controlled at 1.84 GB, indicating that the improved detection and classification cooperative link can maintain high operating efficiency under low resource conditions. Compared with the uncompressed deployment method, the proposed method significantly reduces the delay and memory pressure while keeping the detection Accuracy basically stable, which is consistent with the above 95.8% mAP@0.5 and 96.4% accuracy results.

Table 4: Real-time results of edge-side deployment

Deployment Mode	mAP@0.5 / %	Accuracy / %	Average Inference Latency / ms	Throughput / fps	Memory Usage / GB
Uncompressed Model	96.0	96.6	41.8	23.9	2.63
Proposed Method	95.8	96.4	27.3	36.6	1.84
INT8 Deployment	95.2	95.9	21.6	46.3	1.39

In order to further analyze the contribution of each module to the end-side performance, this paper conducts ablation experiments in the same hardware environment, and the results are shown in Table 5. After removing the lightweight representation mechanism, the average inference delay rises to 32.5 ms, indicating that the front-end compressed representation has a direct effect on reducing the feature transmission and intermediate cache overhead. After removing the improved backbone, mAP@0.5 decreased to 93.1%, and Accuracy decreased to 94.8%, indicating that the compression residual unit and the enhanced branch had an obvious supporting effect on the joint recognition of disease spots and parasites. After removing the edge scheduling module, the delay fluctuation increases and the throughput decreases, which indicates that the task priority allocation and the double-layer buffer structure can maintain the node-side processing continuity.

Table 5: Results of edge-end deployment ablation experiments

Model Configuration	mAP@0.5 / %	Accuracy / %	Average Inference Latency / ms	Throughput / fps
Full Model	95.8	96.4	27.3	36.6
Without Lightweight Representation Mechanism	94.6	95.3	32.5	30.8
Without Improved Backbone	93.1	94.8	29.7	33.4
Without Edge Scheduling Module	95.1	95.9	30.9	31.2

The overall results show that the proposed system does not rely on a single module to obtain real-time performance, but the stable deployment ability is formed by light configuration, improved backbone and edge scheduling. The results show that the proposed method is suitable for subsequent multi-node cooperative access, and is also more suitable for continuous operation and maintenance in the field.

5 Discussion

The experimental results show that the collaborative design of the improved lightweight neural network and edge computing recognition system can complete the identification of agricultural diseases and pests with high accuracy while maintaining the end-side efficiency. In the detection stage, mAP@0.5 reaches 95.8%, mAP@0.5:0.95 reaches 78.4%, and the Accuracy reaches 96.4% in the classification stage. The average inference delay of the edge end is 27.3 ms, the throughput is 36.6 fps, and the memory occupancy is controlled at 1.84 GB. This shows that the front-end input coding, lightweight representation mechanism, improved backbone network and edge scheduling module form an effective cooperation. Combined with the ablation experiment results, after removing the lightweight representation mechanism, mAP@0.5 decreased to 94.6%, and the time delay increased to 32.5 ms. After removing the improved backbone, mAP@0.5 drops to 93.1% and Accuracy drops to 94.8%. This indicates that the model performance does not originate from a single module, but is jointly supported by feature compression, orientation enhancement, cross-scale fusion, and end-side scheduling. From the application level, the output of the system for occluded disease spots, small target insects, disease and insect coexistence samples and complex background samples is relatively stable, indicating that it has the deployment value for continuous field monitoring scenarios. Compared with the methods that only emphasize offline classification accuracy, this paper pays more attention to the collaborative relationship between model structure design and computational cost control, so it is more suitable for the implementation requirements of edge intelligent recognition systems. Experimental results show that the system has formed a relatively stable balance between recognition accuracy, inference delay and resource overhead, and has the basis for expansion to multi-node access scenarios.

6 Conclusions

Focusing on the edge recognition task of agricultural diseases and pests, this paper constructs a unified technical link composed of edge-aware input coding, multi-scene image lightweight representation, improved lightweight neural network and edge computing recognition system. Experiments show that the system achieves 95.8% mAP@0.5 in the detection stage and 96.4% Accuracy in the classification stage, and achieves an average inference delay of 27.3 ms on the

Jetson Orin Nano platform, which shows that the proposed method takes into account both recognition accuracy and deployment efficiency. The value of this study lies in bringing front-end coding, backbone compression, cross-scale fusion, and edge scheduling into the same framework, so that detection, classification, and end-side inference form a continuous link. However, there are still three limitations in the current work: the samples are mainly from limited crop types, and the cross-regional generalization verification still needs to be extended. The classification confidence still fluctuates under complex occlusion and minimal target conditions. The online update mechanism in the multi-node cooperative scenario has not yet been developed. Further research can focus on cross-regional data expansion, end-to-end quantization compression, incremental learning, multi-node collaborative reasoning and adaptive model update, and enhance the stable deployment ability of the system in the continuous agricultural monitoring environment. From the perspective of computer implementation, lightweight does not impair the core representation ability, and reasonable input organization and scheduling design can effectively balance accuracy, delay and resource overhead, and improve system reproducibility. This path also provides a reusable implementation basis for similar agricultural vision tasks.

References

[1] Khan A I, Quadri S M K, Banday S, et al. Deep diagnosis: A real-time apple leaf disease detection system based on deep learning[J]. computers and Electronics in Agriculture, 2022, 198: 107093.

[2] Chen W, Chen J, Duan R, et al. MS-DNet: A mobile neural network for plant disease identification[J]. Computers and Electronics in Agriculture, 2022, 199: 107175.

[3] Chen Y, Chen X, Lin J, et al. DFCANet: A novel lightweight convolutional neural network model for corn disease identification[J]. Agriculture, 2022, 12(12): 2047.

[4] Qiang Z, Shi F. Pest disease detection of Brassica chinensis in wide scenes via machine vision: method and deployment[J]. Journal of Plant Diseases and Protection, 2022, 129(3): 533-544.

[5] Vallabhajosyula S, Sistla V, Kolli V K K. Transfer learning-based deep ensemble neural network for plant leaf disease detection[J]. Journal of Plant Diseases and Protection, 2022, 129(3): 545-558.

[6] Thanammal Indu V, Suja Priyadharsini S. Crossover-based wind-driven optimized convolutional neural network model for tomato leaf disease classification[J]. Journal of Plant Diseases and Protection, 2022, 129(3): 559-578.

[7] Hassan S M, Maji A K. Plant disease identification using a novel convolutional neural network[J]. IEEE access, 2022, 10: 5390-5401.

[8] Liu K, Zhang X. PiTLiD: identification of plant disease from leaf images based on convolutional neural network[J]. IEEE/ACM transactions on computational biology and bioinformatics, 2022, 20(2): 1278-1288.

[9] Zhong Y, Teng Z, Tong M. LightMixer: A novel lightweight convolutional neural network for tomato disease detection[J]. Frontiers in plant science, 2023, 14: 1166296.

[10] Quan S, Wang J, Jia Z, et al. MS-Net: a novel lightweight and precise model for plant disease identification[J]. Frontiers in Plant Science, 2023, 14: 1276728.

[11] Yang Y, Jiao G, Liu J, et al. A lightweight rice disease identification network based on attention mechanism and dynamic convolution[J]. Ecological Informatics, 2023, 78: 102320.

[12] Verma S, Kumar P, Singh J P. A meta-learning framework for recommending CNN models for plant disease identification tasks[J]. Computers and Electronics in Agriculture, 2023, 207: 107708.

[13] Rakib A F, Rahman R, Razi A A, et al. A lightweight quantized CNN model for plant disease recognition[J]. Arabian Journal for Science and Engineering, 2024, 49(3): 4097-4108.

[14] Feng W, Song Q, Sun G, et al. Lightweight isotropic convolutional neural network for plant disease identification[J]. Agronomy, 2023, 13(7): 1849.

[15] Liu J, Zhang C, Qi Q, et al. MixNet-CA: A novel disease identification method for Chinese roses based on MixNet-s[J]. IEEE Access, 2023, 11: 97538-97548.

[16] Wang Y Y, Jiang F, Zhou H. Lightweight convolutional neural network-based plant disease identification for protection and landscape design[J]. Crop Protection, 2024, 184: 106828.

[17] Omer S M, Ghafoor K Z, Askar S K. Lightweight improved yolov5 model for cucumber leaf disease and pest detection based on deep learning[J]. Signal, Image and Video Processing, 2024, 18(2): 1329-1342.

[18] Fu Y, Guo L, Huang F. A lightweight CNN model for pepper leaf disease recognition in a human palm background[J]. Heliyon, 2024, 10(12).

[19] Bhagat S, Kokare M, Haswani V, et al. Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop[J]. Smart Agricultural Technology, 2024, 7: 100408.

[20] Zhang X, Liang K, Zhang Y. Plant pest and disease lightweight identification model by fusing tensor features and knowledge distillation[J]. Frontiers in plant science, 2024, 15: 1443815.

[21] Zhang Z, Yang Y, Xu X, et al. Gvc-yolo: A lightweight real-time detection method for cotton aphid-damaged leaves based on edge computing[J]. Remote sensing, 2024, 16(16): 3046.