IFDA-EMA-YOLOv9: Wheat Disease Detection Integrating Flow Optimization and Auxiliary Supervision

Jiaxiang Fan, Leixiao Li*
College of Intelligent Science and Technology, Inner Mongolia University of Technology, Hohhot, China

***Abstract*—To address the challenges of feature extraction in complex field environments, the limited sensitivity of YOLOv9 to subtle disease features, and the lack of adaptive hyperparameter optimization, this paper proposes an improved high-precision detection model, named IFDA-EMA-YOLOv9. First, to enhance feature extraction capabilities, an Efficient Multi-scale Attention (EMA) mechanism and residual connections are incorporated into the network architecture. This integration effectively suppresses background noise interference and significantly improves the model's ability to aggregate and represent multi-level features of wheat disease lesions. Second, to tackle localization deviations caused by the irregular geometric shapes of disease lesions, an auxiliary box mechanism is integrated into the Complete Intersection over Union (CIoU) loss function, optimizing the regression process to improve the fit of detection boxes. Furthermore, an Improved Flow Direction Algorithm (IFDA) is employed to perform global optimization of the critical model hyperparameters, thereby avoiding the blindness of manual tuning and the local optimum trap. Experimental results on the LWDCD2020 dataset demonstrate that the proposed IFDA-EMA-YOLOv9 significantly outperforms current state-of-the-art (SOTA) methods, achieving substantial improvements in Precision, Recall, and mAP@0.5 by 6.4%, 5.94%, and 6.66%, respectively. These results demonstrate the effectiveness and robustness of the proposed method for wheat disease and pest detection.**

Keywords*—Wheat disease detection; Object detection; YOLOv9; Flow algorithm

I. Introduction

Wheat is one of the world's major staple crops and plays a crucial role in ensuring food security and sustainable agricultural production. However, throughout the full growth cycle of wheat, the frequent occurrence of major pests and diseases such as yellow rust, fusarium head blight, and aphids has become a key factor restricting yield and quality. Statistics indicate that in years with prevalent diseases, the aforementioned disasters can cause wheat yield reductions of 20% to 50%, and in extreme cases, even trigger total seasonal harvest failure, bringing inestimable losses to agricultural production. Therefore, constructing a precise, real-time automated monitoring system possesses significant practical significance for assisting farmers in making scientific decisions, implementing precise pesticide application, and improving pesticide utilization rates.

For a long time, traditional disease monitoring has mainly relied on manual field inspections or simple physical instrument detection. These methods are not only time-consuming and labor-intensive with high subjectivity, but also difficult to adapt to the monitoring demands of large-scale intensive planting, suffering from serious lag. With the rapid development of artificial intelligence technology, utilizing computer vision and image processing technology to realize rapid, non-destructive identification of crop diseases has become an innovative direction in the field of smart agriculture. Compared with traditional means, deep learning-based methods can automatically and efficiently extract disease features from complex backgrounds, significantly enhancing the intelligence level of prevention and control work.

Among numerous deep learning models, the YOLO (You Only Look Once) series algorithms stand out due to their efficient end-to-end detection mechanisms. In particular, the latest YOLOv9 model, by introducing Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), has achieved excellent performance in core computer vision tasks such as general object detection, image classification, and semantic segmentation, demonstrating superior feature extraction capabilities and real-time inference performance [1-3].

Although YOLOv9 performs strongly on general datasets, it still faces numerous challenges that urgently need to be resolved when directly deployed to complex, unstructured wheat field environments: First, the field environment has variable illumination and complex backgrounds, so the applicability and robustness of YOLOv9 in specific agricultural scenarios require in-depth investigation and validation; second, the model's original pooling dimensionality reduction strategy, when modeling cross-channel long-range dependency relationships, easily causes the loss of subtle disease features, thereby damaging the integrity of visual representation; third, the existing CIoU loss function finds it difficult to achieve precise bounding box regression when processing targets like wheat disease lesions, which have extremely variable aspect ratios and irregular geometric shapes; finally, the configuration of model hyperparameters often relies on manual experience and lacks adaptive optimization schemes for specific datasets, limiting the upper bound of model performance.

II. Related Work

A. Progress in Wheat Pest and Disease Detection Technology

In recent years, although the YOLO series has made significant progress in the agricultural field, the research focus has shifted from simple architecture stacking to refined feature

*Corresponding author.

modeling. Qiao et al. [4] enhanced the global feature extraction capability of YOLOv8n by incorporating the C2f-iRMBS module and Inner-EIoU loss function, reaching a mAP of 96.3%. However, addressing the problem of feature redundancy in field environments, the latest research trends have begun to introduce Partial Convolution (PConv) to extract more representative spatial features by reducing redundant calculations. Furthermore, considering the irregularity of wheat lesion morphology, the Large Selective Kernel mechanism has demonstrated advantages in capturing multi-scale lesion features by dynamically adjusting the receptive field. In this context, this paper proposes the EMA-YOLOv9 model, which, by integrating the Efficient Multi-Scale Attention (EMA) mechanism with a residual architecture, aims to solve the negative impact of cross-channel relationship modeling on visual representation under complex backgrounds, further reinforcing the flexibility of feature extraction.

Addressing the difficulty of locating tiny lesions, the traditional IoU loss function has limitations when processing wheat disease lesions of variable sizes. Latest technologies such as Inner-IoU [5] have accelerated convergence by introducing auxiliary bounding box scaling factors, while MPDIoU [6] has effectively solved the problem of inaccurate bounding box overlap measurement using minimum point distance. Drawing on this frontier idea, we introduce an auxiliary frame mechanism into the YOLOv9 framework to optimize the CIoU loss function, effectively balancing localization accuracy among targets of different scales.

B. YOLOv9 Parameter Optimization

YOLOv9 utilizes a reversible information callback mechanism to significantly alleviate feature loss in deep networks. However, its performance is highly dependent on complex hyperparameter configurations. Existing research on YOLOv9 hyperparameter automation remains relatively scarce, with most literature still remaining at optimization schemes for earlier versions.

Reviewing past research, YOLOv5 employed the Genetic Algorithm (GA) [7] to optimize hyperparameters, and Mahmudah et al. [8] utilized the Tree-structured Parzen Estimator (TPE) in YOLOv7 to explore the search space. However, facing the more non-linear parameter space of YOLOv9, traditional Genetic Algorithms, Artificial Bee Colony (ABC) algorithms [9], or Opposition-based Optimization Strategies (OOS) [10] are prone to falling into local optima and are difficult to cover the full hyperparameter set. To excavate the model's potential, this paper innovatively adopts the Improved Flow Direction Algorithm (IFDA) for global optimization and proposes utilizing Chaotic Mapping and Lévy Flight mechanisms to enhance population diversity; the final IFDA-EMA-YOLOv9 model realizes the leap of hyperparameter configuration from empirical tuning to automated evolution.

III. Basic Concepts

A. YOLOv9

YOLOv9 employs a regression strategy to achieve end-to-end simultaneous prediction for target region identification, spatial localization, and category classification, striking a significant balance between detection accuracy and inference efficiency. The overall architecture of the algorithm comprises four core components: the Backbone, the Auxiliary Reversible Branch, the Multi-level Auxiliary Information, and the Head. During the model training phase, YOLOv9 generates a total of six sets of output feature maps for subsequent prediction label inference. Distinct from previous architectures, YOLOv9 innovatively introduces an auxiliary output module. By calculating the correlation between the prediction loss and ground truth labels, this mechanism facilitates the efficient back-propagation of gradient information through the auxiliary reversible path to shallow feature extraction modules. This process optimizes the gradient backflow from deep networks to shallow networks, which is crucial for the effective updating of backbone weights and the maintenance of feature representation capabilities. The core task of the model lies in accurately predicting the spatial locations of targets of different sizes based on multi-scale feature maps. By integrating the Silence module to optimize the inference process, combining standard CBS (Conv-BN-SiLU) modules, and refining the loss function and optimizer strategy, the model achieves significantly enhanced robustness in object detection and localization. Comparative experiments demonstrate that, relative to previous versions such as YOLOv7 and YOLOv8, YOLOv9 achieves improved detection accuracy while significantly reducing the number of parameters and computational complexity (FLOPs). In summary, YOLOv9 represents significant technical progress in the trade-off between processing speed and detection performance.

B. Flow Direction Algorithm (FDA)

1) Algorithm background and d8 mechanism: The Flow Direction Algorithm is inspired by natural surface runoff processes. In hydrology, direct runoff is driven by gravity and converges toward the watershed outlet along the direction of the terrain slope. The classical method for simulating this process involves discretizing the watershed into regular grid cells and determining the runoff path by calculating the elevation difference and slope between rasters. Among these methods, the D8 algorithm (Deterministic-8 Node) proposed by O'Callaghan et al [11]. is currently the mainstream method for determining flow direction. In the D8 model, it is assumed that each central raster has eight adjacent rasters. By calculating the slope between the central raster and its neighborhood rasters, the flow direction is defined as pointing to the adjacent raster with the maximum slope descent (as shown in Fig. 1).

2) Assumptions of FDA: Drawing on the physical mechanism of the D8 algorithm, FDA abstracts the solving process of optimization problems into the process of rainfall converting into runoff and converging toward the outlet. In this mapping system, water flow represents population individuals, position represents the solution vector, and elevation corresponds to the objective function value. Fig. 2 illustrates the flowchart of FDA. Based on literature [12], the operation of FDA is based on the following core assumptions:

- Individual definition: Each flow possesses determined position coordinates and a corresponding elevation (fitness value) in the solution space.

- Neighborhood structure: Each flow possesses β adjacent positions in the space, and each neighborhood position corresponds to an elevation or objective function value.
- Dynamic relationship: The flow velocity of the fluid is proportional to the path slope.
- Velocity definition: The velocity of the fluid moving toward the lowest elevation (the direction of the optimal value) is defined as V.
- Convergence goal: The position of the watershed outlet corresponds to the global optimal solution of the objective function in the search space.

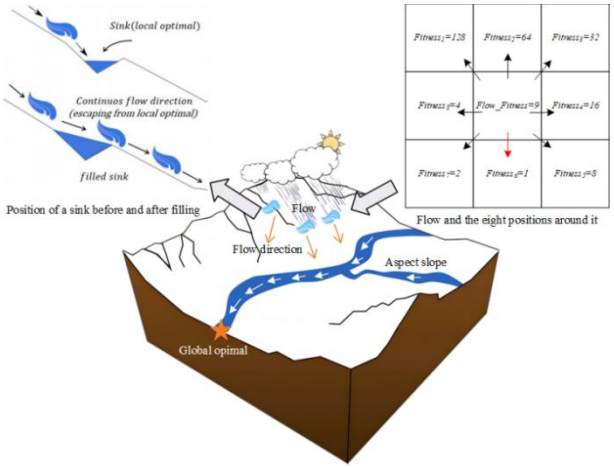

Fig. 1. D8 Method process flow diagram.

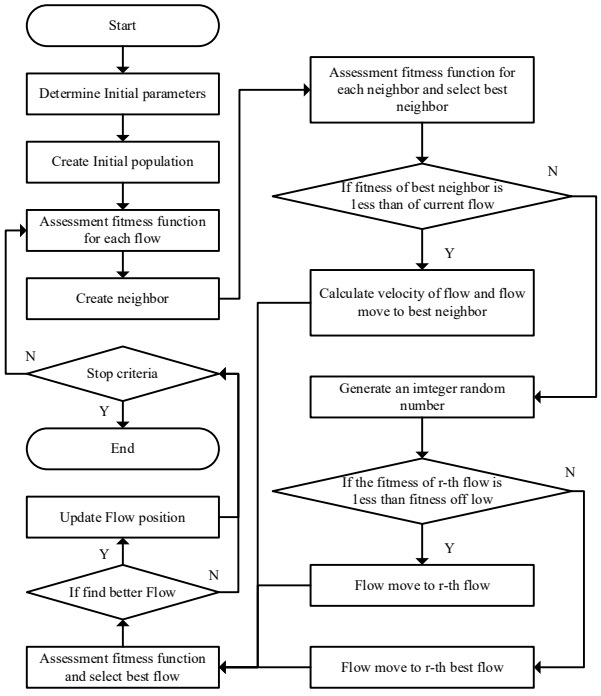

Fig. 2. Flowchart of the original FDA.

3) Initialization of flow positions: The initialization phase of the FDA algorithm involves the setting of key parameters such as population size ($Size_{pop}$), neighborhood quantity ($Size_{near}$), neighborhood radius (R_{near}), and maximum iterations (MaxIter). The initial positions of various fluids in the population are randomly generated within the solution space via the following equation:

$$Flow\_X(i) = lb + r_1 \times (ub - lb) \tag{1}$$

Where $Flow\_X(i)$ denotes the position of the i-th flow, lb and ub represent the lower and upper bounds of the decision variables, respectively; and r_1 is a random number uniformly distributed in [0,1].

4) Neighborhood generation strategy: FDA assumes that a neighborhood with a radius of R_{near} exists around each flow. Its position is generated by the following formula:

$$Neighbor\_X(i) = Flow\_X(i) + r_2 \times R_{near} \tag{2}$$

Here, denotes the position of the j-th neighbor. represents a random number drawn from a normal distribution with a mean of 0 and a standard deviation of 1. A smaller value of restricts the search to a narrow range, whereas a larger value allows for exploration across a broader scope. Searching within a wide range yields more diverse solutions and enhances the probability of identifying near-optimal solutions through global search.

Therefore, FDA adopts a formula to make the value of R_{near} linearly decrease from large to small, so as to achieve a balance between global search and local search. The mathematical formula is as follows:

$$\begin{aligned} R_{near} &= (r_3 \times Rand_X - r_3 \times Flow\_X(i)) \\ &\times \left\| Best_X - Flow\_X(i) \right\| \times W \end{aligned} \tag{3}$$

Where, r_3 is a uniformly distributed random number; $Rand_X$ is a random position generated by (1); W is a non-linear weight that varies randomly between 0 and infinity. The first term of (3) indicates that $Flow\_X(i)$ moves towards the random position ($Rand_X$). As the number of iterations increases, $Flow\_X(i)$ converges towards $Best_X$, causing the Euclidean distance between $Best_X$ and $Flow\_X(i)$ to decrease to zero. Consequently, the local search is optimized. The calculation formula for W is as follows:

$$W = ((1 - \frac{iter}{MaxIter})^{2 \times r_4}) \times (\bar{r_1} \times \frac{iter}{MaxIter}) \times \bar{r_1} \tag{4}$$

Where *iter* represents the current iteration number; r_4 is a uniformly distributed random number; and r is a uniformly distributed random vector.

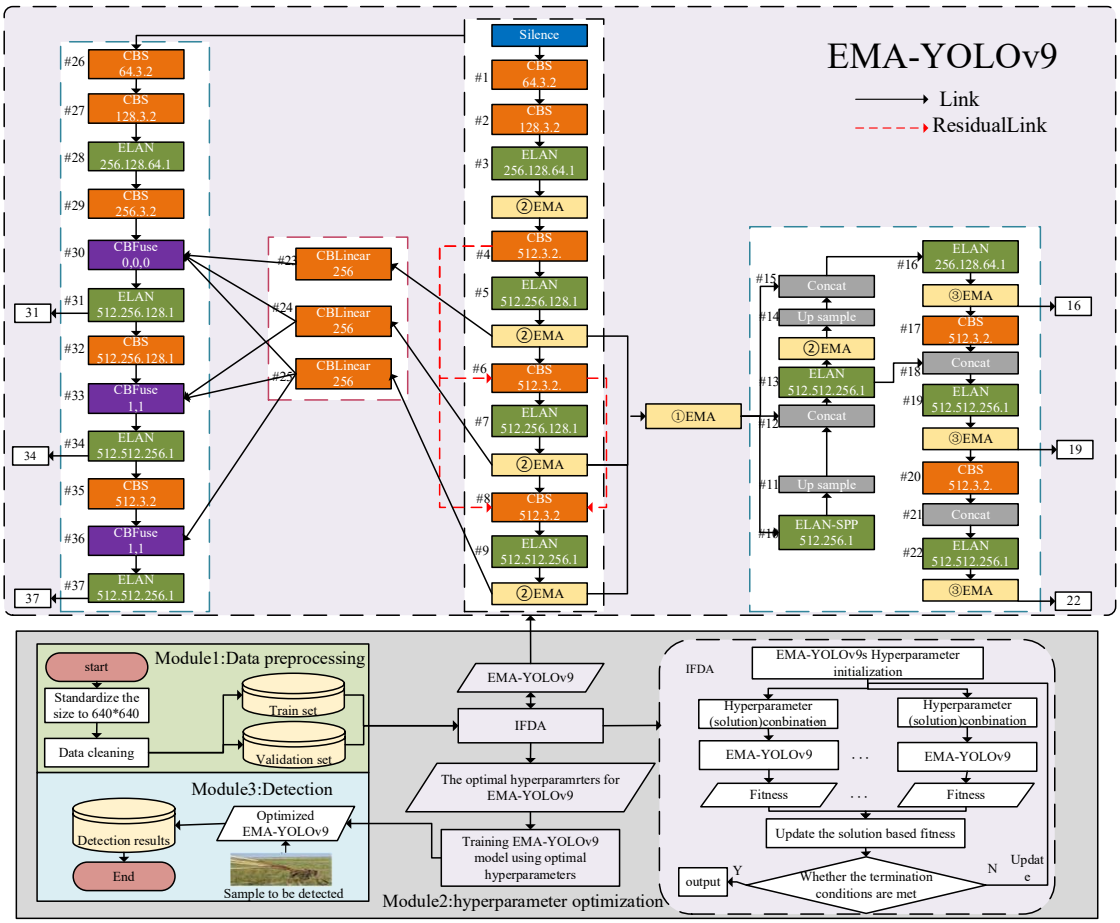

Fig. 3. Overall framework of methods for detecting wheat diseases.

5) Updating flow position: The flow moves at velocity V towards the neighbor with the optimal objective value. V is directly proportional to its slope. The calculation of V is as follows:

$$V = r_5 \times S \tag{5}$$

Where r_5 is a uniformly distributed random number; S represents the slope vector from the neighbor to the current flow position. S for the i-th flow relative to the j-th neighbor is also determined by the following formula:

$$S(i,j,d) = \frac{fitness(Flow\_X(i,d)) - fitness(Neighbor\_X(j,d)))}{\left\| Flow\_X(j,d) - Neighbor\_X(j,d) \right\|} \tag{6}$$

Where fitness() represents the function for calculating the objective value; and d represents the problem dimension.

The new position of the flow $Flow\_X(i)$ is calculated by the following formula:

$$Flow\_newX(i) = Flow\_X(i) + V \times \frac{Flow\_X(i) - Neighbor\_X(j)}{\left\| Flow\_X(i) - Neighbor\_X(j) \right\|} \tag{7}$$

The function value must not be lower than the objective function value of the flow. This concept is analogous to the water-filling process of a sink, which determines the direction of the flow. To simulate the aforementioned state, the FDA randomly selects another flow using the following equation:

$$Flow\_newX(i) = \begin{cases} Flow\_X(i) + \overline{r_2} \times (Flow\_X(r) - Flow\_X(i)) \\ fitness(Flow\_X(r)) < fitness(Flow\_X(i)) \\ Flow\_X(i) + r_6 \times (Best_X - Flow_X(i)) \\ otherwise \end{cases} \tag{8}$$

Where, r is a random integer; r_2 is a random vector uniformly distributed in $[0, 1]$; and r_6 is a uniformly distributed random number. Finally, the fitness value of the newly generated flow is calculated and compared with that of the previous flow to determine its position at iteration Iter+1.

IV. Methodology

A. General Framework

As illustrated in Fig. 3, the comprehensive methodology for identifying wheat diseases and pests utilizing the IFDA-EMA-YOLOv9 approach comprises two primary components:

network architecture design and full-process optimization strategies.

At the network architecture level, the model is built upon YOLOv9 as the baseline. The backbone and neck networks are constructed by integrating CBS, ELAN, and CBFuse modules. Notably, the Enhanced Multi-scale Attention (EMA) mechanism module is introduced to bolster the model's capability in capturing key features and to improve the efficiency of multi-scale feature fusion.

At the system workflow level, the framework establishes a closed-loop processing mechanism: first, Module 1 performs preprocessing tasks, including image dimension standardization and data cleaning; subsequently, Module 2 integrates IFDA into EMA-YOLOv9 for iterative hyperparameter optimization to achieve the optimal model configuration; finally, Module 3 utilizes the optimized model to execute high-precision object detection tasks on the input samples.

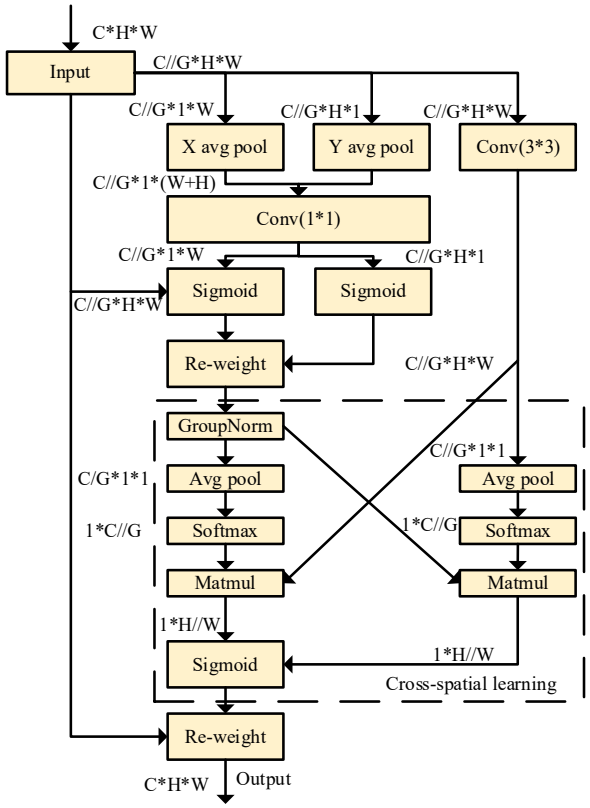

Fig. 4. Structure of EMA.

B. EMA-YOLOv9

1) Multi-level EMA: The relatively small size of wheat and its leaves, coupled with colors that are easily confused with other foliage, poses challenges for traditional YOLO detection models. Incorporating attention mechanisms has emerged as a viable solution to address these issues. IFDA-EMA-YOLOv9 introduces an enhanced multi-scale attention mechanism into YOLOv9, which facilitates the uniform distribution of spatial semantic features across each feature map. The multi-scale architecture of EMA is particularly effective in establishing short-term and long-term dependencies, thereby strengthening the model's feature extraction capabilities and ultimately enhancing object detection performance [13]. The structural representation of EMA is shown in Fig. 4.

EMA is defined as an efficient multi-scale attention mechanism characterized by the absence of dimensionality reduction and its role as a cross-space learning method. It establishes short-term and long-term dependencies through a multi-scale parallel sub-network. By modeling long-term dependencies, the EMA mechanism preserves precise positional information. By fusing contextual information across different scales, the object detection model gains the ability to generate pixel-level attention for feature maps. Compared to traditional Coordinate Attention, EMA utilizes only two convolution kernels (1×1 and 3×3). To aggregate multi-scale spatial structural information, the 3×3 kernel is arranged in parallel with the 1×1 branch to facilitate rapid response. This parallel sub-structure allows the network to avoid extensive sequential and deep processing. EMA enables the learning of effective channel representations without the need for channel dimensionality reduction, thereby generating superior pixel-level attention for high-level feature maps.

As shown in Fig. 5, three methods have been designed to integrate EMA into YOLOv9, including implementations at the backbone and neck/head levels:

Placing EMA at the end of the backbone, as shown in Fig. 5 (a).

Placing EMA after the RepNCSSPELAN4 module in the backbone, as shown in Fig. 5 (b).

Embedding EMA after the RepNCSSPELAN4 module within the Neck and Head structures, as shown in Fig. 5 (c).

This paper aims to investigate the performance of multi-level EMA through experiments and, based on the three aforementioned implementation strategies, determine the optimal configuration for the integration of YOLOv9 and EMA.

2) ICIoU Loss: The original YOLOv9 model employs CIoU (Complete Intersection over Union) loss as its loss function. CIoU loss comprehensively considers three geometric parameters: overlap area, central point distance, and aspect ratio [14]. The calculation formula for CIoU loss is as follows:

$$CIoU = IoU - \left(\frac{\rho^2(b, b^{gt})}{c^2} + \kappa\vartheta\right) \tag{9}$$

Where IoU represents the overlapping area; $\rho^2(b, b^{gt})$ represents the square of the Euclidean distance between the center points of the predicted box and the ground truth box; c^2 represents the square of the diagonal length of the smallest enclosing rectangle covering both boxes; κ and ϑ are aspect ratio parameters, calculated as follows:

$$\kappa = \frac{\vartheta}{1 - IoU + \vartheta} \tag{10}$$

$$\vartheta = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}) \tag{11}$$

Here, w^{gt} and h^{gt} represent the height and width of the ground truth box, while h and w represent the height and width of the predicted box. The CIoU loss function accounts for both central point distance and the consistency of the aspect ratio. However, the variability in the shapes and scales of wheat and its leaves limits the effectiveness of CIoU loss in accurately calculating IoU for irregularly shaped targets. Furthermore, CIoU loss struggles to maintain stable localization accuracy across different target scales, thereby impacting the overall detection accuracy

To address the aforementioned challenges, we combine the Inner IoU with the CIoU loss and introduce an Auxiliary Box into the CIoU framework, thereby proposing the ICIoU loss function. The structural diagram of the auxiliary box is illustrated in Fig. 6.

The center point of the auxiliary box coincides with that of the ground truth box, but their dimensions differ. The auxiliary box serves as a representation of the ground truth box. In cases of high IoU values, the absolute value of the IoU gradient for a smaller-scaled auxiliary box is greater than that of the ground truth box. Conversely, for samples with low IoU values, the absolute value of the IoU gradient for a larger-scaled auxiliary box is smaller than that of the ground truth box. Therefore, adjusting the dimensions of the auxiliary box contributes to accelerating model convergence.

By adjusting the parameter r_{IoU} , the dimensions of the auxiliary box can be customized to adapt to different datasets. This design facilitates more precise fine-tuning of the predicted box position when analyzing wheat pest images with high IoU values. Conversely, when processing images with low IoU values, a larger auxiliary box can be employed to fully cover the target region, thereby mitigating localization errors associated with CIoU loss.

The calculation method for the auxiliary box is described as follows. First, the four boundary coordinates of the actual auxiliary box must be calculated respectively:

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} \times r_{IoU}}{2} \tag{12}$$

$$b_b^{gt} = y_c^{gt} + \frac{h^{gt} \times r_{IoU}}{2} \tag{13}$$

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} \times r_{IoU}}{2} \tag{14}$$

$$b_r^{gt} = x_c^{gt} + \frac{w^{gt} \times r_{IoU}}{2} \tag{15}$$

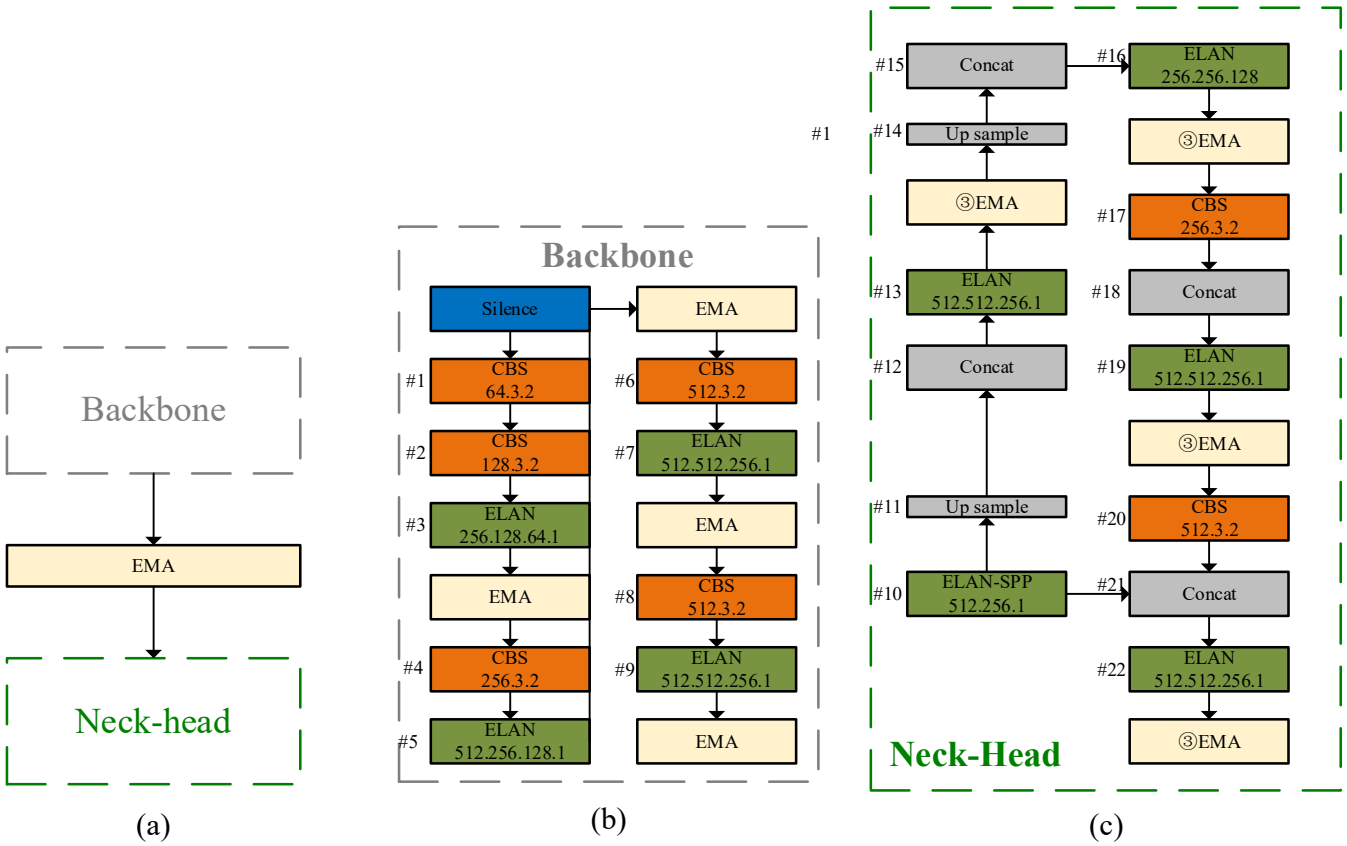

Fig. 5. Diagrammatic sketch of multi-level EMA.

Similarly, the four boundary coordinates of the predicted auxiliary box— b_t 、 b_b 、 b_l and b_r —are calculated respectively. Subsequently, the intersection and union of the auxiliary boxes are calculated:

$$inter = (min(b_r^{gt}, b_r) - max(b_l^{gt}, b_l)) \times (min(b_b^{gt}, b_b) - max(b_t^{gt}, b_t)) \quad (16)$$

$$union = (w^{gt} h^{gt})(r_{IoU})^2 + (wh)(r_{IoU})^2 - inter \quad (17)$$

Subsequently, the intersection-over-union (IoU) term $1 - \frac{inter}{union}$ of the auxiliary box is calculated to serve as the auxiliary box loss. Finally, the loss function of EMA-YOLOv9 is expressed as:

$$ICIoU = IoU - (\frac{\rho^2(b, b^{gt})}{c^2} + \kappa\upsilon) + 1 - \frac{inter}{union} \quad (18)$$

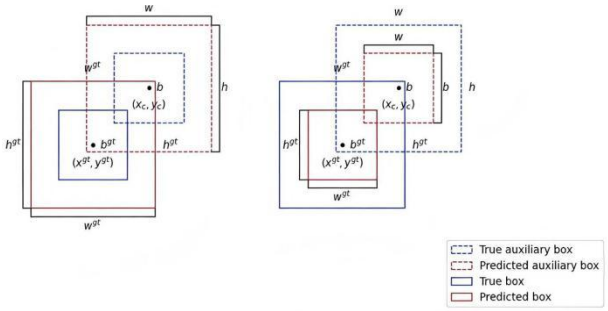

Fig. 6. Schematic diagram of auxiliary frame.

3) Cross-scale feature interaction: To enhance the feature extraction capability of YOLOv9, residual connections were introduced into the backbone network architecture of this model. In the field of object detection, it is generally recognized that low-level features contain richer edge and texture details, whereas high-level features tend to represent more abstract semantic content. Cross-level feature fusion is considered beneficial for improving the model's performance in tasks such as small object detection and fine segmentation. The implementation of residual connections enables YOLOv9 to effectively capture deep information that is difficult to adequately express with single-level features, thereby enriching the hierarchical representation of features and facilitating the acquisition of high-level semantic information. IFDA-EMA-YOLOv9 adopts a residual structure derived from the ResNet architecture, establishing connections among the 4th, 6th, and 8th layers of the backbone network to promote cross-scale feature interactions. The specific residual structure employed is illustrated in Fig. 7. This structure initially utilizes a 1x1 convolution to perform dimensionality reduction on the residual connection, subsequently connects to a 3x3 convolution, and finally increases the dimensionality through another 1x1 convolution.

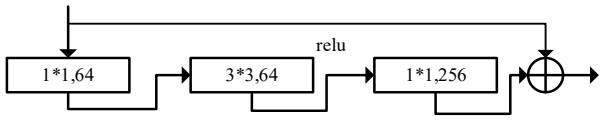

Fig. 7. Matic diagram of residual structure.

C. IFDA

The Improved Flow Direction Algorithm (IFDA) introduces three enhancements based on the original FDA, specifically encompassing solution initialization, the fitness function, and the solution update mechanism.

Meanwhile, to achieve high-precision training of deep learning models, hyperparameters must be adjusted. Notably, evaluating each parameter set requires the model to complete a full training cycle. The formal description of the parameter tuning problem is as follows:

$$\min errorRate = F(P) \quad s.t. \begin{cases} P = \{p_1, p_2, \ldots, p_i\} \\ p_i \in [\mu b_i, lb_i] \\ i = \{1, 2, \ldots, k\} \end{cases} \quad (19)$$

Where *errorRate* denotes the error of the trained model; P represents the set of model hyperparameters; k indicates the number of hyperparameters requiring parameterization; and $[ub_i, lb_i]$ signifies the variation range of parameter p_i.

Early methodologies primarily relied on manual tuning, an approach heavily dependent on the practitioners' expert experience. However, given the inherent uncertainty and time-consuming nature of manual tuning, grid search techniques were gradually adopted. Nevertheless, in the training of deep learning models characterized by numerous parameters and extensive value ranges, grid search often results in high computational complexity and excessive time consumption. This complexity stems from the significantly higher training overhead of deep learning models compared to shallow models such as Support Vector Machines (SVM). Consequently, this paper proposes utilizing heuristic algorithms for parameter optimization.

1) Solution initialization: IFDA must first randomly generate initial solutions within the constrained solution space. This paper summarizes all hyperparameters that may significantly impact the performance of EMA-YOLOv9, as shown in Table I.

The hyperparameters requiring further explanation are as follows:

- Determines the model version of YOLOv9. The original authors of YOLOv9 proposed five different versions, namely YOLOv9-T, YOLOv9-S, YOLOv9-M, YOLOv9-C, and YOLOv9-E, each corresponding to a different number of parameters. Specifically, the YOLOv9-E version contains 57.3 million parameters, while the YOLOv9-T version has the fewest, at only 2 million.

TABLE I. HYPER-PARAMETERS OF EMA-YOLOV9 THAT NEED TO BE OPTIMIZED

Category	Parameter	Symbol	Value range
Model Structure Hyperparameters	Model size	$Size$	$Size \in \{0,1,2,3,4\}$
	Location of EMA	α_{EMA}	$\alpha_{EMA} \in \{0,1,2,3,4,5\}$
	Network width gain	d_m	$d_m \in (0,1]$
	Network width gain	w_m	$w_m \in (0,1]$
Model Training Hyperparameters	Activation function	AF	$AF \in \{0,1,2,3,4,5\}$
	Batch size	BS	$BS \in [8,256], BS \in E^+$
	Warm up epochs	$epochs_{warm}$	$epochs_{warm} \in [1,50], epochs_{warm} \in N^+$
	Initial learning rate	LR	$LR \in [0.0001, 0.001]$
	Final learning rate	LR'	$LR' \in [0.001, 0.1]$
	Momentum of optimizer	$Momentum$	$Momentum \in [0,1]$
	Weight decay coefficient	WD	$WD \in [0,1]$
	Decay rate of moment estimation index	EDR	$EDR \in [0,1]$
	IoU loss weight	w_1	$w_1 \in [0,1]$
	Weight loss of center point distance	w_2	$w_2 \in [0,1]$
	Weight loss of aspect ratio distance	w_3	$w_3 \in [0,1]$
	Tune parameter	r_{IoU}	$r_{IoU} \in [0.5,1.5]$

- α_{EMA} determines the position of the EMA module within the YOLOv9 $\alpha_{EMA}=0$ signifies that no EMA is incorporated; $\alpha_{EMA}=1$ indicates the integration of EMA after the 9th layer (#9); $\alpha_{EMA}=2$ indicates introducing EMA after layers #3, #5, #7, and #9; $\alpha_{EMA}=3$ indicates introducing EMA after layers #13, #16, #19, and #22; $\alpha_{EMA}=4$ indicates introducing EMA at layers #9, #13, #16, #19, and #22; $\alpha_{EMA}=5$ indicates introducing EMA after layers #3, #5, #7, #9, #13, #16, #19, and #22.
- d_m represents the network Depth Gain. This parameter influences the number of Bottleneck CSP modules and SPPCSPC modules within the ELAN architecture of the backbone network. This factor directly determines the overall network depth of YOLOv9.
- w_m represents the network Width Gain. This parameter influences the number of convolutional channels in both the backbone network and the detection head. Consequently, this factor directly determines the overall network width of YOLOv9.
- AF determines the activation function utilized in YOLOv9. The set $\{GeLU, ReLU, Mish, Swish, SiLU, LeakyReLU\}$ is encoded sequentially as $\{0,1,2,3,4,5\}$.
- w_1, w_2, w_3 are the three loss function weight coefficients for ICIoU, whereas the weight for the auxiliary box loss is determined by $1-w_1-w_2-w_3$.

$$ICIoU = w_1 IoU - w_2\left(\frac{\rho^2(b,b^{gt})}{c^2} + w_3\kappa\vartheta\right) + (1-w_1-w_2-w_3)\left(1-\frac{inter}{union}\right) \quad (20)$$

Traditional methods for population initialization typically involve random distribution across the entire space; however, this approach often leads to significant non-uniformity, thereby limiting diversity during the early stages of population evolution and reducing the convergence efficiency of the algorithm [15]. In light of this, we employ various Chaotic Maps to generate initial solutions for different parameters. Utilizing chaotic maps facilitates the generation of more uniformly distributed sequences, which in turn enhances population diversity.We employ the Singer chaotic map to generate the initial solutions for $Momentum$, WD, and EDR, utilize the Gaussian chaotic map to generate the initial solution for $epoch_{warmup}$, and use the Tent chaotic map to generate initial solutions for the other hyperparameters.

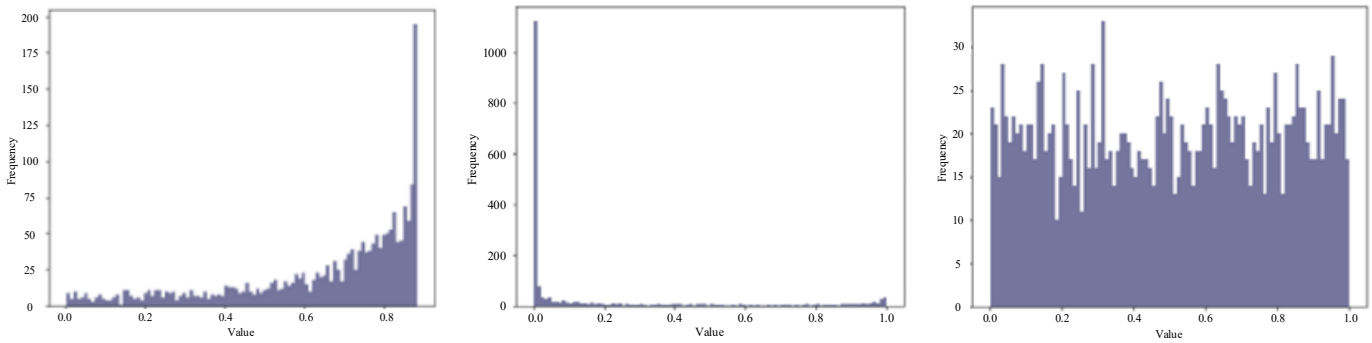

Fig. 8. Distribution curves of three chaotic maps.

The Singer chaotic map is capable of uniformly generating a relatively large number of high values, and its mapping formula is as follows:

$$x_{k+1}^{chaos} = 0.99(7.86_k^{chaos} - 23.32(x_k^{chaos})^2 + 28.75(x_k^{chaos})^3 - 13.302875(x_k^{chaos})^4) \quad (21)$$

The Gaussian chaos mapping can uniformly generate more small values; its mapping formula is:

$$x_{k+1}^{chaos} = \begin{cases} 1, x_k^{chaos} = 0 \\ \dfrac{1}{\mathrm{mod}(x_k^{chaos}, 1)}, x_k^{chaos} \neq 0 \end{cases} \quad (22)$$

The Tent Chaos Mapping can generate uniformly distributed hyperparameters; its mapping formula is:

$$x_{k+1}^{chaos} = \begin{cases} 2x_{k+1}^{chaos}, 0 \leq x_{k+1}^{chaos} \leq \dfrac{1}{2} \\ 2(1 - x_{k+1}^{chaos}), \dfrac{1}{2} \leq x_{k+1}^{chaos} \leq 1 \end{cases} \quad (23)$$

The distribution curves of the three mappings are illustrated in Fig. 8.

In this paper, mAP@0.5 is employed as the metric to evaluate the recognition performance of the model. Additionally, the number of model parameters is utilized as an indicator of model efficiency. To address the multi-objective optimization challenge of simultaneously enhancing recognition performance and reducing the number of model parameters, this paper transforms the problem into two independent single-objective optimization tasks for resolution. *Zoom(Parameters)* refers to the process of normalizing the model parameters into the [0, 1] interval based on established upper and lower bounds; these boundary values can be derived from the value ranges provided in Table I.

Furthermore, the number of additional modules in YOLOv9 also influences the total parameter count of the model. However, since the developers of YOLOv9 determined the quantity of these modules based on empirical knowledge, we did not optimize the BottleneckCSP modules, SPPCSPC modules, or other modules, except for those related to the number of convolutional channels.

2) Fitness function: The fitness function plays a crucial role in evaluating the quality of solutions and serves as the foundation of heuristic algorithms. For the hyperparameter optimization problem of EMA-YOLOv9, a fitness function as shown in (24) was constructed. This function consists of two components: recognition performance and model structural lightweighting.

$$Fitness = mAP_{0.5} - Zoom(Parameters) \quad (24)$$

We employ mAP0.5 as the metric for evaluating the recognition performance of the model. Additionally, the number of model parameters is utilized as a metric to measure model efficiency. To address the multi-objective optimization challenge of simultaneously improving recognition performance and reducing model parameters, this paper transforms it into two independent single-objective optimization problems for solution. Zoom (Parameters) refers to the process of normalizing model parameters to the [0, 1] interval based on established upper and lower bounds; these boundary values can be derived from the value ranges provided in Table I.

Furthermore, the number of additional modules in YOLOv9 also affects the total parameter count of the model. However, considering that the developers of YOLOv9 determined the quantity of these modules based on empirical knowledge, we did not optimize the BottleneckCSP modules, SPPCSPC modules, or other modules, with the exception of those related to the number of convolutional channels.

3) Update mechanism: The process of simulating convergence towards the current optimal flow may cause the algorithm to fall into local optima. Consequently, the Lévy Flight mechanism was introduced, as described in (8). Lévy Flight is characterized by a random walk that employs a heavy-tailed probability distribution for step sizes. This distribution grants a relatively high probability of occurring substantial steps during the random walk process.

Following the introduction of the Lévy Flight mechanism, the update formula for is reformulated as:

$$Flow\_newX(i) = \begin{cases} Flow\_X(i) + \overline{r_2} \times (Flow\_X(r) - Flow\_X(i)) \\ fitness(Flow\_X(r)) < fitness(Flow\_X(i)) \\ Flow\_X(i) + r_{levy} \times (Best_X - Flow_X(i)) \\ otherwise \end{cases} \tag{25}$$

Where represents the Lévy random step size, which is calculated as follows:

$$r_{levy} = \frac{\mu_{levy}}{|v_{levy}|^{\frac{3}{2}}} \tag{26}$$

$$\mu_{levy} \sim N(0, \sigma^2).v_{levy} \sim N(0,1)$$

$$\sigma = \left[\frac{\Gamma(1+1.5)\sin(\frac{3\pi}{4})}{1.5\Gamma(\frac{1+1.5}{2})2^{\frac{0.5}{2}}} \right]^{\frac{3}{2}} \tag{27}$$

Fig. 9. Example image of the LWDCD2020 dataset.

V. Experiments

A. Experimental Details

In this experimental study, the Adam optimizer is utilized within the YOLOv9 model, with the number of training epochs set to 300. The default values for momentum, weight decay (WD), and EDR (Exponential Decay Rate) range from 0.8 to 0.99, while the warmup epochs ($epoch_{warmup}$) are maintained within 10.

B. Dataset and Evaluation Metrics

1) Dataset construction and preprocessing: To comprehensively evaluate the detection performance of the proposed improved algorithm under complex scenarios, the publicly available LWDCD2020 dataset is adopted as the primary experimental dataset (as shown in Fig. 9). This dataset contains three common wheat disease categories—crown and root rot, leaf rust, and loose smut—as well as a healthy class, comprising a total of 12,000 images. Notably, the LWDCD2020 dataset is collected from diverse sources, integrating both field-acquired images and web-crawled data. Consequently, the images exhibit substantial variability in illumination conditions, viewing angles, background complexity, and object scale, enabling effective coverage of a wide range of real-world agricultural scenarios. This diversity endows the dataset with high complexity and representativeness, making it well-suited for evaluating the robustness and generalization capability of detection algorithms. It is worth noting that approximately 20% of the images are obtained via web crawling, which inevitably introduces inconsistencies between images and their annotations. These issues mainly include non-authentic wheat images, mislabeled samples, and images irrelevant to wheat. To ensure data quality and the reliability of experimental results, a rigorous manual verification and cleaning process is conducted to remove contaminated samples, thereby establishing a reliable dataset for subsequent experiments.

Within the disease samples, wheat plants infected with crown and root rot typically exhibit progressive leaf yellowing followed by wilting, accompanied by browning at the stem base, with symptoms intensifying as the disease develops. In contrast, leaf rust is characterized by nearly circular orange-red lesions irregularly distributed on leaf surfaces; as the disease progresses, these lesions may penetrate the leaf tissue, while dark brown to black elliptical lesions appear on the abaxial side. For loose smut, infected wheat spikes are initially covered by a thin gray membrane, which eventually ruptures as the disease advances, releasing black مسحوق-like spores. Diseased spikes are noticeably shorter than healthy ones, and both the main stem and tillers may be affected. Based on these observations, each disease category exhibits distinctive visual characteristics, while significant inter-class differences are maintained. These properties provide effective discriminative cues for detection models and further demonstrate the practical value of applying object detection methods to such tasks.

To enhance experimental efficiency, a parameter parallel mechanism was introduced to accelerate the execution of the IFDA algorithm. This mechanism transforms the water flow population into Resilient Distributed Datasets adapted for the

Spark platform, thereby realizing the parallel computation of the fitness of each individual within the population. This method significantly improves the execution efficiency of the experiments, providing robust technical support for subsequent experimental verification [16].

2) Evaluation metrics: This experiment adopts common evaluation metrics in the field of object detection to quantify the model's performance, primarily including Precision (P), Recall (R), and mean Average Precision (mAP).

The calculation formulas for Precision (P) and Recall (R) are as follows:

$$P = \frac{TP}{TP + FP} \tag{28}$$

$$R = \frac{TP}{TP + FN} \tag{29}$$

Where TP (True Positive) represents the number of correctly detected positive samples, FP indicates the number of falsely detected negative samples, and FN denotes the number of missed positive samples.

To comprehensively evaluate the detection effectiveness, we employ mAP@0.5 and mAP@0.5:0.95 as the core metrics:

mAP@0.5: Represents the mean of the average precision across all categories when the Intersection over Union (IoU) threshold is set to 0.5.

mAP@0.5:0.95: Represents the mean of the average precision when the IoU threshold varies from 0.5 to 0.95 (with a step size of 0.05).

TABLE II. COMPARATIVE EXPERIMENT OF MULTIPLE HEURISTIC EMA-YOLOV9 HYPERPARAMETER OPTIMIZATION ALGORITHM

Algorithm	Best Fitness	Average Fitness	Fitness Standard Deviation	Ranking Based on Average Value
EGA	0.309	0.2824	0.0155	12
GWO	0.4509	0.4329	0.0117	3
MVO	0.4341	0.4130	0.0206	4
SSA	0.4242	0.4053	0.0216	7
SCA	0.4325	0.4122	0.0156	5
BBO	0.3809	0.3304	0.0425	11
JSO	0.4142	0.3940	0.0232	10
MBO	0.4365	0.4047	0.0211	8
BA	0.4251	0.4044	0.0146	9
CSO	0.4203	0.4070	0.0108	6
FDA	0.4574	0.4402	0.0132	2
IFDA	**0.46856**	**0.4486**	**0.0144**	**1**

The population size ($Size_{pop}$) for all heuristic algorithms is set to 20, and the maximum number of iterations ($MaxIter$) is also set to 20. The number of neighbors used in IFDA and FDA is specified as 8. The hyperparameters for the remaining optimization algorithms are all configured to their default settings. Each algorithm is run independently 5 times to calculate the statistics for its optimal value, mean, and standard deviation.

Based on the statistical data analysis in Table II, although all tested algorithms exhibit a certain degree of inherent randomness, and no significant statistical differences are observed in the overall randomness distribution, a horizontal comparison of the baseline algorithms reveals that the Genetic Algorithm (GA) has the relatively lowest optimization effectiveness, whereas the Flow Direction Algorithm (FDA) demonstrates the best comprehensive performance. This empirical result strongly supports the assertion that FDA is the most suitable baseline optimization algorithm for the current EMA-YOLOv9 parameter optimization task.

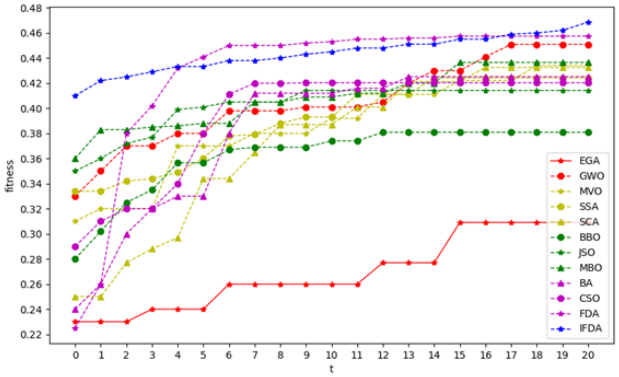

Fig. 10. Optimal fitness curve.

Fig. 10 illustrates the fitness convergence trajectories of the optimal results for each heuristic algorithm. It can be observed that the convergence curve representing IFDA exhibits a significant and steep upward trend during the initial iterations, indicating that, compared to other algorithms, IFDA possesses superior early convergence capability and optimization speed. This performance enhancement is primarily attributed to the improvement of the initialization solution strategy, which effectively elevates the quality of the initial population. Furthermore, the IFDA curve maintains a continuous upward trend in the later stages of iteration, indicating that the algorithm possesses a robust ability to escape local optima traps; the acquisition of this critical characteristic benefits from the introduction and optimization of the Lévy flight mechanism, thereby significantly enhancing the algorithm's global exploitation potential during the later stages of the search.

TABLE III. RESULTS OF ABLATION EXPERIMENT

Algorithm	Optimal fitness	Average adaptability	Accuracy	Recall rate	mAP_0.5
FDA	0.4574	0.4402	95.32	94.55	95.74
Initialization + FDA	0.4552	0.4427	95.10	94.33	95.52
Lévy flight mechanism + FDA	0.4662	0.4468	96.23	95.41	96.62
Initialization + Lévy flight mechanism + FDA	0.4686	0.4486	96.41	95.63	96.86

Specifically, after fine-tuning via the IFDA optimization strategy, the model achieved a significant increase of 5.09% in Precision, 5.11% in Recall, and 5.17% in mAP@0.5. These results fully substantiate the effectiveness of this optimization method in enhancing detection performance for complex object detection tasks.

C. Ablation Study

1) Ablation study of modules: To evaluate the effectiveness of the various improvement strategies within IFDA, an ablation study was conducted in this research. The YOLOv9 architecture and parameter settings remained consistent with those previously described, and the optimal parameters obtained through the optimization of each algorithm were utilized for training. The experimental results are presented in Table III: By combining the initialization strategy, the Lévy flight mechanism, and FDA, the model achieved the highest values in Precision, Recall, and mAP@0.5, confirming the effectiveness of the proposed improvements. Among them, the integration of Lévy flight significantly enhanced the performance of FDA, with the three metrics increasing by 0.95%, 0.91%, and 0.92%, respectively, compared to the original results; the overall performance improved by 0.13%, 0.13%, and 0.14%, respectively, compared to the original algorithm. Furthermore, the results of the ablation study regarding the various improvements of EMA-YOLOv9 are shown in Table IV.

TABLE IV. RESULTS OF ABLATION EXPERIMENT

YOLOv9	EMA	ResNet	IClou	**Accuracy %**	**Recall %**	**mAP_0.5 %**
√	×	×	×	90.80	90.05	90.77
√	√	×	×	91.21	90.26	91.48
√	×	√	×	90.92	90.11	91.29
√	×	×	√	90.98	90.16	91.33

The table systematically presents the performance of the YOLOv9-based object detection model under different strategy combinations, including the incorporation of EMA, ResNet backbone, and the ICIoU loss function, and their effects on Accuracy, Recall, and mAP\@0.5. The baseline YOLOv9 model achieves an Accuracy of 90.80%, Recall of 90.05%, and mAP of 90.77%. Incorporating EMA increases Accuracy to 91.21% and mAP to 91.48%, demonstrating its advantage in training stability and model generalization, with a limited improvement in Recall. Using the ResNet backbone alone results in an Accuracy of 90.92%, Recall of 90.11%, and mAP of 91.29%, indicating that enhanced feature extraction improves both localization and classification performance. Introducing only the ICIoU loss yields an Accuracy of 90.98%, Recall of 90.16%, and mAP of 91.33%, showing that the improved bounding box regression loss significantly optimizes detection precision. When all three strategies are applied simultaneously, the model achieves its highest performance with an Accuracy of 91.74%, Recall of 90.98%, and mAP of 92.09%, indicating a synergistic effect of feature enhancement, training stabilization, and bounding box optimization, thereby achieving overall optimal performance in both classification accuracy and object detection precision and validating the effectiveness of these strategies in enhancing YOLOv9.

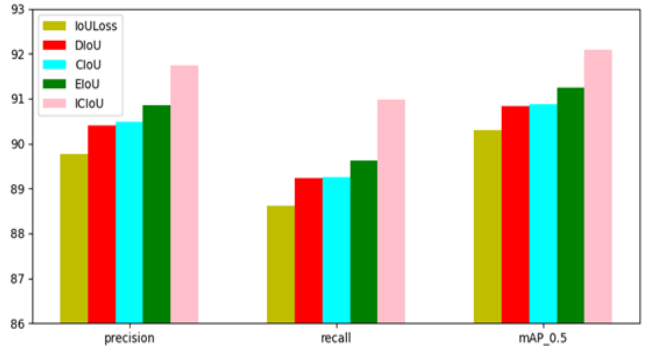

Fig. 11. Comparison of loss function experimental results.

2) Ablation study of loss functions: To verify the effectiveness of the proposed ICIoU loss function in the task of wheat disease and pest identification, we conducted a series of detailed comparative experiments between ICIoU and several other loss functions, including IoU loss, DIoU loss [17], CIoU loss, and EIoU loss [12]. The relevant experimental results are illustrated in Fig. 11. Compared to the baseline IoU loss function, the adoption of the ICIoU loss function improved the model's Precision, Recall, and mAP@0.5 by 2.19%, 2.66%, and 1.99%, respectively. Furthermore, compared to the CIoU loss function, the ICIoU loss function achieved gains of 1.39%, 1.94%, and 1.34% in Precision, Recall, and mAP@0.5, respectively.

Given the No Free Lunch theorem of stochastic optimization, it is impossible to directly determine a specific algorithm universally suited for the hyperparameter optimization of EMA-YOLOv9 [18]. Therefore, this paper compares the proposed IFDA with other heuristic algorithms to validate the advantages of FDA and IFDA in solving the hyperparameter optimization problem for EMA-YOLOv9, as shown in Table V. The heuristic algorithms included in this experiment are: Genetic Algorithm (GA), Grey Wolf Optimizer (GWO) [19], Multi-Verse Optimizer (MVO) [20], Salp Swarm Algorithm (SSA) [21], Sine Cosine Algorithm (SCA) [22], Biogeography-Based Optimization (BBO) [23], Jellyfish Search Optimizer (JSO) [24], Monarch Butterfly Optimization (MBO) [25], Bat Algorithm (BA) [26], and Cockroach Swarm Optimization (CSO) [27].

TABLE V. COMPARATIVE EXPERIMENTAL RESULTS

Method	**Precision %**	**Recall %**	**mAP_0.5 %**
Faster R-CNN[28]	86.73	86.79	87.60
YOLOv5+attention[29]	88.32	87.90	88.73
YOLO v8s+EIOU[30]	92.20	87.80	93.30
YOLOv8+ ShuffleNetV2+Senet[31]	88.90	81.40	88.20
GA-YOLOv5[7]	90.09	89.52	90.43
ABC-YOLOv5[9]	90.29	89.81	90.79
Ours	**96.41**	**95.63**	**96.86**

D. Comparison with SOTA Methods

To verify the advancement of IFDA-EMA-YOLOv9, we compared it with six State-of-the-Art (SOTA) methods. The experimental results are presented in *V*. The selected baseline models are as follows:

Faster R-CNN: Utilizes Faster R-CNN to detect wheat diseases and pests.

YOLOv5+attention: Optimizes YOLOv5 using an improved attention sub-module, and combines BiFPN with fast normalization fusion to weight the fused features.

YOLOv8s+EIOU: Improves YOLOv8s by employing efficient RepGFPN feature fusion and the EIoU loss function.

YOLOv8+ShuffleNet V2+Senet: Employs the lightweight convolutional neural network ShuffleNet V2 as the backbone network of YOLOv8, improving it by adding a small object detection layer and the Senet attention mechanism.

GA-YOLOv5: Represents the optimization of YOLOv5 hyperparameters using its built-in Genetic Algorithm (GA).

ABC-YOLOv5: Represents the optimization of YOLOv5 hyperparameters using the Artificial Bee Colony (ABC) algorithm.

The proposed method demonstrates superior performance compared to GA-YOLOv5 and ABC-YOLOv5, thereby establishing the significant advantages of the IFDA method over traditional optimization algorithms such as the Genetic Algorithm (GA) and Artificial Bee Colony (ABC). Specifically, on the publicly available LWDCD2020 dataset, compared to the best existing method excluding EMA-YOLOv9, the proposed method increases Precision by 6.4% and Recall by 5.94%. Furthermore, the core metric mAP@0.5 is improved by 6.66%. These results indicate that the methodology proposed in this paper has achieved significant progress in related fields. Fig. 12 illustrates the visualized detection results of IFDA-EMA-YOLOv9 across various categories of samples.

(a) Nutritious wheat (b) Crown rot and root rot

(c) leaf rust disease (d) Loose Smut of Wheat

Fig. 12. Visual display of detection result.

VI. Conclusion

This paper proposes a wheat disease detection method based on the improved EMA-YOLOv9 and IFDA. This method integrates multi-scale attention and residual connections to enhance feature extraction, introduces an auxiliary box to optimize localization accuracy, and utilizes the IFDA algorithm—which combines chaotic maps and Lévy flight—to achieve adaptive hyperparameter optimization. Experiments demonstrate that, on the LWDCD2020 dataset, the proposed method improves Precision, Recall, and mAP@0.5 by 6.4%, 5.94%, and 6.66%, respectively, compared to SOTA methods. Additionally, under the experimental environment considered in this study, IFDA-EMA-YOLOv9 achieves a detection frame rate (FPS) of approximately 29.8, which is sufficient to meet the requirements of wheat disease detection tasks. Future work will focus on model lightweighting, exploring its real-time deployment on mobile devices, and its generalized application across multiple scenarios.

References

[1] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in Proc. European Conference on Computer Vision, Cham: Springer Nature Switzerland, 2024.

[2] Li, B., & Fan, J.-W. (2025). Deep Learning-oriented Literature Review and Experimental Analysis of Ship Target Detection. Science Technology and Engineering, 25(27), 11427–11450. https://doi.org/10.12404/j.issn.1671-1815.2409633

[3] Yue, Y. J., et al. "Object detection of tomato fruit based on cascade RCNN." J. Science Technology and Engineering 21 (2021): 2387.

[4] Qiao, S., Yuan, Y., & Qi, R. (2025). Meter-YOLOv8n: A Lightweight and Efficient Algorithm for Word-Wheel Water Meter Reading Recognition. International Journal of Advanced Computer Science & Applications, 16(4).

[5] H. Zhang, C. Xu, and S. Zhang, "Inner-IoU: more effective intersection over union loss with auxiliary bounding box," arXiv preprint arXiv:2311.02877, 2023.

[6] S. Ma and Y. Xu, "Mpdiou: a loss for efficient and accurate bounding box regression," arXiv preprint arXiv:2307.07662, 2023.

[7] L. Popek, R. Perz, G. Galinski, and A. Abratanski, "Optimization of animal detection in thermal images using YOLO architecture," Int. J. Electron. Telecommun., vol. 69, no. 4, pp. 693–700, 2023.

[8] H. Mahmudah, A. S. Aisjah, S. Arifin, and C. A. Prastyantyo, "iYOLOV7-TPE-SS: Leveraging improved YOLO model with multilevel hyperparameter optimization for road damage detection on edge devices," IEEE Access, vol. 13, pp. 1–15, 2025.

[9] A. Araman, I. Pacal, A. Basturk, et al., "Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC)," Expert Syst. Appl., vol. 221, p. 119741, 2023.

[10] Yang, Q.-H., Yang, G.-C., & Zhong, S.-H. (2025). Hyperparameter Optimization of YOLO Model Based on Orthogonal Optimization Strategy. Science Technology and Engineering, 25(4), 1573–1579. https://doi.org/10.12404/j.issn.1671-1815.2309596

[11] J. F. O'Callaghan and D. M. Mark, "The extraction of drainage networks from digital elevation data," Comput. Vis. Graph. Image Process., vol. 28, no. 3, pp. 323–344, 1984.

[12] H. Arami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems," Comput. Ind. Eng., vol. 156, p. 107224, 2021.

[13] D. Uyang, et al., "Efficient multi-scale attention module with cross-spatial learning," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), 2023, pp. 1–5.

[14] Y. F. Hang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," Neurocomputing, vol. 506, pp. 146–157, 2022.

[15] H. Lin and C. Wang, "DIGWO-N-BEATS: An evolutionary time series prediction method for situation prediction," Inf. Sci., vol. 664, p. 120316, 2024.

[16] H. Lin, C. Wang, and Q. Hao, "A novel personality detection method based on high-dimensional psycholinguistic features and improved distributed gray wolf optimizer for feature selection," Inf. Process. Manage., vol. 60, no. 2, p. 103217, 2023.

[17] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 7, 2020, pp. 12993–13000.

[18] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, 1997.

[19] S. N. Makhadmeh et al., "Recent Advances in Grey Wolf Optimizer, its Versions and Applications: Review," in IEEE Access, vol. 12, pp. 22991-23028, 2024, doi: 10.1109/ACCESS.2023.3304889.

[20] L. Abualigah, "Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications," Neural Comput. & Applic., vol. 32, pp. 12381–12401, 2020, doi: 10.1007/s00521-020-04839-1.

[21] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," Neural Comput. Appl., vol. 32, no. 15, pp. 11195–11215, 2020.

[22] Q. S. Hamad, S. A. M. Saleh, S. A. Suandi, et al., "A Review of Enhancing Sine Cosine Algorithm: Common Approaches for Improved Metaheuristic Algorithms," Arch. Computat. Methods Eng., vol. 32, pp. 2549–2606, 2025, doi: 10.1007/s11831-024-10218-z.

[23] W. Guo, M. Chen, L. Wang, et al., "A survey of biogeography-based optimization," Neural Comput. & Applic., vol. 28, pp. 1909–1926, 2017, doi: 10.1007/s00521-016-2179-x.

[24] J. S. Hou and A. Molla, "Recent advances in use of bio-inspired jellyfish search algorithm for solving optimization problems," Sci. Rep., vol. 12, no. 1, p. 19157, 2022.

[25] Y. Feng, S. Deb, G. Wang, A. H. Alavi, "Monarch butterfly optimization: A comprehensive review," Expert Systems with Applications, vol. 168, pp. 114418, 2021, doi: 10.1016/j.eswa.2020.114418.

[26] T. Agarwal, V. Kumar, "A Systematic Review on Bat Algorithm: Theoretical Foundation, Variants, and Applications," Arch. Computat. Methods Eng., vol. 29, pp. 2707–2736, 2022, doi: 10.1007/s11831-021-09673-9.

[27] L. Cheng, et al., "A Review on Cockroach Swarm Optimization Algorithm," in Proceedings of the 12th International Conference on Computer Engineering and Networks, CENet 2022, vol. 961, Springer, Singapore, 2022, doi: 10.1007/978-981-19-6901-0_151.

[28] B. S. Bari, M. N. Islam, M. Rashid, M. J. Hasan, M. A. Razman, R. M. Musa, and A. P. A. Majeed, "A real-time approach of diagnosing rice leaf disease using deep learning-based Faster R-CNN framework," PeerJ Comput. Sci., vol. 7, p. e432, 2021.

[29] H. Wang, S. Shang, D. Wang, X. He, K. Feng, and H. Zhu, "Plant disease detection and classification method based on the optimized lightweight YOLOv5 model," Agriculture, vol. 12, no. 7, p. 931, 2022.

[30] Trinh, D. C., Mac, A. T., Dang, K. G., Nguyen, H. T., Nguyen, H. T., & Bui, T. D. (2024). Alpha-EIOU-YOLOv8: an improved algorithm for rice leaf disease detection. AgriEngineering, 6(1), 302-317.

[31] Cen, X., Lu, S., & Qian, T. (2025). YOLO-LCE: A Lightweight YOLOv8 Model for Agricultural Pest Detection. Agronomy, 15(9), 2022