



Article

GDFC-YOLO: An Efficient Perception Detection Model for Precise Wheat Disease Recognition

Jiawei Qian ¹, Chenxu Dai ¹, Zhanlin Ji ^{2,*} and Jinyun Liu ^{1,*}¹ College of Artificial Intelligence, North China University of Science and Technology, Tangshan 063210, China; qianjiawei@stu.ncst.edu.cn (J.Q.); daichenxu@ncst.edu.cn (C.D.)² College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China

* Correspondence: zhanlin.ji@ncst.edu.cn (Z.J.); liujy23@ncst.edu.cn (J.L.)

Abstract

Wheat disease detection is a crucial component of intelligent agricultural systems in modern agriculture. However, at present, its detection accuracy still has certain limitations. The existing models hardly capture the irregular and fine-grained texture features of the lesions, and the results of spatial information reconstruction caused by standard upsampling operations are inaccurate. In this work, the GDFC-YOLO method is proposed to address these limitations and enhance the accuracy of detection. This method is based on YOLOv11 and encompasses three key aspects of improvement: (1) a newly designed Ghost Dynamic Feature Core (GDFC) in the backbone, which improves the efficiency of disease feature extraction and enhances the model's ability to capture informative representations; (2) a redesigned neck structure, Disease-Focused Neck (DF-Neck), which further strengthens feature expressiveness, to improve multi-scale fusion and refine feature processing pipelines; and (3) the integration of the Powerful Intersection over Union v2 (PloUv2) loss function to optimize the regression accuracy and convergence speed. The results showed that GDFC-YOLO improved the average accuracy from 0.86 to 0.90 when the cross-overmerge threshold was 0.5 (mAP@0.5), its accuracy reached 0.899, its recall rate reached 0.821, and it still maintained a structure with only 9.27 M parameters. From these results, it can be known that GDFC-YOLO has a good detection performance and stronger practicability relatively. It is a solution that can accurately and efficiently detect crop diseases in real agricultural scenarios.

Keywords: crop disease detection; deep learning; image processing; GDFC-YOLO

Academic Editors: Tao Liu and Hanzeyu Xu

Received: 7 June 2025

Revised: 9 July 2025

Accepted: 14 July 2025

Published: 15 July 2025

Citation: Qian, J.; Dai, C.; Ji, Z.; Liu, J. GDFC-YOLO: An Efficient Perception Detection Model for Precise Wheat Disease Recognition. *Agriculture* **2025**, *15*, 1526. <https://doi.org/10.3390/agriculture15141526>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wheat (*Triticum aestivum* L.) is one of the most widely cultivated crops in the world. It is the staple food for over 35% of the global population [1]. Between 2022 and 2023, the area of wheat affected by diseases in China reached 41.66 million hectares, while the total area under control reached 71.48 million hectares. Although the overall severity of wheat diseases across the country remained at a medium level, some diseases have shown severe outbreaks in the northwest region [2]. There are many common pests and diseases, such as stripe rust, leaf rust, loose black smut, aphids and yellow rust. These pests and diseases pose a great threat to the yield and quality of wheat and eventually affect food security and farmers' income. Therefore, it is significant to identify wheat diseases accurately and efficiently. In recent years, deep learning has been widely applied in agricultural scenarios such as pest and disease identification, precise fertilization, and intelligent temperature

control [3]. And the YOLO series has become a research hotspot in disease detection due to its characteristics of real-time performance and high efficiency.

In recent years, in order to achieve efficient object detection in an environment with limited resources, various lightweight models have been proposed. Qiu et al. [4] introduced LGWheatNet, which integrates the SeCUIB and DWDown modules and adopts the strategy of slice inference, resulting in a significant improvement in inference efficiency. The computing cost is merely 5.0 GB floating-point operations. Within the range of the intersection point-super-binding threshold of 0.5–0.95 and the step size of 0.05, the average accuracy of this method during the flowering period of wheat can be maintained at 0.662. Yao et al. [5] developed YOLO-Wheat by combining the C2f-DCN module with the SCNet attention mechanism. Although the model is 23.94 MB in size, mAP@0.5 has an astonishing accuracy rate of 93.28%. Although these methods have achieved a significant reduction in the complexity of the model and the computational cost, the ability to extract fine-grained features of tiny or morphologically diverse lesions in complex natural scenes is still relatively limited. This reveals the contradiction that has always existed between the demand for high-precision feature representation and the constraint of computational efficiency in the natural environment.

To solve the problem of fine-grained modeling of irregular lesion structures, many studies have combined the Transformer architecture with multi-scale feature enhancement techniques. Kumar et al. [6] proposed the Cat-YOLOv9 algorithm, which integrates the class attention converter and the multi-head attention mechanism. It has greatly improved the localization ability of densely distributed lesions, and its accuracy rate reached 95%. Zhong et al. [7] introduced a multi-scale edge enhancement module and a global context module within the YOLOv8 framework and embedded a ViT layer, achieving a mAP@0.5 result of 89.5%. Bao et al. [8] utilized unmanned aerial vehicle-based remote sensing images and spatial attention mechanisms. The detection rate of early small lesions increased to 83.2% mAP@0.5. However, these methods still rely heavily on traditional convolutional layers and standard upsampling operations, which limits their spatial reconstruction ability for irregular lesion edges. As a result, it is very difficult to capture fine-grained geometric details.

In the fields of data improvement and application-specific optimization, some studies have explored the application of generative adversarial networks and multi-stage detection frameworks in alleviating data scarcity and addressing task diversity. Volety et al. [9] combined YOLOv8 and gan to expand the training dataset, achieving a classification accuracy rate of 97.2%. Doroshenko et al. [10] developed an intelligent microscopic scanning system that integrates electron microscopy, coordinate staging, and convolutional neural networks to automatically assess the spore concentration of wheat leaf diseases. The maps of the YOLOv8 and RT-DETR models reached 98.4% and 98.2%, respectively. Jiang et al. [11] proposed a stage awareness detection framework for the different growth stages of wheat. At multiple development stages, the mAP values ranged from 66.55% to 82.8%. Although these improvements have been made, most existing methods rely on loss functions based on the joint of intersection ratios, such as CIoU and GIoU. These methods are particularly sensitive to the size of the anchor box and are difficult to adapt to the dynamic damage scale. This will mostly affect the convergence speed and positioning accuracy of the regression.

In the specialized plant disease detection task, the multi-scale feature fusion technology and the instance segmentation technology were combined for use, which improved the detection accuracy. Sharma et al. [12] successfully achieved accurate detection of wheat stripe rust with the help of the YLo-based model, and the detection accuracy rate reached 97.56%. Önlér et al. [13] used YOLOv8m to detect wheat powdery mildew, and the final

f1 score was 79%. Mao et al. [14] introduced the DAE-Mask model and combined edge enhancement and dense connection networks, achieving an average accuracy of 96.02% on the MSWDD2022 dataset. Sharma et al. [15] once again applied YOLOv8 to the instance segmentation work of powdery mildew, achieving a detection accuracy rate of 99.37%. Although these methods performed well in multi-scale feature fusion and attention-based modeling, they relied on *ou* regression. There are still relatively prominent deviations in the localization of large lesions, and the sensitivity to scale changes is also relatively limited. This imposes restrictions on the stability and convergence efficiency of the model.

To address these challenges, this study proposes the high-precision detection model GDFC-YOLO based on YOLOv11s. The main contributions of this paper are as follows: The GDFC module was designed. This module can enhance the extraction ability of lesion textures under natural field conditions. Meanwhile, it can also keep the structure lightweight. The feature fusion structure is optimized by relying on DF-Neck to improve the local perception ability and spatial reconstruction ability of the model. The PIoUv2 loss function is introduced. It solves the instability problem that exists in the traditional *iou* regression, accelerates the convergence speed, and improves the accuracy of the boundary box in complex lesion detection scenarios.

2. Materials and Methods

2.1. Dataset Construction

2.1.1. Data Sources

The wheat disease images used in this study were collected at the Yuanyang Modern Agricultural Science and Technology Research Base of Henan Agricultural University. Relevant data were collected from February to late May 2023, with image acquisition performed from 8:00 a.m. and 6:00 p.m. each day. The equipment used included a Xiaomi 10S smartphone (Xiaomi Corporation, Beijing, China) and a Canon EOS 1500D camera (Canon Inc., Tokyo, Japan), and all images were saved in JPG format. All photographs were taken under natural field conditions, and the backgrounds that included soil, sky, and wheat fields reflected realistic agricultural environments. To enhance the diversity of the dataset and improve the generalization capability of the model, images were captured under various angles and lighting conditions, and image resolutions ranged from 600×800 to 1536×2048 .

Wheat diseases mainly affect the spike axis and leaves. After diagnosis and verification by experts, a total of 4156 images were screened out. These images were used to construct a dataset containing various disease categories. Specifically, this dataset contains 632 images of leaf rust and 435 stripe images. There were 1276 cases of *Fusarium* head blight, also known as scab disease; 468 cases of leaf spot disease; and 400 cases of powdery mildew. In addition, this dataset also contains 501 pictures of healthy wheat ears and 444 pictures of healthy wheat leaves. Some representative examples can be seen in Figure 1.

2.1.2. Dataset Production

This paper aims to adapt to the multi-instance annotation scenario. Here, it needs to be explained that the multi-instance annotation scenario refers to the situation where a single image may contain multiple disease instances. A hierarchical data partitioning strategy based on annotation instances is adopted. This paper uses the method of hierarchical random sampling to divide the data set in an 8:1:1 ratio, dividing it into the training set, the validation set, and the test set. Among them, the training set has 3359 images, including 21,487 annotated instances; the validation set has 377 images, including 2402 instances; and the test set has 420 images, including 2683 instances. This approach ensures that the

category distribution within each subset is statistically consistent with the original dataset. Table 1 gives a detailed description of the distribution of classes.

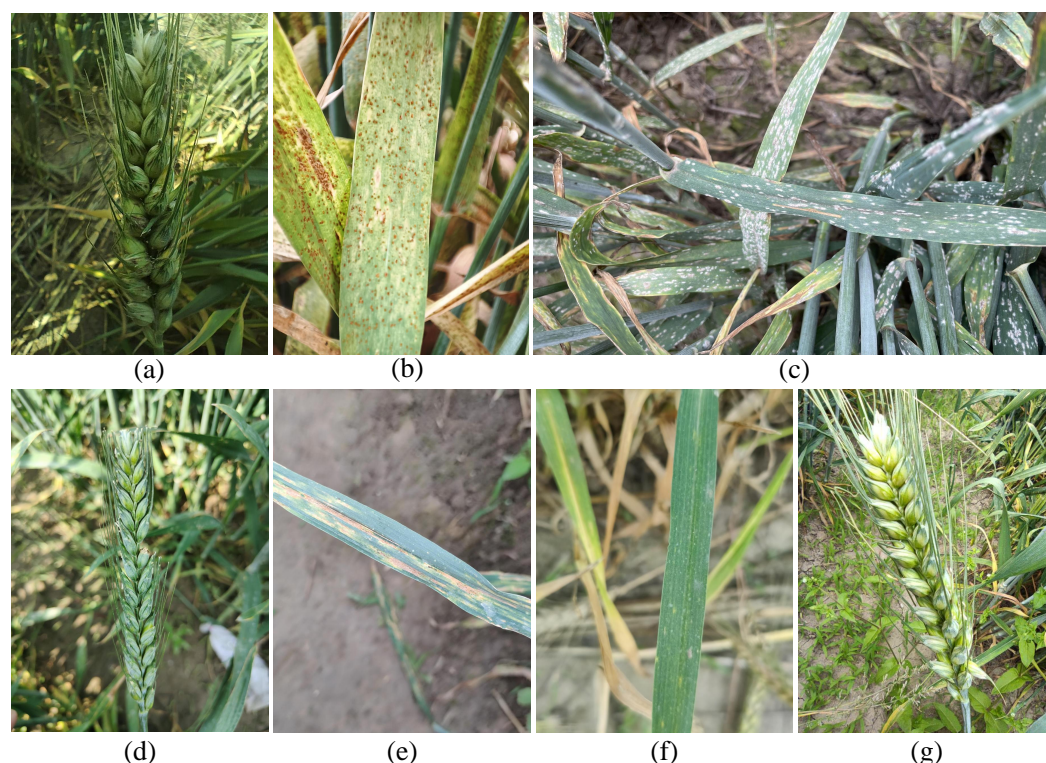


Figure 1. Representative samples from the wheat disease image dataset. (a) Typical symptoms of glume blotch. (b) Typical symptoms of leaf rust. (c) Typical symptoms of powdery mildew. (d) Typical symptoms of scab. (e) Typical symptoms of stripe rust. (f) Healthy wheat leaf. (g) Healthy wheat ear.

Table 1. Statistics of the wheat disease dataset splits.

Disease Category	Training Set	Validation Set	Test Set	Total Instances	Category Proportion
Powdery Mildew	810	102	108	1020	17.45%
Scab	1060	125	110	1295	22.16%
Leaf Rust	827	107	102	1036	17.73%
Stripe Rust	470	63	60	593	10.15%
Glume Blotch	351	36	41	428	7.32%
Wheat Ear	491	81	52	624	10.68%
Wheat Leaf	680	66	103	849	14.53%
Total	4689	580	576	5845	100%

Note: The splitting ratio is calculated based on the total number of instances (5845). The actual number of images in the training, validation, and test sets are 3359, 377, and 420, respectively.

Figure 2a presents the histogram of the centroid coordinates of the object. This histogram can reveal the spatial prior situation and the status of positional deviation. Such a revelation can promote the modeling work of context-dependent and spatial attention mechanisms. In Figure 2b, the edge distribution of the object width and the edge distribution of the object height are visualized. This kind of visualization display can provide relatively detailed statistical insights into the variability of the object scale. These statistical data lay the foundation for developing more effective data improvement strategies, such as random cropping strategies and scale-aware transformation strategies. At the same time, they can also provide some useful information for the acceptance of the field calibration of the detection backbone. Overall, these visual diagnostic contents constitute a relatively comprehensive multi-dimensional dataset quality assessment framework. With this frame-

work, data-driven improvements can be achieved during the process of model design and optimization.

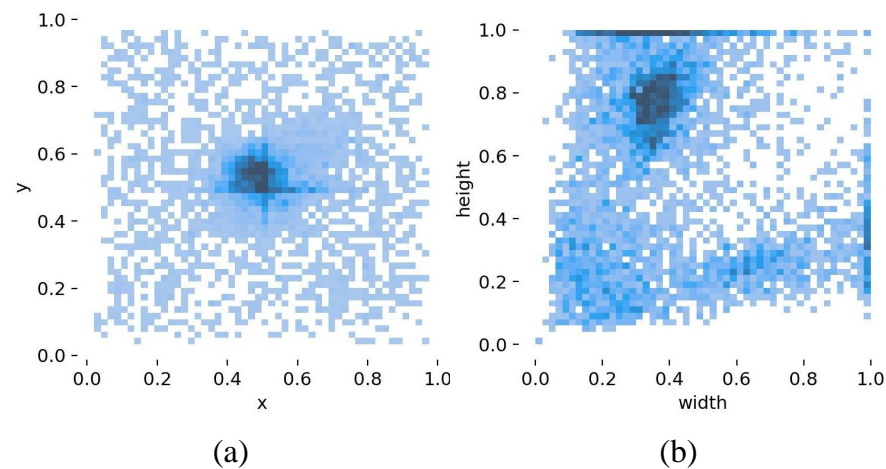


Figure 2. Dataset annotation file statistics and visualization charts. (a) Histograms of the dataset variables X and Y. (b) Histograms of the dataset variables width and height.

All images were manually annotated using the LabelImg tool (v1.8.6). Disease regions were marked using the largest possible horizontal bounding rectangles, and annotations were saved in PASCAL VOC XML format. Each image contains at least one annotated instance. To prevent model overfitting and enhance its generalization capability, data augmentation techniques were employed, primarily through online augmentation. The procedure involved (1) resizing all training images to 640×640 pixels, (2) applying random vertical and horizontal flips, and (3) normalizing the pixel values using the dataset mean and standard deviation. The annotation files were subsequently converted from the VOC XML format to the YOLO TXT format, forming the final input that is compatible with YOLO-based training frameworks.

2.2. YOLOv11 Convolutional Neural Network

YOLOv11 is a stable next-generation YOLO series model officially released by Ultralytics, represents one of the latest advancements in object detection models, and embodies the current state of the art (SOTA) in the field. Compared with earlier versions in the YOLO series, YOLOv11 achieves substantial improvements in both detection accuracy and inference efficiency. The overall architecture of the model is illustrated in Figure 3.

Building upon the core advantages of the YOLO series, YOLOv11 achieves a significant performance breakthrough through modular innovations and an end-to-end detection strategy. The architecture incorporates two key modules: C3K2 and C2PSA. The C3K2 module enhances local feature extraction by nesting and integrating the C2f and C3 structures and embedding a dual 3×3 convolutional kernel unit (C3k). The C2PSA module, based on the C2f framework, introduces an improved multi-head attention mechanism (PSA). This is achieved by replacing LayerNorm with activation-free convolutional layers and reconstructing the MLP using dual convolutional layers to form the PSABlock, thereby significantly improving global feature modeling capabilities.

Based on the end-to-end no-NMS design of YOLOv10, this paper enables YOLOv11 to adopt a dual-head detection strategy for joint optimization. The multi-positive matching branches enhance the supervision signal through multi-label allocation, and the one-to-one allocation branches that combined with depth-separable convolution, namely dsc, can promote more accurate classification and localization during the reasoning process. And YOLOv10 only retain one-to-one branches to eliminate post-processing delays.

YOLOv11 supports multi-scale deployment across five model sizes. By leveraging the synergy between the C3K2 and C2PSA modules, it reduces the number of parameters by 15%. On the COCO dataset, it achieves an AP of 47.6%, which is 4.2% higher than that of YOLOv8, while also increasing the inference speed by 22%. This model demonstrates strong robustness. In the scenario of small object detection, it thus provides an effective solution for real-time applications.

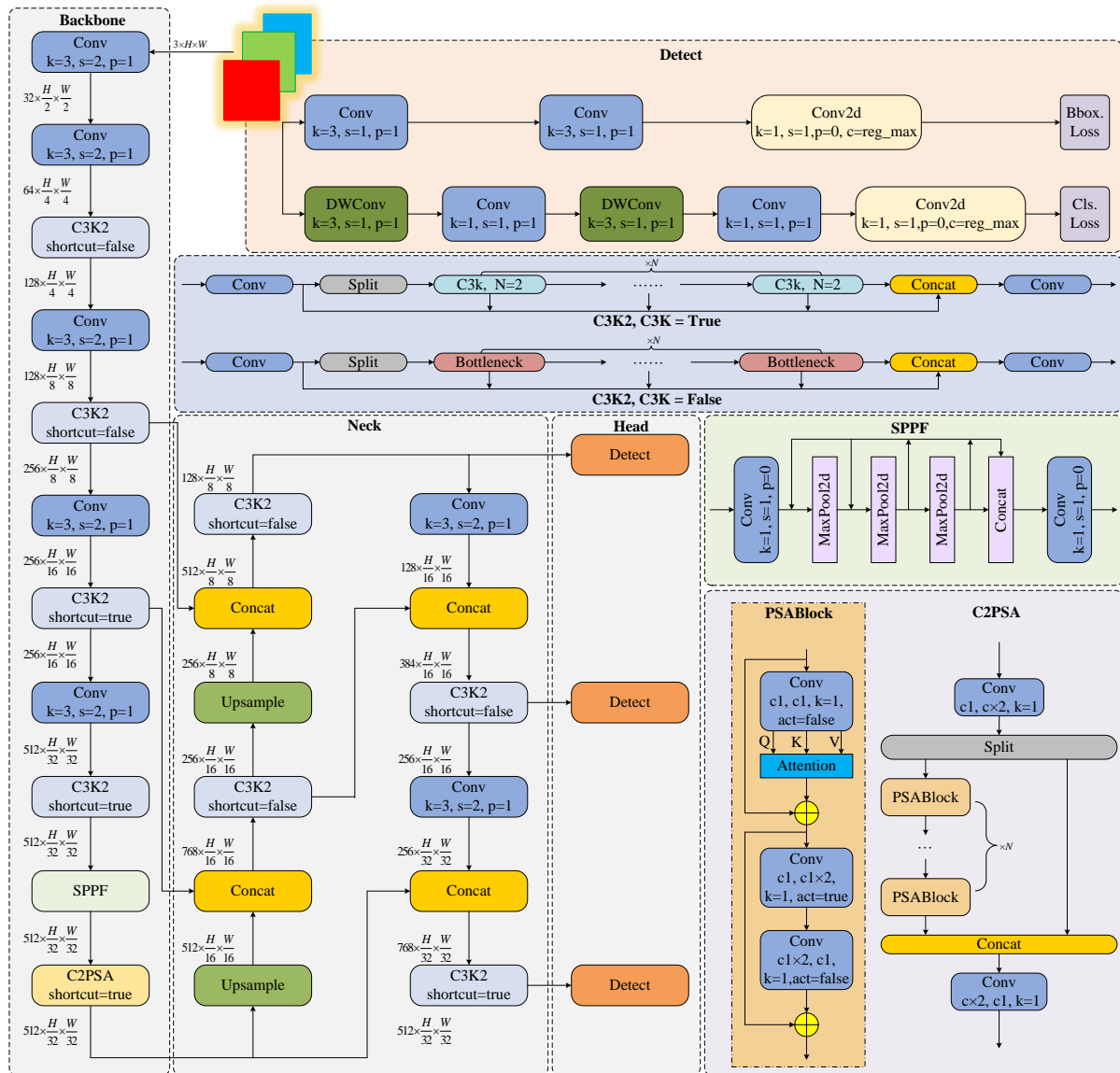


Figure 3. Architecture of YOLOv11.

2.3. YOLOv11 Network Improvements

To address the challenges faced in the efficient detection of wheat diseases in complex natural environments, this paper designs a framework called GDFC-YOLO. This framework integrates many advanced technologies to improve the detection accuracy. Just as shown in Figure 4, the dynamic feature extraction module GDFC is embedded in the YOLO backbone network. This is performed to enhance the extraction of disease characteristics. In the neck part of the model, the newly designed network structure will collaboratively optimize together with the Feature Enhanced Bottleneck module to refine the feature processing pipeline. This network component is called DF-Neck. In addition, the PIoUv2 loss function is used to solve the problem of unbalanced penalties for different object sizes existing in the traditional IoU loss function. This loss function also optimizes the regression path

of the boundary box, significantly improving the detection accuracy and the accuracy of the boundary box. With these innovative improvements, GDFC-YOLO has achieved high accuracy and efficiency in the real-world crop disease detection scenarios, providing strong support for intelligent agricultural management.

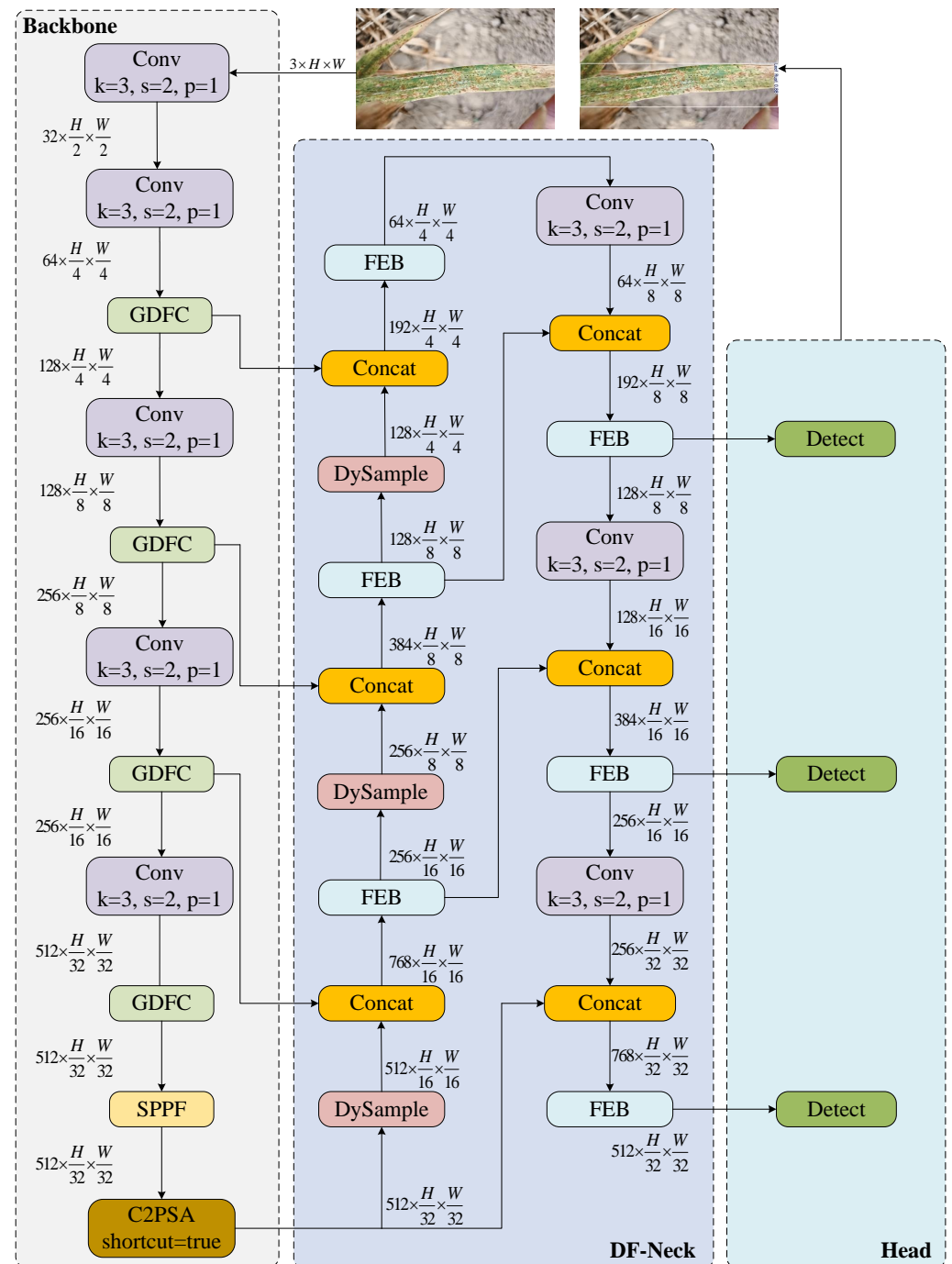


Figure 4. Architecture of GDFC-YOLO.

2.3.1. Ghost Dynamic Feature Core

Wheat disease detection tasks in natural backgrounds require high-precision feature extraction while facing the practical challenges of high computational demands and large model parameters. To overcome these limitations, this study proposes an innovative module design that integrates an improved dynamic convolution architecture [16] with Sequence-to-Sequence with Self-Attention (SCSA) attention mechanisms [17], named the GDFC module. By combining dynamic convolution, attention mechanisms, and lightweight design principles,

the GDFC module enhances the feature extraction capability and efficiency of the model, thereby improving disease detection accuracy and deployment adaptability.

The overall structure of the GDFC module is shown in Figure 5a. Firstly, the input feature map $X \in \mathbb{R}^{C \times H \times W}$ undergoes a 1×1 convolution for channel compression, resulting in an intermediate feature :

$$F_0 = \text{Conv}_{1 \times 1}(X), \quad (1)$$

The intermediate feature is split along the channel dimension into a retained branch A_0 and an enhanced branch B_0 . The enhanced branch is sequentially fed into N switchable submodules. When the switching parameter *Switch* is set to True, the C3k_Ghost Dynamic Feature Core Block (C3k_GDCBlock), a more complex structure, is used. Otherwise, the lightweight Ghost Dynamic Feature Core Block (GDCBlock) is employed. This recursive process can be expressed as follows:

$$B_i = \begin{cases} \text{C3k_GDCBlock}(B_{i-1}), & \text{Switch} = \text{True}, \\ \text{GDCBlock}(B_{i-1}), & \text{Switch} = \text{False}, \end{cases} \quad (2)$$

where the *Switch* parameter is a manually preset static flag (not input-adaptive), fixed during both training and inference. When higher accuracy is paramount and computational resources are unconstrained, set *Switch* = True, and the complex C3k_GDCBlock is activated. When a lightweight model is needed for edge deployment, set *Switch* = False, and the lightweight GDCBlock is activated.

Finally, the output obtained from each branch is connected in a channel dimension manner, and then fusion processing is carried out by using the 1×1 convolution method to restore the feature map to the feature map of the C channel:

$$Y = \text{Conv}_{1 \times 1}(\text{Concat}[A_0, B_1, \dots, B_N]), \quad (3)$$

Just as shown in Figure 5b, the C3k_GDCBlock proposed in this paper inherits the dual-branch architecture of the original C3 module. To elaborate, the input feature map X is compressed along the main branch to $M = \text{Conv}_{1 \times 1}(X)$, retaining the original semantic information. The remaining branches will rely on the two stages of the GDCBlock module in sequence. In this way, a richer context representation can be extracted:

$$R = \text{GDCBlock}(\text{GDCBlock}(\text{Conv}_{1 \times 1}(X))), \quad (4)$$

The outputs of the main branch and the remaining branches will eventually be connected along the channel dimension. After the connection, they are fused together through convolution operations:

$$Y = \text{Conv}_{1 \times 1}(\text{Concat}[M, R]), \quad (5)$$

This “preservation-improvement-fusion” design approach balances the retention of shallow features and the enhancement of deep features. That is, it retains shallow features while enhancing deep features. Its modular structure is beneficial for scalability and can make the overall design more scalable.

As shown in Figure 5c, the GDCBlock is the core part of the module mentioned above. It is composed of two parallel branches, namely the dynamic convolution branch and the depthwise separable convolution (DSC) branch. In the dynamic convolution branch, the input features will first be modulated with the help of the SCSA attention mechanism. The appearance of this mechanism is just like that shown in Figure 6. This mechanism integrates spatial attention, namely Shareable Multi-Semantic Spatial Attention (SMSA),

and channel attention, namely Progressive Channel-wise Self-Attention (PCSA), to carry out the refinement work of weighted features:

$$X' = W_s \odot (W_c \odot X), \quad (6)$$

where \odot denotes the element-wise multiplication, while W_s represents the spatial attention map generated by the SMSA module. The computation is defined as follows

$$W_s = \sigma(\text{GN}(\text{Concat}[\text{DSC} \text{Conv}_k(X_h), \text{DSC} \text{Conv}_k(X_w)])), \quad (7)$$

where X_h and X_w represent the sequences obtained by applying pooling operations along the height and width dimensions, respectively. $\text{DSC} \text{Conv}_k$ denotes the DSC with a kernel size of k . GN stands for group normalization, and σ refers to the Sigmoid activation function.

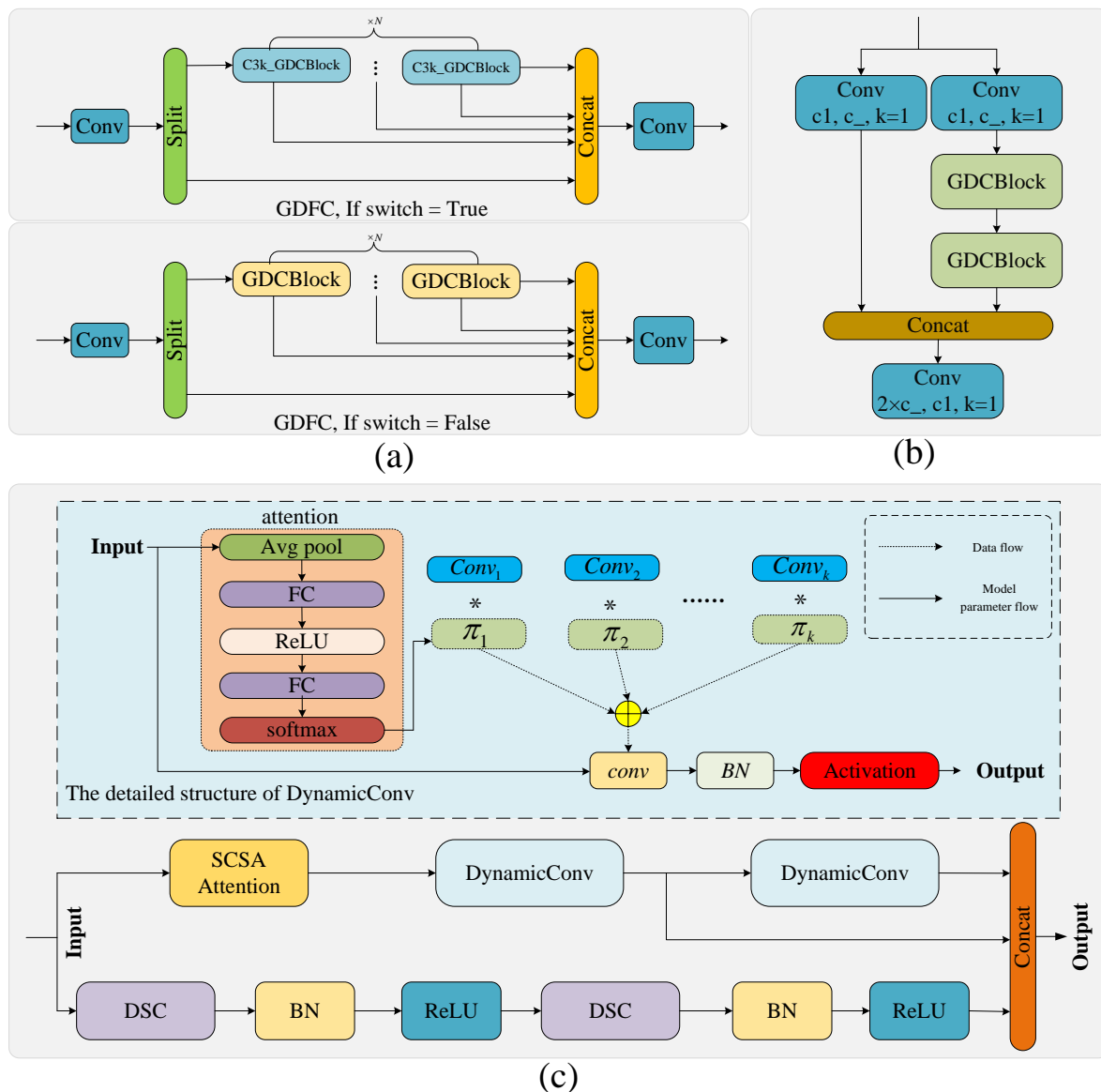


Figure 5. (a) Structure of the GDFC module within GDFC-YOLO. (b) Structure of the C3k_GDCBlock. (c) Structure of the GDCBlock.

The channel attention W_c is generated by the PCSA module, which utilizes global average pooling followed by a self-attention mechanism to compute the attention weight:

$$W_c = \sigma(GAP(Softmax(\frac{QK^T}{\sqrt{d}})V)), \quad (8)$$

where Q , K , and V represent the query, key, and value vectors, respectively. d denotes the scaling factor. GAP stands for global average pooling.

After the attention-weighted feature map is obtained, it is then fed into the Dynamic-Conv module to generate sample-adaptive convolution kernels. Specifically, the dynamic routing weight X is first generated from the input feature α as follows:

$$\alpha = \sigma(W_r \cdot GAP(X)), \quad (9)$$

where W_r represents the learnable parameter of the fully connected (FC) layer. The weight α is then used to perform a weighted aggregation over the K expert convolution kernel $\{W_i\}$, which is formulated as

$$W_d = \sum_{i=1}^K \alpha_i W_i, \quad (10)$$

Finally, the dynamic convolution kernel W_d is used to perform the following convolution operation:

$$X'' = Conv(X; W_d), \quad (11)$$

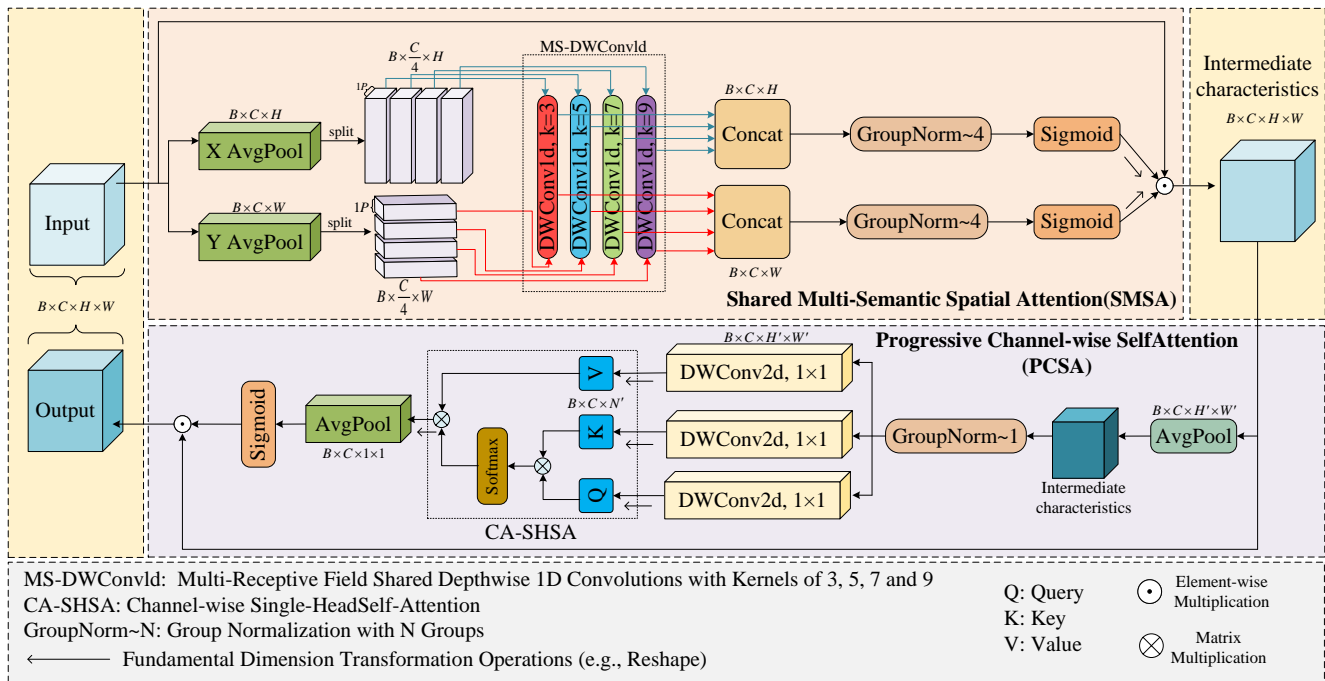


Figure 6. Structure of the SCSA attention mechanism.

The other branch consists of two layers of DSCs, as shown in Figure 7. It includes two consecutive depthwise + pointwise convolutions, reducing the parameter count from $O(C^2k^2)$ in the traditional convolution to $O(Ck^2 + C^2)$, thus significantly improving efficiency. Ultimately, the GDCBlock combines the outputs of the two branches by addition or concatenation to form richer features.

In summary, the GDFC module introduces a multi-scale context, attention mechanisms, and dynamic modeling through a switchable architecture, achieving a superior accuracy–speed trade-off in complex scenarios and enhancing the practicality and flexibility of the model.

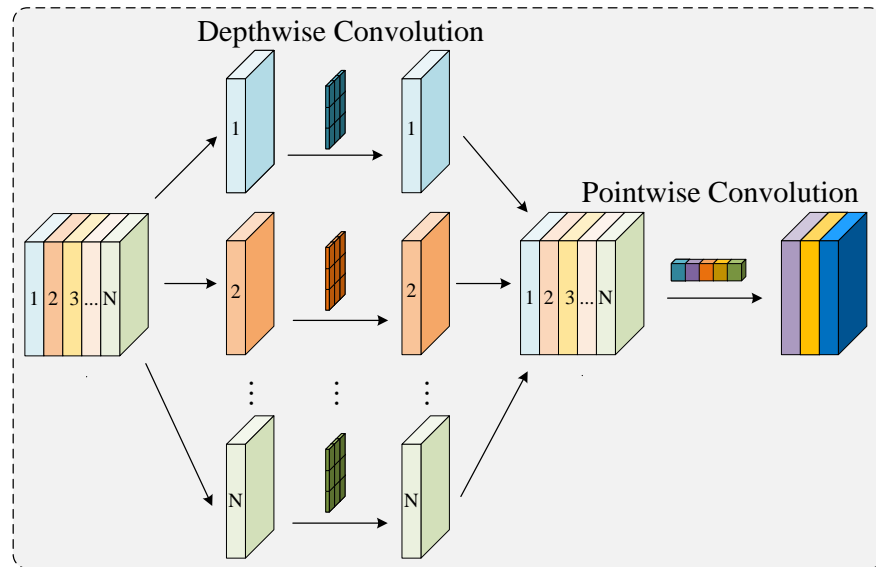


Figure 7. Detailed architecture of the depthwise separable convolution.

2.3.2. Neck Network Architecture and Feature Enhancement Module Co-Optimization

In the YOLO neck, a DF-Neck architecture was designed, which integrated the ultra-lightweight dynamic upsampling module DySample [18] and the first-level wavelet transform [19]. DF-Neck enhances the conventional feature pyramid by adding a shallow compensation pathway that injects high-resolution features from the second layer of the backbone into the neck, thereby improving the edge localization accuracy of small lesions. Within the parallel structure, DySample handles spatial reconstruction while the feature-enhanced bottleneck (FEB) focuses on frequency domain enhancement. The final output is a multi-scale feature fusion result. The overall architecture is illustrated in Figure 8.

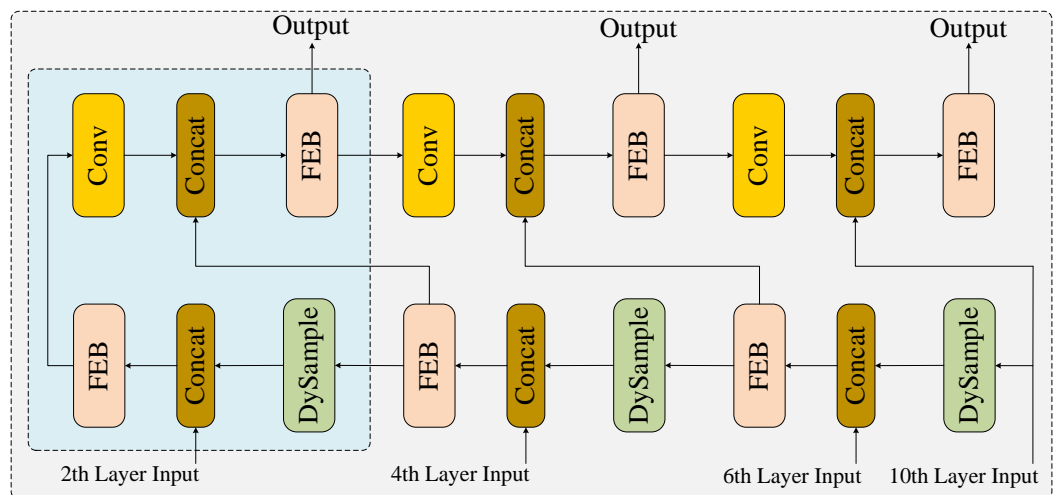


Figure 8. Schematic illustration of the DF-Neck architecture.

DySample is an ultra-lightweight dynamic upsampling module designed to preserve the fine details of high-resolution feature maps to the greatest extent while maintaining computational efficiency. Given an input feature map $X \in \mathbb{R}^{C \times H \times W}$ and an upsampling factor $r \in \mathbb{N}^+$, the offset tensor is first generated via a linear projection:

$$\Delta = W_{proj}(X), \quad (12)$$

where W_{proj} denotes the learnable projection matrix and $W_{proj} \in \mathbb{R}^{C \times (2r^2)}$, which maps each channel to $2r^2$ offset components. Subsequently, the PixelShuffle operation (denoted by $PS(\cdot)$) reorganizes Δ into the coordinate offset field $\Delta p = PS(\Delta) \in \mathbb{R}^{2 \times rH \times rW}$, corresponding to the displacements along the x and y directions for each upsampling point. Let the sampling grid coordinates for standard bilinear interpolation be $p \in \mathbb{R}^{2 \times rH \times rW}$; then the dynamic sampling locations are defined as:

$$p' = p + \Delta p, \quad (13)$$

Finally, in this paper, $Y = \text{grid_sample}(X', p')$ will be used to carry out the operation of feature resampling, and ultimately the output $Y \in \mathbb{R}^{C \times (rH) \times (rW)}$ will be obtained. Here, $\text{grid_sample}(\cdot)$ represents the result after the bilinear interpolation of X at the specified sampling coordinates.

Figure 9 compares two offset generation examples, namely the LP style and the PL style. In the LP style, an offset vector Δ of size $2r^2$ is first generated along the channel dimension, and then this offset vector is extended to the spatial dimension, that is, $\Delta p = PS(\Delta)$, with the help of PixelShuffle. In this way, fine-grained control can be achieved with relatively high parameter efficiency at the channel level. In the PL style, PixelShuffle first rearranges the feature mapping into a spatial representation of size $\frac{C}{r^2} \times rH \times rW$. Then, the linear layer W_{proj} directly generates the offset field Δp of the same dimension, creating a more lightweight solution that is suitable for deployment under highly resource-constrained conditions. Both of these styles have complementary advantages and can ensure efficient and flexible upsampling in different deployment environments.

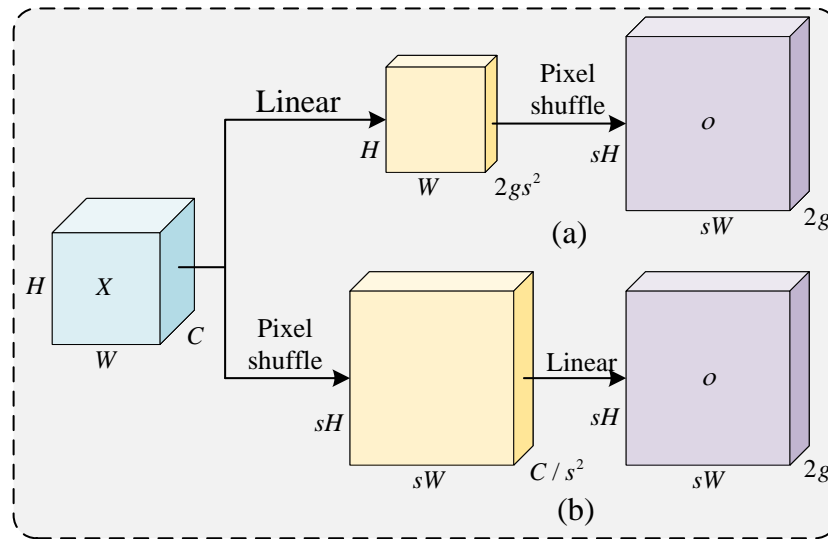


Figure 9. Offset generation styles in DySample. (a) Linear + pixel shuffle. (b) Pixel shuffle + linear.

Figure 10 provides a detailed illustration of the range-factor design: in the static variant, the offsets are uniformly scaled by a predefined constant $\alpha \in (0, 1]$:

$$\Delta p_{static} = \alpha \Delta p, \quad (14)$$

where the smaller the constant α , the closer the operation approximates the standard bilinear interpolation. The dynamic variant instead employs an additional linear layer W_{dyn} together with a Sigmoid activation $\sigma(\cdot)$ to produce a learnable scaling factor:

$$\begin{aligned} \beta &= \sigma(W_{dyn}(X)) \in (0, 1)^{2 \times rH \times rW}, \\ \Delta p_{dynamic} &= \beta \odot \Delta p, \end{aligned} \quad (15)$$

where \odot denotes the element-wise multiplication, which enables the adaptive amplification of displacements in detail-rich regions (e.g. along object edges) and attenuation in smooth regions, thereby balancing flexibility and consistency.

Furthermore, to further enhance the representational capacity, the channel dimension can be divided into g groups. g denotes the number of groups into which the channels of the feature map are evenly divided. To balance representational capacity and computational overhead, our dynamic module partition the channel dimension into g groups, independently generates offsets for each group, and completes adaptive upsampling via the `grid_sample` function. As Liu et al. [18] pointed out, when $g = 4$, an improvement of 0.8 mIoU in performance is observed compared to $g = 1$. Therefore, in this work, $g = 4$ is adopted for the standard (LP) architecture.

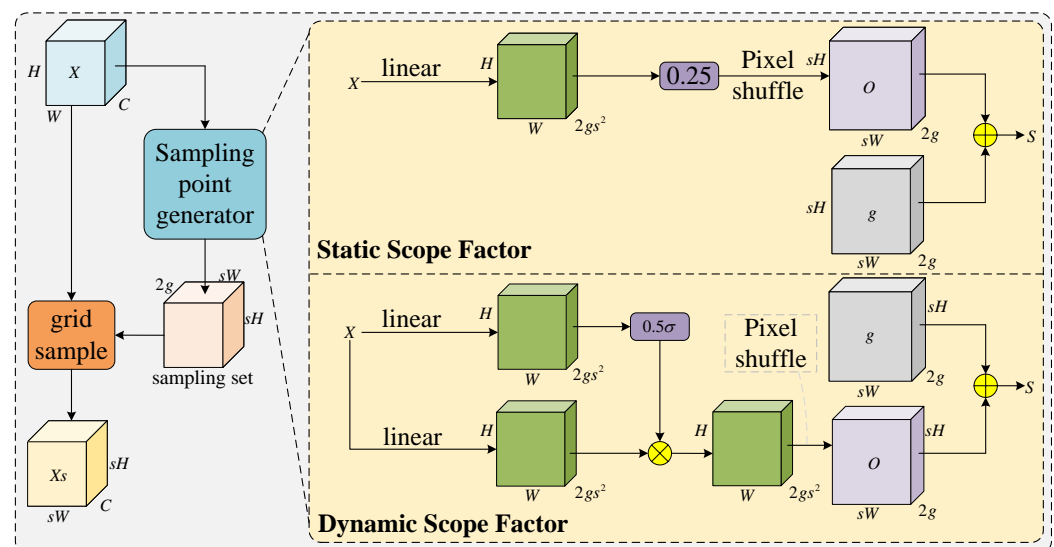


Figure 10. Detailed structural diagram of DySample.

Figure 11a illustrates the outer structure of the FEB module. FEB first uses a 1×1 convolution to reduce the channel dimension to C' , producing the feature F_0 . The feature F_0 is then evenly divided into M parts. After the first part is retained, the remaining parts sequentially pass through C3k_WaveBottleneck modules, cumulatively enhancing the features. Finally, the outputs are concatenated and pass through a convolution layer to restore the channel dimension.

Just as shown in Figure 11b, the C3k_WaveBottleneck module adopts the design concept of “retain–downsample–concatenate”. The input data will be divided into two branches along the channel dimension, one is the fast branch, while the other is the main branch. Here, for the main branch, it will rely on two consecutive WaveBottleneck blocks to carry out spatial downsampling and improvement operations.

Figure 11c presents the specific design of the WaveBottleneck block. This design combines the frequency decomposition ability of the wavelet transform with the efficiency of dsc. It adopts a parallel structure, integrating the main path and the remaining paths. Such a design has many advantages. It can retain the global structural information. It can also improve the response effect on high-frequency textures.

WTConv2d is the core component of the WaveBottleneck, as illustrated in Figure 12. The fundamental idea behind WTConv2d is to decompose the input into multiple frequency components using a first-level wavelet transform (WT). Lightweight convolutions are then applied separately on each frequency band, followed by the reconstruction of the output via an inverse wavelet transform (IWT).

Specifically, for the first-level wavelet transform, the procedure is as follows: the input feature map $X \in \mathbb{R}^{C \times H \times W}$ undergoes a first-level 2D Haar wavelet transform. Then, the Haar wavelet transform uses a fixed set of 2×2 filters to decompose the input into four frequency components:

$$[X_{LL}, X_{LH}, X_{HL}, X_{HH}] = WT(X), \quad (16)$$

where $WT(\cdot)$ denotes the Haar wavelet transform. The low-frequency component X_{LL} captures global structural information. The horizontal high-frequency component X_{LH} captures horizontal edges or texture variations. The vertical high-frequency component X_{HL} captures vertical edges or texture variations. The diagonal high-frequency component X_{HH} captures diagonal details. The Haar filter kernels are defined as follows:

$$\begin{aligned} f_{LL} &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, f_{LH} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \\ f_{HL} &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, f_{HH} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \end{aligned} \quad (17)$$

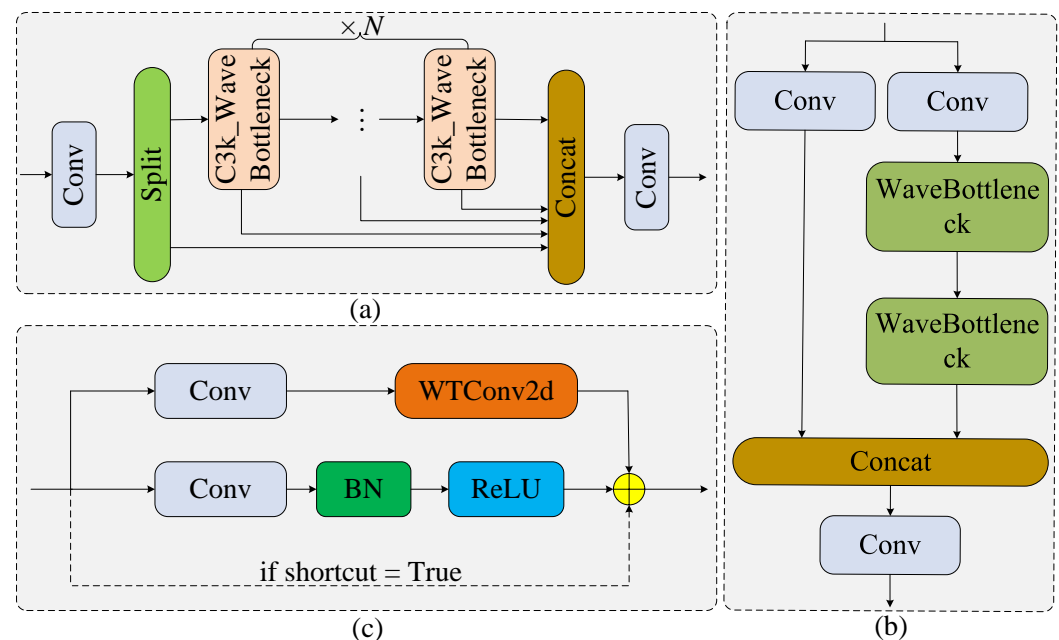


Figure 11. Architecture of the FEB. (a) Overall structure of the FEB, (b) architecture of the C3k_WaveBottleneck module, and (c) schematic illustration of the WaveBottleneck.

Each filter kernel operates on the non-overlappin 2×2 blocks of the input image, generating corresponding sub-band feature maps. Subsequently, DSCs are applied to these multi-frequency components in the wavelet domain. After convolution, the multi-frequency features are mapped back to the original spatial domain via the IWT. The IWT reconstructs the sub-band components to the original resolution. The filter kernels for the inverse transform are the transposes of the original filters, and the reconstruction process is implemented through a transposed convolution (also known as deconvolution), which is mathematically expressed as

$$X_{recon} = Conv - Transposed \left(\left\{ f_{LL}^T, f_{LH}^T, f_{HL}^T, f_{HH}^T \right\}, [X_{LL}, X_{LH}, X_{HL}, X_{HH}] \right), \quad (18)$$

Finally, to preserve local high-frequency information, the output of the conventional convolution branch is added to that of the wavelet-domain branch:

$$\text{Output} = \text{Conv}_{\text{base}}(X) \odot \beta + \text{IWT}, \quad (19)$$

where $\text{Conv}_{\text{base}}$ denotes the standard depthwise convolution, and $\beta \in \mathbb{R}^C$ represents the scaling factor applied to the base convolution.

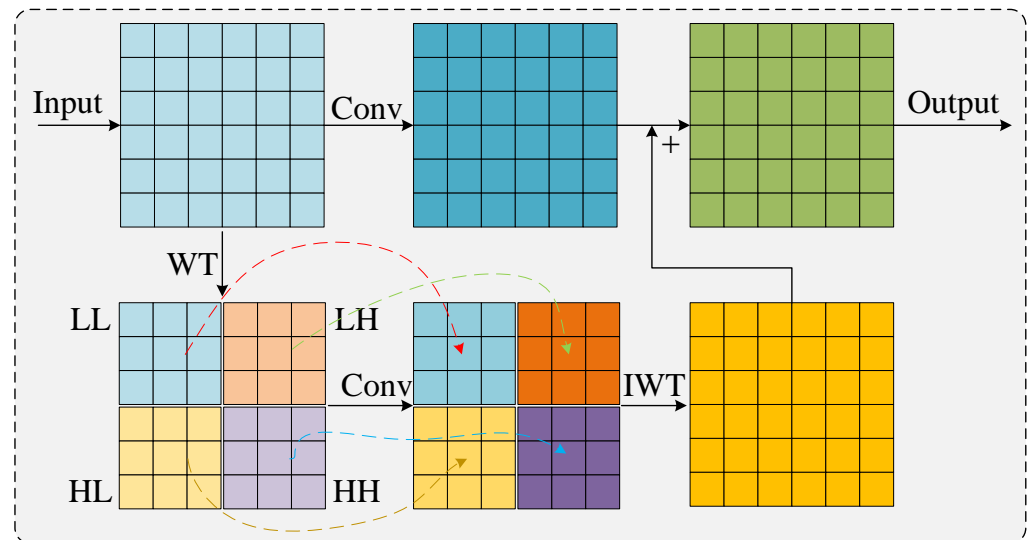


Figure 12. Detailed architecture diagram of WTConv2d with the single-level wavelet transform.

2.3.3. Powerful Intersection over Union v2

In object detection tasks, traditional IoU-based loss functions (such as GIoU [20], CIoU [21], EIoU [22], and SIoU [23]) often use the minimum enclosing box size (as illustrated in Figure 13a) or complex geometric factors (Figure 13b) as normalization terms. This approach causes the loss to decrease simply by enlarging the anchor box when there is no overlap with the target, resulting in a convoluted regression path and slow convergence. Furthermore, the penalty factors are not adaptive to the target size, making it difficult to suppress abnormal anchor box expansion. This issue significantly impairs localization accuracy, especially in wheat disease and pest detection under complex backgrounds.

To address the aforementioned issues, this study introduces a bounding box regression loss function based on PIoUv2 [24]. The core innovation lies in the incorporation of a size-adaptive penalty factor and a nonlinear gradient modulation mechanism. PIoU fundamentally eliminates anchor box inflation while adaptively adjusting to the box size of the ground truth. It is defined by three steps, as illustrated in Figure 13c.

(1) Size -Adaptive Penalty Factor

$$P = \frac{1}{4} \left(\frac{dw_1}{w_{gt}} + \frac{dw_2}{w_{gt}} + \frac{dh_1}{h_{gt}} + \frac{dh_2}{h_{gt}} \right), \quad (20)$$

where d_{w1} and d_{w2} represent the absolute distances between the predicted box's left and right edges and the corresponding edges of the ground truth box, respectively. d_{h1} and d_{h2} denote the absolute distances between the predicted box's top and bottom edges and the corresponding edges of the ground truth box, respectively. w_{gt} and h_{gt} correspond to the width and height of the ground truth box, respectively. Notably, $P \in [0, +\infty)$ depends solely on the ground truth box dimensions—anchor box enlargement does not affect P —thus preventing area inflation.

(2) Nonlinear Gradient Modulation Function

$$f(P) = 1 - e^{-P^2}, \quad (21)$$

Its gradient is

$$f'(P) = 2Pe^{-P^2}. \quad (22)$$

When $P \gg 1$ (extreme anchor boxes), $f'(P)$ is small, suppressing harmful gradients. When $P \approx 1$ (moderate quality), $f'(P)$ reaches its maximum, accelerating anchor box convergence. When $P \rightarrow 0$ (high quality), $f'(P)$ decreases, stabilizing the alignment.

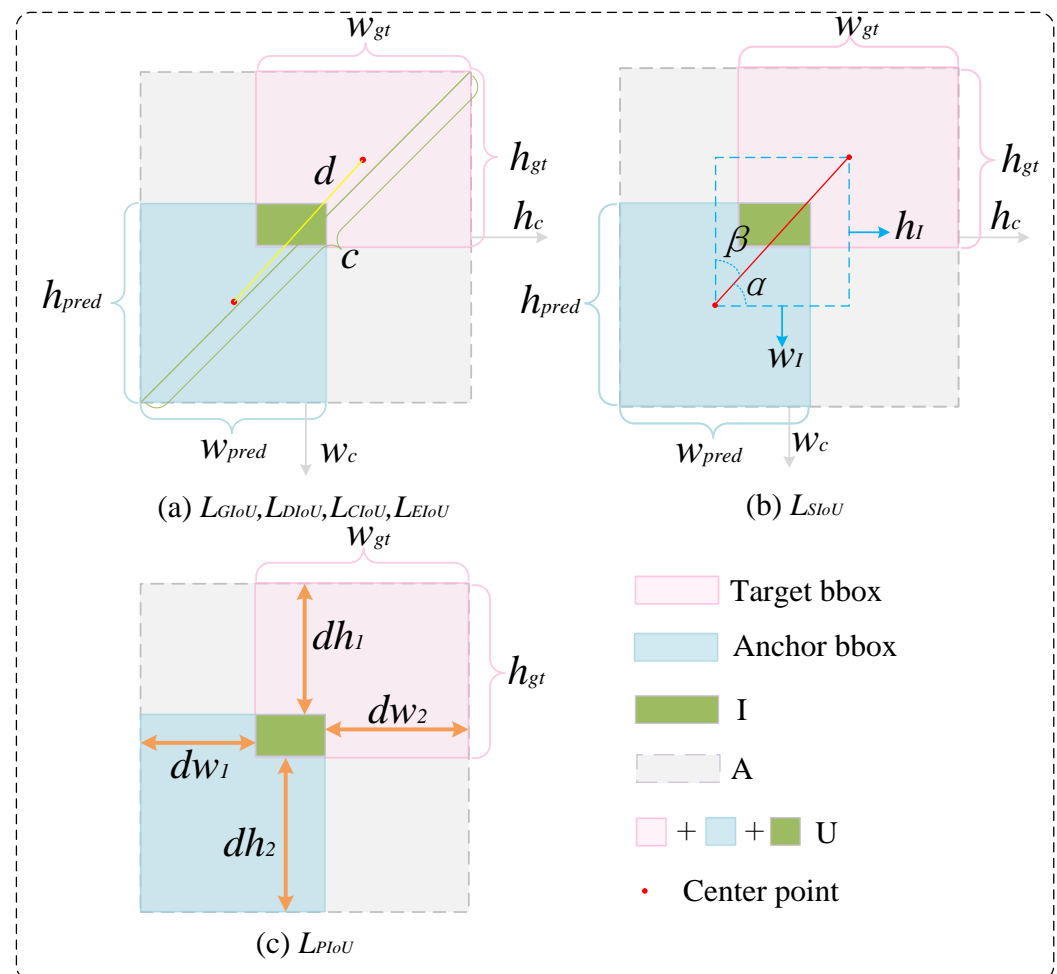


Figure 13. IoU-based loss functions. The loss functions in (a,b) utilize dimensional information—such as the diagonal length or area of the minimum enclosing box (gray dashed box) covering both the anchor and target boxes—as the denominator in the loss factor. In contrast, the PIoU loss in (c) employs only the side length of the target box as the denominator in its loss factor.

(3) Definition of PIoU Loss

The “Powerful IoU” metric is introduced:

$$PIoU = IoU - f(P), \quad -1 \leq PIoU \leq 1, \quad (23)$$

The final PIoU loss is defined as

$$\mathcal{L}_{PIoU} = 1 - PIoU = 1 - IoU + f(P), \quad 0 \leq PIoU \leq 2, \quad (24)$$

where $IoU = \frac{I}{U}$ and I represent the intersection areas between the predicted box and the ground truth box, while U denotes their union area. To further address the issue of imbalanced gradient contributions caused by varying training sample quality, PIoUv2 introduces a non-monotonic attention mechanism based on PIoU. This mechanism focuses primarily on medium-quality anchors while suppressing the negative impact of low-quality samples. The quality metric is defined as

$$q = e^{-P}, q \in (0, 1], \quad (25)$$

The non-monotonic attention function is defined as

$$u(\lambda q) = 3(\lambda q)e^{-(\lambda q)^2}, \quad (26)$$

Therefore, the final PIoUv2 loss is defined as

$$\mathcal{L}_{PIoU_{v2}} = u(\lambda q)\mathcal{L}_{PIoU} = 3\lambda qe^{-(\lambda q)^2}(1 - IoU + f(P)), \quad (27)$$

where λ is the sole tuning hyperparameter that controls the peak position of the attention function. q represents the similarity metric between the anchor box and the target box, where a q value closer to 1 represents higher quality. The function curve of $u(\lambda q)$ exhibits a non-monotonic shape, reaching its maximum response at intermediate values, thereby assigning the greatest optimization weight to the anchor boxes of moderate quality.

2.4. Model Training and Testing

2.4.1. Test Environment and Parameter Settings

The experiments were conducted on Windows 10 with hardware consisting of an NVIDIA GeForce RTX 3060 GPU (12 GB VRAM), an Intel Core i5 CPU, and 16 GB of RAM. The software environment was established based on Python 3.10.14, PyTorch 2.1.2+cu118, and CUDA 11.8. The experiments utilized the YOLOv11s framework. The training parameters included an input resolution of 640×640 , batch sizes of 16 or 32, an initial learning rate of 0.01, and a total of 300 training epochs. To evaluate the best-performing model rather than the final checkpoint during training, the model with the lowest validation loss, saved as best.pt, was used for testing.

2.4.2. Evaluation Indicators

When evaluating the performance of the detection model, a series of key metrics are computed to reflect the model's effectiveness in identifying target objects. Typically, these metrics are calculated based on the four fundamental components of the confusion matrix: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). In this study, a standardized evaluation framework encompassing the following metrics was employed.

- (1) Precision: Precision measures the proportion of correctly detected instances among all detected instances. It is defined as the ratio of TPs to the total number of detections. It is formulated as

$$P = \frac{TP}{TP + FP} \quad (28)$$

- (2) Recall: Recall primarily evaluates the model's ability to detect all relevant instances within the test dataset. It is defined as the ratio between the number of correctly

detected instances and the total number of actual instances present. Its computation is given by

$$R = \frac{TP}{TP + FN} \quad (29)$$

- (3) mAP@0.5: The mean average precision (AP) at an IoU threshold of 0.5 represents the average of the AP values computed for all categories, and if the IoU between the predicted bounding box and the ground truth box is greater than or equal to 0.5, the detection is considered correct. This metric provides a comprehensive assessment of the model's overall detection capability under a relatively lenient localization requirement. The calculation is defined as

$$\text{mAP@0.5} = \frac{1}{N} \sum_{c=1}^N AP_c \quad (30)$$

where N denotes the total number of categories and AP_c represents the precision for category c . The metric AP is obtained by integrating the precision–recall (P-R) curve, which is computed as the area under the curve: $AP = \int_0^1 P(R) dR$.

- (4) mAP@[0.5:0.95]: It refers to the mAP computed across multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. This metric comprehensively evaluates the model's performance under varying localization precision requirements. A higher value indicates that the target object is localized more accurately. Its calculation is defined as

$$\text{mAP@[0.5 : 0.95]} = \frac{1}{N} \sum_{c=1}^N \left(\frac{1}{10} \sum_{k=1}^{10} AP_c^{(k)} \right) \quad (31)$$

where N denotes the total number of categories and $AP_c^{(k)}$ represents the precision of category c at the k IoU threshold. AP is computed by integrating the P-R curve, namely by calculating the area under the curve: $AP = \int_0^1 P(R) dR$.

3. Results and Analysis

3.1. GDFC Experiments

To verify that the GDFC module can achieve high-precision feature extraction in natural backgrounds and to demonstrate the model's ability to improve feature extraction accuracy while controlling computational complexity and parameter count, a comparative study was conducted on the wheat disease dataset. The comparison results are presented in Tables 2 and 3.

This paper finds that the computing efficiency of the GDFC module has been significantly improved. The specific data can be seen in Table 2. The number of GFlops has decreased from the original 21.6 to 19.9, which is a reduction of approximately 7.9%. The number of parameters has also decreased from 9.43 M to 8.89 M, with a reduction ratio of 5.7%. And the size of the model file was reduced from 18.3 MB to 17.5 MB, a reduction of 4.4%. From these results, it can be seen that this method not only improves the detection performance but also reduces the complexity of the calculation and the requirements for storage, thereby enhancing its practicability.

As indicated in Table 2, the introduced GDFC module results in a latency trade-off, with Frames Per Second (FPS) decreasing from 150.7 to 68.79, attributed to its sequential dynamic convolution operations. Nevertheless, this design offers a crucial balance for agricultural disease detection. The 4.1% increase in precision reduces false-positive diagnoses in field applications, where misclassifying healthy plants as diseased is more costly than missed detections. Moreover, despite increased latency, the reduction in GFLOPs by 7.9% and model size by 4.4% improves deployability on edge devices. Additionally, with a

throughput of 68.79 FPS, the model exceeds the minimum 15 FPS requirement for real-time agricultural detection systems, facilitating seamless deployment in latency-sensitive field operations such as drone-based monitoring and automated scouting.

Regarding the choice of recursion depth N , Table 2 shows that increasing N steadily raises the computational cost while sharply reducing inference speed. When $N = 1$, GFLOPs drop from 21.6 to 19.9, the model size shrinks by 4.4%, and the model still runs at 68.79 FPS. When $N = 2$, GFLOPs rise to 20.9, the model file nearly doubles, and FPS falls to 45.46. When $N = 3$, GFLOPs climb further to 21.9, the model size grows to 39.2 MB, and FPS sinks to 33.73.

Meanwhile, Table 2 confirms that the marginal mAP@0.5 improvements for $N = 2$ and 3 are negligible compared to the increase in GFLOPs, Params, and model file size. In particular, $N = 2$ incurs a 33.3% drop in FPS and almost doubles the model size versus $N = 1$, making it unsuitable for real-time deployment on resource-constrained edge devices. Therefore, $N = 1$ is adopted as the default recursion depth for the GDFC block.

Table 2. Computational efficiency of the GDFC module on the wheat disease dataset.

Model	GFLOPs	Params (M)	Model File Size	FPS	mAP@0.5
Bsaeline	21.6	9.43	18.3 MB	150.7	0.860
GDFC ($N = 1$)	19.9	8.89	17.5 MB	68.79	0.886
GDFC ($N = 2$)	20.9	9.48	36.8 MB	45.46	0.897
GDFC ($N = 3$)	21.9	10.07	39.2 MB	33.73	0.887

Compared with the baseline YOLOv11s model, integrating the GDFC module into the backbone network leads to a significant performance improvement, as shown in Table 3. Its precision (P) increases from 0.853 to 0.888, representing a gain of 4.1%. Although its recall (R) shows a marginal decrease from 0.825 to 0.819, the significant gains in precision and the more comprehensive mAP overshadow this minor dip. The slight dip in recall is likely a consequence of the precision–recall trade-off. The focus of the GDFC module on enhancing feature discrimination may have made the model marginally more selective, favoring higher confidence detection, which may increase model precision but can occasionally induce the missing of borderline TPs. The mAP@0.5 increases from 0.860 to 0.886, yielding an increase of 3.0%. The mAP@[0.5:0.95] rises from 0.681 to 0.696, reflecting an increase of 2.2%.

Table 3. Object detection performance of the GDFC module on the wheat disease dataset.

Model	P	R	mAP@0.5	mAP@[0.5:0.95]
Bsaeline	0.853	0.825	0.860	0.681
GDFC	0.888	0.819	0.886	0.696

3.2. DF-Neck Experiments

As illustrated in Figure 14, to validate the enhancement capability of the FEB module within DF-Neck for irregular small lesion texture features, HiResCAM heatmaps were generated on the final-layer feature maps of the neck for the same wheat disease image using both the baseline model (which included C3K2 and standard upsampling) and DF-Neck. The qualitative visualization results indicate that the baseline model exhibits relatively sparse activation in regions corresponding to irregular lesion edges, whereas the heatmap of the improved model demonstrates significantly higher responses along lesion edges and minute spots.

As shown in Figure 15, the average P–R curves across all categories on the entire wheat disease test set are presented for both the baseline model and the DF-Neck module. It is evident that DF-Neck achieves a higher precision and recall at all confidence thresholds.

The overall AP increases from 0.860 to 0.886. Moreover, the DF-Neck curve shifts upwards and to the right, closely hugging the top-right corner, with notable gains in the medium-to-high recall region. These results convincingly demonstrate the effectiveness of the DySample upsampling structure and the FEB feature enhancement module in improving the spatial information reconstruction of lesions and enhancing detection performance.

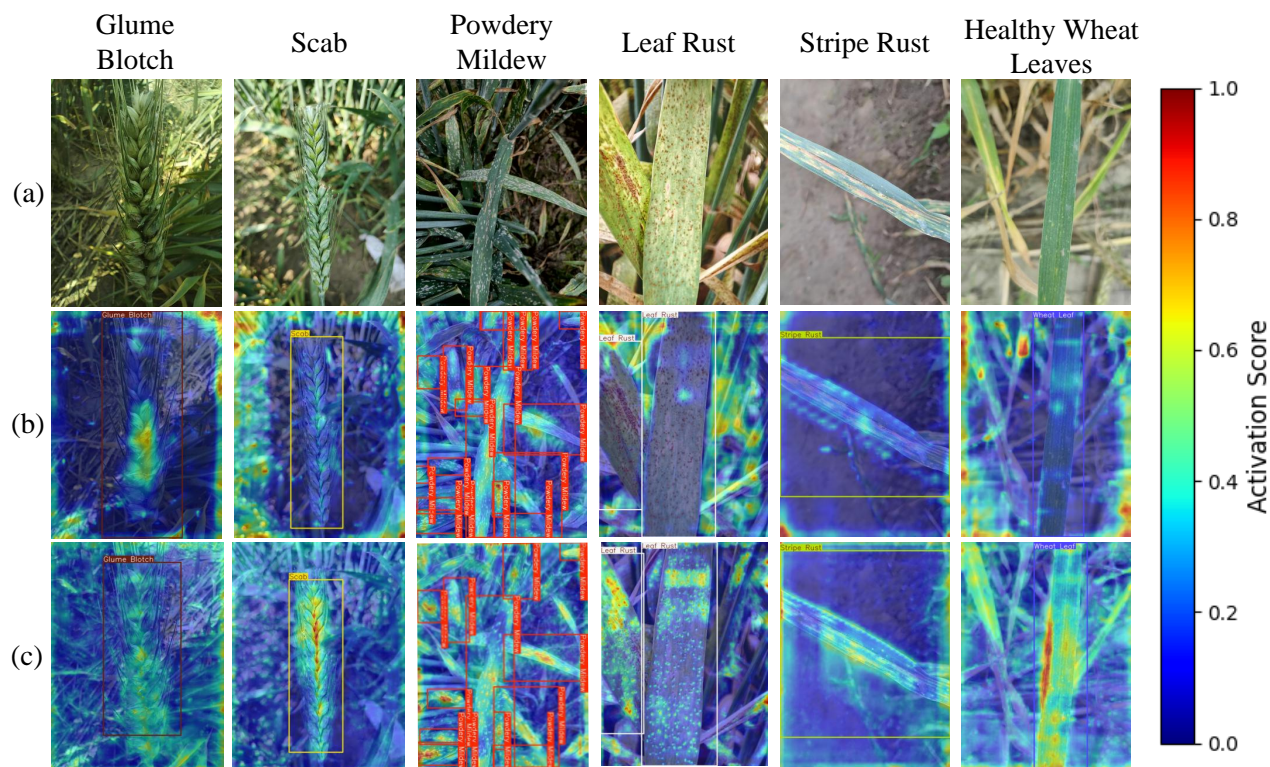


Figure 14. Comparison of HiResCAM heatmaps between the baseline model and the proposed DF-Neck module on the wheat disease dataset. (a) Original image. (b) Heatmap generated by the baseline model. (c) Heatmap generated by DF-Neck.

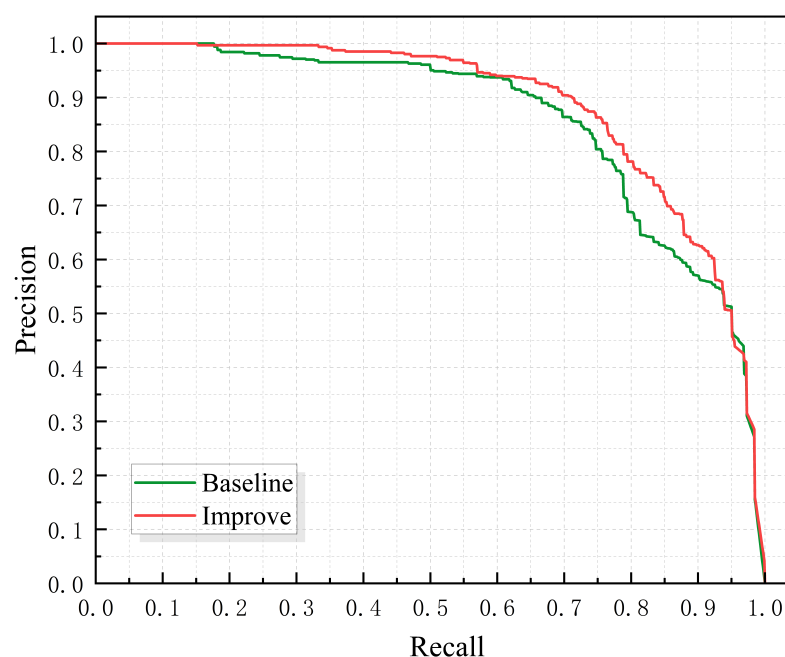


Figure 15. Average P–R curves of all categories on the wheat disease test set for the baseline model and the proposed DF-Neck module.

This paper aims to verify the overall performance of the DF-Neck module. Therefore, on the wheat disease dataset, it was compared with the baseline model. The results obtained after the comparison can be seen in Table 4.

Table 4. Object detection performance of the DF-Neck module on the wheat disease dataset.

Model	P	R	mAP@0.5	mAP@[0.5:0.95]
Bsaeline	0.853	0.825	0.860	0.681
DF-Neck	0.891	0.816	0.886	0.672

Compared with the baseline model, the introduction of the DF-Neck module has significantly improved the detection performance. The original P-value was 0.853, and now it has increased to 0.891, an increase of approximately 4.5%. The mAP@0.5 has increased from 0.86 to 0.886, an increase of 3.0%. This indicates that this model has advantages in reducing false detection and improving the ability of medium-precision detection. R decreased from 0.825 to 0.816, and the mAP@[0.5:0.95] decreased from 0.681 to 0.672. These decreases are within an acceptable range and have little impact on the overall detection performance. Experimental results demonstrate that by optimizing multi-scale feature fusion, the DF-Neck module improves comprehensive detection accuracy. This enhancement enables the network to prioritize high-confidence targets while suppressing ambiguous boundary cases, although it incurs a slight reduction in recall rate for some samples. In agricultural disease detection scenarios, this approach effectively balances high precision with low recall rates.

3.3. PIoUv2 Experiments

To validate the effectiveness of the proposed PIoUv2 regression loss function in terms of the convergence speed and regression accuracy, this study conducted comparative experiments with mainstream IoU-based loss functions, as illustrated in Figure 16. This value of hyperparameter λ was determined through systematic optimization by Liu et al. [24], and the PIoUv2 model works best when $\lambda = 1.3$, demonstrating robustness and generalizability. Therefore the λ of the PIoUv2 loss function was set to 1.3 in this study. Our analysis was carried out from the following two perspectives:

(1) Convergence speed analysis

The results show that for the val/box_loss metric, PIoUv2 reached its optimal value as early as epoch 122, while for the mAP@0.5 metric, PIoUv2 reached its best performance as early as epoch 115. As for the mAP@[0.5:0.95] metric, its best advantage is that it occurs at epoch 134. Also, PIoUv2 reached its peak at epoch 91. Only the precision metric is an exception. For this metric, the performance of EIoU is slightly better. It reached its best state at epoch 219. Overall, PIoUv2 shows a faster convergence speed in most indicators. It can achieve stable performance earlier than those competing loss functions, which indicates that it is superior in terms of training convergence efficiency.

(2) Regression Accuracy Analysis

From the perspective of the final regression accuracy, PIoUv2 also demonstrates a very prominent advantage. By comparing the end-of-training metrics including mAP@0.5, mAP@[0.5:0.95], accuracy, and recall, it is easy to see that PIoUv2 performs well in key metrics such as mAP@0.5, mAP@[0.5:0.95], and recall, all achieving the highest scores. Although its accuracy is slightly lower than that of EIoU, the difference is very small. Its overall performance is still superior. During the entire training process of PIoUv2, its fluctuation is relatively small, the curve is smoother, and its regression characteristics are more stable. In order to comprehensively compare the detection

accuracy of different regression loss functions, Table 5 summarizes the key metrics obtained after training each method: mAP@0.5, mAP@[0.5:0.95], precision, and recall.

Table 5. Final performance comparison of different regression loss functions.

Experiments	P	R	mAP@0.5	mAP@[0.5:0.95]
CIoU	0.853	0.825	0.860	0.681
EIoU	0.923	0.793	0.866	0.667
SIoU	0.901	0.792	0.868	0.675
DIoU	0.878	0.815	0.876	0.674
GIoU	0.885	0.801	0.869	0.677
PIoUv2	0.916	0.805	0.885	0.692

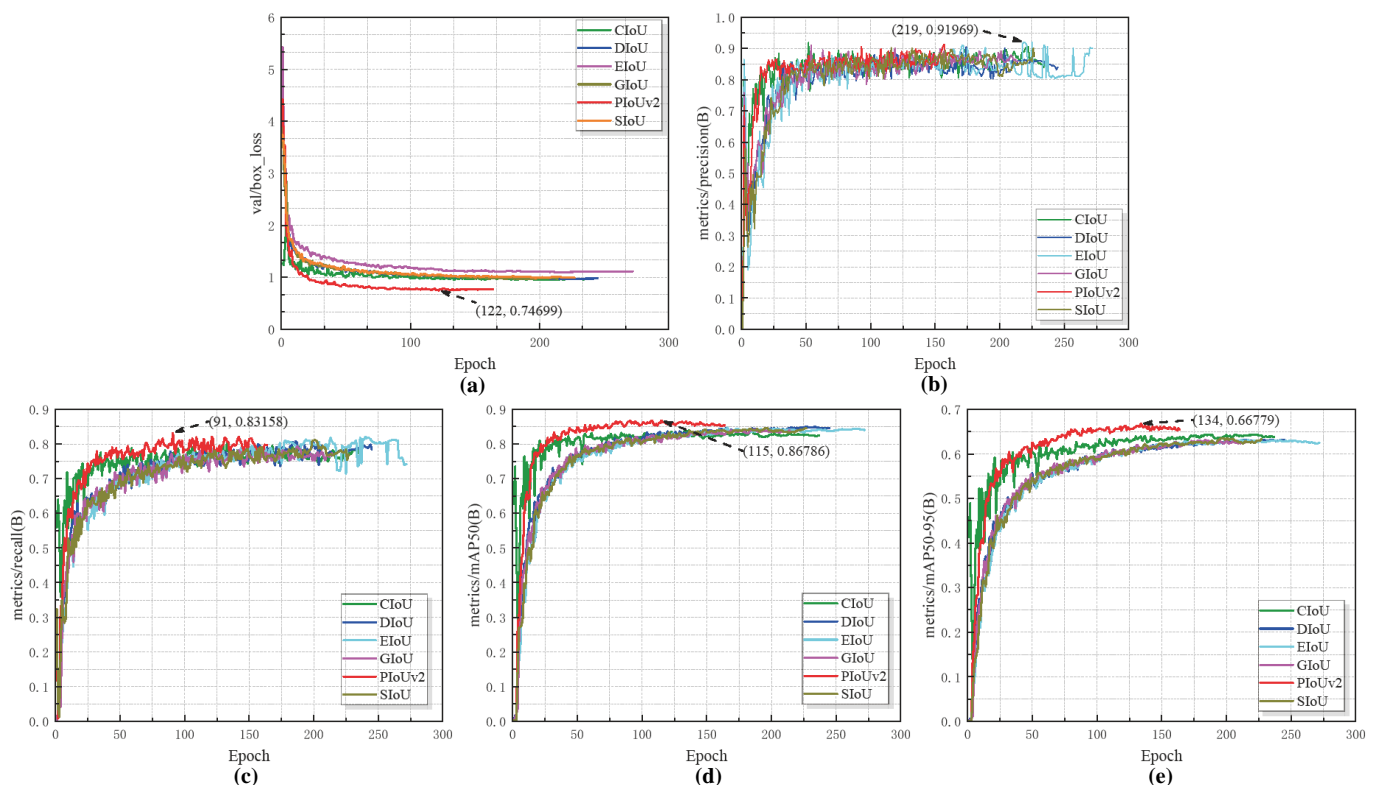


Figure 16. Comparison between PIoUv2 and mainstream IoU loss functions in regression performance and convergence speeds. (a) Comparison of validation box regression losses. (b) Comparison of the precision metric. (c) Comparison of the recall metric. (d) Comparison of the mAP@0.5 metric. (e) Comparison of the mAP@[0.5:0.95] metric.

As can be seen from Table 5, PIoUv2 achieved the best results in both the mAP@0.5 and mAP@[0.5:0.95] metrics, with mAP@0.5 reaching 0.885 and mAP@[0.5:0.95] reaching 0.692. In terms of recall, PIoUv2 performed relatively stably. Its recall rate value is 0.805. Although this value is slightly lower than 0.825 of CIoU and 0.815 of DIoU, it is still at a relatively high level. These results clearly show that in terms of the regression accuracy of boundary boxes, PIoUv2 is superior to traditional loss functions such as GIoU, DIoU and CIoU. It performs better.

The precision of PIoUv2 is 0.916, which is slightly lower than that of EIoU (0.923), but overall, PIoUv2 strikes a favorable balance between precision and recall, effectively balancing detection accuracy and completeness.

These findings further validate the effectiveness of PIoUv2 in practical object detection tasks: it not only accelerates model convergence but also achieves comprehensive

improvements in final performance, demonstrating a strong generalization ability and practical value.

3.4. Ablation Study

To evaluate the specific contribution of each improved module to the overall model performance, several ablation experiments were conducted on the wheat disease dataset, as shown in Table 6.

Firstly, after the GDFC module is introduced, the parameter count of the model decreases to 8.89 M (a reduction of 5.7%), and its computational cost drops to 19.9 GFLOPs (a reduction of 7.9%), while the P increases by 4.1% to 0.888, demonstrating the effectiveness of the lightweight feature fusion design. Secondly, the collaborative optimization of the neck network structure and the feature enhancement module (DF-Neck) further increases its precision to 0.891 by strengthening local feature extraction capabilities. However, due to the increased architectural complexity, the parameter count rises to 9.81 M, indicating a trade-off that requires a balance with other modules. By further adopting the PIoUv2 localization loss function, the P metric significantly increases to 0.916 (+7.4%), and the mAP@[0.5:0.95] metric returns to 0.692, indicating a substantial improvement in bounding box regression accuracy. Finally, when all three modules are jointly optimized, the model achieves the best overall performance, with the mAP@0.5 metric surpassing 0.9 (+4.7%) and parameters controlled at 9.27 M, validating the complementarity of global feature fusion, lightweight design, and precise loss optimization. These experiments demonstrate that the proposed modules, by optimizing feature representation, reducing computational redundancy, and enhancing localization capabilities, provide a high-precision and efficient solution for wheat disease detection.

Table 6. Ablation study results.

Baseline	GDFC	DF-Neck	PIoUv2	P	R	mAP@0.5	mAP@[0.5:0.95]	Params (M)	GFLOPs
✓				0.853	0.825	0.860	0.681	9.43	21.6
✓	✓			0.888	0.819	0.886	0.696	8.89	19.9
✓		✓		0.891	0.816	0.886	0.672	9.81	24.3
✓			✓	0.916	0.805	0.885	0.692	9.43	21.6
✓	✓	✓		0.872	0.829	0.894	0.691	9.27	22.5
✓	✓		✓	0.870	0.835	0.889	0.693	8.89	19.9
✓		✓	✓	0.893	0.813	0.889	0.697	9.81	24.3
✓	✓	✓	✓	0.899	0.821	0.900	0.695	9.27	22.5

3.5. Model Comparison Experiments

To validate the overall performance of the wheat disease detection model proposed in this study, some comparative experiments were conducted under identical experimental conditions against mainstream algorithms. The results are presented in Table 7.

Compared with two-stage detectors, the proposed model demonstrates significant advantages in lightweight design and computational efficiency. Although Cascade R-CNN achieves the highest mAP@[0.5:0.95] (0.711), its parameter count (69.17 M) and computational cost (99.4 GFLOPs) are approximately 7.5 and 4.4 times those of the proposed model, respectively, making it impractical for real-world deployment. Among single-stage models, YOLOv12s attains a precision (P) of 0.919 with 9.1 M parameters and 19.6 GFLOPs, but its mAP@0.5 (0.872) and mAP@[0.5:0.95] (0.669) are both lower than those of our model (0.9 and 0.695), indicating that the improvements in our modules effectively enhance multi-scale disease feature representation.

A further comparison among YOLO variants shows that the mAP@0.5 of our model increases by 4.7% compared with YOLOv11s (from 0.86 to 0.9), and the parameter count decreases by 0.16 M (from 9.43 M to 9.27 M). The computational cost (22.5 GFLOPs) is

also notably lower than that of YOLOv10s (24.8 GFLOPs) and YOLOv8s (23.6 GFLOPs). Notably, our model maintains a high recall rate ($R = 0.821$) while achieving a precision score ($P = 0.899$) superior to most compared models, only slightly lower than YOLOv12s (0.919) and YOLOv9s (0.911). However, its $mAP@[0.5:0.95]$ (0.695) surpasses that of YOLOv12s (0.669) and YOLOv9s (0.667), validating the synergistic optimization effect of global dynamic feature fusion and precise localization loss.

Table 7. Comparison results with other state-of-the-art algorithms on the wheat disease dataset.

Method	P	R	mAP@0.5	mAP@[0.5:0.95]	Params (M)	GFLOPs
rtdetr	0.871	0.798	0.851	0.645	42.78	130.5
Cascade R-CNN	0.889	0.835	0.886	0.711	69.17	99.4
Faster R-CNN	0.882	0.843	0.888	0.710	41.38	71.6
RetinaNet	0.865	0.815	0.859	0.652	19.90	46.8
SSD	0.597	0.651	0.594	0.341	24.55	105.5
YOLOv12s	0.919	0.780	0.872	0.669	9.10	19.6
YOLOv11s	0.853	0.825	0.860	0.681	9.43	21.6
YOLOv10s	0.895	0.785	0.857	0.671	8.07	24.8
YOLOv9s	0.911	0.797	0.871	0.667	6.32	22.7
YOLOv8s	0.894	0.789	0.868	0.664	9.84	23.6
YOLOv6s	0.887	0.799	0.867	0.667	15.99	43.0
YOLOv5s	0.889	0.806	0.871	0.677	7.83	19.0
Ours	0.899	0.821	0.900	0.695	9.27	22.5

The results of the experiment clearly show that this model has achieved the best balance in terms of accuracy and efficiency and can provide a reliable solution for the real-time detection of wheat diseases under complex field environmental conditions.

As shown in Figure 17, in this paper, $mAP@0.5$ is used as an evaluation metric. In the test set, the proposed GDFC-YOLO model is compared with the current mainstream object detection models. The results show that GDFC-YOLO achieves an AP as high as 0.900. This value is better than all the models used for comparison. Among these comparative models, there is the baseline YOLOv11 model, whose AP is 0.860, as well as other YOLO variants, whose AP ranges from 0.857 to 0.872. Additionally, there are models like Faster R-CNN, whose AP is 0.888, and Cascade R-CNN. The AP is 0.886 for these classic detectors. Compared with the baseline model, the AP of GDFC-YOLO has increased by 0.040, and its P-R curve has shifted towards the upper-right corner, getting closer to the upper-right corner of the coordinate system. This indicates that this model has achieved a better balance between the accuracy and recall rate at different confidence thresholds. Within the medium and high recall range, the benefits it has obtained are very prominent. From these results, it can be fully proved that the GDFC-YOLO model proposed in this paper has superior performance when detecting targets in the natural disease environment of wheat.

The analysis of the confusion matrix in Figure 18 clearly shows that the wheat disease detection model proposed in this study has significant advantages in the recognition of key agronomic traits. Figure 18a shows that the model has a precision rate of 100% for scab (all 123 cases correctly identified) and a recall rate of 95% for stripe rust (only three cases missed), highlighting its accurate ability to capture highly harmful diseases. Figure 18b further confirms that the model achieves recall rates of 94% and 81% for glume blotch and wheat ear, respectively, indicating its strong ability to represent small disease spots and organ structures.

It is worth noting that the overall F1-score of the three important diseases, scab, stripe rust, and glume blotch, exceeds 85%, which is significantly better than existing field detection methods. The current main limitation is the misjudgment of powdery mildew under complex background interference (32% background confusion). In the

future, the generalization ability of the model can be further improved by embedding a background suppression module. Overall, this model provides reliable technical support for the early and accurate prevention and control of wheat diseases.

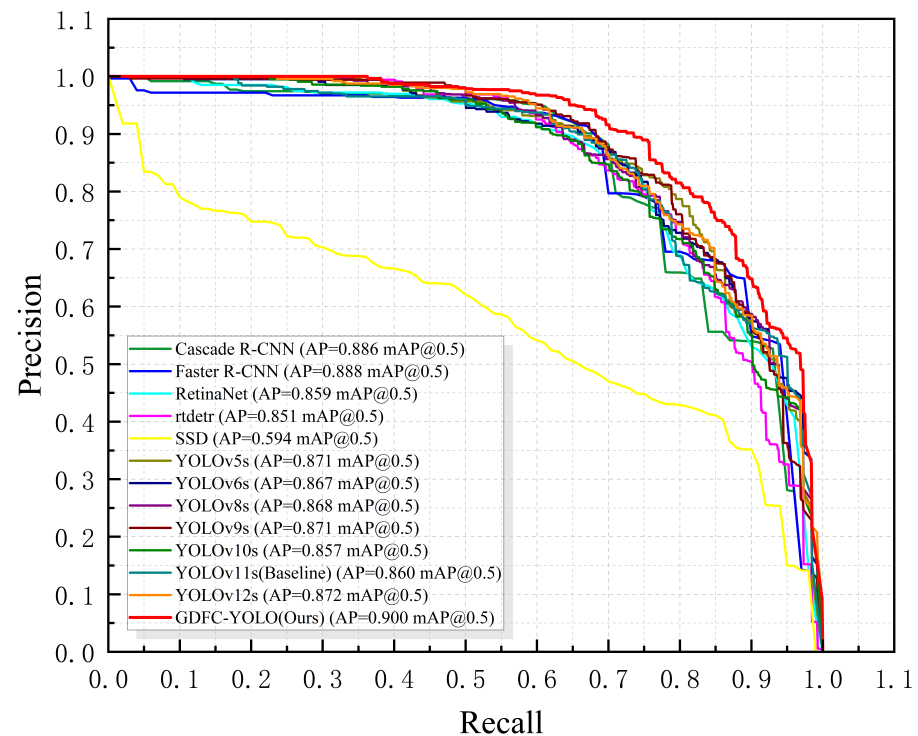


Figure 17. Average P-R curves of GDFC-YOLO and other state-of-the-art algorithms across all categories on the wheat disease test set.

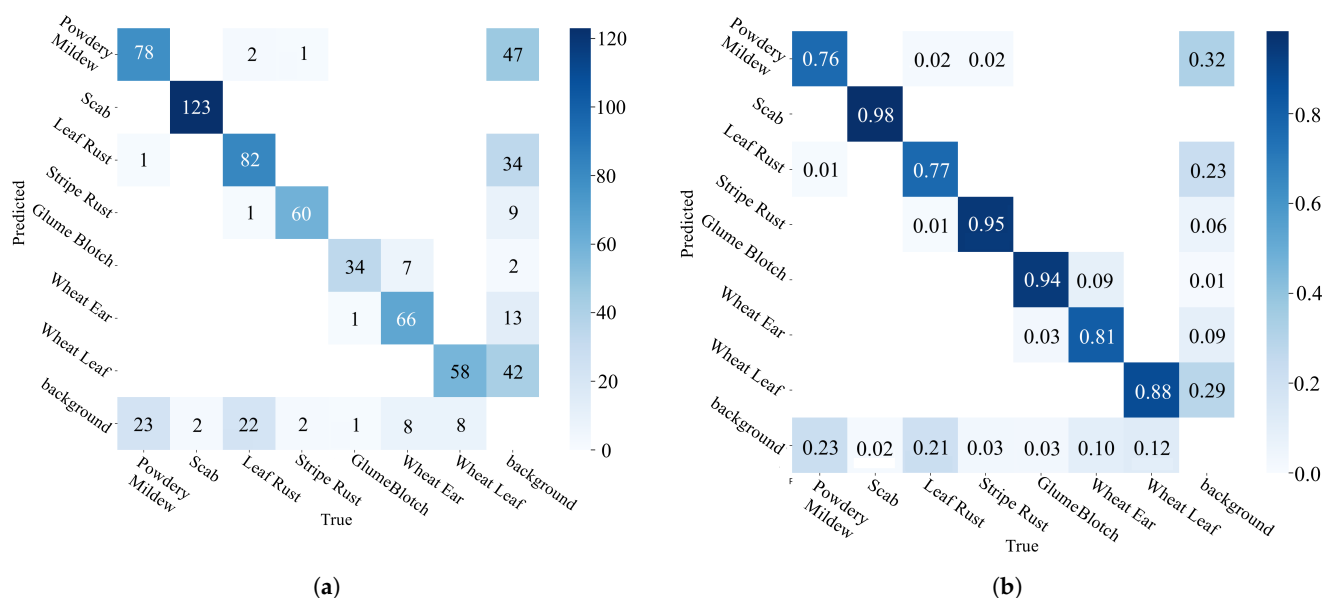


Figure 18. Confusion matrix plot. (a) Confusion matrix. (b) Confusion matrix normalized.

3.6. Model Generalization Experiments

Generalization experiments were conducted to verify the generalization ability of the proposed GDFC-YOLO model on three public plant disease datasets, namely PlantVillage [25], PlantDoc [26], and LWDCD 2020 [27]. The result obtained from the experiment is shown in Table 8.

Table 8. Performance comparison of GDFC-YOLO on other public datasets.

Dataset	P	R	mAP@0.5	mAP@[0.5:0.95]	Params (M)	GFLOPs
Plant-Village	0.854	0.832	0.916	0.902	9.28	22.6
PlantDoc	0.874	0.811	0.918	0.719	9.28	22.6
LWDCD 2020	0.891	0.845	0.92	0.631	9.27	22.5

In order to determine whether the disease representation learned goes beyond the specific features of wheat, the GDFC-YOLO model was applied to the comprehensive PlantVillage dataset (38 disease categories and 54,312 fruit/vegetable images) and PlantDoc dataset (29 categories). It achieved 91.6% and 91.8% for mAP@0.5, 85.4% and 87.4% for precision, individually. It confirmed that the proposed model can effectively identify diseases across a wide range of plant species. To study the generalizability of the developed model in different wheat-growing regions, the LWDCD 2020 dataset (11 wheat diseases and 1130 real field images) was used. It obtained 92.0% for mAP@0.5 and 89.1% for precision. The performance proves its strong adaptability to the complex situations in the actual wheat field environment.

These multi-faceted validation results demonstrate the developed model's excellent generalization capacity. Its consistent higher precision across different testing scenarios strongly supports its suitability for deployment in different categories and varied wheat-growing regions.

4. Conclusions

In the detection of wheat diseases, there exist situations such as natural background interference, small lesion areas, and blurred edges. This study proposes a high-precision detection model called GDFC-YOLO. This model uses the GDFC module to achieve dynamic weighting of multi-scale features, which significantly improves the detection accuracy in complex backgrounds. Accuracy increased by 4.1% and mAP@0.5 to 0.886. The parameters were also reduced by 5.7%, from 9.43 M to 8.89 M, and the computing cost was also reduced by 7.9%, from 21.6 GFLOPs to 19.9 GFLOPs. Thus, its efficiency and representation ability were effectively balanced.

DF-Neck integrates lightweight dsc and an innovative upsampling strategy, enabling the capture of small lesion textures and blurred edges. On the LWDCD 2020 dataset, it implemented 0.92 mAP@0.5. The improved bounding box regression loss function PIoUv2 introduced adaptive size penalties and dynamic normalization, improved localization accuracy, increased mAP@[0.5:0.95] to 0.692, and also accelerated training convergence.

The experimental results show that GDFC-YOLO achieved 0.9 and 0.695, respectively, for mAP@0.5 and mAP@[0.5:0.95] on the wheat disease dataset with only 9.27 M parameters and 22.5 GFLOPs. Compared with YOLOv11s, the accuracy of GDFC-YOLO is 5.4% higher. The parameters have decreased by 0.16 M. Cross-dataset evaluation was conducted on the Plant-Village and PlantDoc datasets, and the mAP@0.5 results were 0.916 and 0.918, respectively, which verified the model's strong generalization ability.

GDFC-YOLO has improved the detection accuracy of irregular lesions under complex natural conditions and contributed to the development of crop disease monitoring. Its enhanced feature representation and precise positioning capabilities can support more reliable disease diagnosis, which is crucial for making informed decisions in crop management. This model has strong generalization ability on different datasets and can be widely used in various agricultural scenarios. Ultimately, it can help strengthen crop health monitoring and increase crop yields.

Author Contributions: Conceptualization, J.Q. and C.D.; methodology, J.Q.; software, J.Q. and C.D.; validation, J.Q. and Z.J.; formal analysis, J.Q. and J.L.; investigation, J.Q. and C.D.; resources, Z.J. and J.L.; data curation, C.D. and Z.J.; writing—original draft preparation, J.Q.; writing—review and editing, J.Q., C.D., Z.J. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2017YFE0135700) and Zhejiang A&F University Research Startup Funds (203402004501).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in Zenodo at: <https://doi.org/10.5281/zenodo.15621359>, accessed on 13 July 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Grote, U.; Fasse, A.; Nguyen, T.T.; Erenstein, O. Food security and the dynamics of wheat and maize value chains in Africa and Asia. *Front. Sustain. Food Syst.* **2021**, *4*, 617009. [CrossRef]
2. Li, Y.; Liu, W.C.; Zhao, Z.H. The occurrence and management of wheat insect pests and diseases in China in 2022 and reflections on pest control measures. *China Plant Prot.* **2023**, *43*, 52–54.
3. Ghazal, S.; Munir, A.; Qureshi, W.S. Computer vision in smart agriculture and precision farming: Techniques and applications. *Artif. Intell. Agric.* **2024**, *13*, 64–83. [CrossRef]
4. Qiu, Z.; Wang, F.; Li, T.; Liu, C.; Jin, X.; Qing, S.; Shi, Y.; Wu, Y.; Liu, C. LGWheatNet: A Lightweight Wheat Spike Detection Model Based on Multi-Scale Information Fusion. *Plants* **2025**, *14*, 1098. [CrossRef] [PubMed]
5. Yao, X.; Yang, F.; Yao, J. YOLO-Wheat: A Wheat Disease Detection Algorithm Improved by YOLOv8s. *IEEE Access* **2024**, *12*, 133877–133888. [CrossRef]
6. Kumar, D.; Kukreja, V. CaiT-YOLOv9: Hybrid Transformer Model for Wheat Leaf Fungal Head Prediction and Diseases Classification. *Int. J. Inf. Technol.* **2025**, *17*, 2749–2763. [CrossRef]
7. Zhong, D.; Wang, P.; Shen, J.; Zhang, D. Detection of Wheat Pest and Disease in Complex Backgrounds Based on Improved YOLOv8 Model. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 1080. [CrossRef]
8. Bao, W.; Huang, C.; Hu, G.; Su, B.; Yang, X. Detection of Fusarium Head Blight in Wheat Using UAV Remote Sensing Based on Parallel Channel Space Attention. *Comput. Electron. Agric.* **2024**, *217*, 108630. [CrossRef]
9. Volety, D.R.; RamanThakur, S.; Mishra, S.; Goel, S.; Garg, R.; Yamsani, N. Wheat Disease Detection Using YOLOv8 and GAN Model. In *Innovative Computing and Communications*; Springer: Singapore, 2024; pp. 349–363.
10. Doroshenko, O.V.; Golub, M.V.; Kremneva, O.Y.; Shcherban', P.S.; Peklich, A.S.; Danilov, R.Y.; Gasiyan, K.E.; Ponomarev, A.V.; Lagutin, I.N.; Moroz, I.A.; et al. Automated Assessment of Wheat Leaf Disease Spore Concentration Using a Smart Microscopy Scanning System. *Agronomy* **2024**, *14*, 1945. [CrossRef]
11. Jiang, Q.; Wang, H.; Sun, Z.; Cao, S.; Wang, H. YOLOv5s-Based Image Identification of Stripe Rust and Leaf Rust on Wheat at Different Growth Stages. *Plants* **2024**, *13*, 2835. [CrossRef] [PubMed]
12. Sharma, J.; Kumar, D.; Chattopadhyay, S.; Kukreja, V.; Verma, A. A YOLO-Based Framework for Accurate Identification of Wheat Mosaic Virus Disease. In Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 14–15 March 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–4.
13. Önlü, E.; Köycü, N.D. Wheat Powdery Mildew Detection with YOLOv8 Object Detection Model. *Appl. Sci.* **2024**, *14*, 7073. [CrossRef]
14. Mao, R.; Zhang, Y.; Wang, Z.; Hao, X.; Zhu, T.; Gao, S.; Hu, X. DAE-Mask: A novel deep-learning-based automatic detection model for in-field wheat diseases. *Precis. Agric.* **2024**, *25*, 785–810. [CrossRef]
15. Sharma, J.; Kumar, D.; Chattopadhyay, S.; Kukreja, V.; Verma, A. Wheat Powdery Mildew Automatic Identification Through YOLOv8 Instance Segmentation. In Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 14–15 March 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–5.
16. Han, K.; Wang, Y.; Guo, J.; Wu, E. ParameterNet: Parameters Are All You Need for Large-scale Visual Pretraining of Mobile Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 15751–15761.
17. Si, Y.; Xu, H.; Zhu, X.; Zhang, W.; Dong, Y.; Chen, Y.; Li, H. SCSA: Exploring the synergistic effects between spatial and channel attention. *Neurocomputing* **2025**, *634*, 129866. [CrossRef]

18. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 6027–6037.
19. Finder, S.E.; Amoyal, R.; Treister, E.; Freifeld, O. Wavelet Convolutions for Large Receptive Fields. In *Computer Vision—ECCV 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15112, pp. 363–380.
20. Rezaatofghi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
21. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
22. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [[CrossRef](#)]
23. Gevorgyan, Z. SloU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740. [[CrossRef](#)]
24. Liu, C.; Wang, K.; Li, Q.; Zhao, F.; Zhao, K.; Ma, H. Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism. *Neural Netw.* **2024**, *170*, 276–284. [[CrossRef](#)] [[PubMed](#)]
25. Gao, C.; Guo, W.; Yang, C.; Gong, Z.; Yue, J.; Fu, Y.; Feng, H. A fast and lightweight detection model for wheat fusarium head blight spikes in natural environments. *Comput. Electron. Agric.* **2024**, *216*, 108484. [[CrossRef](#)]
26. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A Dataset for Visual Plant Disease Detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 5–7 January 2020; pp. 249–253.
27. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.