

# Lightweight Rice Leaf Disease Detection Method Based on Improved YOLOv8

Yingan Fu, Yong Zhang\*

**Abstract**—Rice, a major global food crop, is severely impacted in terms of yield and quality by leaf diseases. In view of the issues of high computational complexity and low detection accuracy due to the complex rice growth environment in the current rice leaf disease detection model, this paper proposes a lightweight rice leaf disease detection model YOLOv8-AMD. Firstly, in real rice growth scenarios, the area of rice leaves affected by diseases is relatively small, leading to potential pixel distortion and loss of fine-grained details during the recognition process. For this reason, we introduce the lightweight convolutional ADown module to replace the convolutional Conv module in the original model to enhance the detection ability of low-resolution images. Secondly, to better perceive the information about leaf diseases in the input image and make the model more integrated with the features related to leaf diseases, we propose the C2f\_MLCA module. By introducing a lightweight Mixed Local Channel Attention mechanism (MLCA) into C2f, the expressive and detection effects of the algorithm are better improved. Finally, to reduce the model's computational burden and training time, we introduce Dysample, a lightweight dynamic upsampler that replaces the original upsampling operator, Upsample. This replacement significantly reduces computational requirements and training time. The YOLOv8-AMD model achieved an 88.8% on the precision, 88.7% and 92.5% on the recall and mAP50, respectively. Experiments conducted on the datasets of corn and tomato leaf diseases further confirm that the proposed model not only optimizes parameters and computational load but also augments the detection performance in multiple dimensions. This model, serving as a high-performance and lightweight solution for detecting rice leaf diseases, is capable of furnishing farmers and researchers with accurate and timely prediction outcomes.

**Index Terms**—rice leaf disease, YOLOv8, attention mechanism, lightweight

## I. INTRODUCTION

**R**ICE, a globally crucial food crop, is significantly impacted by foliar diseases in terms of yield and quality. Foliar diseases such as rice blasts, brown spots, and sheath blight are caused by the infection of pathogenic microorganisms like fungi, bacteria, and viruses. Traditionally, manual observation and empirical judgment are employed to identify and diagnose rice leaf diseases, but this approach is time-consuming, labor-intensive, and prone to misjudgment. In recent years, deep learning have attained extraordinary accomplishments within the domain of image recognition and categorization[1], presenting new possibilities for addressing the problem of rice leaf disease detection and diagnosis. Currently, in the realm of deep learning, target detection methods

are mainly divided into one-stage and two-stage categories. The core concept of the one-stage method is to directly predict the bounding box and category of the target from the input image without going through a separate region proposal step. In contrast, the two-stage method first generates a series of region proposals (i.e., candidate frames) and then conducts classification and edge regression on these proposals. In the one-stage method, a Region Proposal Network (RPN) is typically used, which slides directly over the entire image and predicts both the location and category of the target. In the two-stage approach, the first stage involves generating region proposals, while the second stage entails further refinement and processing of these proposals using a detection network. One-stage methods are more efficient and faster compared to two-stage methods as they avoid the cumbersome steps of region proposal generation and processing. However, two-stage methods may slightly outperform one-stage methods in terms of accuracy as they have more opportunities to fine-tune and improve the region proposals.

The automatic identification and diagnosis of rice leaf diseases through deep learning techniques offer multiple advantages. Firstly, it can remarkably enhance the accuracy and efficiency of identification and diagnosis while significantly reducing the time required for disease detection. Secondly, the deep learning model can automatically learn and adapt to the characteristics of different diseases, possessing excellent generalization ability. Consequently, it can also exhibit good recognition ability when confronted with new disease types[2], [3]. Additionally, deep learning technology can be integrated with other sensors and agricultural management systems to achieve real-time monitoring and early warning of rice diseases. This enables the implementation of timely preventive and curative measures, thus lessening the influence of diseases on the yield and quality of rice[4], [5]. In conclusion, the utilization of deep learning in the realm of identifying and diagnosing rice diseases identification and diagnosis not only improves production efficiency but also helps ensure the stability and healthy development of the rice industry. For example, Kumar VS et al. proposed a multiscale YOLOv5 detection network using DenseNet-201 as the backbone network and Bi-directional Feature Attention Pyramid Network (Bi-FAPN) utilized for extracting features from segmented images. The obtained values were 82.8 for average precision, 94.87 for accuracy, 75.81 for average recall, 0.71 for IoU, 0.017 for inference time, and 92.45 for F1 score respectively[6]. Li D et al. put forward a video detection architecture grounded in deep learning. This architecture features a customized backbone and is specifically designed for the detection of plant pests and diseases within videos.[7]. Haque ME et al. proposed a YOLOv5 deep learning method for the classification and detection of rice leaf diseases. The YOLOv5 model was trained and evalu-

Manuscript received August 21, 2024; revised December 23, 2024.

Yingan Fu is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (e-mail: 19923052958@163.com).

Yong Zhang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, PR China (\*Corresponding author, phone: 86-0412-889699; e-mail: zy9091@163.com).

ated, achieving key performance metrics with the following recognition levels: accuracy at 90%, precision at 67%, recall at 76%, mean Average Precision (mAP) at 81%, and an F1 score of 81%[8]. Jia L et al. presented a rice pest and disease recognition model. It's based on the enhanced YOLOv7, using lightweight MobileNetV3 for feature extraction to cut down parameters. Also, it integrates the CA and SIOU loss functions to enhance recognition accuracy. An accuracy of 92.3% was achieved with mAP @.5 reaching 93.7%[9]. Mishra N et al. designed a new Ant lion-based YOLO-V5 (AL-YOLOv5) system to improve the functionality of the system for detecting rice leaf diseases[10]. Li S et al. utilized residual feature enhancement to reduce semantic gaps in rice pest and disease image features and added a convolutional block attention module to the backbone network, enhancing pest and disease traits and suppressing the background. The proposed system permits unattended operation, high detection precision, objective results, and data traceability[11]. Aziz F et al. employed convolutional neural networks to predict bounding boxes and recognize objects in rice leaf disease detection. This scheme achieved a high accuracy precision of 94% at a detection speed of 0.028 seconds for 1 image[12]. Pan P et al. put forward the Xo-YOLO model. A specific network is integrated into its backbone for better UAV-based disease detection. The neck is enhanced via the GSConv hybrid convolution module, cutting computational cost and parameter numbers. A rotation angle is appended to the head layer output to handle the elongated and rotated appearance of diseases seen from drones. It achieved a mAP of 94.95%[13]. Bari BS, et al. used a network to conduct real-time detection of rice leaf diseases. They also incorporated an advanced Region Proposal Network (RPN) architecture within the algorithm. This RPN network can precisely process target locations, thereby facilitating the generation of candidate regions[14]. Trinh DC et al. devised a two-stage approach for recognizing rice leaf diseases using an artificial intelligence (AI) algorithm. In the initial phase, photographs of diseases on rice leaves were automatically collected from the field. During the second step, after training the YOLOv8 model using the proposed image dataset, the trained model was deployed on Internet of Things (IoT) devices to detect and recognize rice leaf diseases[15]. Zhu R et al. developed a loss calculation algorithm that exhibits insensitivity to sorting, thereby strengthening the model's robustness. A loss scaling factor grounded in polygon perimeter was incorporated to heighten the discovery of tiny targets[16]. Liu X et al. incorporated Spatial Pyramid Pooling and Dilated (SPPD) structures, which possess different gap rates and are composed of GELU activation functions, into the Backbone network. This augmentation expanded the receptive field of the network. Moreover, they integrated Coordinate Attention (CA) to assist the model in concentrating on the characteristics of rice diseases and enhancing detection accuracy. The model attained a mean average precision at 0.5 (mAP @0.5) of 89.3% and reached a Frames Per Second (FPS) rate of 217[17]. Yang Y et al. put forward a novel module. This particular module was integrated into the tail end of the YOLOv8n architecture's backbone, leading to the formation of a target detection model. The newly introduced model has the capacity to reach a mAP50 of 93.6%. The network training weights are limited to 6.7MB, and only 16.8 ms is

needed for the detection of a single pest image[18].

In the realm of rice disease detection methods, although existing literature has made significant progress in achieving accurate detection, most approaches for detecting rice leaf diseases still have notable deficiencies. These shortcomings include high computational complexity in current rice leaf disease detection models and reduced detection accuracy due to the complexity of the background. To address these difficulties, we propose a modified form of the YOLOv8 algorithm. The key achievements of this research are detailed as follows:

(1) In real rice growth scenarios, the area of rice leaves affected by diseases is relatively small, leading to potential pixel distortion and loss of fine-grained details during the recognition process. For this reason, we introduce the lightweight convolutional ADown module to replace the convolutional Conv module in the original model to enhance the detection ability of low-resolution images. The ADown module offers greater advantages in detection accuracy and model performance due to its diverse feature extraction and efficient downsampling approach.

(2) To better perceive the information of leaf diseases in the input image, improve the detection accuracy of rice leaf diseases, and make the model more integrated with the features related to leaf diseases, we propose the C2f\_MLCA module. By introducing a lightweight Mixed Local Channel Attention (MLCA) into C2f, which is capable of integrating both channel and spatial information, as well as the effect of localized information expression.

(3) To reduce the computational burden and training time of the model, we introduce Dysample, a lightweight dynamic upsampler that replaces the original upsampling operator, Upsample. This replacement significantly reduces computational requirements and training time.

(4) In this dataset, we conducted rice leaf disease detection experiments utilizing the enhanced YOLOv8 algorithm. Our results demonstrate significant improvements, achieving a mAP50 of 92.5%, precision of 88.8%, and recall rate of 88.7%. Moreover, greatly reduced computational effort. The model is simplified to 2.6M, which outperforms most current classical detection algorithms, including Faster R-CNN, YOLOv3-tiny, YOLOv5s, YOLOv6n, and YOLOv8n.

The paper is structured as follows. Section II focuses on the dataset preparation and an overview of YOLOv8. Section III pertains to the enhancements made to YOLOv8. Section IV presents the experimental outcomes. The conclusion is examined in Section V.

## II. MATERIALS AND METHODS

### A. Datasets

1) *Datasets source*: The dataset employed in this research was obtained from the complimentary and open-source platforms furnished by Roboflow. To ensure the data's quality, we manually selected images that featured clear and complete targets, resulting in a total of 2098 sample images. As shown in Fig. 1, there are six species of diseases, namely Rice Bacterial Blight (350 images), Rice Blast (300 images), Rice Brown Spot (330 images), Rice Sheath Blight (350 images), Rice Leaf Scald (360 images), and Rice Tungro (408 images).

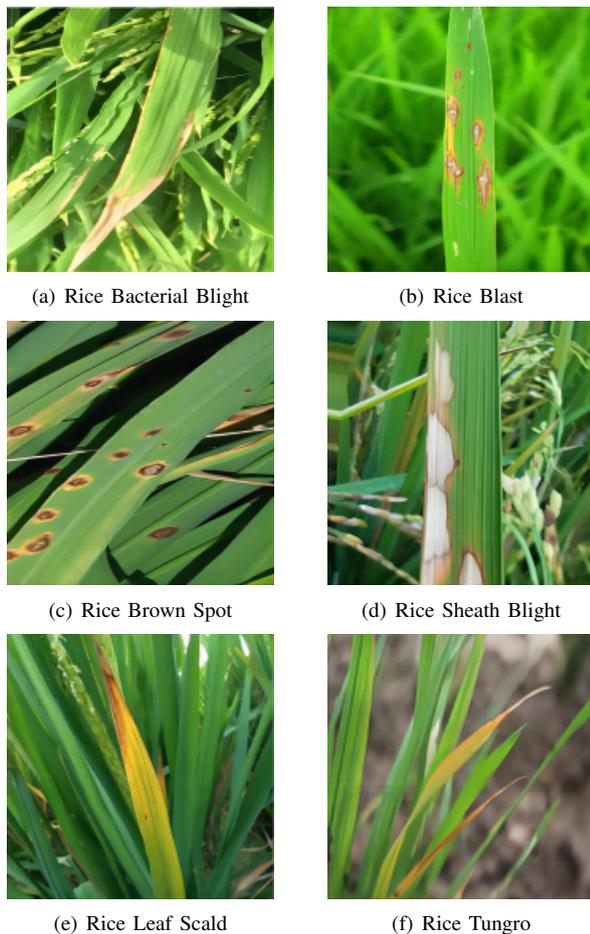


Fig. 1: Image of rice leaf disease.

2) *Data preprocessing*: The sample images in the dataset were labeled using the Labeling labeling software, resulting in a labeled sample set[19]. The labeling process involved using a maximum horizontal rectangular box to encompass the area of disease occurrence within each image. The labels were stored in XML files following the VOC format, with each image having at least one disease label. To mitigate the risk of overfitting and enhance generalization, data augmentation techniques were employed to augment the dataset. Two types of data augmentation techniques were utilized, namely online data augmentation and offline data augmentation[11], [20]. In this study, offline enhancement techniques were employed to modify the image properties while preserving the image content. Geometric transformations, including Image Crop (Random Resize Crop), Image Translation, Image Rotation, etc., were utilized[21]. Content transformations, including color dithering and adding noise, were also applied. The combination of geometric and content transformations facilitated the comprehensive modification of image properties, thereby enhancing the sturdiness and universality of the model. Through data augmentation, a total of 10,490 labeled images were generated from the initial dataset of 2,098 labeled images. The augmented dataset was then divided into three parts: the training set, the validation set, and the test set. The division was performed randomly, with a ratio of 8:1:1, resulting in 8,392 images for the training set, 1,049 images for the testing set, and 1,049 images for the validation set. The number of pictures for the six types of

diseases is shown in Table I, The training set is employed to train the model by optimizing its parameters and updating its weights based on the provided training data. The validation set is utilized to fine-tune the model's hyperparameters, allowing for adjustment of various settings and configurations to enhance its performance. Furthermore, the validation set serves the purpose of conducting a preliminary assessment of the model's capabilities, providing valuable insights into its effectiveness. Conversely, the testing set is exclusively used to evaluate the model's detection accuracy and assess its generalization capabilities. Comprising unseen data, this set offers a reliable measure of the model's performance on new and unfamiliar examples, enabling an evaluation of its ability to effectively detect rice leaf diseases in real-world scenarios.

TABLE I: Datasets Information.

Disease types	The number of pictures
Rice Bacterial Blight	1750
Rice Blast	1500
Rice Brown Spot	1650
Rice Sheath Blight	1750
Rice Leaf Scald	1800
Rice Tungro	2040

### B. YOLOv8 networks

The YOLO family comprises a set of deep learning-based algorithms for real-time target detection. These algorithms utilize a single forward propagation pass over an image to directly predict the target's location and class. YOLOv8, created by the identical team behind YOLOv5[22], stands as a state-of-the-art (SOTA) model. It takes advantage of the triumphs of its antecedent and brings in novel characteristics and refinements to boost its efficacy and versatility. YOLOv8 excels in its speed, accuracy, and user-friendliness, rendering it a remarkable option for a diverse range of tasks, including object detection and tracking, instance segmentation, image classification, and pose estimation. Within the YOLOv8 framework, different variants, namely n, s, l, m, and x are available to cater to different usage scenarios. These variants increase in network depth, leading to improved detection accuracy. Among them, YOLOv8n fulfills our requirements in terms of accuracy, inference speed, and lightweight characteristics. Therefore, YOLOv8n is selected as the research focus for rice leaf disease detection in this project. The YOLOv8 network consists of four principal elements: Input, Backbone, Neck, and Head. These components work together to form the overall model structure, as illustrated in Fig. 2.

### III. IMPROVED METHOD FOR RICE LEAF DISEASE DETECTION

Due to the issues of elevated computational complexity and diminished detection accuracy resulting from the intricate rice growth environment in existing rice leaf disease detection, this research puts forward a lightweight rice leaf disease detection approach grounded in the enhanced YOLOv8. It primarily undertakes optimization and enhancement in the following three aspects to attain superior detection capabilities and reduced computational expenditure.

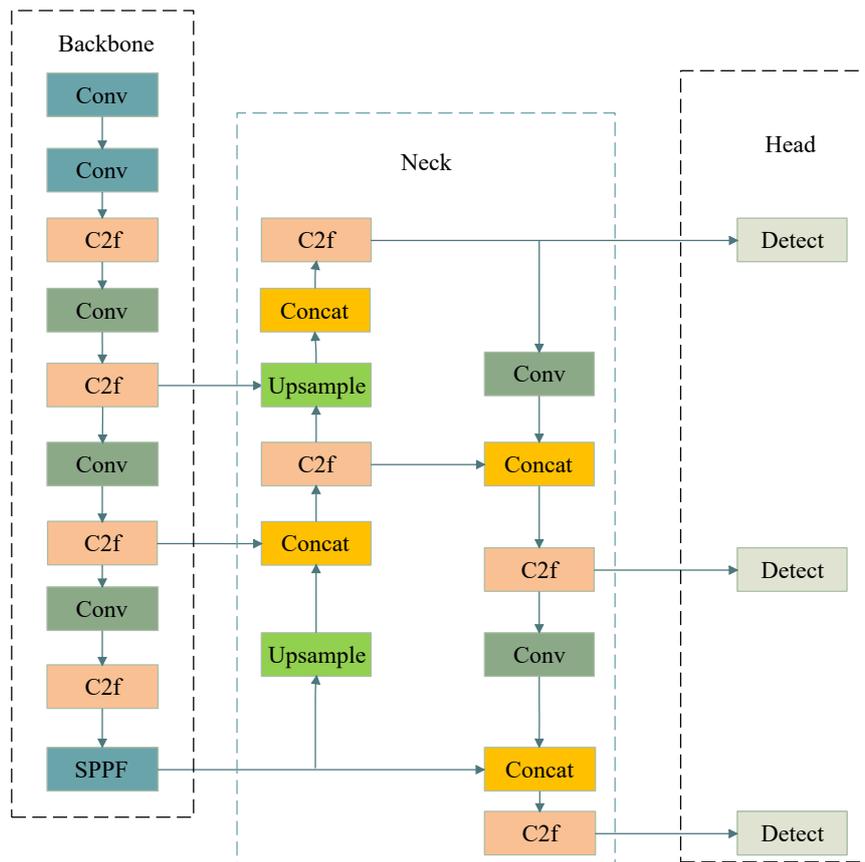


Fig. 2: YOLOv8 structure diagram.

Firstly, in real-world rice growth scenarios, certain images of rice leaf diseases on the smaller side and might encounter problems like pixel distortion[23], This leads to the loss of minute and detailed information and poses challenges for feature learning during the recognition process. To deal with this situation, a lightweight convolutional ADown module is incorporated to substitute the original Conv module within the model. The ADown module offers significant advantages in detection accuracy and model performance due to its diverse feature extraction and effective downsampling techniques. Additionally, the ADown module curtails the quantity of parameters and computational expenses while effectively retaining and extracting image features. This lightweight convolutional operation improves the model's accuracy without increasing computational requirements, enabling faster and more accurate detection of rice leaf diseases.

Secondly, to improve the feature fusion effect, increase the target localization accuracy, and suppress the background interference. In this experiment, a lightweight Mixed Local Channel Attention (MLCA) module is proposed to enhance the performance of the object detection network. The MLCA is incorporated into the C2f of the backbone part to form a new C2f\_MLCA module, which is capable of integrating the channel information and spatial information as well as the expression effect of local information at the same time. It has a more enhanced perception of the leaf disease details within the input image and boosts the precision rate of rice leaf disease detection, so that the model focuses more on the features related to leaf disease and reduces the response to irrelevant information in the background. It contributes to

enhancing the model's stability and interference resistance.

Finally, to address the computational burden and time constraints during the training process, we introduce Dysample, a lightweight dynamic upsampler that replaces the original upsampling operator, Upsample. Dysample eliminates the need for dynamic convolutions and avoids complex dynamic convolution processing, resulting in a significant reduction in computational burden and time requirements. In this research, the enhanced model is designated as the YOLOv8-AMD network, and its model architecture is illustrated in Fig. 3.

#### A. ADown module

During the transmission of images, spatial feature information undergoes a conversion process that gradually converts it into channel information, necessitating the compression of the height and width dimensions of the feature map. In the YOLOv8 model, this conversion is facilitated by constructing a deep network structure via the piling up of multiple convolutional layers. However, these convolutional layers often possess a high number of parameters and computational complexity, which can potentially hinder their efficacy in extracting features. In particular, when dealing with targets that exhibit complex texture and shape features, such as rice leaf diseases, conventional convolutional operations may struggle to capture intricate details effectively. This limitation can lead to compromised detection performance[24]. Therefore, we introduce a new convolutional module called ADown. In comparison to traditional convolutional layers, the ADown module offers significant advantages in terms of

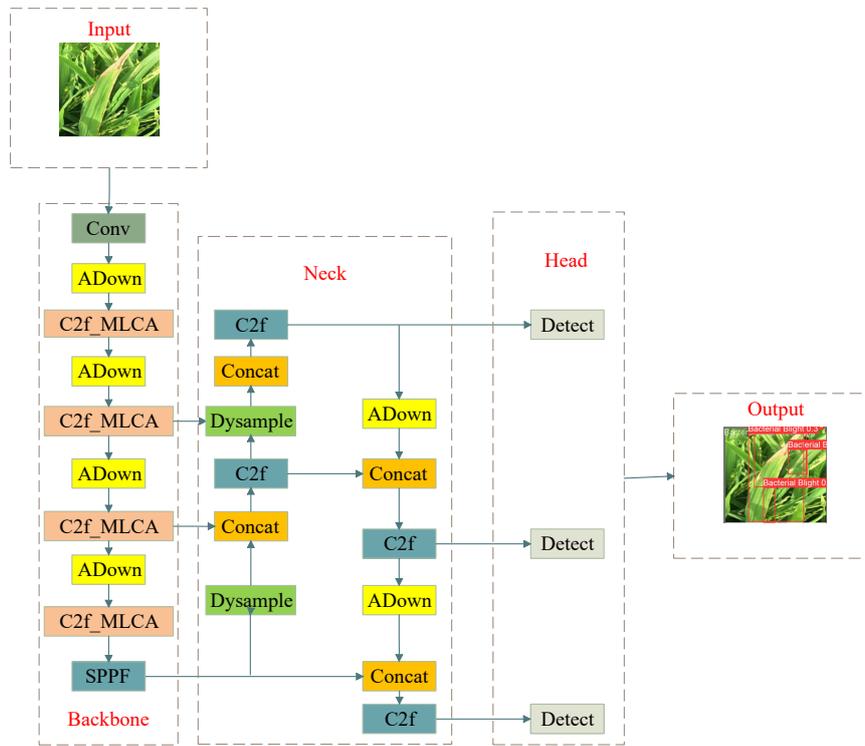


Fig. 3: YOLOv8-AMD structure diagram.

detection accuracy and model performance[25]. This is primarily attributed to its diverse feature extraction techniques and efficient down-sampling methods. These advantages hold particular significance in the context of rice leaf disease detection, especially in resource-limited environments. By incorporating the ADown module, the efficiency of modeling operations can be improved. The lightweight convolutional operation of the ADown module enhances model accuracy without imposing additional computational burden, thereby enabling faster and more accurate detection of rice leaf diseases. Fig. 4 visually presents the structure of the ADown lightweight convolutional module.

*B. C2f\_MLCA module*

The C2f module serves as a channel attention mechanism module, assigning weights based on the correlation among different channels. However, It should be emphasized that the C2f module solely considers the interdependence among channel dimensions and overlooks the spatial dimensions in its calculations. To address the problems that most channel attention mechanisms disregard spatial feature details, which leads to inferior model representation or target detection performance[26], along with the issue of the spatial attention module having a high computational expense and the YOLOv8 model being inadequate for the rice leaf disease detection assignment, we propose the C2f\_MLCA module. This module introduces a lightweight Mixed Local Channel Attention (MLCA) mechanism to better enhance the representation and detection capabilities of the target detection algorithm. The main idea of its MLCA attention mechanism is to aggregate contextual information at different levels and scales and use attention weights to adjust the importance of this information. It has the capacity to concurrently integrate channel and spatial details, along with both local and global

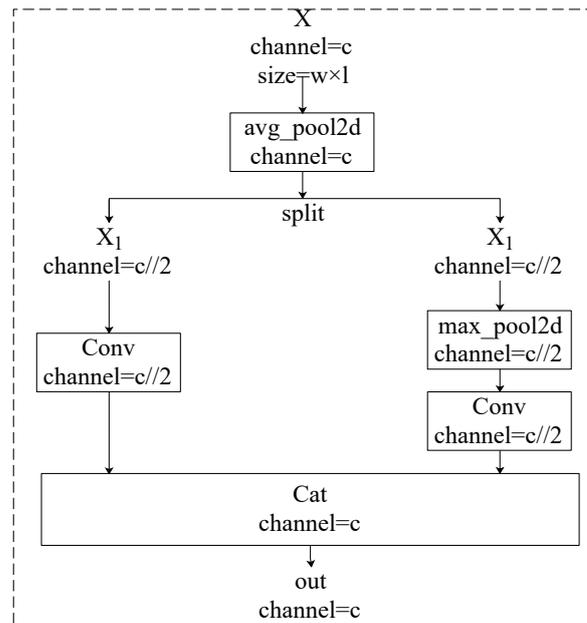


Fig. 4: Structure of ADown lightweight convolution Module.

aspects, thereby enhancing the network’s representational capabilities[26]. At the same time, it does not significantly expand the number of parameters or computational volume of the network and maintains its original real-time performance. The MLCA module is shown in Fig. 5.

The C2f\_MLCA module is designed to analyze channel information, spatial details, and both local and global elements simultaneously by integrating local geographic data into the channel attention mechanism. It employs a two-step pooling process along with one-dimensional convolution.

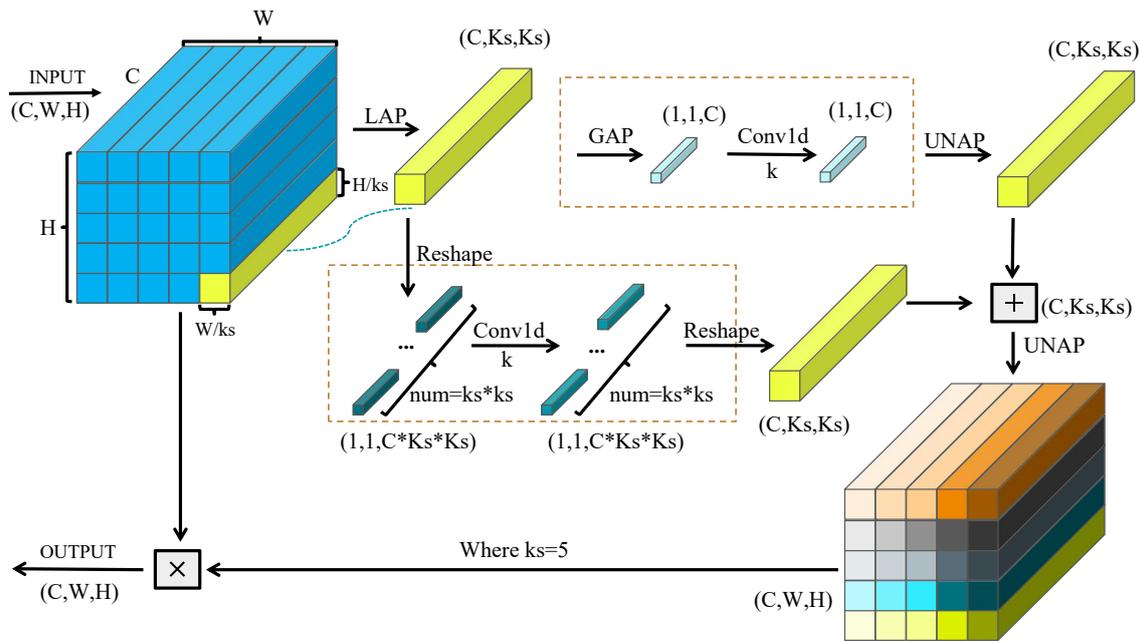


Fig. 5: Structure of ADOWN lightweight convolution Module.

This approach not only improves processing speed but also prevents the accuracy loss typically associated with reducing channel dimensions. Additionally, this method enhances the expandability of the attention mechanism[26]. Fig. 6 illustrates the structure of the C2f\_MLCA module.

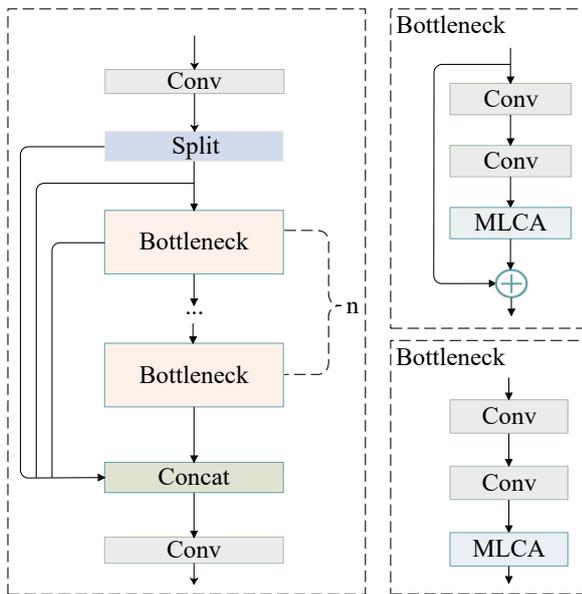


Fig. 6: Structure of C2f\_MLCA module.

It is capable of addressing the subsequent issues associated with the YOLOv8 model.

(1) The YOLOv8 model demands extensive annotated data for training. Nevertheless, rice leaf disease datasets within intricate natural settings are frequently arduous to amass. This constraint restricts the model’s capacity to generalize and adapt;

(2) The YOLOv8 employs an anchorless detector that directly predicts the location and dimensions of the target. Although this design curtails the quantity of model parameters, it might give rise to erratic detection outcomes, especially

when dealing with leaf diseases exhibiting irregular forms and diverse sizes;

(3) The backbone network of YOLOv8 utilizes a cross-stage partial connectivity method. This strategy strikes a balance between the network’s depth and width and augments the efficiency of feature extraction. However, it could potentially result in an inadequate flow of information among feature maps, thereby impinging on the capture of minute details of rice leaf diseases[27].

### C. Dysample module

When it comes to target detection tasks, the utilization of upsampling operations is crucial for resizing the input feature mapping to match the dimensions of the original image. This allows the model to efficiently recognize targets with varying sizes and at different distances. Nevertheless, conventional upsampling techniques typically depend on bilinear interpolation[28], which has some constraints and might lead to the omission of essential image details. Additionally, these conventional kernel-based upsampling approaches demand substantial computational resources and parameter overhead, making them unsuitable for implementing lightweight network architectures[29].

To reduce a significant computational burden and time, this paper introduces Dysample[30], an innovative dynamic upsampler. Dysample primarily employs a point-sampling approach instead of the traditional kernel-based approach, which improves resource efficiency. Dysample can remarkably cut down the computational cost and latency as it doesn’t require time-consuming dynamic convolution and extra sub-networks for generating dynamic kernels. Its lightweight nature, along with fewer parameters and lower GPU memory requirements, makes it highly efficient. Notably, Dysample outperforms other upsamplers in several intensive prediction tasks, including semantic segmentation and target detection, demonstrating its versatility and effectiveness in diverse applications.

Its advantages:

(1) Avoiding dynamic convolution: Dysample does not use complex dynamic convolution processing, which reduces a lot of computational burden and time.

(2) Point sampling method: It implements upsampling by point sampling instead of the traditional kernel method, which is simpler and more efficient.

(3) Fewer parameters and memory requirements: Compared to previous methods, Dysample requires fewer parameters and reduces the memory requirements of the GPU, which means it is lighter and runs faster. With its innovative point sampling method, Dysample improves processing speed and application range while maintaining high efficiency.

Fig. 7 illustrates the network structure of Dysample. The sampling set  $S$  consists of the original sampling grid ( $O$ ) and generated offsets ( $G$ ). Offsets are created via the “linear + pixel shuffle” method. For instance, in static factor sampling with a  $c \times h \times w$  feature map and upsampling factor  $s$ , the map passes through a linear layer (input  $c$ , output  $2s^2$  channels), then reshaped by pixel shuffle to  $2 \times sh \times sw$  (for  $x$  and  $y$  coords). Finally, an upsampled  $c \times sh \times sw$  map is generated[31]. Here,  $x$  is input,  $x'$  up-sampled features,  $G$  offsets,  $O$  original grid,  $\sigma$  is sigmoid,  $sh$  is sampling height,  $sw$  sampling width, and  $gs^2$  channels after linear layer.

#### IV. TEST AND ANALYSIS

##### A. Test environment and parameter configuration

All experiments in this paper were conducted using Pytorch, a deep learning framework, on an Ubuntu 22.04 system equipped with a 12th Gen Intel(R) Core(TM) i5-12400F CPU@2.5GHz and an NVIDIA GeForce RTX 3090 GPU. The Python version used was 3.9, and the Pytorch version was 2.2.1. The specific configuration details of the experimental environment are provided in Table II. For training and inference, a single GPU was utilized, and all networks were trained with a batch size of 16. Other hyperparameters, including the learning rate, weight decay, and data augmentation, were set to their default values, which are specified in Table III.

TABLE II: The configuration of environment parameters.

Environmental parameter	Configuration
CPU	12th Gen Intel(R) Core(TM) i5-12400F
GPU	NVIDIA GeForce RTX 3090
Operating system	Ubuntu22.04
Programming language	Python3.9
Deep learning framework	Pytorch2.2.1

TABLE III: The configuration of hyperparameters.

Hyperparameter	Configuration
Learning Rate	0.01
Image Size 640*640	640*640
Momentum	0.937
Optimizer	SGD
Batch Size	16
Epoch	100
Weight Decay	0.0005S
close_mosaic	10
Learning rate	0.01

##### B. Model evaluation indexes

In this study, several evaluation metrics were employed to assess the efficacy of the YOLOv8-AMD network model in rice leaf disease detection. These metrics include Precision (P), Recall (R), and mean Average Precision (mAP). Additionally, the number of model Parameters (Parameters), Giga Floating-point Operations Per Second (GFLOPs), and the processing capacity in terms of Frames Per Second (FPS) were chosen as evaluation indices for the target detection model[32].

Precision, denoted as P, is a metric that quantifies the ratio of samples accurately categorized as positive classes by the classifier in relation to the overall quantity of samples designated as positive classes. Eq. (1) presents the calculation formula for precision.

$$P = \frac{TP}{TP + FP} \quad (1)$$

Where  $TP$  stands for the quantity of true positive samples (that is, positive samples that have been correctly classified), while  $FP$  stands for the number of false positive samples (namely, samples that are wrongly classified as positive despite actually being negative).

Recall, denoted as R, is a metric that measures the proportion of positive class samples that are correctly recognized as positive by the classifier compared to the entire number of positive class samples. Eq. (2) presents the calculation formula for the Recall.

$$R = \frac{TP}{TP + FN} \quad (2)$$

Where  $TP$  stands for the quantity of true positive samples (that is, positive samples that have been correctly classified), and  $FN$  stands for the count of false negative samples (that is, samples which are wrongly categorized as negative while they are in fact positive).

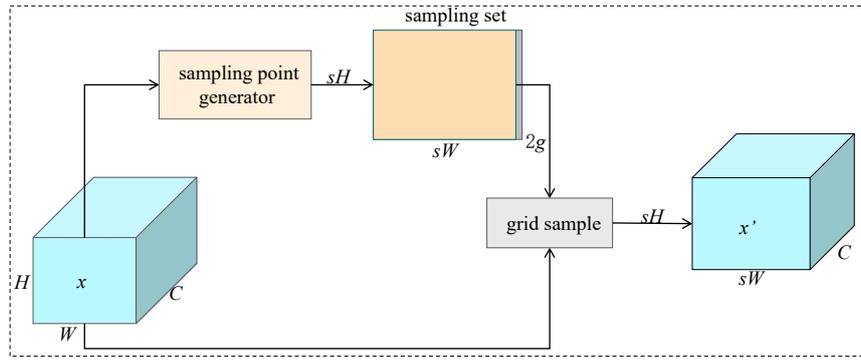
mAP is a metric that is employed to gauge the average precision of a detection model among different categories. In the target detection task, the average precision (AP) is initially computed for each category, and subsequently, the AP values are averaged over all categories to derive the mAP score. Eq. (3) shows the calculation formula for mAP.

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (3)$$

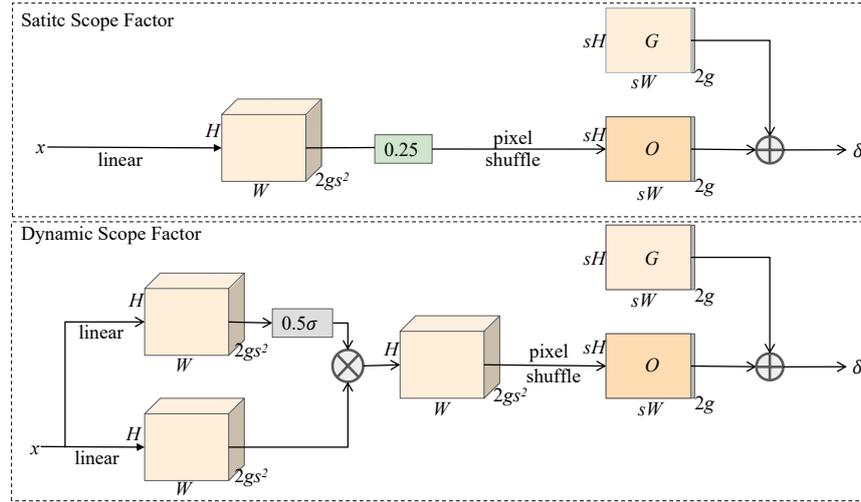
Where  $n$  indicates the overall number of categories, and  $AP_i$  represents the average precision for category  $i$ . The mAP value offers a comprehensive evaluation of the detection model’s efficacy across numerous categories.

Parameters denote the overall quantity of trainable parameters that make up the deep learning model. A higher parameter count often indicates a more intricate model. While a greater quantity of parameters can potentially increase the model’s learning capacity[33], it also carries the risk of overfitting and can result in increased computational requirements. It is crucial to maintain an equilibrium between model complexity and efficiency for the purpose of guaranteeing optimal performance.

FPS serves as a metric for gauging the processing speed of the target detection model. It reflects the quantity of images



(a) Sampling-based dynamic upscale



(b) Sampling point generator in Dysample

Fig. 7: Dysample network structure.

that the model is capable of analyzing within one second. A greater FPS value implies a quicker detection speed. Eq. (4) shows the calculation formula for FPS.

$$FPS = \frac{1}{T_{total}} = \frac{1}{T_{pre} + T_{det} + T_{post}} \quad (4)$$

where  $T_{total}$  is the total process time;  $T_{pre}$  is the pre-processing time;  $T_{det}$  is the detection time; and  $T_{post}$  is the post-processing time.

### C. Comparison experiment of adding different attention mechanisms in C2f

In order to assess the impact of the MLCA attention mechanism when added to the C2f module, we conducted experiments using A-C2f as the base model. While keeping other parameters unchanged, we integrated six different attention mechanisms, namely Cascaded Group Attention (CGA)[34], Sea-Attention (Sea)[35], Inverted Residual Mobile Block (iRMB)[36], Spatial Attention (SA)[37], Iterative Attention Feature Fusion (iAFF)[38], and Efficient Channel Attention (ECA)[39], into the same position of the C2f module. This resulted in six optimized models: A-C2f\_CGA, A-C2f\_Sea, A-C2f\_iRMB, A-C2f\_SA, A-C2f\_iAFF, and A-C2f\_ECA. After training and testing the six optimized models, along with the base model A-C2f, employing the identical dataset[40], evaluation metrics, and experimental

environment, we compare and analyze their performance. Table IV presents the results of these comparisons. where A is the Yolov8-ADown-Dysample model.

TABLE IV: Integration of diverse attention mechanisms within C2f

Models	P	R	mAP50	Params(M)	GFLOPs	FPS
A-C2f	87.0	87.3	92.1	2.60	7.3	200
A-C2f_CGA	87.4	87.9	92.4	2.65	7.5	143
A-C2f_Sea	87.5	87.5	92.4	2.65	7.3	130
A-C2f_iRMB	88.2	87.7	<b>92.8</b>	2.72	10.7	50
A-C2f_SA	87.9	88.2	92.3	2.65	7.3	208
A-C2f_iAFF	87.5	<b>89.3</b>	92.2	2.71	7.5	208
A-C2f_ECA	87.6	87.7	92.4	2.70	7.3	188
A-C2f_MLCA	<b>88.8</b>	88.7	92.5	<b>2.60</b>	<b>7.3</b>	<b>250</b>

In Table IV, we observe that the models with the attention mechanism added in the C2f module (A-C2f\_CGA, A-C2f\_Sea, A-C2f\_iRMB, A-C2f\_SA, A-C2f\_iAFF, A-C2f\_ECA, A-C2f\_MLCA) exhibit higher values of precision, recall rate, and mAP50 in comparison to the base model (A), denoting that the attention mechanism enhances feature fusion and boosts the performance and precision in detecting rice leaf diseases. This verifies the effectiveness of the attention mechanism in augmenting the model's precision regarding the detection and categorization of rice leaf diseases.

Among the attention mechanisms evaluated in Table IV,

C2f\_MLCA stands out with a higher accuracy rate of 88.8%. Additionally, its number of parameters is smaller than the other attention mechanisms at 2.6 M. Notably, C2f\_MLCA also demonstrates a significantly faster detection speed at 250 FPS compared to the other mechanisms. Although C2f\_iRMB achieves the highest mAP50 of 92.8%, its detection speed is much lower at 50 FPS compared to C2f\_MLCA. On the other hand, C2f\_iAFF exhibits the highest recall rate at 89.3%. However, C2f\_MLCA surpasses C2f\_iAFF in respect of precision, mAP50, quantity of parameters, GFLOPs and FPS, indicating superior overall performance.

In a word, C2f\_MLCA succeeds in attaining a harmony between precision and rapidity, all the while diminishing the computational load and inference duration of the model. It boosts the model's capacity to perceive leaf disease-related information in input images, improves the precision of rice leaf disease detection, and emphasizes features relevant to leaf diseases while reducing responses to irrelevant background information.

#### D. Ablation Experiments

In the ablation experiments, we aimed to ensure a fair and reasonable comparison by training all models using the same dataset and under the same parameter environment. The objective was to evaluate the enhancement in the performance of the YOLOv8 model following the integration of the enhanced ADown lightweight convolution module and the dynamic upsampler Dysample, and the C2f\_MLCA module. To conduct the ablation experiments, we systematically removed or replaced these components and evaluated the resulting models. Table V shows the comparison outcomes of these ablation experiments, which provides insights into the performance impact of each component and their combined effect on the YOLOv8 model[41].

Table V presents the preliminary experimental results of the base model YOLOv8, serving as a reference for the subsequent eight experimental sets. The outcomes consist of a precision value of 86.3%, a recall level of 85.5%, a mAP50 figure of 91.3%, a parameter magnitude of 3.00 million, a floating-point volume of 8.1 G, and an FPS rate of 222.

In the second experimental group, when compared to the base model YOLOv8, the lightweight convolutional ADown module yields notable enhancements in precision (1.4%), recall rate (1.3%), and mAP50 (1.0%). Furthermore, there is a reduction in computation by 0.8 G, a decrease in parameter count by 0.41 million, and an improvement in processing speed by 56 FPS. These results imply that the lightweight convolutional ADown module enhances the model's proficiency in detecting rice leaf diseases within low-resolution images or smaller targets, and at the same time, decreases the incidences of false positives and false negatives. Consequently, rice leaf disease detection has become faster and more accurate. In the third experimental group, the incorporation of the dynamic upsampler Dysample leads to improvements in precision, recall rate, and mAP50 by 0.7%, 1.1%, and 0.4%, respectively. Additionally, the computational GFLOPs are reduced from 8.1 G to 7.5 G. These results indicate that Dysample effectively reduces computational workloads and processing delays.

In the fourth experimental group, the introduction of C2f-MLCA resulted in improvements in precision (0.5%), recall

rate (1.4%), and FPS (41), while the mAP50, computational effort, and number of parameters remained unchanged. This indicates that C2f-MLCA enhances the representation and detection of target detection algorithms for rice leaf diseases, leading to more accurate and efficient detection of such diseases.

The eighth experimental group showcases the results of incorporating all the proposed improvements. In comparison to the base model YOLOv8, the YOLOv8-AMD model demonstrates significant enhancements. Precision is improved by 2.5%, while the recall rate and mAP50 metrics show improvements of 3.2% and 1.2%, respectively. Moreover, the FPS is increased by 28. Notably, the computational complexity is substantially reduced, with the computational GFLOPs dropping from 8.1 G to 7.3 G. Additionally, the quantity of parameters is reduced from 3.00 M to 2.60 M, with reduction percentages of 9.88% and 13.33% respectively.

The method proposed in this paper surpasses the base model YOLOv8n in terms of performance, achieving lightweight characteristics and enhancing the accuracy of the model for faster and more precise detection of rice leaf diseases. Fig. 8 illustrates the results of the ablation experiments, showcasing precision, recall rate, and mAP50 through the Linear graphs.

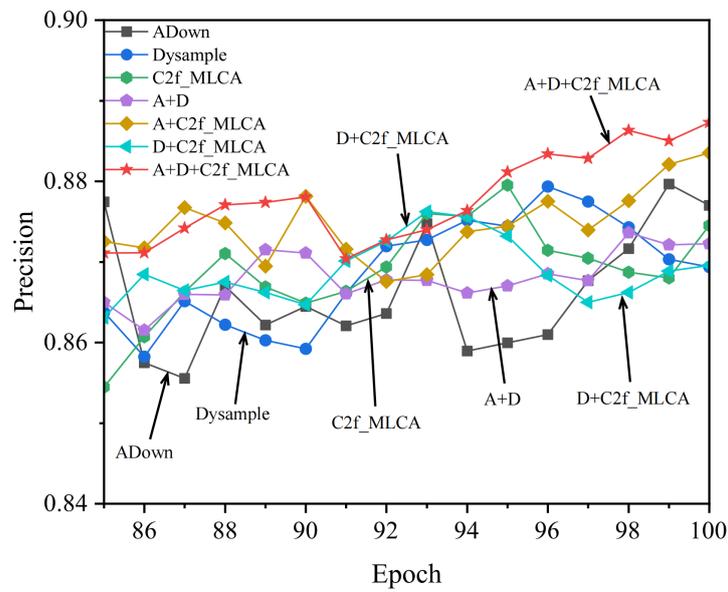
#### E. Comparison Experiments of Different Models

To assess the efficacy of the proposed improved algorithms in this research, we conducted comparative experiments with Faster R-CNN, YOLOv3-tiny, YOLOv5s, YOLOv6n, and YOLOv8n. These experiments were conducted under identical settings, utilizing the same equipment, datasets, and data augmentation methods, while ensuring an equal distribution between the training and test sets[42]. The experiments were carried out for 100 iterations. Table VI presents a comparison of the performance metrics, including P, R, mAP50, Parameters, GFLOP and FPS.

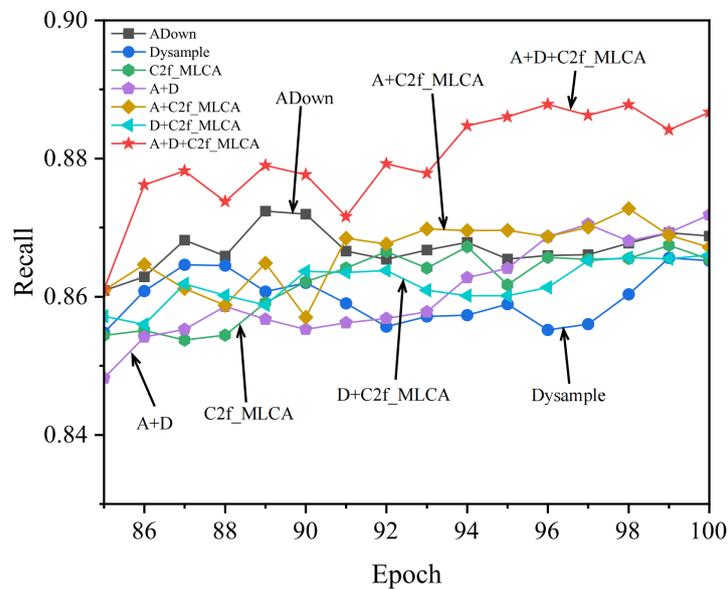
Table VI exhibits the outstanding performance of the algorithm proposed in this research, particularly in terms of mAP50. Specifically, YOLOv8-AMD achieves an impressive mAP50 of 92.5% under the same experimental conditions. In comparison, significant differences in mAP50 are observed with Faster R-CNN (66.5%), YOLOv3-tiny (83.6%), YOLOv5s (90.6%), YOLOv6n (87.9%), and the standard YOLOv8n (91.3%). Notably, YOLOv8-AMD achieves a detection rate of 250, surpassing other one-stage and two-stage algorithms. Furthermore, the detection performance metrics exhibit substantial improvements, with Precision and Recall rates reaching 88.8% and 88.7%, respectively. These findings are effectively visualized in the radar plot depicted in Fig. 9, This evidently demonstrates the method's preeminence in detection accuracy and dependability. Overall, the model surpasses other algorithms within the target detection field, thereby further highlighting its efficacy.

#### F. Model generalizability experiment

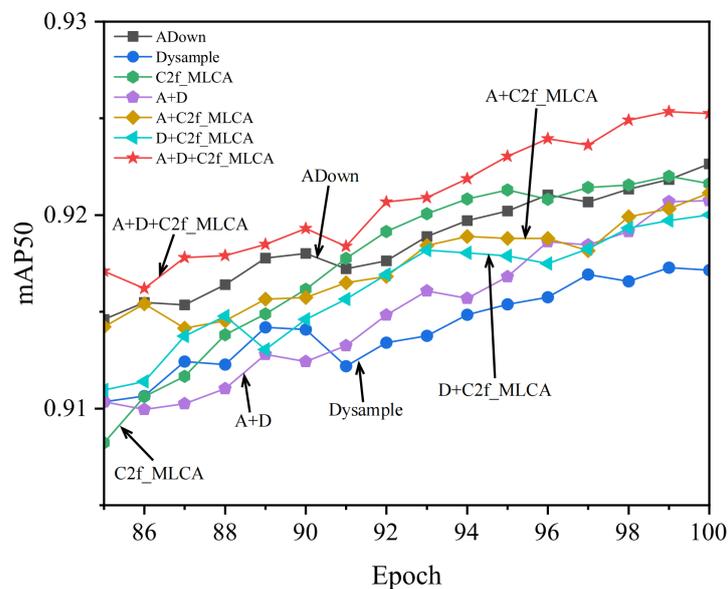
To evaluate the universality of the enhanced YOLOv8-AMD algorithm model, random samples from the traffic signs dataset and blood cells dataset were selected for model generalization experiments using Roboflow's public dataset.



(a) Linear graphs on Precision



(b) Linear graphs on Recall



(c) Linear graphs on mAP50

Fig. 8: Linear graphs

TABLE V: Ablation experiments.

Yolov8	ADown	Dysample	C2f_MLCA	P	R	mAP50	Params(M)	GFLOPs	FPS
✓				86.3	85.5	91.3	3.00	8.1	222
✓	✓			87.7	86.8	92.3	<b>2.59</b>	7.3	278
✓		✓		87.0	86.6	91.7	3.02	7.5	189
✓			✓	86.8	86.9	91.3	3.00	8.2	263
✓	✓	✓		87.0	87.3	92.1	2.60	7.3	200
✓	✓		✓	88.4	86.7	92.1	2.59	7.3	278
✓		✓	✓	87.0	86.6	92.0	3.02	8.2	<b>312</b>
✓	✓	✓	✓	<b>88.8</b>	<b>88.7</b>	<b>92.5</b>	2.60	<b>7.3</b>	250

TABLE VI: Comparison experiments of different target detection algorithms.

Models	P	R	mAP50	Params(M)	GFLOPs	FPS
Faster R-CNN	70.9	58.6	66.5	68.47	122.7	73
YOLOv3-tiny	79.4	77.3	83.6	103.69	283.0	103
YOLOv5s	85.2	85.1	90.6	2.54	<b>7.1</b>	244
YOLOv6n	84.8	83.0	87.9	4.23	11.8	233
YOLOv8n	86.3	85.5	91.3	3.00	8.1	222
YOLOv8-AMD	<b>88.8</b>	<b>88.7</b>	<b>92.5</b>	<b>2.60</b>	7.3	<b>250</b>

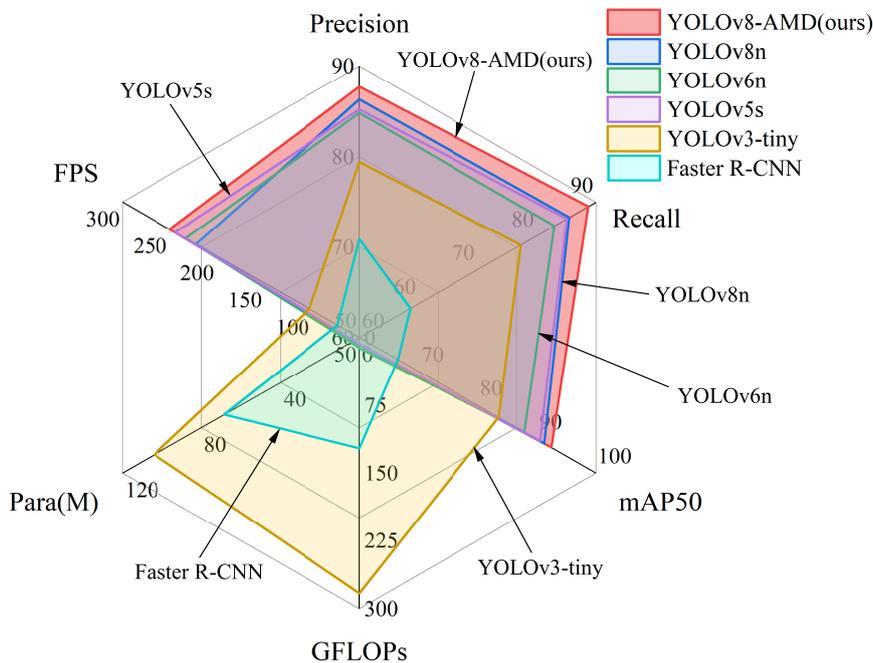


Fig. 9: Radar chart.

These experiments were conducted under consistent parameter settings for both YOLOv8 and YOLOv8-AMD. The results of the traffic sign identification by the YOLOv8-AMD model are presented in Table VII, where A is the YOLOv8-AMD model and B is the YOLOv8 model, while the comparative experiments between YOLOv8 and YOLOv8-AMD for traffic sign detection are displayed in Table VIII. Additionally, Table IX showcases the identification results of blood cell detection using YOLOv8-AMD, where A is the YOLOv8-AMD model and B is the YOLOv8 model, and Table X presents the comparative experiments between YOLOv8 and YOLOv8-AMD for blood cell detection.

Table VII demonstrates the detection results of traffic signs using both YOLOv8 and YOLOv8-AMD models. YOLOv8-AMD achieves improved accuracy in detecting traffic signs compared to YOLOv8. Specifically, YOLOv8-AMD achieves a remarkable accuracy of 100% for Speed Limit 120 and Speed Limit 50, while also achieving a 100% recall rate for Speed Limit 110 and Stop. Moreover, both YOLOv8 and

YOLOv8-AMD achieve a high mAP50 of 99.5% for the Stop traffic sign.

Table VIII presents the comparison between the base model YOLOv8 and YOLOv8-AMD under the same parameter settings. YOLOv8 achieves precision, recall rate, and mAP50 of 94.7%, 85.9%, and 93.5%, respectively. In contrast, YOLOv8-AMD demonstrates improvements in precision by 1.2%, recall rate by 0.7%, and mAP50 by 0.1%. Additionally, YOLOv8-AMD exhibits a remarkable growth in FPS by 23. Notably, the computational GFLOPs are reduced from 8.1 G to 7.3 G, and the model parameters decrease from 3.00 M to 2.61 M, indicating reductions of 9.88% and 13.33%, respectively. These findings suggest that the YOLOv8-AMD module effectively reduces computational requirements and inference time while achieving a balance between model accuracy and speed.

Table IX illustrates the precision of YOLOv8-AMD outperforming YOLOv8 for all three cell types. Notably, both YOLOv8 and YOLOv8-AMD achieved a perfect recall rate

TABLE VII: YOLOv8 and YOLOv8-AMD traffic sign identification results.

Classes	P		R		mAP50	
	A	B	A	B	A	B
Green Light	89.2	92.1	60.5	68.0	75.7	80.6
Red Light	90.6	91.7	60.7	61.1	78.4	79.8
Speed Limit of 100	94.0	96.0	92.3	90.4	96.9	97.2
Speed Limit of 110	89.5	84.5	94.1	100	98.5	96.6
Speed Limit of 120	99.0	100	92.3	92.1	99.2	98.7
Speed Limit of 20	99.0	99.1	97.5	98.2	98.6	98.6
Speed Limit of 30	97.0	95.3	94.6	93.2	96.7	97.0
Speed Limit of 40	96.9	97.0	86.8	89.1	97.3	96.4
Speed Limit of 50	93.6	100	81.7	80.0	95.0	94.2
Speed Limit of 60	94.4	98.9	85.5	86.8	94.2	94.8
Speed Limit of 70	97.3	99.2	96.2	94.9	97.9	97.8
Speed Limit of 80	95.1	95.2	87.5	92.9	96.1	96.5
Speed Limit of 90	92.7	95.9	73.7	65.8	85.5	83.0
Stop	97.8	98.4	98.9	100	99.5	99.5

TABLE VIII: YOLOv8 and YOLOv8-AMD traffic sign comparison experiments.

Data set	Models	P	R	mAP50	Params(M)	GFLOPs	FPS
traffic signs	YOLOv8	94.7	85.9	93.5	3.00	8.1	67
	YOLOv8-AMD	<b>95.9</b>	<b>86.6</b>	<b>93.6</b>	<b>2.60</b>	<b>7.3</b>	<b>80</b>

TABLE IX: YOLOv8 and YOLOv8-AMD hemocyte identification results.

Classes	P		R		mAP50	
	A	B	A	B	A	B
Platelets	78.4	81.2	89.5	94.7	90.7	91.1
RBC	74.8	75.6	88.6	86.8	88.5	88.7
WBC	94.6	96.1	100	100	98.4	98.5

of 100% for WBC.

Table X demonstrates remarkable improvements in precision (1.7%), recall rate (1.2%), and mAP50 (0.3%) achieved by YOLOv8-AMD compared to the base model YOLOv8. Moreover, YOLOv8-AMD exhibits reductions in computational GFLOPs from 8.1 G to 7.3 G and in parameters from 3.00 M to 2.61 M, representing reductions of 9.88% and 13.33%, respectively. Notably, YOLOv8-AMD achieves a significantly higher FPS of 263 compared to YOLOv8's 119, resulting in a substantial improvement of 144 FPS.

In conclusion, YOLOv8-AMD demonstrates superior performance compared to YOLOv8 under the same parameter settings for different datasets. It exhibits improvements in P, R, mAP50, and FPS. It should be noted that the parameter quantity of the enhanced YOLOv8 model remains at 2.60M, and the size of the computational volume of GFLOPs is kept at 7.3G, which achieves a lighter weight and faster detection speed. lightweight and accelerated detection speed, which illustrates the superiority of the improved YOLOv8 method proposed in this paper in terms of detection accuracy and reliability.

### G. Multi-crop disease detection experiments

To comprehensively evaluate the performance and applicability of the model proposed in this paper on different crops, we selected the corn disease dataset and the tomato leaf disease dataset from the Roboflow open-source platform. This kind of selection is intended to guarantee the dependability and variety of the datasets, in order to assess the model's capabilities in a more thorough manner. Among them, corn includes six categories of diseases such as Brow Spot, Corn Rust, Corn Smut, Downy Mildew, Grey Leaf Spot, Leaf Blight, as shown in Fig. 10. In addition, tomato also includes

six types of diseases such as Bacterial Spot, Early Blight, Healthy, Late Blight, Leaf Miner, and Spider Mites, as shown in Fig. 11.

We ensure that experiments are performed on YOLOv8 and YOLOv8-AMD under the same parameter environment. The experiment is shown in Table XI. Detailed records of the experimental indices and results of different models on diverse datasets facilitate in-depth analysis and comparison of the performance of the two models in detecting maize and tomato leaf diseases, thereby furnishing solid data support and theoretical foundation for subsequent research and application. The comparison outcomes of corn and tomato diseases are presented in Fig. 12 and Fig. 13 respectively.

From the data shown in Table 11, firstly, YOLOv8-AMD is able to maintain the parameter at 2.6M and the computation volume at 7.3G on both tomato and corn datasets. Secondly, YOLOv8-AMD has a significant advantage over YOLOv8, especially in tomato disease detection, with a P improvement of 20.3%, and notably, significant improvements in R (11.4%) and FPS (32). Finally, for the corn dataset, despite a slight decrease in recall, there was a significant improvement in accuracy (8.1%) and mAP50 (0.9%). In summary, YOLOv8-AMD presents multiple advantages. It optimizes model parameters and computational complexity, simultaneously boosting the detection precision and retaining excellent real-time detection capabilities, thus furnishing more efficient technical assistance for disease surveillance and management in the agricultural sector.

### H. Analysis of YOLOv8-AMD experiment results

The confusion matrix acts as a significant instrument for evaluating the performance of a classification model[43]. It presents the classification outcomes of the model in a tabular

TABLE X: YOLOv8 and YOLOv8-AMD blood cell comparison experiments.

Data set	Models	P	R	mAP50	Params(M)	GFLOPs	FPS
BCCD	YOLOv8	82.6	92.7	92.5	3.00	8.1	119
	YOLOv8-AMD	<b>84.3</b>	<b>93.9</b>	<b>92.8</b>	<b>2.60</b>	<b>7.3</b>	<b>263</b>

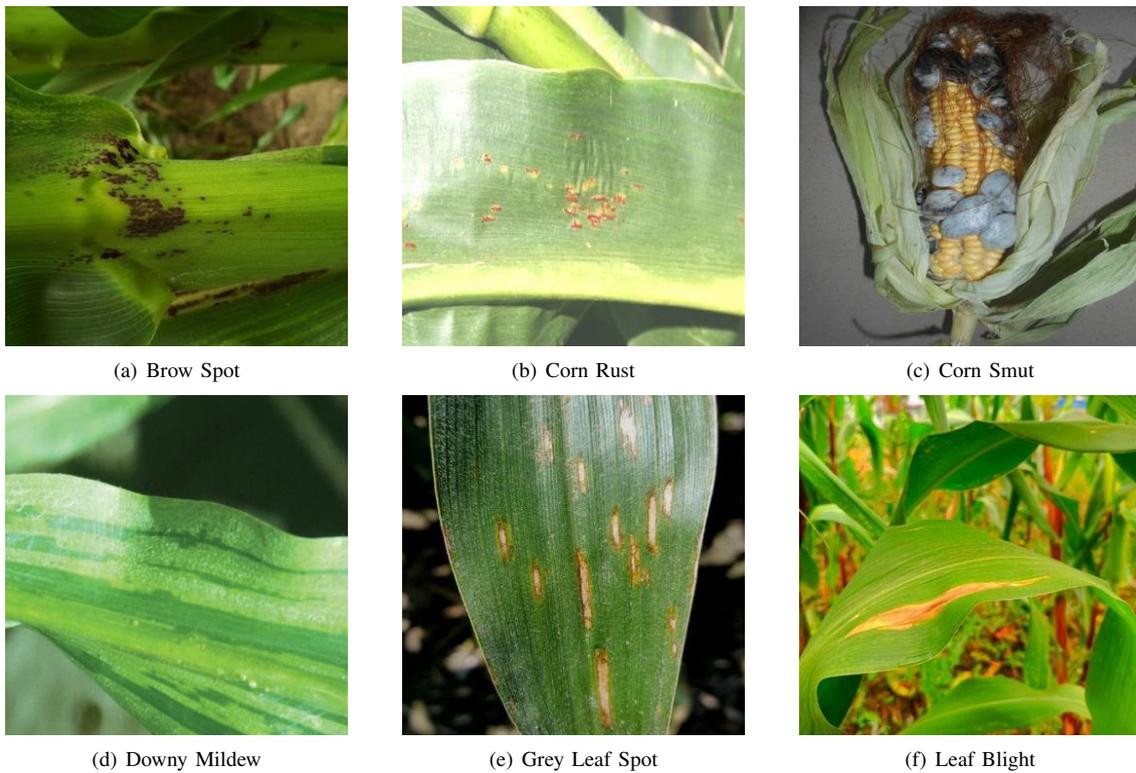


Fig. 10: Image of corn disease.

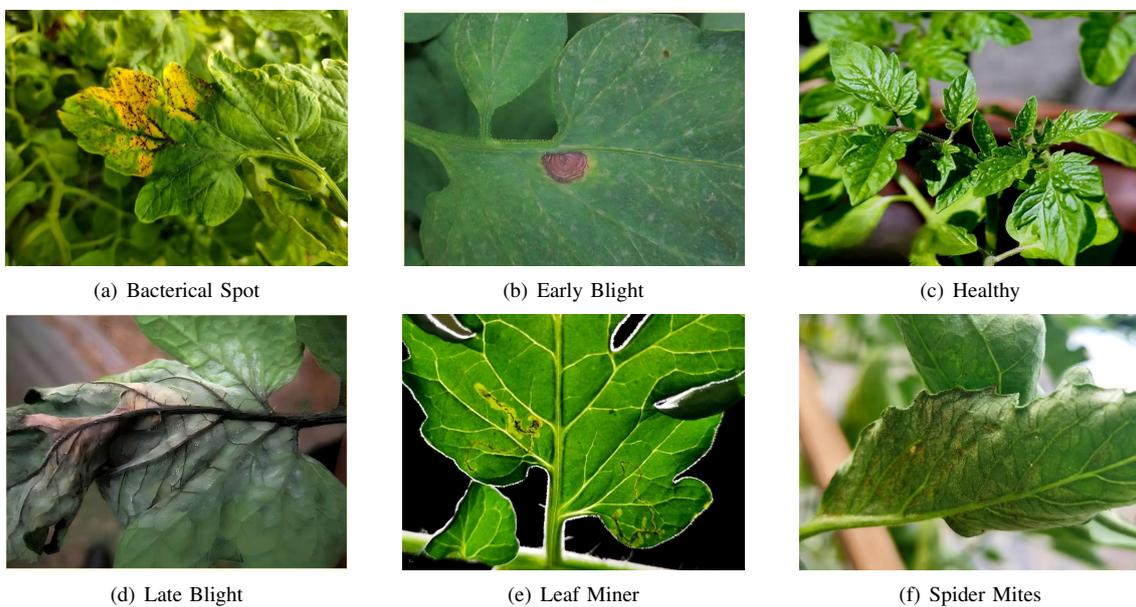


Fig. 11: Image of tomato disease.

TABLE XI: Tomato and corn disease experiments.

Data set	Models	P	R	mAP50	Params(M)	GFLOPs	FPS
Tomato	YOLOv8	65.3	33.1	41.1	3.00	8.1	87
	YOLOv8-AMD	<b>85.9</b>	<b>42.5</b>	<b>41.5</b>	<b>2.60</b>	<b>7.3</b>	<b>119</b>
Corn	YOLOv8	78.9	<b>84.1</b>	85.2	3.00	8.1	149
	YOLOv8-AMD	<b>87.0</b>	79.5	<b>86.1</b>	<b>2.60</b>	<b>7.3</b>	<b>154</b>

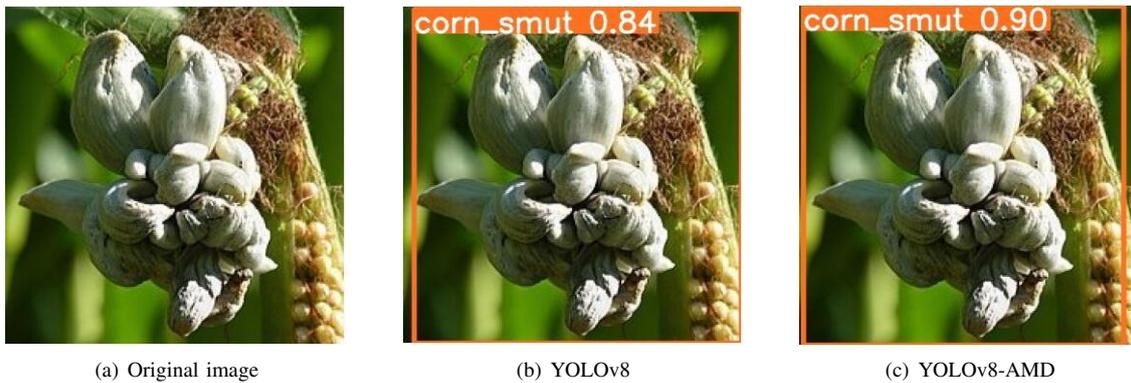


Fig. 12: Corn disease detection.

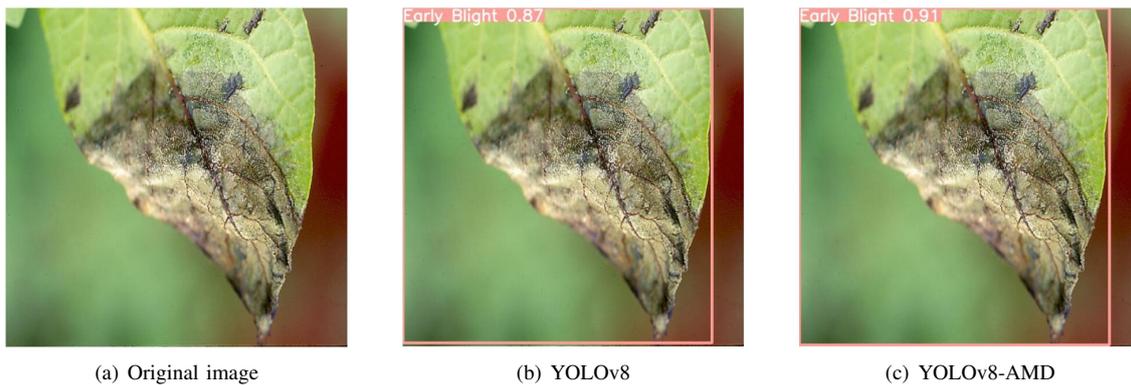


Fig. 13: Tomato disease detection.

format, enabling a comprehensive understanding of its performance across different categories. In this experiment, the confusion matrix is employed to evaluate the effectiveness of the YOLOv8-AMD detection algorithm. Constructed as a two-dimensional matrix, it represents the actual categories in the rows and the predicted categories in the columns. By examining the prediction results within different categories, various metrics including accuracy, recall, and false positive rate can be derived. Fig. 14 displays the confusion matrix, while Table 11 presents the recognition outcomes of YOLOv8-AMD for each category.

Based on the analysis of the confusion matrix presented in Fig. 14, this experiment categorized rice leaf diseases into six distinct classes: Bacterial Blight, Blast, Brown Spot, Sheath Blight, Leaf Scald, and Tungro. Through repeated experiments, the YOLOv8-AMD model achieved impressive detection accuracy for these six rice leaf diseases. Notably, the accuracy for leaf disease detection was as high as 98% for Tungro. Taking Blast as an example, the model exhibited a recognition accuracy of 90%. The false negative rate, indicating instances where rice blast was incorrectly predicted as the background, was 10%. Meanwhile, the false positive rate, representing cases where background diseases were wrongly identified as Blast, was 31%.

Upon analyzing Table XII, it becomes evident that the YOLOv8-AMD model achieved the highest P, R, and mAP50 scores for Tungro, with values of 92.6%, 97.5%, and 98.4, respectively. However, the accuracy of identifying Bacterial Blight was slightly lower at 81.5%. The recall rate and mAP50 scores were the lowest for Brown Spot, measuring 70.6% and 82.0%, respectively.

TABLE XII: Identification effect of YOLOv8-AMD on rice leaf diseases.

Classes	P	R	mAP50
Bacterial Blight	81.5	91.4	92.8
Blast	85.2	85.5	90.4
Brown Spot	90.8	70.6	82.0
Sheath Blight	91.0	90.8	93.7
Leaf Scald	91.7	96.3	98.0
Tungro	92.6	97.5	98.4

## V. DISCUSSIONS AND CONCLUSIONS

In this paper, we investigate the problem of detecting leaf diseases in rice in complex natural environments and propose an improved YOLOv8 lightweight detection algorithm called YOLOv8-AMD. Specifically, the lightweight convolutional ADown module is used to replace the convolutional Conv module in YOLOv8. The ADown module has a greater advantage in detection accuracy and model performance due to its diverse feature extraction and effective down-sampling approach; In addition, a lightweight hybrid local channel attention mechanism, MLCA, is introduced into C2f, called C2f\_MLCA, which is able to simultaneously integrate channel information and spatial information, as well as the representation effect of local information. It better perceives the information of leaf disease in the input image, improves the accuracy of rice leaf disease, and makes the model pay more attention to the features related to leaf disease and reduces the response to irrelevant information in the background. In addition, a lightweight dynamic upsampler, Dysample, is used instead of the original Upsample upsam-

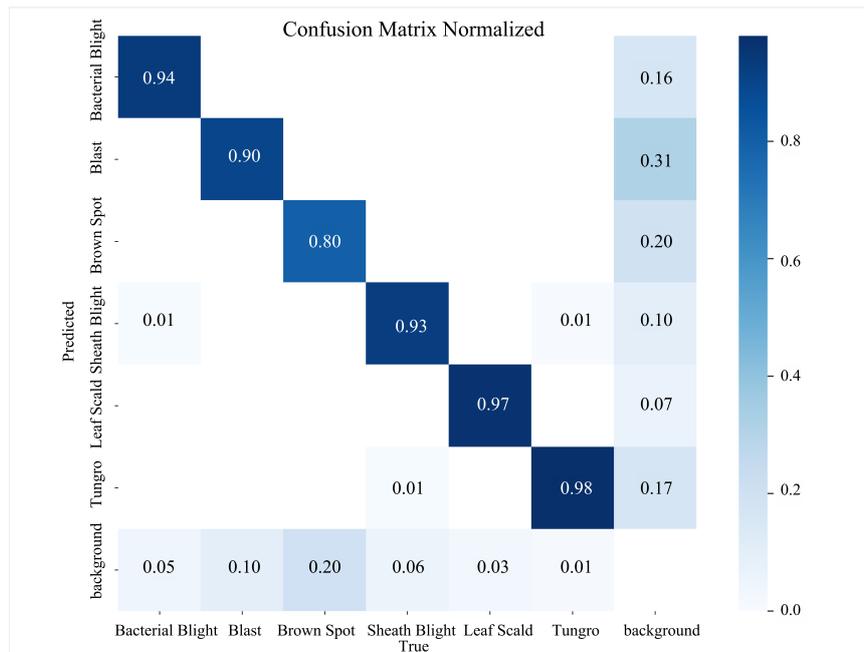


Fig. 14: confusion matrix.

pling operator to reduce the model size and improve the model accuracy while reducing the number of model parameters and floating-point computations. The YOLOv8-AMD algorithm exhibits excellent performance and considerable competitiveness, achieving an accuracy of 88.8%, a recall rate of 88.7%, and an mAP of 92.5% on the rice dataset.

In our future research, we will explore new and advanced techniques to further improve the accuracy of disease detection. For example, we will study the super-resolution technique to improve the detection of rice diseases with very small size and complex background interference, and strive to find better solutions.

REFERENCES

[1] S. Tan, H. Lu, J. Yu, M. Lan, X. Hu, H. Zheng, Y. Peng, Y. Wang, Z. Li, L. Qi *et al.*, "In-field rice panicles detection and growth stages recognition based on RiceRes2Net," *Computers and Electronics in Agriculture*, vol. 206, no. C, p. 107704, 2023.

[2] G. Ganesan and J. Chinnappan, "Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model," *Journal of Field Robotics*, vol. 39, no. 7, pp. 1085–1109, 2022.

[3] K. Kiratiratanapruk, P. Temniranrat, A. Kitvimonrat, W. Sinthupinyo, and S. Patarapuwadol, "Using deep learning techniques to detect rice diseases from images of rice fields," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, vol. 12144, 04 September, 2020, pp. 225–237.

[4] S. Lee, Y. Jeong, S. Son, and B. Lee, "A Self-Predictable Crop Yield Platform (scyp) Based on Crop Diseases Using Deep Learning," *Sustainability*, vol. 11, no. 13, p. 3637, 2019.

[5] A. Sanaeifar, M. L. Guindo, A. Bakhshipour, H. Fazayeli, X. Li, and C. Yang, "Advancing precision agriculture: The potential of deep learning for cereal plant head detection," *Computers and Electronics in Agriculture*, vol. 209, no. C, p. 107875, 2023.

[6] V. S. Kumar, M. Jaganathan, A. Viswanathan, M. Umamaheswari, and J. Vignesh, "Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLOv5 model," *Environmental Research Communications*, vol. 5, no. 6, p. 065014, 2023.

[7] D. Li, R. Wang, C. Xie, L. Liu, J. Zhang, R. Li, F. Wang, M. Zhou, and W. Liu, "A Recognition Method for Rice Plant Diseases and Pests Video Detection Based on Deep Convolutional Neural Network," *Sensors*, vol. 20, no. 3, p. 578, 2020.

[8] M. E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, and M. Paul, "Rice Leaf Disease Classification and Detection Using YOLOv5," *ArXiv*, 2022.

[9] L. Jia, T. Wang, Y. Chen, Y. Zang, X. Li, H. Shi, and L. Gao, "MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection," *Agriculture*, vol. 13, no. 7, p. 1285, 2023.

[10] N. Mishra, J. Seetha, A. G. D. Kumar, S. Menon, and A. Ravuri, "Design an Ant Lion-Based Yolo-V5 Model for Prediction and Classification of Paddy Leaf Disease," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 6s, pp. 599–612, 2023.

[11] S. Li, Z. Feng, B. Yang, H. Li, F. Liao, Y. Gao, S. Liu, J. Tang, and Q. Yao, "An intelligent monitoring system of diseases and pests on rice canopy," *Frontiers in Plant Science*, vol. 13, p. 972286, 2022.

[12] F. Aziz, F. Ernawan, M. Fakhredin, and P. W. Adi, "YOLO Network-Based for Detection of Rice Leaf Disease," in *2023 International Conference on Information Technology Research and Innovation (ICITRI)*, 16-16 August, 2023, pp. 65–69.

[13] P. Pan, W. Guo, X. Zheng, L. Hu, G. Zhou, and J. Zhang, "Xoo-YOLO: a detection method for wild rice bacterial blight in the field from the perspective of unmanned aerial vehicles," *Frontiers in Plant Science*, vol. 14, p. 1256545, 2023.

[14] B. S. Bari, M. N. Islam, M. Rashid, M. J. Hasan, M. A. M. Razman, R. M. Musa, A. F. Ab Nasir, and A. P. A. Majeed, "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework," *PeerJ Computer Science*, vol. 7, p. e432, 2021.

[15] D. C. Trinh, A. T. Mac, K. G. Dang, H. T. Nguyen, H. T. Nguyen, and T. D. Bui, "Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection," *AgriEngineering*, vol. 6, no. 1, pp. 302–317, 2024.

[16] R. Zhu, F. Hao, and D. Ma, "Research on Polygon Pest-Infected Leaf Region Detection Based on YOLOv8," *Agriculture*, vol. 13, no. 12, p. 2253, 2023.

[17] X. Liu, S. Zhou, Z. Peng, H. Zhang, and Q. Hu, "Detection of Rice Leaf Diseases Based on Improved YOLOv8n," *Academic Journal of Science and Technology*, vol. 10, no. 1, pp. 272–277, 2024.

[18] Y. Yang, Y. Xiao, Z. Chen, D. Tang, Z. Li, and Z. Li, "Fcbyolo: A lightweight and high-performance fine grain detection strategy for rice pests," *IEEE Access*, vol. 11, pp. 101 286–101 295, 2023.

[19] L. Wu, N. Lu, X. Yao, and Y. Yang, "Enhanced Helmet Wearing Detection Using Improved YOLO Algorithm," *IAENG International Journal of Computer Science*, vol. 51, no. 10, pp. 1426–1439, 2024.

[20] A. K. Sangaiyah, F.-N. Yu, Y.-B. Lin, W.-C. Shen, and A. Sharma, "UAV T-YOLO-Rice: An Enhanced Tiny YOLO Networks for Rice Leaves Diseases Detection in Paddy Agronomy," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5201–5216, 2024.

[21] D. Cheng, Z. Zhao, and J. Feng, "Rice Diseases Identification Method Based on Improved YOLOv7-Tiny," *Agriculture*, vol. 14, no. 5, p. 709, 2024.

- [22] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October, 2021, pp. 2778–2788.
- [23] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliūnas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," *Expert Systems*, vol. 38, no. 7, p. e12746, 2021.
- [24] J. Chen, W. Wang, D. Zhang, A. Zeb, and Y. A. Nanekhan, "Attention embedded lightweight network for maize disease recognition," *Plant Pathology*, vol. 70, no. 3, pp. 630–642, 2021.
- [25] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *ArXiv*, 2024.
- [26] D. Wan, R. Lu, S. Shen, T. Xu, X. Lang, and Z. Ren, "Mixed local channel attention for object detection," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106442, 2023.
- [27] Y. Zhang and Q. Ni, "A Novel Weld-Seam Defect Detection Algorithm Based on the S-YOLO Model," *Axioms*, vol. 12, no. 7, p. 697, 2023.
- [28] Y. Lan, K. Huang, C. Yang, L. Lei, J. Ye, J. Zhang, W. Zeng, Y. Zhang, and J. Deng, "Real-Time Identification of Rice Weeds by UAV Low-Altitude Remote Sensing Based on Improved Semantic Segmentation Model," *Remote Sensing*, vol. 13, no. 21, p. 4370, 2021.
- [29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October, 2021, pp. 10012–10022.
- [30] W. Liu, H. Lu, H. Fu, and Z. Cao, "Learning to Upsample by Learning to Sample," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October, 2023, pp. 6027–6037.
- [31] T. Jiang, J. Zhou, B. Xie, L. Liu, C. Ji, Y. Liu, B. Liu, and B. Zhang, "Improved YOLOv8 Model for Lightweight Pigeon Egg Detection," *Animals*, vol. 14, no. 8, p. 1226, 2024.
- [32] A. M. Roy and J. Bhaduri, "A Deep Learning Enabled Multi-Class Plant Disease Detection Model Based on Computer Vision," *Ai*, vol. 2, no. 3, pp. 413–428, 2021.
- [33] Y. He, Z. Zhou, L. Tian, Y. Liu, and X. Luo, "Brown rice planthopper (*Nilaparvata lugens* Stal) detection based on deep learning," *Precision Agriculture*, vol. 21, no. 6, pp. 1385–1402, 2020.
- [34] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory Efficient Vision Transformer with Cascaded Group Attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June, 2023, pp. 14420–14430.
- [35] Q. Wan, Z. Huang, J. Lu, G. Yu, and L. Zhang, "Seaformer: Squeeze-Enhanced Axial Transformer for Mobile Semantic Segmentation," *ArXiv*, 2023.
- [36] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking Mobile Block for Efficient Attention-based Models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1389–1400.
- [37] Q.-L. Zhang and Y.-B. Yang, "Sa-Net: Shuffle Attention for Deep Convolutional Neural Networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2235–2239.
- [38] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional Feature Fusion," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, January, 2021, pp. 3560–3569.
- [39] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June, 2020, pp. 11534–11542.
- [40] S. Deari and S. Ulukaya, "A Hybrid Multistage Model Based on YOLO and Modified Inception Network for Rice Leaf Disease Analysis," *Arabian Journal for Science and Engineering*, vol. 49, no. 5, pp. 6715–6723, 2024.
- [41] Q. Liu, W. Huang, X. Duan, J. Wei, T. Hu, J. Yu, and J. Huang, "DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n," *Electronics*, vol. 12, no. 18, p. 3892, 2023.
- [42] C. Chen, B. Wu, and Y. Shi, "An improved YOLOv5 algorithm for detecting target," *IAENG International Journal of Computer Science*, vol. 51, no. 10, pp. 1454–1461, 2024.
- [43] I. Diakhaby, M. L. Ba, and A. D. Gueye, "Pest Birds Detection Approach in Rice Crops Using Pre-trained YOLOv4 Model," in *International Conference on Innovations and Interdisciplinary Solutions for Underserved Areas*, February, 2023, pp. 223–234.