IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 June 2024, accepted 28 August 2024, date of publication 2 September 2024, date of current version 12 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3453379

RESEARCH ARTICLE

YOLOv8-ACCW: Lightweight Grape Leaf Disease Detection Method Based on Improved YOLOv8

ZUXING CHEN, JUNJIE FENG, (Member, IEEE), KUN ZHU, ZHENYAN YANG, YANHONG WANG, AND MINGYUE REN

School of Physics and Electrical Engineering, Liupanshui Normal University, Liupanshui 553004, China

Corresponding authors: Zuxing Chen (13718996928@163.com) and Junjie Feng (2651034399@qq.com)

This work was supported in part by Guizhou Provincial Department of Higher Education Science Research Project (Youth Project), Qian Jiao Ji [2022], under Grant 345; in part by the Liupanshui Science and Technology Bureau Science and Technology Innovation Talent Team Project (Liupanshui Intelligent Identification Technology Science and Technology Innovation Talent Team) under Grant 52020-2023-0-2-20; in part by the First Class Undergraduate Major Construction Project of Liupanshui Normal University under Grant LPSSYYlzy2202; in part by Guizhou Province Natural Science Foundation of China through Qiankehe Fundamentals-ZK[2021] General under Grant 322; and in part by Guizhou Provincial Department of Education Higher Education Science Research Project (Youth Project), Qian Jiao Ji [2022], under Grant 346.

ABSTRACT Grape black root, black measles, and blight are three common grape leaf diseases that significantly impact grape yield. However, current research lacks real-time detection methods for grape leaf diseases, which cannot ensure the healthy growth of grape plants. To improve the accuracy of grape leaf disease detection and enable easy deployment of the model on mobile devices, this study proposes a lightweight grape leaf disease detection method based on improved YOLOv8.Firstly, the AKConv module is employed to enable arbitrary sampling of targets of various sizes, replacing traditional convolutional (Conv) modules, thereby reducing model parameters and enhancing disease detection. Secondly, the Coordinate Attention (CA) mechanism is introduced at the end and neck of the backbone network, embedding positional information into channel attention to strengthen feature extraction capabilities and suppress irrelevant feature interference. Next, the lightweight Content-aware Reassembly of Features (CARAFE) module is introduced to improve the model's ability to extract important features. Lastly, the Wise-IoU (Weighted Interpolation of Sequential Evidence for Intersection over Union) boundary loss function replaces the original loss function, enhancing the network's bounding box regression performance and detection accuracy for small target diseases. The experimental results on a self-constructed dataset demonstrate that the improved YOLOv8-ACCW exhibits strong detection capabilities for small target disease regions. In the identification of grape leaf lesions, the model achieved F1 scores, mAP50, and mAP50-95 values of 92.4%, 92.8%, and 73.8%, respectively. Compared to the original algorithm, these metrics represent improvements of 3.1%, 3.1%, and 5.6%, respectively. The model's parameter size is only 2.8M, and its computational cost is merely 7.5G, reflecting reductions of 6.6% and 8.5%, respectively. The algorithm's detection speed reaches 143 FPS, meeting the requirements for real-time detection and enabling the rapid and accurate identification of grape leaf diseases. Through comparison with other mainstream object detection algorithms, the effectiveness and superiority of this method have been verified. This advancement can provide references for the deployment and application of mobile detection equipment such as grape leaf disease detection robots. It offers a valuable pathway to enhance the grape industry in Guizhou and ensure its healthy development.

INDEX TERMS Image recognition, deep learning, object detection, lightweight, grape leaf diseases, YOLOv8.

I. INTRODUCTION

In China, grape cultivation and consumption occupy the top position in the world. This fruit is not only an economically important crop but is also widely cultivated in

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

various regions globally [1]. Liupanshui in Guizhou, with its unique geographical location, has become a key area for grape cultivation. The grape industry has a significant impact on the local economy and social development. Nevertheless, the occurrence of grape diseases often leads to a substantial reduction in yield, posing a threat to the economic benefits of vineyards. Therefore, timely and accurate identification of diseases and the implementation of effective management and control measures are crucial [2].Currently, the detection of grape leaf diseases is still primarily conducted through manual observation. Relevant personnel need to observe the spots on the leaves with the naked eye and rely on their experience to determine whether a specific disease is present. The accuracy of disease identification depends on the experience of the personnel. Additionally, some diseased areas on grape leaves are very small, which not only increases the difficulty of detection but also greatly affects the ability of personnel to identify them, leading to a higher rate of misdiagnosis [3], [4].

With the rapid advancement and widespread application of technologies such as computer vision, machine learning, and artificial intelligence, these high-tech tools have become core elements of smart agriculture. They can automatically collect and analyze various types of information in agricultural production [5], [6], [7], [8], [9], [10]. For example, support vector machines (SVM) and K-means clustering have been employed in this context [11], [12], [13], [14], [15], [16], [17]. However, due to the complexity of image preprocessing and feature extraction, the detection efficiency of these methods remains low. Convolutional Neural Networks (CNNs), developed as end-to-end deep learning methods, fully leverage large image datasets to directly discover discriminative features from raw images. This approach avoids complex image preprocessing and reduces memory usage. Inspired by CNN breakthroughs in pattern recognition, using CNNs to identify early plant leaf diseases has become a new focus in smart agriculture [18], [19], [20], [21], [22].In recent years, deep learning-based object detection algorithms have developed rapidly. Currently, research on plant disease recognition based on computer vision can be roughly divided into two categories: the two-stage detection algorithms represented by R-CNN [23], Fast R-CNN [24], and Faster R-CNN [25], which first generate candidate regions and then recognize these regions. For instance, in 2017, Fuentes [26] proposed a deep learning-based object detection method to identify and locate tomato diseases. In this method, Faster R-CNN, R-FCN, and SSD were used to detect and identify tomato diseases and pests, with R-FCN achieving the highest mAP (mean Average Precision) of 85.98%. In 2018, Liu [27] first removed the background of grape disease images to reduce interference from non-disease areas and employed Fast-RCNN technology to extract disease spots on grape leaves, making the leaves more suitable for detection. For common grape diseases, this method achieved an average mAP of 75.52%. In 2021, He [28] used SE attention mechanism and asymmetric mixed convolution modules in Mask R-CNN to detect three types of apple diseases, achieving an average Intersection over Union (IoU) of 94.7%, and reducing the memory required for training. These studies indicate that image classification technology has made significant progress in grape disease recognition, with high accuracy in identifying specific diseases, but they primarily focus on partial disease identification and have not achieved precise annotation of disease regions.

Another category is single-stage object detection algorithms represented by SSD [29] and the YOLO series [30]. These algorithms do not require generating candidate frames and instead convert the bounding box problem into a regression problem. They utilize features extracted from the network to predict lesion locations and categories, characterized by high accuracy, fast speed, short training time, and low computational cost.In 2019, Qi [31] proposed a real-time grape detection model using the YOLOv3 model as the basic framework. This model replaced the Darknet-53 backbone of YOLOv3 with EfficientNet to effectively balance image resolution with the depth and width of the training network, ultimately achieving a recognition rate of 97.29%. In 2020, Jiang [32] expanded the apple leaf disease dataset and combined the Inception structure in GoogleNet with the SSD network. Experimental results showed that the mAP reached 78.80%, with a detection speed of 23.13 fps. In 2022, Zhuo [33] proposed a real-time apple detection model, YOLOV4-CA. In this model, MobileNet v3 was used as the backbone feature extractor, and depthwise separable convolution was introduced into the feature fusion network to improve performance, achieving an average precision of 92.23%. Huang [34] achieved citrus fruit recognition in natural environments using artistic means and proposed a citrus recognition method based on an improved YOLOv5 model. Test results showed that the model's average accuracy was 91.3%.These studies confirm the feasibility of object detection technology in plant disease and pest detection. However, they have paid less attention to grape diseases and face challenges such as the difficulty of collecting large-scale disease image data, low detection accuracy of existing models, and the inapplicability of improved methods to detect early small lesions.

In 2023, YOLOv8 rapidly emerged, and many researchers achieved good results using YOLOv8 for object detection. Dehuan [35] proposed the Light-SA YOLOV8 (Lightweight Self-Attention YOLOV8) model, which addresses challenges posed by complex backgrounds by incorporating the BRA self-attention mechanism module before the SPPF layer in the backbone. Additionally, to simplify computational complexity, the backbone network integrated FasterNet blocks. To improve the accuracy and computational efficiency of citrus pest and disease detection, an AFPN (Asymptotic Feature Pyramid Network) structure was introduced in the neck. The average detection accuracy for six pest and disease categories in the test dataset was 92.6%.Ye [36] proposed an improved

YOLOv8 lesion detection method focusing on detecting tea leaf wilt, tea white spot, tea anthracnose, and tea ring spot diseases. The method enhances the feature extraction capability of the YOLOv8 network framework by introducing a Receptive Field Concentrated Attention Module (RFCBAM) to replace C2f in the backbone network. Additionally, a Mixed Pooling SPPF (MixSPPF) module was proposed to enhance information mixing between different level features. In the neck network, the RepGFPN module replaced the C2f module to further enhance feature extraction. A dynamic head module embedded in the detection head applied multiple attention mechanisms to improve multi-scale spatial localization and multi-task perception capabilities. The internal IoU loss function was adopted to replace the original CIoU, enhancing the learning ability for small lesion samples. Furthermore, the AKConv module replaced the traditional Conv module, allowing for arbitrary sampling of targets of various sizes, thus reducing model parameters and enhancing disease detection.

To integrate computer vision technology with dynamic identification of grape leaf diseases, this study focuses on grape black root, black measles, and blight as research subjects. A lightweight grape leaf disease detection method based on an improved YOLOv8 is proposed. This method, based on the YOLOv8 network framework, introduces several enhancements. Firstly, the AKConv module replaces the traditional Conv module, allowing for arbitrary sampling of targets of various sizes, thereby reducing model parameters and strengthening disease detection. Secondly, the Coordinate Attention (CA) mechanism is introduced at the end of the backbone network and in the neck network. This mechanism embeds positional information into channel attention, enhancing feature extraction capabilities and suppressing interference from irrelevant features. Thirdly, the lightweight general upsampling content-aware reassembly CARAFE module is introduced to improve the model's ability to extract important features. Finally, the Wise-IoU boundary loss function replaces the original loss function, enhancing the network's bounding box regression performance and improving the detection of small target diseases.

II. RELEVANT PRINCIPLES

A. YOLOv8 ALGORITHM STRUCTURE AND WORKING PRINCIPLE

The YOLOv8 network is mainly composed of Backbone, Neck, and Head components. The network structure is shown in Figure 1.

The backbone is primarily used for feature extraction. YOLOv8 replaces the Cross Stage Partial (CSP) module from YOLOv5 with a lightweight C2f module, enhancing feature expression capabilities through a dense residual structure. It changes the number of channels via split and concatenate operations based on the scaling factor, reducing computational complexity and model size. The end of the backbone employs a fast spatial pyramid pooling layer (SPPF) to increase the receptive field and capture multi-level feature information from the scene.

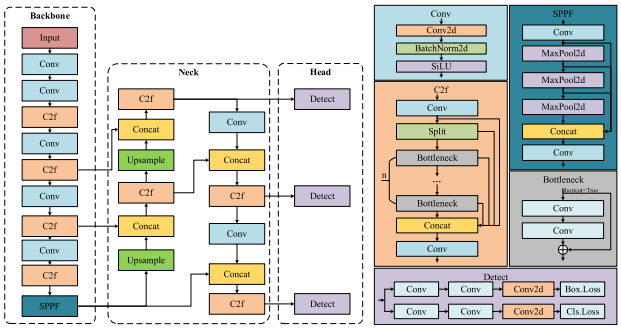


FIGURE 1. YOLOv8n network architecture.

The neck part is mainly used for feature fusion, utilizing a path aggregation network and C2f module to fuse feature maps of different scales output by the three stages of the backbone, aiding in the aggregation of shallow information into deeper features.

The head part uses a decoupled head structure, divided into classification and localization prediction branches, to alleviate the conflict between classification and localization tasks. An anchor-free framework is adopted to improve detection performance, which is advantageous for detecting targets with irregular aspect ratios.

Loss calculation includes classification loss (Varifocal Loss) and regression loss (Complete IoU Loss + Distribution Focal Loss). Varifocal Loss, based on the focal loss function, handles class imbalance better, improving the model's detection accuracy. CIOU Loss handles the overlap between predicted and ground truth boxes more effectively. DFL Loss addresses class imbalance and background class issues, enabling the network to quickly focus on areas close to the target location's distribution.

B. PROBLEMS WITH THE ALGORITHM

When applying the YOLO algorithm to the rapid identification of grape leaf diseases, it is necessary to consider the limited hardware performance of mobile devices. Simply choosing the YOLOv8x algorithm is impractical because, although it offers the best detection performance, it cannot meet the real-time requirements of grape leaf disease detection tasks. Therefore, this study selects the smallest weight model of the YOLOv8n algorithm. However, during practical detection, several issues arise with the YOLOv8n algorithm. Firstly, the YOLOv8n algorithm uses a large number of standard convolutions and C2f modules, which, while improving accuracy, reduces running speed and increases the model's parameters. Secondly, rapidly changing scenes in mobile detection require sufficient detection precision. However, the YOLOv8n algorithm does not perform ideally in detecting grape leaf diseases, often resulting in false positives and missed detections.

III. YOLOv8 MODEL IMPROVEMENT

To enhance the accuracy of grape leaf disease detection, a lightweight grape leaf disease detection method based on improved YOLOv8 is proposed. Table 1 lists the number of layers, parameter count, module names, and parameters for YOLOv8-ACCW. Figure 2 illustrates the modified YOLOv8n structure. Firstly, the AKConv module replaces the traditional convolution Conv module, allowing arbitrary sampling of targets of various sizes, thereby reducing model parameters and enhancing disease detection. Secondly, a coordinate attention mechanism (CA) is introduced after C2f in the neck part of YOLOv8n, embedding positional information into channel attention to strengthen feature extraction and suppress irrelevant features. Next, the lightweight general up-sampling content-aware reassembly (CARAFE) module is introduced to improve the model's capability to extract important features. Lastly, the Wise-IoU boundary loss function replaces the original loss function to enhance network bounding box regression performance and improve detection accuracy for small target diseases.

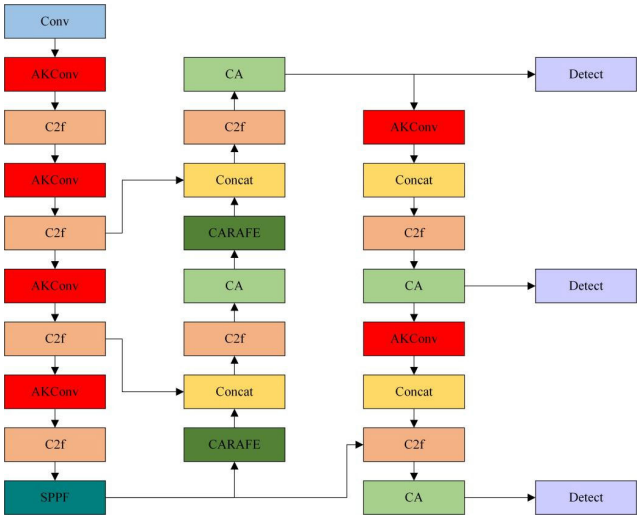


FIGURE 2. YOLOv8-ACCW network architecture.

A. AKConv MODULE

Currently, convolutional neural networks (CNNs) utilizing convolution operations have made remarkable strides in the field of deep learning. However, traditional convolution operations still face limitations. Firstly, the fixed shape and size of the sampling window restrict the convolution operation to a local window, hindering the capture of information from other locations. Secondly, the fixed size of the convolution kernel, set at k×k square, leads to exponentially increasing parameter calculations with size, posing challenges for constructing lightweight models. To address these issues, this study introduces Adaptive Kernel Convolution (AKConv), which allows the convolution kernel to use an arbitrary number of parameters and sampling shapes. This not only enhances model performance but also reduces the number of model parameters. The structure is illustrated in Figure 3.

TABLE 1. Parameters of YOLOv8-ACCW layers.

	from	n	params	module	arguments
0	-1	1	464	Conv	[32, 64, 6, 2]
1	-1	1	4876	AKConv	[16, 32, 6, 2]
2	-1	1	7360	C2f	[32, 32, 1, True]
3	-1	1	15884	AKConv	[32, 64, 6, 2]
4	-1	2	49664	C2f	[64, 64, 2, True]
5	-1	1	56332	AKConv	[64, 128, 6, 2]
6	-1	2	197632	C2f	[128, 128, 2, True]
7	-1	1	210956	AKConv	[128, 256, 6, 2]
8	-1	1	460288	C2f	[256, 256, 1, True]
9	-1	1	164608	SPPF	[256, 256, 5]
10	-1	1	74312	CARAFE	[256]
11	[-1, 6]	1	0	Concat	[1]
12	-1	1	148224	C2f	[384, 128, 1]
13	-1	1	3352	CA	[128, 128]
14	-1	1	66120	AKConv	[128]
15	[-1, 4]	1	0	Concat	[1]
16	-1	1	37248	C2f	[192, 64, 1]
17	-1	1	1688	CA	[64, 64]
18	-1	1	31628	AKConv	[64, 64, 6, 2]
19	[-1, 13]	1	0	Concat	[1]
20	-1	1	123648	C2f	[192, 128, 1]
21	-1	1	3352	CA	[128, 128]
22	-1	1	112396	AKConv	[128, 128, 6, 2]
23	[-1, 9]	1	0	Concat	[1]
24	-1	1	493056	C2f	[384, 256, 1]
25	-1	1	6680	CA	[256, 256]
26	[17, 21, 25]	1	752092	Detect	[4, [64, 128, 256]]

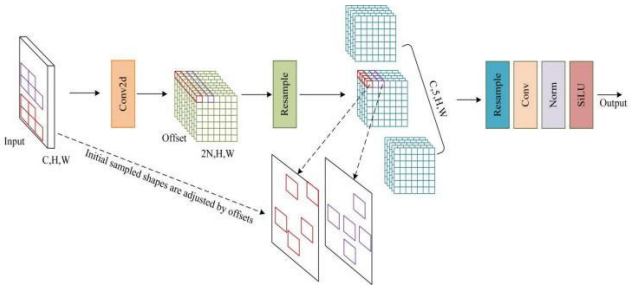


FIGURE 3. AKConv structure.

In AKConv, the input image size is set to (C, H, W), where C represents the number of channels, and H and W represent the height and width of the image, respectively. The convolution operation initially applies the initial sampling shape of the convolution kernel to the input image using Conv2d. Subsequently, the initial sampling shape is adjusted by learned offsets, a critical step in AKConv that enables the convolution kernel's shape to dynamically adapt to image features. Following this adjustment, AKConv resamples the feature map based on the modified sampling shape. The resampled feature map undergoes reshaping, convolution, normalization, and activation by the SiLU function to generate the final output.

AKConv innovatively improves the design of convolution kernels to enhance the adaptability and efficiency of convolutional networks. Unlike traditional convolution kernels, these kernels do not have fixed sizes and shapes but can dynamically adjust based on the density and characteristics of diseases to determine the required number of parameters. Considering different types of diseases with varying sizes and distributions of lesions, AKConv automatically adjusts the size of the convolution kernel during processing to effectively capture the sizes and shapes of various diseases, thereby enhancing feature extraction efficiency. Adaptive sampling shapes are illustrated in Figure 4. Furthermore, by designing different initial sampling shapes for a 5 × 5 sampling grid, AKConv accurately covers and processes different image regions, thereby improving the accuracy of feature extraction, as shown in Figure 5. Additionally, AKConv can adjust the position of the convolution kernel using offsets to adapt to local feature variations in different positions of the target image, better accommodating non-rigid deformations, occlusions, and complex backgrounds. This capability forms a solid foundation for enhancing disease detection, as depicted in Figure 6.

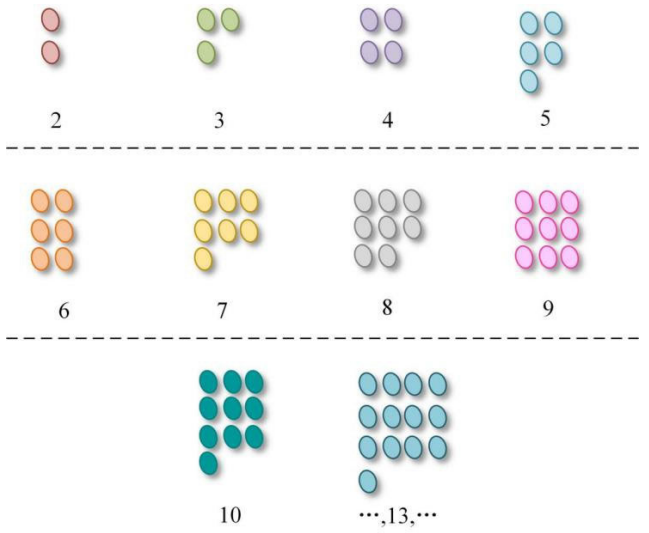


FIGURE 4. Initial sampling shape.

B. CA MODULE

Due to the complex and variable environment of grape leaf diseases, to enhance the model's ability to express features of grape leaf diseases, the CA attention mechanism was added between every two feature fusions at the end and neck of the YOLOv8 backbone network. This local feature enhancement enables the network to ignore irrelevant information interference and ensures that the fused feature maps contain more effective information. Attention mechanisms are generally divided into channel attention mechanisms, spatial attention mechanisms, and combinations of both. Traditional attention mechanisms like Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM) have many shortcomings. SE attention focuses only on building interdependencies between channels, ignoring spatial features. CBAM introduces large-scale convolution kernels to extract spatial features but overlooks long-range dependency issues. While other attention modules without these problems have good performance, they have too many parameters and are not suitable for application deployment.

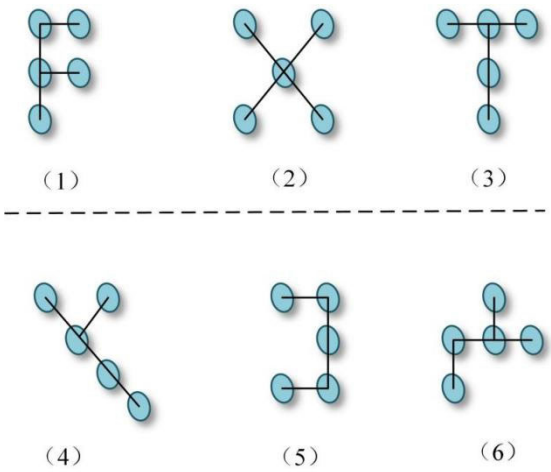


FIGURE 5. The 5 × 5 different initial sample shapes.

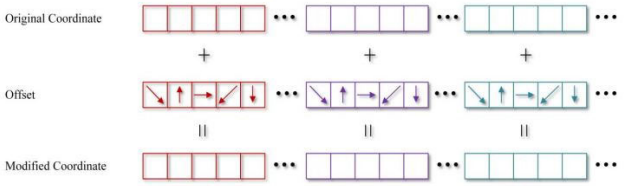


FIGURE 6. Offset adjusts the sample shape.

The CA module is a new attention module proposed for channel attention, which not only captures inter-channel information but also directional and positional perception. This helps the model to more accurately locate and identify the target of interest. The CA module is highly flexible and can be added at multiple positions in existing models without introducing excessive parameters or computational overhead. Its structure is depicted in Figure 7, where inputs are horizontally and vertically pooled to maintain long-distance dependencies in both directions. Subsequently, the stitched information from both directions undergoes segmentation and convolution to simultaneously focus on horizontal and vertical directions. The two resulting feature maps precisely indicate the rows and columns of the target object of interest.

C. CARAFE UPSAMPLING

In the YOLOv8 object detection network model, the feature pyramid structure uses nearest-neighbor interpolation for upsampling. However, this method relies solely on the spatial positions of pixels to determine the upsampling kernel,

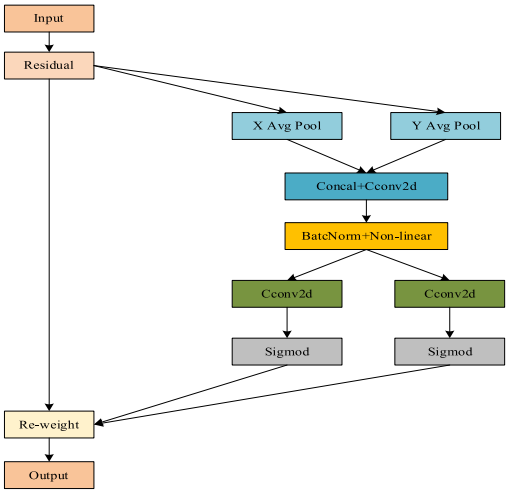


FIGURE 7. CA module.

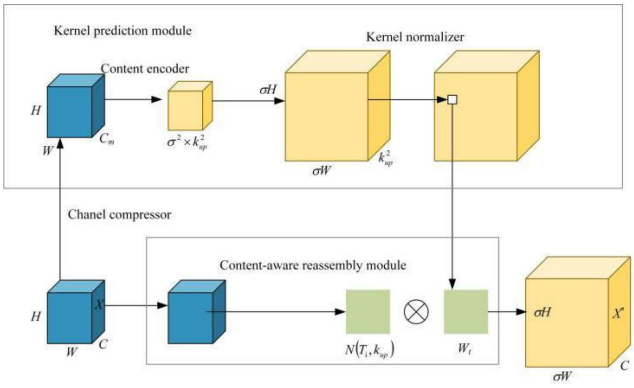
lacking mechanisms to fully utilize feature information. To address this issue, Wang et al. proposed the content-aware feature reassembly (CARAFE) structure, which incorporates a larger receptive field. CARAFE not only aggregates contextual information better and utilizes feature details based on the input feature map but also links the upsampling kernel with semantic information, enhancing perception of important content while maintaining model lightweight due to its reduced computational overhead. Through efficient upsampling operations, CARAFE preserves fine-grained image details effectively, thereby improving the model's accuracy in recognizing disease features.

CARAFE upsampling consists primarily of the kernel prediction module and the content-aware reassembly module. In the kernel prediction module, initially, the number of channels of input feature map $H \times W \times C$ is reduced to C_m, resulting in feature map $C_m \times H \times W$, which helps in reducing subsequent computational workload. Then, content encoding is performed using a $k_{up} \times k_{up}$ convolutional kernel to generate the reassembled kernel, producing a $\sigma^2 \times k_{up}^2$ feature map, as shown in Equation (1).

$$C_m = \sigma^2 \times k_{up}^2 \tag{1}$$

In the equation, C_m represents the number of channels in the dimensionally reduced feature layer, σ denotes the upsampling factor (typically 2), and $\sigma H \times \sigma W \times k_{up}^2$ indicates the predicted size of the upsampling kernel. Next, the channels are expanded in the spatial dimension and rearranged to form a $4 \times 44 \setminus$ times 44×4 upsampling kernel, followed by softmax normalization to ensure the weights sum up to 1.

In the equation, C_m represents the number of channels in the dimension-reduced feature layer, σ is the upsampling factor (usually 2), and k_{up} denotes the predicted size of the upsampling kernel. Subsequently, the channels are expanded in the spatial dimension, then rearranged to form an upsampling kernel of size $\sigma H \times \sigma W \times k_{up}^2$, followed by Softmax normalization to ensure their weights sum to 1. Within the content-aware reassembly module, the obtained upsampling kernel is utilized for feature reassembly to extract target features. For each target position in the output feature map, a region of size $k_{up} \times k_{up}$ centered on the target is taken, and a dot product is performed with the predicted upsampling kernel at that point, mapping it back to the input feature map to obtain a feature map of size $\sigma H \times \sigma W \times C$. The CARAFE upsampling network structure is illustrated in Figure 8.



Note: H represents the height of the input feature map, W denotes the width of the input feature map, C_m indicates the number of channels in the dimension-reduced feature layer, C is the length of the original input feature map, X represents the original input feature map, σ stands for the upsampling factor, k_{up} denotes the size of the reassembly kernel, X' signifies the reassembled feature map, T_i refers to the position in the input feature map, and N represents a square region.

FIGURE 8. CARAFE upsampling network structure.

Due to CARAFE upsampling's ability to generate distinct upsampling kernels for different features and to focus on the distribution of features across the global feature map, this study replaces the original nearest-neighbor interpolation upsampling module in the YOLOv8n model with CARAFE upsampling. This replacement enhances the model's ability to recognize important features during upsampling without adding extra parameters or computational overhead, thereby improving the network's capability to extract disease-related features.

D. THE WISE-IoU LOSS FUNCTION

In grape leaf disease detection tasks, black measles disease, characterized by small pathological features, constitutes a significant proportion. Detecting these small disease features is crucial for the entire disease detection task. Traditional loss functions only consider the intersection over union (IoU) between predicted and ground truth bounding boxes during calculation, neglecting classification information. Therefore, by designing an appropriate loss function, model detection accuracy can be improved. YOLOv8 employs DF Loss and CIoU Loss to compute bounding box regression losses. CIoU uses a monotonic focal mechanism but lacks consideration for balancing hard and easy examples. When the target detection training set contains low-quality examples, this can degrade model performance. The computation of CIoU is shown

in equation (2).

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2\left(b^A, b^B\right)}{c^2} + \alpha v \tag{2}$$

In the equation, b^A and b^B respectively represent the centers of the predicted and ground truth bounding boxes, ρ denotes the Euclidean distance between the two points, c represents the diagonal length of the minimum bounding rectangle of the predicted and ground truth boxes; α is a balancing parameter; v is used to calculate the consistency of aspect ratios between predicted and target bounding boxes. This study introduces the Wise-IoU loss function with a dynamic non-monotonic focusing mechanism to balance samples. By replacing IoU with outlier-aware anchor box quality assessment through a dynamic non-monotonic focusing mechanism, the model avoids excessive penalties from geometric factors (such as distance and aspect ratio), as shown in equations (3) to (5).

$$L_{\mathrm{WIoU}} = rR_{\mathrm{WIoU}}L_{\mathrm{IoU}}, r = \frac{\beta}{\delta\alpha^{\beta-\alpha}} \tag{3}$$

$$\beta = \frac{L^*_{\mathrm{IOU}}}{\overline{L_{\mathrm{IoU}}}} \in [0, +\infty) \tag{4}$$

$$R_{\mathrm{WIoU}} = \exp\left(\frac{\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2}{\left(c_w^2 + c_h^2\right)^*}\right) \tag{5}$$

In the equation, $L_{\mathrm{IoU}} \in [0, 1]$ represents the IoU loss, which weakens the penalty for high-quality anchor boxes, enhances focus on the distance between centers when the overlap between anchor and predicted boxes is high; $R_{\mathrm{WIoU}} \in [1, \exp]$ represents the penalty term in Wise-IoU, strengthening the loss for average-quality anchor boxes. The superscript $*$ indicates non-participation in backpropagation, effectively preventing gradients that may prevent the network model from converging. $\overline{L_{\mathrm{IoU}}}$ acts as a normalization factor, representing the incremental moving average. β represents the outlier score, where a lower value indicates higher anchor box quality, assigning a smaller gradient boost. Simultaneously, smaller gradient boosts are allocated to predicted boxes with larger outlier values, effectively reducing harmful gradients for low-quality training samples. This ensures that the bounding box regression loss focuses on average-quality anchor boxes, thereby enhancing overall network performance.

IV. EXPERIMENTAL DESIGN AND RESULT ANALYSIS

A. DATA COLLECTION

This paper collected grape leaf disease images from the Grape Industry Demonstration Park in Zhongshan District, Liupanshui City, Guizhou Province, China, totaling 10,000 images. The images were captured using a Canon EOS 550D camera with a resolution of 640 × 640 pixels and saved in.JPG format. Among the 10,000 images, there are four types: black measles, black root, blight, and healthy leaves. All 10,000 images are of high quality and have been annotated. The grape leaf disease annotation work was performed using the labelImg annotation software, marking the positions of different types of lesions in the images. The dataset was then randomly divided into training, validation, and test sets in an 8:1:1 ratio, and formatted in YOLO format. Specifically, 8,000 images were used for training, 1,000 for validation, and 1,000 for testing. Annotations and visual experiments were conducted for the four types as shown in Figure 10. Each matrix cell in the figure represents a label used in model training, with cell color depth reflecting the correlation between respective labels. Darker cells indicate stronger learning of correlations between labels, while lighter cells indicate weaker correlations. Panel A shows a histogram of category counts in the dataset; B illustrates the lengths and widths of each labeled box when all x and y values of the labels are set to the same position; C depicts the distribution of x and y values in the images; D shows the ratio of label width to height in the dataset; and E provides detailed information on the distribution of labels in the original dataset. Analysis reveals an uneven distribution of diseases in the self-built dataset. The precise localization of rectangular annotation boxes indicates the proposed method's suitability for detecting grape leaf diseases in the Liupanshui region.

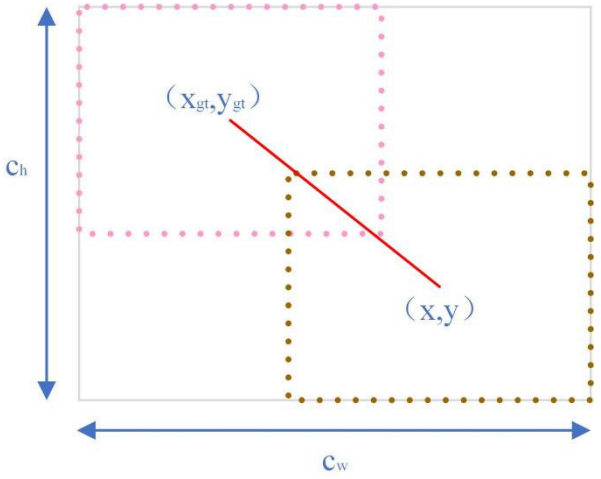


FIGURE 9. Schematic diagram of Wise-IoU parameters.

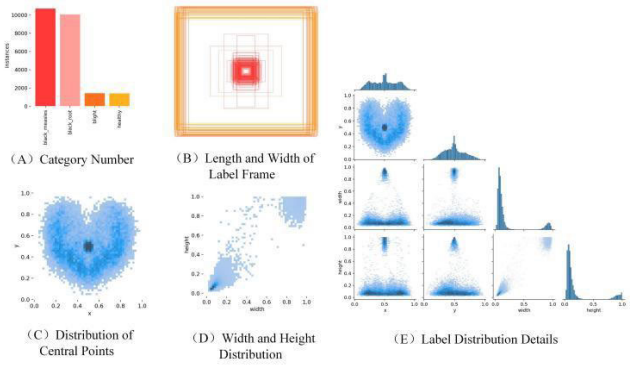


FIGURE 10. Dataset annotation file statistics and visualization.

Sample images of grape leaf disease are shown in Figure 11.

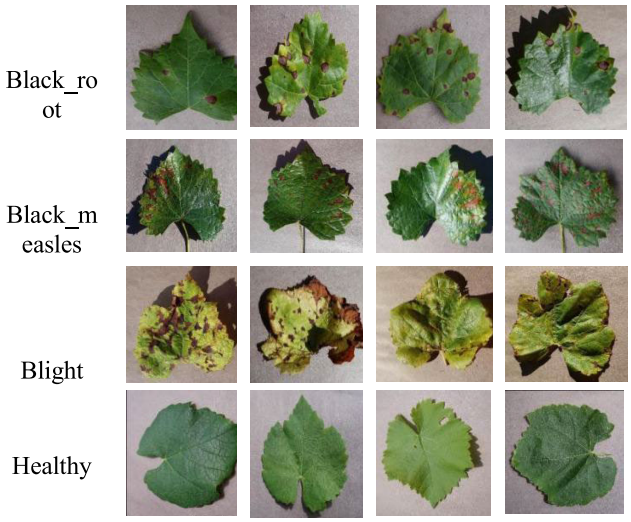


FIGURE 11. Examples of grape leaf disease images.

B. PRODUCTIONEXPERIMENTAL ENVIRONMENT AND HYPERPARAMETER SETTINGS

C. EVALUATION METRICS

This paper uses evaluation metrics including F1 score, mean Average Precision (mAP), number of parameters (Params), Giga Floating-point Operations per second (GFLOPs), and Frames Per Second (FPS). Among these, precision and recall are used as basic metrics, with F1 score and mAP calculated based on precision and recall serving as the final evaluation metrics to measure model recognition accuracy. GFLOPs measure the complexity of the model or algorithm, while Params indicate the model size. Generally, smaller Params and GFLOPs indicate lower computational requirements and easier construction on low-end devices with lower hardware performance demands. FPS refers to the number of frames detected per second, influenced by algorithm weights and experimental hardware configurations.

TABLE 2. Hyperparameter settings.

Hyperparameters	Configuration
Image size	640×640
Epoch	100
Batch size	8
lr0	0.01
lrf	0.01

Precision is the ratio of correctly predicted positive observations among all predicted positive results, defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

Recall is the ratio of correctly predicted positive observations to all observations in the class, defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

where TP is the number of correctly predicted targets, FP is the number of incorrectly predicted targets, and FN is the number of targets that were not predicted correctly.

The formula for average precision across n classes is:

$$\text{mAP} = \frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1} Precison(Recall)d(Recall) \tag{8}$$

The F1-score combines precision and recall, providing a comprehensive measure of overall network performance. It is computed as the harmonic mean of the two metrics:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

Model complexity is measured by the number of parameters (Params) and computational complexity (GFLOPs), while inference speed is measured by Frames Per Second (FPS). FPS represents the number of frames detected per second, influenced not only by algorithmic weights but also by the hardware configuration of the experimental setup.

TABLE 3. Experimental setup.

Environment Configuration	Name	Information
Hardware configuration	CPU	Intel（R）Core（TM）i5-13600
	GPU	NVIDIA RTX 3060Ti
	Memory	8.0GB
Software configuration	Operating system	Windows 11
	System environment	PyCharm
	Python version	3.10.9
	Pytorchversion	1.12.1
	CUDAversion	11.7

D. THE IMPACT OF DIFFERENT ATTENTION MECHANISMS ON NETWORK PERFORMANCE

Attention mechanisms can generally be divided into channel attention mechanisms, spatial attention mechanisms, and their combinations. Traditional attention mechanisms like Squeeze-and-Excitation attention (SE) and Convolutional Block Attention Module (CBAM) have several shortcomings. SE attention focuses solely on constructing inter-channel dependencies and ignores spatial features. Although CBAM introduces large convolutional kernels to extract spatial

features, it neglects long-range dependency issues. Other attention modules that address this problem often have a high number of parameters, making them unsuitable for application deployment.Experiments were conducted comparing the SE attention module and CBAM attention module with the CA module. The experimental results are shown in Table 4.

As seen in Table 4, compared to other attention mechanism modules studied, such as the SE and CBAM attention modules, the inclusion of the CA attention module significantly improved the network's detection accuracy. This is because the CA attention module, proposed specifically for channel attention, captures not only cross-channel information but also directional and positional awareness, aiding the model in more accurately locating and recognizing targets. By adding the CA attention mechanism between every two feature fusions at the end of the YOLOv8 backbone and neck, local feature enhancement is achieved, allowing the network to ignore irrelevant information interference and resulting in feature maps that contain more effective information.

TABLE 4. Comparison results of multiple attention mechanisms.

Algorithms	F1	mAP50	mAP50-95	Params/M	GFLOPs
YOLOv8n	89.3	89.7	68.2	3	8.2
YOLOv8+SE	89.8	88.7	68.1	3.1	8.2
YOLOv8+CBAM	89.7	89.3	69.4	3.2	8.3
YOLOv8+CA (ours)	91.2	89.9	71.6	3.1	8.2

E. THE IMPACT OF THE AKConv MODULE ON NETWORK PERFORMANCE

To verify the relationship between the introduction of the AKConv module at different positions and network performance, an ablation experiment was designed as shown in Table 5. AKConv-backbone refers to replacing the Conv modules in the backbone network with AKConv modules while keeping the neck network unchanged. AKConv-neck indicates that the Conv modules in the neck network are replaced with AKConv modules, with the Conv modules in the backbone network remaining unchanged. AKConv-all means replacing the Conv modules in both the backbone network after the first Conv module and the neck network with AKConv modules.The introduction of the AKConv module into the YOLOv8n backbone network mainly provides a flexible convolution mechanism. AKConv allows convolution kernels to have any number of parameters, enabling them to be adjusted in size and shape according to actual needs, thereby adapting more effectively to changes in targets. For different sizes of convolution kernels, AKConv proposes a new algorithm to generate initial sampling coordinates, further enhancing its flexibility in handling various target sizes. To adapt to different changes in targets, AKConv adjusts the sampling positions of irregular convolution kernels based on obtained offsets, thereby improving the accuracy of feature extraction. AKConv supports the linear increase and decrease of convolution parameters, helping to optimize performance in hardware environments, especially suitable for lightweight model applications. This improvement breaks through the limitations of traditional convolutions, which are restricted to fixed local windows and fixed sampling shapes, allowing convolution operations to adapt more precisely to different datasets and target positions.Experimental results indicate that replacing the Conv modules in both the backbone network after the first Conv module and the neck network with AKConv modules yields the best performance.

TABLE 5. Comparison results of different positions of C2FGHOST.

Algorithms	F1	mAP50	mAP50-95	Params/M	GFLOPs
YOLOv8n	89.3	89.7	68.2	3	8.2
AKConv-backbone	86.4	86.9	68.5	2.6	7.3
AKConv-neck	88.2	88.8	69.4	2.5	6.9
AKConv-all(ours)	91.4	89.5	71.8	2.3	6.4

F. ABLATION EXPERIMENT COMPARISON

To validate the accuracy of the proposed improved algorithm, experiments were conducted by establishing several models: YOLOv8n, YOLOv8n-A, YOLOv8n-AC, YOLOv8n-ACC, and YOLOv8n-ACCW. Among them, YOLOv8n-A employs the AKConv module in YOLOv8n, allowing arbitrary sampling for targets of various sizes, replacing the traditional Conv module, thus reducing model parameters and enhancing disease detection. YOLOv8n-AC utilizes the AKConv module to replace the traditional Conv module and incorporates the Coordinate Attention (CA) mechanism at the end of the backbone network and in the neck network to embed positional information into channel attention, thereby enhancing feature extraction and suppressing irrelevant features. YOLOv8n-ACC uses the AKConv module, replaces the traditional Conv module, introduces the CA mechanism at the end of the backbone network and in the neck network, and then incorporates the lightweight general upsampling content-aware reassembly (CARAFE) module to improve the model's ability to extract important features. YOLOv8n-ACCW uses the AKConv module, replaces the traditional Conv module, introduces the CA mechanism at the end of the backbone network and in the neck network, incorporates the CARAFE module, and finally uses the Wise-IoU boundary loss function to replace the original loss function. YOLOv8n-ACCW is the algorithm proposed in this paper.

As shown in Table 6, the improved algorithm adopts a more efficient network structure to enhance the YOLOv8n network structure, improving accuracy and reducing model parameters and computational complexity. It also proves that the AKConv module does not reduce the algorithm's accuracy but rather decreases model parameters and computational load. The introduction of the lightweight general upsampling content-aware reassembly (CARAFE) module improves the model's ability to extract important features,

TABLE 6. Results of ablation experiment.

Algorithms	AKConv	CA	CARAFE	Wise-IoU	F1	mAP50	mAP50-95	Params/M	GFLOPs	FPS
YOLOv8n	×	×	×	×	89.3	89.7	68.2	3	8.2	147
YOLOv8n-A	√				89.8	90.1	69.3	2.4	6.4	145
YOLOv8n-AC	√	√			89.8	91.5	70.4	2.6	7.6	156
YOLOv8n-ACC	√	√	√		90.2	92.3	72.7	2.7	7.3	140
YOLOv8n-ACCW	√	√	√	√	92.4	92.8	73.8	2.8	7.5	143

while the CA attention mechanism only slightly increases the parameters, effectively enhancing detection accuracy. The use of the Wise-IoU boundary loss function plays a crucial role in detecting small target diseases. Combining these improvements with the YOLOv8n algorithm minimizes the model size, with parameters reduced to only 2.8M and computational complexity to 7.5G, respectively reducing by 6.6% and 8.5%. This effectively lowers the difficulty and cost of deploying the model on mobile terminals, and while meeting real-time requirements, significantly enhances accuracy.

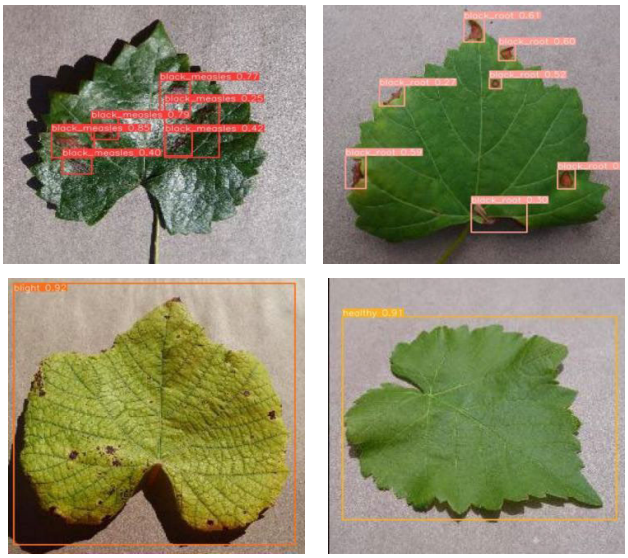
G. PRACTICAL APPLICATION DETECTION EVALUATION

Diseases were captured in the grape leaf disease dataset. Figure 12 shows the detection results of the YOLOv8-ACCW algorithm on the grape leaf disease dataset. The improved algorithm identifies more instances of grape leaf diseases compared to the original algorithm and provides more precise detection. The results indicate that the improved YOLOv8-ACCW algorithm can effectively detect grape leaf disease targets and accurately identify their locations, demonstrating strong robustness and accuracy.

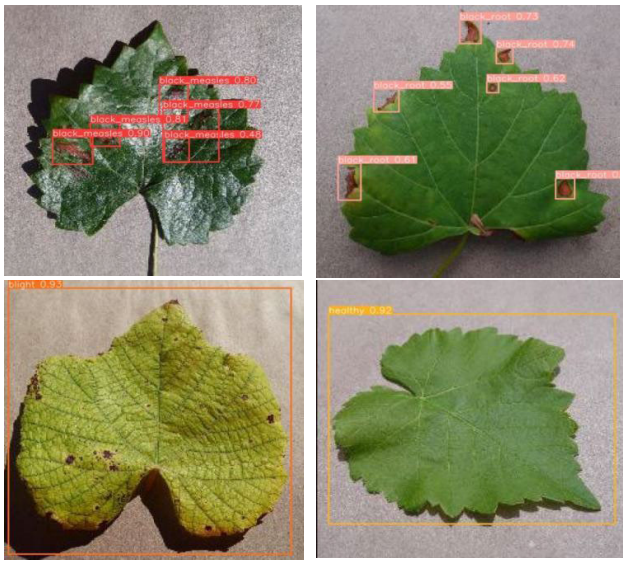
To further validate the model's detection performance on grape leaf disease targets, Table 7 compares the performance of YOLOv8n and the improved YOLOv8-ACCW model in detecting grape leaf diseases. The data show that YOLOv8-ACCW has higher detection accuracy for grape leaf diseases than YOLOv8n. Compared to YOLOv8n, the mAP50 and mAP50-90 improved by 3.1% and 5.2%, respectively.

Based on YOLOv8n, the training and validation process curves of the grape leaf disease detection model are shown in Figure 13. From Figure 13, it can be observed that during the training and validation processes, the loss curve initially decreases rapidly and then gradually stabilizes. Meanwhile, the validation metrics such as accuracy, recall rate, and mAP (mean Average Precision) show a rapid increase followed by a tendency to plateau and stabilize.

Based on YOLOv8-ACCW, the training and validation process curves of the grape leaf disease detection model are shown in Figure 14. From Figure 14, it can be observed that the convergence of the relevant curves is well demonstrated, which confirms the reliability of the grape leaf disease detection model based on YOLOv8-ACCW.



(a) Original algorithm detection results



(b) Improved algorithm detection results

FIGURE 12. Detection effect.

The experiment provides a comparison of the training and validation process curves between the YOLOv8n and YOLOv8-ACCW models, as shown in Figure 13. It can be

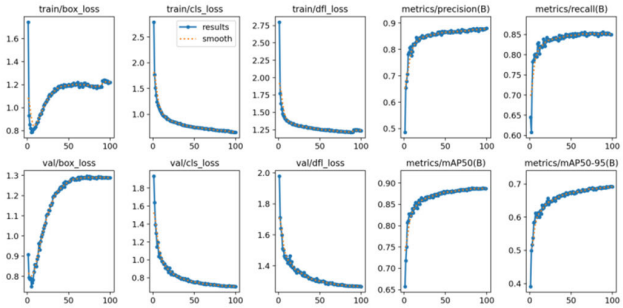


FIGURE 13. Training and validation process curves of grape leaf disease detection model based on YOLOv8n.

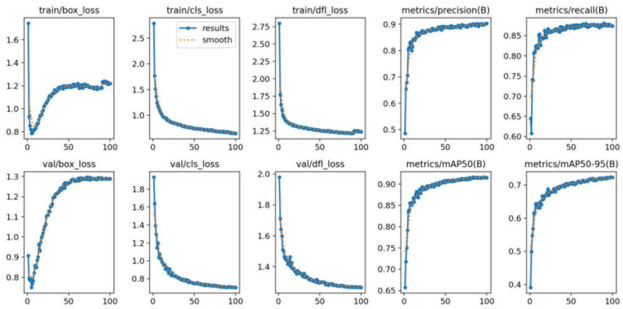


FIGURE 14. Training and validation process curves of the grape leaf disease detection model based on YOLOv8-ACCW.

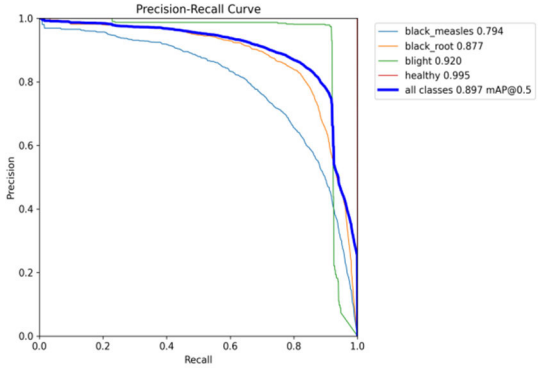
seen from Figure 13 that the improved convergence of the relevant curves is evident, with a greater improvement in accuracy. The model's predicted results are closer to the true values. Table 7 presents a comparison of grape leaf disease detection results from the grape leaf disease dataset.

TABLE 7. Comparison of grape leaf disease detection results on the grape leaf disease dataset.

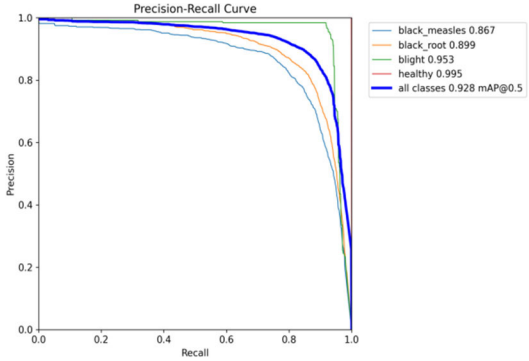
Algorithms	Type	mAP50	mAP50-95
YOLOv8n	Black-root	87.7	48.7
	Black_measles	79.4	50.5
	Blight	92	81.4
	Healthy	99.5	92.2
YOLOv8-ACCW	Black-root	89.7	51.4
	Black_measles	86	58.1
	Blight	94.5	86.5
	Healthy	99.5	97.5

In this study, under the same experimental environment and dataset conditions, traditional YOLOv8 model and the improved YOLOv8-ACCW model were trained using consistent experimental methods.

Figure 15 shows the Precision-Recall (PR) curves of both the traditional YOLOv8 model and the improved YOLOv8-ACCW model. It can be observed that the improved algorithm model has increased the mAP50 (mean Average Precision at IoU 0.5) for various grape leaf disease detections to some extent. The overall mAP50 has improved from the original 89.7% to 92.8%, representing an increase of 3.1%.



(a) YOLOv8n



(b) YOLOv8-ACCW

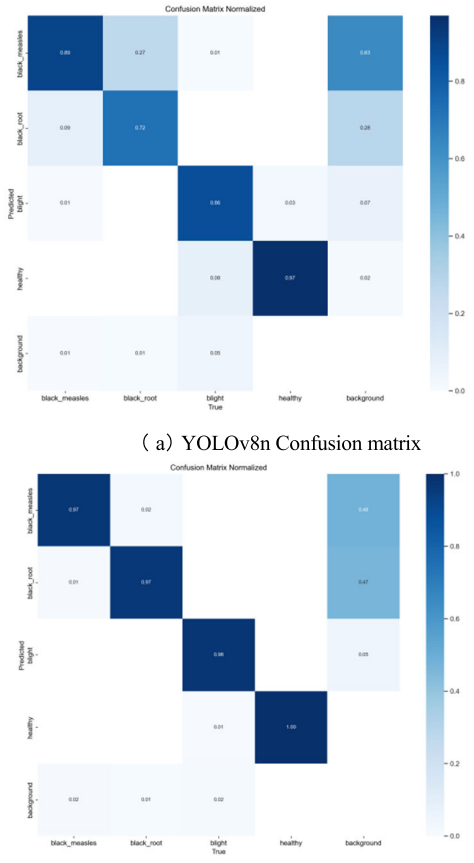
FIGURE 15. Precision–recall diagram of different methods.

Figure 16 displays the confusion matrices for the YOLOv8n and YOLOv8-ACCW models, used to classify four categories of grape leaf diseases.

H. COMPARISON WITH DETECTION RESULTS FROM DIFFERENT ALGORITHMS

To further validate the algorithm's performance, this study compares YOLOv8-ACCW with other mainstream object detection algorithms using the grape leaf disease dataset. The results are shown in Table 8. Compared to two-stage algorithms like Faster-RCNN and Cascade-RCNN, and single-stage algorithms like Efficientnet and CenterNet, YOLOv8-ACCW achieves improvements of 1.6%, 0.5%, 3.5%, and 4.4% in mAP50, and improvements of 1.3%, 2.5%, 5.6%, and 6.3% in mAP50-95. Significant improvements in FPS (frames per second) are also observed.

In comparison with the real-time object detection algorithm RT-DETR, YOLOv8-ACCW shows almost no



（a）YOLOv8n Confusion matrix

（b）YOLOv8-ACCW Confusion matrix

FIGURE 16. Confusion matrices of the YOLOv8n and YOLOv8-ACCW models.

decrease in F1 and mAP50, while having only 7.2% and 6.7% of the parameters and computations of RT-DETR, respectively. Moreover, there are significant improvements in mAP50-95 and FPS.Additionally, to assess its performance within the YOLO series, YOLOv8-ACCW is compared with lightweight networks such as YOLOv3tiny, YOLOv4tiny, YOLOv5s, YOLOv6n-ReLU, YOLOv7tiny, and YOLOv8n. As shown in Table 8, YOLOv8-ACCW achieves improvements of 8.2%, 7.3%, 4.1%, 3.9%, 3%, and 3.1% in mAP50 compared to these networks, while reducing computation and parameter sizes to varying degrees. Similarly, improvements in mAP50-95 are also notable across these comparisons.Furthermore, based on the same improvement approach applied to YOLOv7tiny, YOLOv7-ACCW is designed, showing enhanced detection accuracy compared to YOLOv7tiny. However, improvements on YOLOv8n are more significant than on YOLOv7tiny, primarily due to YOLOv8's better scalability and potential for performance enhancement through improvements in model structure and hyperparameters.

In conclusion, experimental results demonstrate that compared to other mainstream object detection algorithms, YOLOv8-ACCW achieves high detection accuracy with minimal model size, lower usage costs, higher detection efficiency, and simpler deployment. This makes it efficient and superior for practical applications.

TABLE 8. Comparison results of different algorithms.

Algorithms	F1	mAP50	mAP 50-95	Params/M	GFLOPs	FPS
Faster-RCNN	90.4	91.2	72.5	139.6	348.8	42
Cascade-RCNN	91.5	92.3	71.3	98.9	117.6	45
EfficientNet-b0	89	89.3	68.2	67	14.1	84
CenterNet	89.6	88.4	67.5	17.6	19.7	81
RT-DETR-L	89.5	90.1	69.4	38.9	112	50
YOLOv3tiny	85.9	84.6	66.7	12.1	19	137
YOLOv4tiny	87.8	85.5	66.2	7.3	18.9	146
YOLOv5s	89.6	88.7	67.3	8.2	18	126
YOLOv6n-ReLU	87.1	88.9	68	9.3	13.2	134
YOLOv7tiny	87.8	89.8	69.6	6.7	13.8	116
YOLOv7-ACCW	91.2	92	71.8	6.9	14.1	138
YOLOv8n	89.3	89.7	68.2	3	8.2	147
YOLOv8-ACCW	92.4	92.8	73.8	2.8	7.5	143

V. CONCLUSION

This paper focuses on three common grape leaf diseases: black root, black measles, and blight. Addressing the issue of poor detection performance for small lesions, a lightweight grape leaf disease detection method based on improved YOLOv8 is proposed, overcoming challenges faced by traditional YOLOv8n in object detection applications. The AKConv module replaces traditional convolutional modules, allowing arbitrary sampling of targets of various sizes to reduce model parameters and enhance disease detection. Additionally, a coordinate attention mechanism (CA) is introduced into the backbone and neck networks of YOLOv8 to embed positional information into channel attention, thereby strengthening feature extraction and suppressing irrelevant features. Furthermore, a lightweight universal upsampling content-aware reassembly module (CARAFE) is introduced to improve the model's ability to extract important features. Finally, the Wise-IoU boundary loss function is used to enhance boundary box regression performance and improve detection of small target diseases.Experimental results demonstrate that YOLOv8-ACCW offers advantages such as fewer parameters, lower computational requirements, and higher detection accuracy, meeting real-time requirements. Compared to the original YOLOv8, the proposed algorithm achieves a 3.1% increase in mean average precision (mAP) and performs well in both natural and simple environments. This method achieves high detection accuracy while reducing demands on platform computing and storage capabilities, making it easy to deploy on resource-constrained devices.

This study acknowledges limitations in disease category detection on grape leaves. To validate the robustness of the proposed model, expanding the categories in the grape disease dataset is necessary to obtain more comprehensive results. Moreover, the algorithm can be applied to detect other agricultural diseases. Future research will focus on deploying the improved model on resource-constrained embedded detection devices and refining the proposed algorithm for grape leaf disease recognition under conditions of strong reflection and extremely low light in practical applications.

REFERENCES

[1] M. Fraiwan, E. Faouri, and N. Khasawneh, "Multiclass classification of grape diseases using deep artificial intelligence," *Agriculture*, vol. 12, no. 10, p. 1542, Sep. 2022.

[2] X. Yin, W. Li, Z. Li, and L. Yi, "Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning," *Int. J. Agricult. Biol. Eng.*, vol. 15, no. 3, pp. 184–194, 2022.

[3] A. Adeel, M. A. Khan, M. Sharif, F. Azam, J. H. Shah, T. Umer, and S. Wan, "Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion," *Sustain. Comput., Informat. Syst.*, vol. 24, Dec. 2019, Art. no. 100349.

[4] N. Agrawal, J. Singhai, and D. K. Agarwal, "Grape leaf disease detection and classification using multi-class support vector machine," in *Proc. Int. Conf. Recent Innov. Signal Process. Embedded Syst. (RISE)*, Bhopal, India, Oct. 2017, pp. 238–244.

[5] P. Goncharov, G. Ososkov, and A. Nechaevskiy, "Disease detection on the plant leaves by deep," *Remote Sens.*, vol. 15, no. 17, p. 4356, 2019.

[6] A. E. Ibrahimi and N. E. Akchioui, "A review on plant diseases detection using artificial intelligence techniques," in *Proc. AIP Conf.*, 2023, p. 40018.

[7] 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, Jun. 2023.

[8] Z. Salman, A. Muhammad, M. J. Piran, and D. Han, "Crop-saving with AI: Latest trends in deep learning techniques for plant pathology," *Frontiers Plant Sci.*, vol. 14, Aug. 2023, Art. no. 1224709.

[9] A. Thakur, S. Venu, and M. Gurusamy, "An extensive review on agricultural robots with a focus on their perception systems," *Comput. Electron. Agricult.*, vol. 212, Sep. 2023, Art. no. 108146.

[10] Z. Jia, J. Hao, Y. Hou, R. Wang, R. Zhang, and S. Yao, "Study on rapid detection and identification of multi category apple leaf disease," *INMATEH Agricult. Eng.*, vol. 67, no. 6, pp. 67–76, Aug. 2022.

[11] Y. Es-saady, I. El Massi, M. El Yassa, D. Mammass, and A. Benazoun, "Automatic recognition of plant leaves diseases based on serial combination of two SVM classifiers," in *Proc. Int. Conf. Electr. Inf. Technol. (ICEIT)*, May 2016, pp. 561–566.

[12] G. Owomugisha and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 158–163.

[13] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," in *Proc. Conf. Adv. Signal Process. (CASP)*, Jun. 2016, pp. 175–179.

[14] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, "Identification of alfalfa leaf diseases using image recognition technology," *PLoS ONE*, vol. 11, no. 12, Dec. 2016, Art. no. e0168274.

[15] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–4.

[16] E. Dickinson, M. J. Rusilowicz, M. Dickinson, A. J. Charlton, U. Bechtold, P. M. Mullineaux, and J. Wilson, "Integrating transcriptomic techniques and k-means clustering in metabolomics to identify markers of abiotic and biotic stress in medicago truncatula," *Metabolomics*, vol. 14, no. 10, p. 126, Oct. 2018.

[17] K. Tian, J. Li, J. Zeng, A. Evans, and L. Zhang, "Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm," *Comput. Electron. Agricult.*, vol. 165, Oct. 2019, Art. no. 104962.

[18] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional neural networks for the automatic identification of plant diseases," *Frontiers Plant Sci.*, vol. 10, p. 941, Jul. 2019.

[19] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Frontiers Plant Sci.*, vol. 10, p. 611, May 2019.

[20] G. Polder, P. M. Blok, H. A. C. de Villiers, J. M. van der Wolf, and J. Kamp, "Potato virus Y detection in seed potatoes using deep learning on hyperspectral images," *Frontiers Plant Sci.*, vol. 10, p. 209, Mar. 2019.

[21] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, Oct. 2019.

[22] Y. Zhu, W. Sun, X. Cao, C. Wang, D. Wu, Y. Yang, and N. Ye, "TA-CNN: Two-way attention models in deep convolutional neural network for plant recognition," *Neurocomputing*, vol. 365, pp. 191–200, Nov. 2019.

[23] C. Dong, K. Zhang, Z. Xie, and C. Shi, "An improved cascade RCNN detection method for key components and defects of transmission lines," *IET Gener., Transmiss. Distrib.*, vol. 17, no. 19, pp. 4277–4292, Oct. 2023.

[24] X. Wang, A. Shrivastava, and A. Gupta, "A-Fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3039–3048.

[25] R. Liu, Z. Yu, D. Mo, and Y. Cai, "An improved faster-RCNN algorithm for object detection in remote sensing images," in *Proc. 39th Chin. Control Conf. (CCC)*, Jul. 2020, pp. 7188–7192.

[26] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, p. 2022, Sep. 2017.

[27] T. Liu, Q. Feng, and S. Yang, "Grape leaf disease detection method based on convolutional neural network," *Northeast. Agric. Univ.*, vol. 49, pp. 73–83, Jan. 2018.

[28] Z. He, J. Huang, Q. Liu, and Y. Zhang, "Segmentation of apple leaf disease based on asymmetric mixing convolutional neural network," *Trans. Soc. Agric.*, vol. 52, pp. 221–230, Mar. 2021.

[29] S. Zhai, D. Shang, S. Wang, and S. Dong, "DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion," *IEEE Access*, vol. 8, pp. 24344–24357, 2020.

[30] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Proc. Comput. Sci.*, vol. 199, pp. 1066–1073, Jan. 2022.

[31] X. Qi, "Construction and application research of grape pest and disease recognition model based on deep learning," *Xian Build. Univ. Sci. Technol.*, no. 2, p. 69, 2022.

[32] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[33] W. Zhuo, W. Jian, X. Wang, S. Jia, X. Bai, and Y. Zhao, "Lightweight detection method of apple in natural environment based on improved YOLO v4," *Trans. Chin. Soc. Agric. Mach.*, vol. 53, pp. 294–302, Jan. 2022.

[34] T. Huang, H. Huang, Z. Li, S. Lü, X. Xue, Q. Dai, and W. Wen, "Citrus fruit recognition method based on improved YOLOv5 model," *J. Huazhong Agric. Univ.*, vol. 41, pp. 170–177, Oct. 2022.

[35] D. Luo, Y. Xue, X. Deng, B. Yang, H. Chen, and Z. Mo, "Citrus diseases and pests detection model based on self-attention YOLOV8," *IEEE Access*, vol. 11, pp. 139872–139881, 2023.

[36] R. Ye, G. Shao, Y. He, Q. Gao, and T. Li, "YOLOv8-RMDA: Lightweight YOLOv8 network for early detection of small target diseases in tea," *Sensors*, vol. 24, no. 9, pp. 1424–8220, May 2024.

ZUXING CHEN was born in Bijie, Guizhou, in 1994. He received the Bachelor of Engineering degree in automation from the School of Robotics, Beijing United University, in July 2018, and the master's degree in control engineering from the School of Engineering, Beijing Forestry University, in July 2020. In September 2020, he joined the School of Physics and Electrical Engineering, Liupanshui Normal University, as a full-time Teacher. He is a Lecturer. His main research interest includes artificial intelligence.

JUNJIE FENG (Member, IEEE) was born in Tangshan, Hebei. He received the Ph.D. degree in communication and information systems from Nanjing University of Aeronautics and Astronautics, in June 2017. He is a Professor, the Doctoral Student, an Expert in urban management in Liupanshui City, the Master's Supervisor with Dalian University, and the Course Leader of "Automatic Control Principles" in the provincial "Golden Course" (first-class course). Currently, he is the Vice President of the Institute of Physics and Electrical Engineering, and also engaged with the School of Physics and Electrical Engineering, Liupanshui Normal University. He is mainly engaged in research work, such as compressed sensing, sparse signal processing, and radar signal processing.

KUN ZHU was born in Panxian, Guizhou. He received the bachelor's degree from the Department of Physics and Electronic Science, Guizhou Normal University, in 2000, and the master's degree in theoretical physics from the School of Physics and Technology, Yunnan University, in 2010. He is currently engaged with the School of Physics and Electrical Engineering, Liupanshui Normal University. He is the Master's Supervisor with Liaoning Normal University and the Executive Director of Guizhou Physical Society. He is the Dean, a Professor, and the Master of Physics. He is mainly engaged in research in the field of teaching theory in optics and physics. He has rich research experience in the radiation characteristics of WGM laser in cylindrical microcavities.

ZHENYAN YANG was born in Liupanshui, Guizhou, in April 1995. She received the degree from the Condensed Matter Physics Program, School of Physical Science and Technology, Southwest University, in June 2020. She joined the School of Physics and Electrical Engineering, Liupanshui Normal University, in September 2020. She is currently engaged in teaching theoretical and experimental courses related to physics. Her research interests include magnetic materials and magnetic storage technology.

YANHONG WANG was born in Weining, Guizhou, in 1988. She received the master's degree major in circuits and systems from Guizhou University, in July 2014. Since 2021, has been with Liupanshui Normal University, as a full-time Teacher.

MINGYUE REN was born in Henan, China. She received the master's degree from Beijing Jiaotong University, in 2017. She is currently an Automation Teacher with Liupanshui Normal University. Her main research interest includes control optimization research based on intelligent algorithms.