Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-17957-4

The improved YOLOv8 algorithm based on EMSPConv and SPE-head modules

Guihao Wen[1] · Ming Li[1] · Yonghang Luo[1] · Chaoshan Shi[1] · Yunfei Tan[1]

Received: 20 September 2023 / Revised: 6 December 2023 / Accepted: 18 December 2023

Abstract
Addressing the challenges of high model complexity, low generalization capability, and suboptimal detection performance in most algorithms for crop leaf disease detection, the paper propose a lightweight enhanced YOLOv8 algorithm. First, by incorporating the advantages of GhostNet's feature redundancy reduction and MobileNet's ability to fuse diverse channel features using the concept of Group Convolution, the paper enhance the backbone network. This improves the network's ability to extract critical features from a multitude of similar redundant feature maps. Second, to improve detection accuracy while reducing model parameters and computational load, the paper introduce the slim-Neck module. Finally, addressing the issue where detection head parameters and computations account for over half of the model, the paper restructure the Head using the concept of shared parameters and integrate convolution blocks that enhance multi-scale information recognition. Results from multiple experiments, averaged for consistency, indicate that compared to the original YOLOv8 algorithm, the improved algorithm achieves an increase in mAP50 from 86% to 87.3% and mAP50:95 from 67% to 68.6%. The model's size is a mere 5.45 MB, and the computational parameter GFLOPs is reduced from 8.1 to 5.5, even lower than the most lightweight YOLOv5. In comparison to other large-model algorithms, this model also demonstrates strong competitiveness in detection accuracy.

Keywords Leaf disease detection · YOLOv8 · Lightweight · Group convolutional

✉ Ming Li
55613163@qq.com

Guihao Wen
810525595@qq.com

Yonghang Luo
1834673531@qq.com

Chaoshan Shi
2468493724@qq.com

Yunfei Tan
517945859@qq.com

[1] Computer Science and Information Sciences, Chongqing Normal University, Shapingba, Chongqing 401331, China

Published online: 02 January 2024

1 Introduction

With the continuous advancement of agricultural technology and the growing demand for increased crop yields, crop leaf disease detection has become a significant research area in both the agricultural and computer vision domains [1]. However, as modern agriculture undergoes digital transformation, crop leaf disease detection continues to face a series of challenges:

(1) Model Complexity and Computational Resource Issues:
Crop leaf disease detection algorithms often entail extensive computational and parameter requirements, resulting in high costs that do not necessarily align with the benefits in practical agricultural applications. The substantial demand for computational resources restricts the widespread adoption of such algorithms. Therefore, there is a need to explore more lightweight models and algorithms to reduce costs and enhance deployability.
(2) Feature Limitations Due to Occlusion:
Leaf disease targets are typically subject to occlusion by vegetation and leaves [2–4], leading to an abundance of redundant features. This severely restricts the visibility of critical features of the targets and affects the feature extraction capabilities of computer vision models. The phenomenon of occlusion complicates leaf disease detection, requiring an algorithm that is more adaptive and robust to identify partially occluded leaf disease targets.
(3) Noise Interference Issues:
Images used in crop leaf disease detection may contain various noise interferences originating from soil, weeds, variations in lighting, and multiple types of leaf diseases. This makes it challenging for computer vision models to accurately classify and locate the features of different leaf diseases, often leading to false positives. Effective noise suppression and feature denoising techniques are essential to enhance the accuracy and robustness of crop leaf disease detection algorithms.

As a result, the urgent development of modern smart agriculture necessitates more efficient, lightweight, occlusion-adaptive, and noise-robust crop leaf disease detection algorithms. These algorithms can provide practical solutions for smart agriculture to ensure the quality and yield of crops.

In addressing the challenges associated with crop leaf disease detection, several researchers have explored improvements to YOLO-based models for disease detection. For instance, Liu et al. [5] utilized YOLOv3 for detecting tomato diseases and pests. Their approach incorporated multi-scale feature detection, target bounding box dimension classification, and multi-scale training, resulting in an accuracy of 92.39% and high real-time detection capabilities. Li et al. [6] introduced an enhanced YOLOv4-based micro-framework for real-time grape disease diagnosis. Their framework integrated techniques such as subsampling fusion structures, Mish activation functions, attention mechanisms, and Soft-NMS to improve the accuracy of real-time grape disease diagnosis.

Mathew and colleagues [7] presented a deep learning approach for identifying bacterial spot disease in bell pepper plants. Their method involved capturing random images of these plants using a smartphone on the farm, followed by inputting these images into the YOLOv5 model. The model predicted the presence of disease based on observed symptoms on the leaves. While these methods improved crop leaf disease detection through

feature fusion or enhancements to fusion modules, there remains room for improvement in the extraction of multi-scale information related to various leaf diseases. Additionally, the analysis of computational requirements and the benefits of deploying these models in practical applications is limited.

Therefore, to comprehensively address the challenges faced in crop leaf disease detection, this paper focuses on enhancing multi-scale occluded leaf disease features, reducing irrelevant noise interference, improving the positioning capabilities of the detection head, and enhancing detection accuracy. Building upon the YOLOv8 model, this paper proposes a lightweight and improved YOLOv8-based crop leaf disease detection algorithm. The main contributions of this paper are as follows:

(1) **EMSPConv Module:** Inspired by the concept of Group Convolution, this paper combines the advantages of GhostNet in handling redundant features with MobileNet's ability to fuse different channel feature information. A novel convolution module, named EMSPConv (Efficient Multi-Scale Conv), is designed. The EMSPConv module inherits the ability to handle redundant features from GhostNet and can fuse features extracted from different channels. Compared to standard convolutions, this module has fewer parameters, lower computational burden, and retains the richness of feature information. In this paper, EMSPConv replaces some of the standard convolutions in the backbone network.
(2) **Slim-Neck Module:** To accelerate inference predictions, this paper introduces the GSConv and VoVGSCSP modules within the Slim-Neck module in the neck network. GSConv's computational complexity is approximately half that of standard convolutions while maintaining similar learning capabilities. The VoV-GSCSP module reduces computational requirements and simplifies the structure of the Neck network, while ensuring adequate accuracy.
(3) **Redesigned Detection Head:** To address the issue of substantial computational resource consumption in the original YOLOv8 model's detection head, this paper employs the concept of shared parameters and introduces the lightweight and efficient EMSPConv convolution. The detection head of YOLOv8 is restructured with a significant reduction in parameter count. Leveraging multiple EMSPConv convolution blocks, the detection head's ability to extract multi-scale information features is further enhanced. In ablation experiments, the new detection head is not only lightweight but also demonstrates higher detection accuracy. The new detection head is named SPE-Head.

2 Design of enhanced YOLOv8 algorithm

2.1 Analysis of the YOLOv8 algorithm

Firstly, this paper chooses to enhance the YOLOv8 algorithm for the detection of crop diseases. The selection of algorithms in the field of crop disease detection, according to this paper, must meet three criteria: model lightweightness, inference speed, and detection accuracy. Model lightweightness is crucial since crop disease detection often takes place in outdoor fields where energy supply and implementation space for equipment are limited. The cost relationship between facility equipment and the inherent income in the field of crop cultivation is a significant concern. Therefore, algorithms applied in the field

of leaf disease detection need to have smaller computational requirements and lower implementation costs. Inference speed is essential in disease detection, following the principle that, similar to animal diseases, the earlier a disease is detected, the better. Early detection leads to timely intervention, minimizing losses and maximizing benefits. Detection accuracy holds a paramount position in the field of detection. In leaf disease detection, algorithms must simultaneously meet the requirements for detecting both normal and small targets due to the presence of subtle lesions or patterns. In the current algorithmic landscape, many outstanding algorithms exist. For instance, the Visual Prompt Tuning (VPT) algorithm in [8] excels in model capacity and training data scale, often outperforming complete fine-tuning while reducing storage costs for each task. However, in some cases, VPT performs slightly below complete fine-tuning. Addressing this, the E2VPT algorithm in [9] introduces a set of learnable key-value prompts and visual prompts in the self-attention layer and input layer, enhancing the effectiveness of model fine-tuning. The algorithm also includes a prompt pruning mechanism, systematically pruning low-importance prompts, significantly lightening the model. In [10], VPT is applied to the video subtitle generation domain, proposing a Video Language Prompt Adjustment (VL-Prompt) method. Experimental results demonstrate that VL-Prompt outperforms several state-of-the-art baseline methods. Despite exhibiting characteristics such as model lightweightness and superior performance, VPT has limited generalization ability, especially in the diverse reasoning of leaf disease detection. Considering the aforementioned three essential criteria, as well as the generalization performance of the model, this paper ultimately selects the YOLOv8 algorithm.

YOLOv8 is an advanced object detection model that builds upon the success of its predecessor YOLO versions, offering comprehensive improvements and innovations to enhance performance and flexibility. This model introduces a novel backbone network, an Anchor-Free detection head, and a loss function, enabling efficient operation across various hardware platforms (as shown in Fig. 1). YOLOv8 provides object detection networks with different resolutions, supports instance segmentation models, and offers multiple scale models to meet diverse scene requirements. The backbone network and Neck section have undergone careful fine-tuning, adopting the C2f structure to enhance gradient propagation, leading to a significant boost in performance. The Head section adopts a decoupled head structure, separating classification and detection tasks, while incorporating

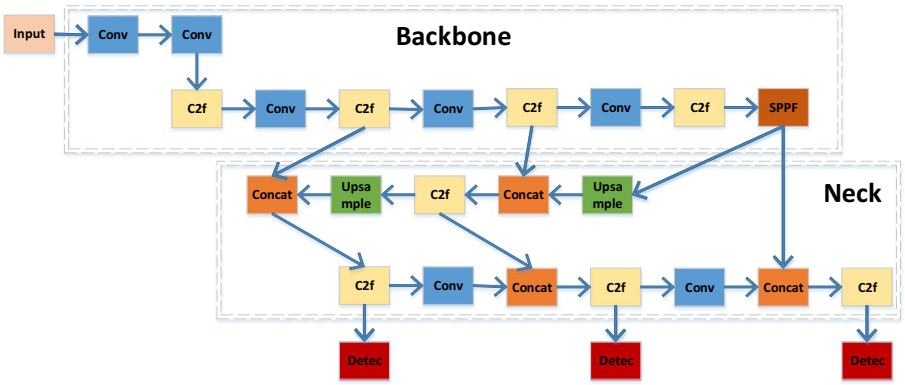

Fig. 1 YOLOv8 structure

Anchor-Free methods and Distribution Focal Loss for the loss function. These improvements and innovations position YOLOv8 as the State-of-the-Art (SOTA) model in the field of object detection, delivering substantial breakthroughs in performance. YOLOv8, short for "You Only Look Once," unquestionably excels in inference speed. The YOLOv8 algorithm combines the strengths of its predecessors in the YOLO series while ensuring model lightweightness. Although not as lightweight as YOLOv5, YOLOv8 outperforms YOLOv5 in overall performance. This is why this paper chose v8 over v5. While comparative experiments with the YOLOv5 model were conducted in the experimental section, the results indicate that the modified YOLOv8 in this paper achieves superior lightweight metrics while maintaining accuracy. Another factor influencing the choice of the YOLOv8 algorithm in this paper is the significant potential for future development in the field of leaf disease detection. Considering the accuracy requirements and the need for small target detection in the leaf disease domain, YOLOv8's Neck network can be extended to incorporate layers specifically designed for small target detection. Additionally, improvements can be made to the detection head, such as adding components for small target detection or enhancing the generalization capabilities of the detection head. This approach aims to propose an algorithm that aligns perfectly with the requirements of the leaf disease domain—a lightweight, highly accurate, efficient, and versatile algorithm. Of course, this is a goal for future work.

However, the drawbacks of YOLOv8 are quite apparent. As seen in Fig. 1, the YOLOv8 network architecture extensively employs numerous convolutional blocks and C2f blocks, inevitably leading to an increase in both model computation and parameter count. Although YOLOv8 is relatively lightweight, it still falls short compared to YOLOv5. The detection head poses a challenge due to its substantial consumption of computational resources, with its computation and parameter count accounting for approximately 50% of the entire model. YOLOv8 exhibits accuracy limitations in detecting small objects, and there is room for further optimization in the feature extraction capability of the backbone and the feature fusion ability of the Neck network. Therefore, this paper presents a series of improvements to the YOLOv8 network structure.

2.2 EMSPConv module

EMSPConv (Efficient Multi-Scale Conv) is a novel convolutional module introduced in this paper. Compared to regular convolutions, EMSPConv not only reduces the number of parameters and computational load but also enhances detection accuracy, achieving lightweight and efficient target detection. Firstly, EMSPConv is a streamlined convolutional module that incorporates multi-scale information. Its primary design inspiration is drawn from the concept of Group Convolution [11, 12], the unique insights on redundant features as presented in GhostNet [13], and the channel-wise feature fusion used in MobileNet.

GhostNet highlights the presence of highly similar features within intermediate feature maps generated by each residual block, signifying the existence of redundant features. To enhance feature extraction capabilities while incurring minimal overhead, it is crucial to extract critical feature information from these redundant features. Since feature information is extracted from channels of varying sizes, and the information within each channel is independent, we leverage the channel-wise feature fusion technique inspired by MobileNet's pointwise convolution. As illustrated in the EMSPConv structural diagram in Fig. 2, the process begins with an ordinary convolution operation applied to feature maps. These maps are then divided into k groups, and linear operations Φ are performed on each

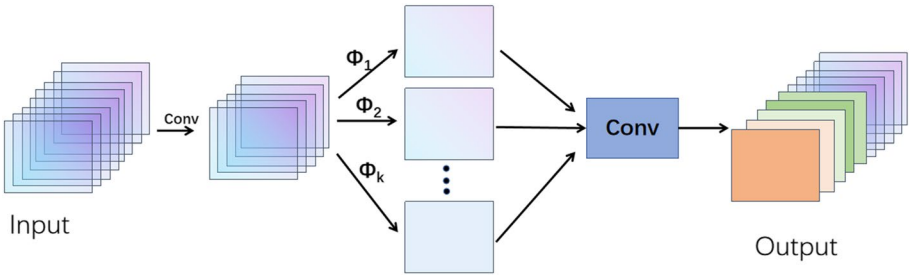

Fig. 2 EMSPConv network structure

group to produce complete feature maps (typically, Φ involves 1×1, 3×3, 5×5, or 7×7 convolutional kernels). Subsequently, channel-wise feature fusion is performed using a pointwise convolution, resulting in the output.

Complexity Analysis FLOPs (floating point operations): This metric is used to gauge the complexity of an algorithm or model (where K represents the convolution kernel size). The calculation formula is as follows:

$$\text{FLOPs} = \text{Cout} \times \text{Hout} \times \text{Wout} \times \text{Cin} \times \text{K} \times \text{K} \tag{1}$$

Assuming there is one identity transformation and s-1 EMSPConv feature maps (each feature map having s-1 redundancies), the FLOPs decrease by a factor of s:

$$\begin{aligned} r_s &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} h' \cdot w' \cdot c \cdot k \cdot k + (s-1)\frac{n}{s} h' \cdot w' \cdot d \cdot d} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \end{aligned} \tag{2}$$

In this equation, the numerator represents the complexity of the standard convolution, and the denominator represents the complexity of the EMSPConv module. The left side of the denominator is the FLOPs generated by the standard convolution (the first step), while the right side represents the FLOPs produced by the linear operations (the second step). Specifically, here, 's' denotes the total mappings generated for each channel (the hyperparameters 's' and 'k' can be adjusted; 's' is used to control the number of feature maps in the traditional convolution method, and 'd' represents the size of the linear transformation convolution kernel). This includes one intrinsic feature map and s-1 EMSPConv feature maps. 'c' represents the total number of input feature maps, which is typically large, while 's' is usually much smaller than 'c'. 'n/s' indicates the number of intrinsic feature maps output after the standard convolution, and 'd×d' represents the average kernel size of the linear operation, which is usually similar in size to 'k×k'.

2.3 SlimNeck module

The Neck network in YOLOv8 is positioned between the backbone and the head to better leverage the features extracted by the backbone, playing a role in feature fusion. In [14], the TFPN structure is introduced to address the weaknesses of traditional FPN. This structure incorporates three efficient modules: an adaptive extraction module with richer top-down features, a feature calibration module for calibrating upsampled features, and a feature feedback module that returns communication channels from the feature pyramid to the bottom-up backbone network. The TFPN structure enhances the ability to learn multi-scale

features in detection tasks. In YOLOv8, a PAN dual-pyramid structure is employed instead. While traditional FPN transfers strong semantic features from top (small size, more convolutions, rich semantic information) to bottom (large size, fewer convolutions, less semantic information), enhancing the entire pyramid, it only strengthens semantic information and does not convey localization information. PAN addresses this limitation by introducing a bottom-up (fewer convolutions, large size) pyramid appended behind the traditional FPN, forming a dual-pyramid structure. PAN incorporates a path aggregation approach, aggregating shallow and deep feature maps, and passing feature information along specific paths to enhance the representation of strong localization features from lower layers. Compared to TFPN, the feature fusion pyramid in this paper's YOLO demonstrates a stronger ability to express multi-scale features. This is attributed to the PAN and EMSPConv modules in the algorithm, both of which collectively extract multi-scale feature information.

The DenserNet network structure in [15] is relatively lightweight. It is a novel CNN-based architecture that aggregates more densely packed features from multiple semantics to achieve powerful image representation. However, the DenserNet network structure has a certain impact on detection accuracy. The SG-Net network in [16] is also a highly powerful network structure, combining original Self-Attention Network SAN representation with syntax-guided SAN representation. It possesses numerous advantages such as high precision, efficiency, and lightness. However, SG-Net has certain memory requirements during inference, which is not conducive to the subsequent development of the field of leaf disease detection. To achieve a better balance between the lightweight nature and accuracy of YOLOv8 model object detection, this paper introduces the Slim-Neck module. This module ensures lightweight while also improving detection accuracy (experimental details can be found in Section 3.4). The GSConv and VoVGSCSP components from Slim-Neck are embedded into the original algorithm's Neck network.

As shown in Fig. 3, the workflow of GSConv begins with subsampling the input using a regular convolution, followed by employing DWConv (Depthwise Convolution) [17]. The results of these two convolution operations are then concatenated, one from SC (Standard Convolution) and the other from DSC (Depthwise Separable Convolution). Finally,

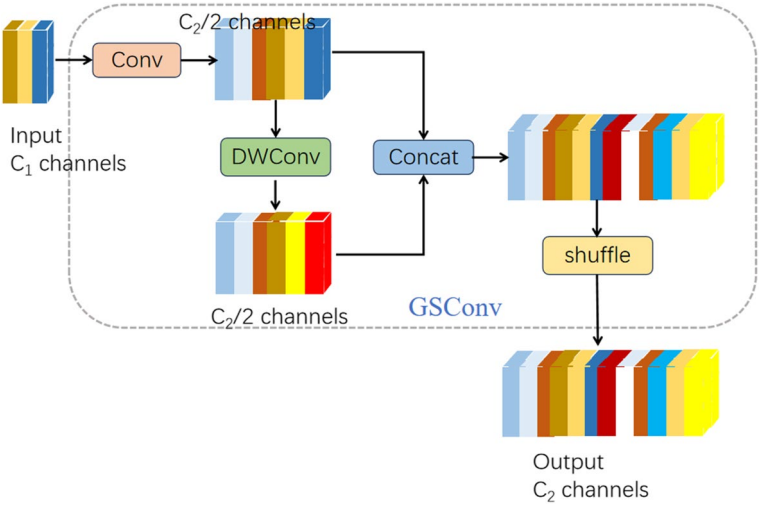

Fig. 3 GSConv structure diagram

a shuffle operation is performed using a uniform mixing strategy to ensure that the output information from SC and DSC is evenly combined. This shuffle operation effectively achieves an even exchange of information from SC output across different channels, enabling the uniform distribution and mixing of local feature information, allowing the information from SC to be fully integrated into the output of DSC.

To accelerate the algorithm's inference prediction, input images undergo a similar transformation process in the backbone network, where spatial information is gradually migrated to the channels. The spatial dimensions of feature maps progressively decrease while the number of channels increases, potentially leading to the loss of semantic information. The time complexity for SC, DSC, and GSConv is as follows:

$$\mathrm{Time}_{\mathrm{sc}} \sim \mathrm{O}(\mathrm{W} \times \mathrm{H} \times \mathrm{K}_1 \times \mathrm{K}_2 \times \mathrm{C}_1 \times \mathrm{C}_2) \tag{3}$$

$$\mathrm{Time}_{\mathrm{DSC}} \sim \mathrm{O}(\mathrm{W} \times \mathrm{H} \times \mathrm{K}_1 \times \mathrm{K}_2 \times 1 \times \mathrm{C}_2) \tag{4}$$

$$\mathrm{Time}_{\mathrm{GSConv}} \sim \mathrm{O}[\mathrm{W} \times \mathrm{H} \times \mathrm{K}_1 \times \mathrm{K}_2 \times \frac{\mathrm{C}_2}{2}(\mathrm{C}_1 + 1)] \tag{5}$$

where W and H represent the width and height of the output feature map, respectively. $\mathrm{K1} \times \mathrm{K2}$ is the size of the convolution kernel, C1 denotes the number of channels for each convolution kernel, and C2 is the number of channels in the output feature map.

GSConv demonstrates more significant benefits for lightweight models. By combining DSC layers with shuffle, it enhances the model's non-linear representation capabilities. However, if GSConv is extensively used throughout the entire model, it may result in a very deep model, which can limit data flow and increase inference time. Therefore, GSConv is exclusively applied to the neck section, where the received feature maps have the maximum channel count and minimum spatial dimensions. In such cases, the feature maps contain fewer redundant information, and no additional compression is required, allowing the attention module to work more effectively.

Additionally, to enhance the learning capacity of Convolutional Neural Networks (CNN), reduce computational complexity, simplify network structure, and ensure the model maintains sufficient accuracy, this paper introduces the VoVGSCSP module in the Neck network section [18]. The VoVGSCSP module combines theoretical concepts from DensNet, VoVNet, CSPNet, and others, as shown in Fig. 4. As seen, the VoVGSCSP module employs the lightweight convolution approach GSConv to replace standard convolution (SC) and further introduces GSbottleneck on top of GSConv, enhancing the model's learning capacity, reducing computational complexity, simplifying network structure, all while maintaining ample accuracy.

2.4 SPE-head reconstruction

The original YOLOv8 algorithm introduces significant changes to the Head section compared to previous YOLO algorithms. It has evolved from the original coupled head to a decoupled head and transitioned from YOLOv5's Anchor-Based approach to an Anchor-Free method [19]. Its structure is depicted in Fig. 5, with the Head section consisting of two branches: a decoupled classification branch and a regression branch. It's worth noting that the regression branch adopts the integral form representation proposed in Distribution Focal Loss.

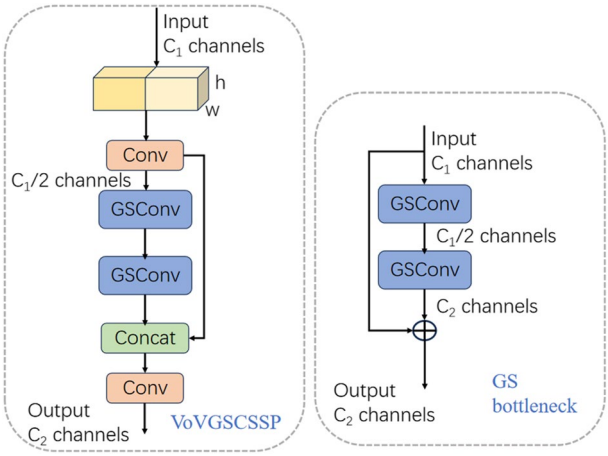

Fig. 4 VOVGSCSP network structure

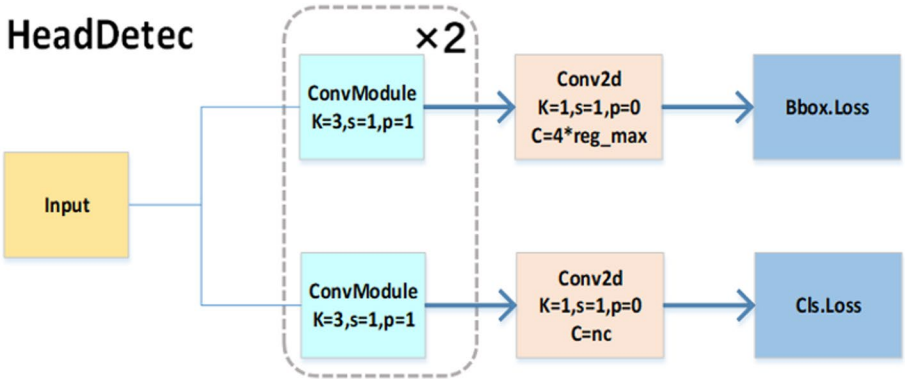

Fig. 5 Structure of the original YOLOv8 detection head

You can observe that in the Head section of YOLOv8, each branch consists of two 3×3 convolution layers. Furthermore, in the entire network structure of YOLOv8, there are a total of 3 Detect layers from Input to the Neck section. This means that in the entire network structure, the Head section alone contains 12 layers of 3×3 convolutions. This is one of the reasons why the Head section of the YOLOv8 algorithm accounts for nearly half of the parameter and computational load. Considering that this paper focuses on the research of crop disease detection algorithms for rural revitalization and that these detection tasks are typically conducted in outdoor environments, lower computational complexity and a lighter model are advantageous for practical deployment. In this context, this paper adopts the idea of shared parameters [20, 21] and redesigns the Convolution part of the original Head section, as illustrated in Fig. 6. It postpones the branches in the Head section, adding a shared parameter module after the Input in the Head section. Then, it proceeds with the decoupled classification and regression branches. Within the shared parameter module, this paper no longer uses the original 3×3 Convolution from the Head but incorporates EMSPConv.

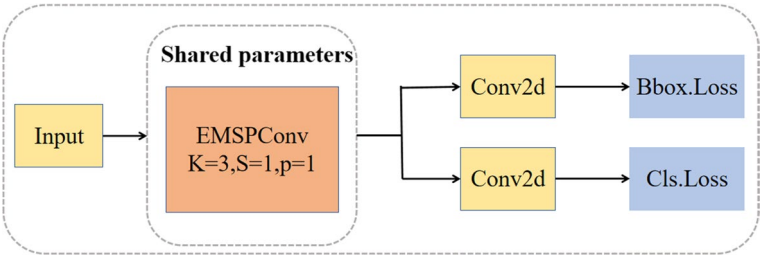

Fig. 6 Reconstructed detection head structure

3 Experimental results and analysis

3.1 Experimental environment

The experiments in this paper were conducted using the PyTorch framework, and GPU acceleration was employed. The experimental setup and fundamental training parameters are outlined in Table 1.

3.2 Experimental data

The dataset used in the experiments of this paper is a leaf disease dataset provided by Roboflow, an end-to-end computer vision platform. This dataset is of high quality, comprising a total of 5,494 images and encompassing 12 distinct categories: Beans Angular Leaf Spot, Beans Rust, Strawberry Angular Leaf Spot, Strawberry Anthracnose Fruit Rot, Strawberry Blossom Blight, Strawberry Gray Mold, Strawberry Leaf Spot, Strawberry Powdery Mildew Fruit, Strawberry Powdery Mildew Leaf, Tomato Blight, Tomato Leaf Mold, and Tomato Spider Mites. For training the leaf disease detection model, this experiment randomly divided the dataset into training, validation, and test sets in a 5:3:2 ratio.

3.3 Evaluation indicators

The primary evaluation metrics in this paper's experiments include model accuracy (Precision), recall (Recall), mean average precision (mAP) [22], and an analysis of model complexity represented in GFLOPs (Giga Floating Point Operations per Second) [23]. Assuming the number of true positives in the predicted results is TP (True Positives), the number

Table 1 Experimental parameter configuration

Name	Configuration
Operating system	Windows 11
Development system	CUDA 11
GPU	NVIDIA GeForce GTX 3050 4GB
Epochs	200
Batch-size	4

of false positives is FP (False Positives), and the number of missed true positives in the predicted results is FN (False Negatives), precision (P), recall (R), and mean average precision (mAP) can be calculated using the following formulas:

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

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

$$\text{AP} = \int_0^1 \text{P(R)dR} \tag{8}$$

$$\text{mAP} = \frac{1}{\text{C}} \sum_{\text{i=1}}^{\text{C}} \text{AP}_\text{i} \tag{9}$$

GFLOPs (Giga Floating Point Operations per Second) is a unit of computer performance measurement, indicating the number of billion floating-point operations that can be executed per second. GFLOPs is commonly used to assess the computational efficiency of models. In general, lower GFLOPs values are more favorable because lower GFLOPs imply that a model can accomplish more computational tasks in the same amount of time [24].

3.4 Ablation study

In this paper, the algorithm introduces Slim-Neck into the Neck network of the original YOLOv8n model and replaces the original detection head with the reconstructed detection head proposed in this paper. We conduct ablation experiments to assess the impact of each module on the algorithm, as shown in Table 2. Here, A represents the original YOLOv8n model algorithm, B represents the addition of Slim-Neck to A, C represents the inclusion of the EMSPConv module in the backbone network of A, D represents the use of the reconstructed SPE-head on top of A, and E (the improved YOLOv8 algorithm presented in this paper) represents the combination of Slim-Neck, EMSPConv, and the SPE-head added to A.

From Table 2, it is evident that the original YOLOv8 model has significant potential for improving both detection accuracy and model complexity on the leaf disease dataset. Building upon this foundation, the gradual introduction of the aforementioned modules further enhances detection performance, while also achieving significant model lightweight optimization. Through repeated experiments and averaging the results, the performance of

Table 2 Ablation experiment

Model	*P*%	R%	mAp50	mAP50:95	GFLOPs
A	85.3%	82.2%	86%	67%	8.1
B	86.9%	82.7%	87.2%	68.3%	7.3
C	88.2%	81.2%	87.5%	68.1%	7.9
D	86.9%	82.2%	87.4%	68.5%	6.4
E	86.8%	82.4%	87.3%	68.6%	5.5

Table 3 Comparative experiment (1)

	YOLOv3tiny	YOLOv5n	Yolov8n	Our
Precision	81.1%	81.8%	85.3%	86.8%
Recall	82.3%	81%	82.2%	82.4%
$mAP_{@0.5}$	83.7%	84.4%	86%	87.3%
$mAP_{@0.5:0.95}$	57%	64%	67%	68.6
GFLOPs	18.9	7.1	8.1	5.5

Table 4 Comparative experiment (2)

	Faster-RCN	SSD	YOLOv5s	Yolov8s	Our
Precision	62.4%	85.3%	88.3%	88.1%	86.8%
Recall	80.1%	56.1%	84.3%	84.7%	82.4%
$mAP_{@0.5}$	74.2%	63.5%	88.5%	88.6%	87.3%
GFLOPs	412.4	76.4	23.9	28.5	5.5

Algorithm E, which incorporates the fusion of Slim-Neck, EMSPConv, and the new detection head, has shown remarkable improvement. Specifically, mAP50 increased by 1.3%, mAP50:95 increased by 1.6%, and GFLOPs decreased to 5.5. This lightweight-improved YOLOv8 algorithm achieves both model lightweightness and a significant enhancement in detection accuracy.

3.5 Comparative experiments

To compare the detection performance of the detection networks before and after the improvements, six models, including YOLOv3tiny, YOLOv5n, YOLOv5s, YOLOv8n, YOLOv8s, and our enhanced YOLOv8n (referred to as "Our" hereafter), were trained and validated on the leaf disease dataset. The results obtained through iterative experiments and averaged results are presented in Tables 3 and 4.

The experimental results in Table 3 indicate that the enhanced YOLOv8 model in this paper shows improvements in various metrics compared to YOLOv3tiny, YOLOv5n, and the original YOLOv8 model at both IOU thresholds. When the IOU threshold is set at 0.5, the proposed approach exhibits a mAP increase of 3.6%, 2.9%, and 1.3% compared to YOLOv3tiny, YOLOv5n, and YOLOv8 models, respectively. Within the IOU threshold range of [0.5:0.95], our method outperforms the other three models by 11.6%, 4.6%, and 1.6%, respectively. In terms of model computational complexity, as indicated by GFLOPs, our algorithm reduces the computational load by 13.4%, 1.6%, and 2.6% compared to YOLOv3tiny, YOLOv5n, and YOLOv8, respectively. As shown in Table 4, our improved YOLOv8 algorithm maintains significant competitiveness in terms of detection accuracy when compared to larger models like Faster R-CNN,SSD, YOLOv5s and YOLOv8s. Furthermore, our enhanced algorithm exhibits a significant reduction in computational complexity and parameter count, being only 1/4 and 1/5 of YOLOv5s and YOLOv8s, respectively, highlighting its lightweight characteristics.

In summary, through comprehensive experimental validation, this paper's method has achieved a significant performance boost compared to the original algorithm and demonstrates strong competitiveness when compared to other state-of-the-art detection models.

Yolov3iny:

YOLOv5n:

Fig. 7 Prediction results

YOLOv8n:

Our:

Fig. 7 (continued)

3.6 Prediction result validation

A portion of models trained with the algorithm was utilized for predictions, and the prediction results are depicted in Fig. 7. The results indicate that this paper's algorithm performs well in detection while remaining lightweight. It exhibits strong generalization capabilities and effectively addresses issues related to model false positives and false negatives.

4 Conclusion

The experimental results demonstrate the robust performance of the improved YOLOv8 model proposed in this paper for leaf disease detection. This model efficiently detects various types of leaf diseases while maintaining a lightweight nature and provides precise localization information. By optimizing the YOLO algorithm, this paper not only achieves model lightweighting but also successfully accomplishes efficient and accurate leaf disease detection tasks. This work offers valuable insights for research and applications in the field of crop disease monitoring [25].

This paper presents a lightweight improved YOLOv8 algorithm model, which has undergone various optimizations and enhancements compared to the original YOLOv8. Firstly, by combining the advantages of GhostNet in handling redundant features with MobileNet's ability to fuse different channel feature information using Group convolution, the backbone network is improved, enhancing the network's capability to extract crucial features from numerous similar redundant feature maps. Secondly, to increase the algorithm's detection accuracy while reducing model parameters and computational complexity, the Slim-Neck module is introduced. This not only improves detection accuracy but also reduces the number of model parameters, while expediting the feature fusion process. Lastly, addressing the issue where the detection head parameters and computational complexity account for over half of the model, the Head is restructured using the concept of shared parameters. Additionally, convolutional blocks that enhance multi-scale information recognition are incorporated to improve detection accuracy. After conducting repeated experiments and averaging the results, the improved algorithm achieves an mAP50 and mAP50:95 of 87.3% and 68.6% on the leaf disease dataset, respectively, which is a 1.3% and 1.6% improvement over the original network. The GFLOPs metric for computational complexity is reduced to 5.5, which is 2.6 GFLOPs lower than the original model and 1.6 GFLOPs lower than the current lightest YOLOv5n model. The size of the trained model is only 5.45 MB. The overall outcome of the improved algorithm is a lightweight model that enhances detection accuracy, demonstrating its strong competitiveness in recent object detection networks.

In our future work, we will continue to optimize the network architecture to further enhance detection accuracy while maintaining model lightweight. We aim to introduce interpretable mechanisms and attributes, regularizing the network's representations to enhance its interpretability [26], and improve the model's detection generalization.

Data availability Data will be made available on reasonable request.

Declarations

Conflict of interest The authors declare that they do not have any conflicts of interest that influence the work reported in this paper.

Ethical approval No animals were involved in this study. All applicable international, national, and/or institutional guidelines for the care and use of animals were followed.

References

1. Orchi H, Sadik M, Khaldoun M, Sabir E (2023) Real-time detection of crop leaf diseases using enhanced YOLOv8 algorithm. In: 2023 International Wireless Communications and Mobile Computing (IWCMC). Marrakesh, Morocco 1690–1696
2. Terven J, Cordova-Esparza D (2023) A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint arXiv:2304.00501
3. Bhosale YH, Zanwar SR, Ali SS, Vaidya NS, Auti RA, Patil DH (2023) Multi-plant and multi-crop leaf disease detection and classification using deep neural networks, machine learning, image processing with precision agriculture - A review. In: 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, pp 1–7. https://doi.org/10.1109/ICCCI56745.2023.10128246
4. Li Y, Fan Q, Huang H, Han Z, Gu Q (2023) a modified yolov8 detection network for UAV aerial image recognition. Drones 7(5):304
5. Liu J, Wang X (2020) Tomato diseases and pests detection based on improved yolo v3 convolutional neural network. Front Plant Sci 11:898
6. Li H, Li C, Li G, Chen L (2021) A real-time table grape detection method based on improved yolov4-tiny network in complex back-ground. Biosys Eng 212:347–359
7. Mathew MP, Mahesh TY (2022) Leaf-based disease detection in bell pepper plant using yolo v5. Signal Image Video Process 16(3):841–847. https://doi.org/10.1007/s11760-021-02024-y
8. Jia M, Tang L, Chen BC, Cardie C, Belongie S, Hariharan B, Lim SN (2022) Visual prompt tuning. In: European conference on computer vision, vol 13693. LNCS, pp 709–727. https://doi.org/10.1007/978-3-031-19827-4_41
9. Han C, Wang Q, Cui Y, Cao Z, Wang W, Qi S, Liu D (2023) E2VPT: An effective and efficient approach for visual prompt tuning. arXiv preprint arXiv:2307.13770
10. Yan L, Han C, Xu Z, Liu D, Wang Q (2023) Prompt learns prompt: exploring knowledge-aware generative prompt collaboration for video captioning. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI). International Joint Conferences on Artificial Intelligence Organization, vol 180. https://doi.org/10.24963/ijcai
11. Wang X, Kan M, Shan S, Chen X (2019) Fully learnable group convolution for acceleration of deep neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9041–9050. https://doi.org/10.1109/CVPR.2019.00926
12. Zhang T, Qi GJ, Xiao B, Wang J (2017) Interleaved group convolutions. In: Proceedings of the IEEE international conference on computer vision, pp 4383–4392. https://doi.org/10.1109/ICCV.2017.469
13. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165
14. Liu D, Liang J, Geng T, Loui A, Zhou T (2023) Tripartite feature enhanced pyramid network for dense prediction. IEEE Trans Image Processing 32:2678–2692. https://doi.org/10.1109/TIP.2023.3272826
15. Liu D, Cui Y, Yan L, Mousas C, Yang B, Chen Y (2021) Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In Proceedings of the AAAI Conference on Artificial Intelligence 35(7):6101–6109
16. Liu D, Cui Y, Tan W, Chen Y (2021) Sg-net: Spatial granularity network for one-stage video instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9811–9820. https://doi.org/10.1109/CVPR46437.2021.00969
17. Li H, Li J, Wei H, Liu Z, Zhan Z, Ren Q (2022) Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. arXiv preprint arXiv:2206.02424

18. Cohen T, Welling M (2016) Group equivariant convolutional networks. In International conference on machine learning ICML 6:4375–4386
19. Dai X, Chen Y, Xiao B, Chen D, Liu M, Yuan L, Zhang L (2021) Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7369–7378. https://doi.org/10.1109/CVPR46437.2021.00729
20. Moehrs S, Del Guerra A, Herbert DJ, Mandelkern MA (2006) A detector head design for small-animal PET with silicon photomultipliers (SiPM). Phys Med Biol 51(5):1113
21. Albert PS, Follmann DA (2008) Shared-parameter models. In: Longitudinal data analysis. Chapman and Hall/CR, pp 447–466. https://doi.org/10.1201/9781420011579.CH19
22. Aboah A, Wang B, Bagci U, Adu-Gyamfi Y (2023) Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 5350–5358. https://doi.org/10.1109/CVPRW59228.2023.00564
23. Talaat FM, ZainEldin H (2023) An improved fire detection approach based on YOLO-v8 for smart cities. Neural Comput Applications 35(28):20939–20954. https://doi.org/10.1007/s00521-023-08809-1
24. Lou H, Duan X, Guo J, Liu H, Gu J, Bi L, Chen H (2023) DC-YOLOv8: Small-size object detection algorithm based on camera sensor. Electronics 12(10):2323
25. Hussain M (2023) YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines 11(7):677
26. Wang W, Cheng H, Zhou T et al (2023) Visual recognition with deep nearest centroids. arXiv:2209.07383 [cs.CV]