Weed and rice seedling detection in field based on MAL-YOLOv5 model

Yuqing Yang[1], Dequan Zhu[1], Kai Zhang[1], Minhui Chen[2], Ruixing Xing[3], Wei Xiong[1], Yu Zou[4], Juan Liao[3*]

(1. *School of Mechanical and Vehicle Engineering, Anhui Agricultural University, Hefei 230036, China*;
2. *School of Water Resources and Civil Engineering, China Agricultural University, Beijing 100083, China*;
3. *School of Electronics and Electrical Engineering, Anhui Agricultural University, Hefei 230036, China*;
4. *Rice Research Institute, Anhui Academy of Agricultural Sciences, Hefei 230031, China*)

Abstract: Weeds severely reduce rice yield and quality, making reliable in-field detection of weed and rice seedling essential for automated weed management. Although deep learning-based object detection techniques have shown significant potential in automatically distinguishing crops from weeds, existing models often suffer from large model sizes, high computational complexity, and insufficient feature extraction. To address these issues, this study proposes a lightweight multi-angle object detection model named MAL-YOLOv5 (Multi-Angle Lightweight YOLOv5), based on the YOLOv5 (You Only Look Once version 5) framework, which effectively reduces model complexity while maintaining detection performance. Specifically, the lightweight MobileNetV3 architecture is adopted to replace the original backbone network, significantly decreasing the number of parameters without compromising accuracy. Furthermore, the neck network of MAL-YOLOv5 is enhanced by integrating spatial and channel reconstruction convolution (SCConv) and a single-shot feature aggregation module (SCCSP), which reduces spatial and channel redundancies in the convolutional module, thereby compressing the neck network and improving feature representation. Additionally, a rotated bounding box with angular information is introduced for annotating and detecting rice seedling and weed, which effectively mitigates the interference from background and non-target objects, enabling more precise identification. Experimental results show that the precision, recall, and mAP of the MAL-YOLOv5 model are 93.1%, 91.9%, and 93.4%, respectively. Compared to YOLOv5s_obb, the MAL-YOLOv5 model reduces the number of parameters by 80.1% and computational cost in GFLOPs (Giga Floating-Point Operations) by 81.5%, significantly minimizing model size with only marginal loss in accuracy, conserving computational and storage costs while lowering the hardware requirements for intelligent mechanical weeding equipment.

Keywords: rice seedling and weed detection, YOLOv5, lightweight, SCConv, SCCSP, rotated bounding box
DOI: 10.25165/j.ijabe.20261903.9496

Citation: Yang Y Q, Zhu D Q, Zhang K, Chen M H, Xing R X, Xiong W, et al. Weed and rice seedling detection in field based on MAL-YOLOv5 model. Int J Agric & Biol Eng, 2026; 19(3): 256–266.

1 Introduction

The assurance of rice yield and quality is paramount for national food security, given its role as a staple crop[1]. However, its productivity is severely affected by interspecific competition with weeds for fundamental resources, such as soil nutrients, growing space, and water[2]. Additionally, weeds can provide habitat for pests, which cause various diseases, seriously affecting the quality and yield of rice[3]. Conventional weed control predominantly relies on the blanket application of chemical herbicides[4]. Although widely used, this approach is not only labor-intensive but also poses serious risks, including the emergence of herbicide-resistant weeds, environmental pollution, and pesticide residues[5,6]. Consequently, the development of precise targeted weeding strategies is imperative for sustainable rice production. With the development of agricultural mechanization and machine vision, the way for intelligent weeding equipment is paved as a promising solution[7]. The core of this technology lies in the accurate identification and localization of weeds versus crops, enabling machinery to perform weeding or targeted spraying, thereby enhancing efficiency and reducing chemical usage[8]. Moreover, due to the growth characteristics of rice crops, the critical window for effective mechanical weeding in rice fields spans from the seedling transplanting stage until the canopy closure[9]. Hence, it is vital to achieve rapid and accurate detection of weeds in rice fields.

Early research on plant discrimination mainly relied on machine learning techniques using hand-crafted features, such as color, texture, and shape. Methods including random forests[10], bag of visual words with HOG features[11], artificial neural networks[12], and support vector machines[13] have achieved high accuracy (often >95%) under controlled or specific conditions. However, these methods are often limited by their dependence on manual feature engineering, meticulous image preprocessing, and they often lack robustness and generalization in complex real-field environments.

The advent of deep learning has revolutionized the field,

Received date: 2024-11-04 **Accepted date:** 2026-05-15
Biographies: Yuqing Yang, MS, research interest: crop phenotype detection, Email: 23720754@stu.ahau.edu.cn; **Dequan Zhu**, PhD, Professor, research interest: agricultural machinery information perception and intelligent control, Email: zhudequan@ahau.edu.cn; **Kai Zhang**, MS, research interest: crop detection, Email: zhangk@stu.ahau.edu.cn; **Minhui Chen**, MS, research interest: crop image segmentation, Email: 20720712@stu.ahau.edu.cn; **Ruixing Xing**, MS, research interest: rice disease detection, Email: 2454402649@qq.com; **Wei Xiong**, PhD, Lecturer, research interest: planting technology and equipment research, Email: xiongwei_ahau@ahau.edu.cn; **Yu Zou**, PhD, Associate Research Fellow, research interest: rice genetics and breeding, Email: zouyu0308@126.com.

***Corresponding author: Juan Liao**, PhD, Associate Professor, research interest: perception of crop phenotype. School of Electronics and Electrical Engineering, Anhui Agricultural University, Hefei 230036, China. Email: liaojuan@ahau.edu.cn.

offering superior feature extraction capabilities and adaptability. Naik et al.[14] employed a region-based convolutional neural network (R-CNN) to classify sesame and weeds, which achieved a detection accuracy of 96.84% and a weed classification accuracy of 97.79%. Daşkın et al.[15] developed an ensemble model based on transfer learning for maize and weed detection, yielding better performance than individual VGG16 and InceptionV3 models. Jiang et al.[16] developed a GCN-ResNet-101 method that builds graphs from CNN features and Euclidean distances, leveraging both labeled and unlabeled data. The method achieved accuracies of 97.80%, 99.37%, 98.93%, and 96.51% on four weed datasets. Li et al.[17] introduced the YOLOv7-FWeed model for weed detection based on an improved YOLOv7, which enhances recognition accuracy by incorporating the F-ReLU (Funnel Rectified Linear Unit) activation function and a maxPool multi-head self-attention (M-MHSA) module. More recently, object detection frameworks like the YOLO series have been favored for their speed and efficiency. Subsequent improvements, including the integration of attention mechanisms[18] and advanced feature fusion strategies[19], have further improved the detection performance. Despite these advances, many high-performance models remain computationally heavy, with large parameter sizes and high computational demands (FLOPs), making them unsuitable for lightweight embedded systems, particularly on resource-constrained intelligent machinery. Moreover, the horizontal rectangular bounding box is employed to determine object positions and categories. However, the rice seedling and weed in images exhibit overlapping or intersecting regions. Using horizontal bounding boxes often leads to significant inclusion of background pixels and severe box overlaps when objects are close, impairing accurate localization.

To address the aforementioned issues, this study presents a lightweight multi-angle detection model MAL-YOLOv5 for rice seedling and weed detection. The backbone network of YOLOv5 is replaced with MobileNetV3 to achieve a lightweight design. Additionally, a Slim-neck structure is constructed by integrating SCConv and SCCSP modules, which reduces feature redundancy and further decreases the number of parameters. Furthermore, it introduces rotated bounding boxes for precise localization, where the primary stem direction of the rice seedling or weed serves as the reference and the rotation angle θ is defined as the angle between the bounding box's long edge aligned with the plant's primary stem and the positive x-axis. This approach can effectively mitigate background interference and resolve the localization ambiguities common with horizontal bounding boxes in dense plant scenarios.

2 Materials and methods

2.1 Construction of rice seedling and weed datasets

2.1.1 Image acquisition

In this study, the rice seedling and weed images were collected at the Quanjiao Experimental Base of Anhui Academy of Agricultural Sciences in June 2022, and the Agricultural Garden of Anhui Agricultural University in April 2023. Using a Canon EOS 850D camera, 2034 images were obtained under diverse lighting and viewing angles. A rigorous quality check led to the exclusion of 74 images that were blurred due to motion or defocus, resulting in a final curated set of 1960 images. Given that the native resolution of 3456×2304 pixels is computationally prohibitive and exceeds the input size of typical detection models, all images were cropped and resized to 512×512 pixels to optimize GPU memory usage and processing speed. Figure 1 illustrates a subset of the collected images.

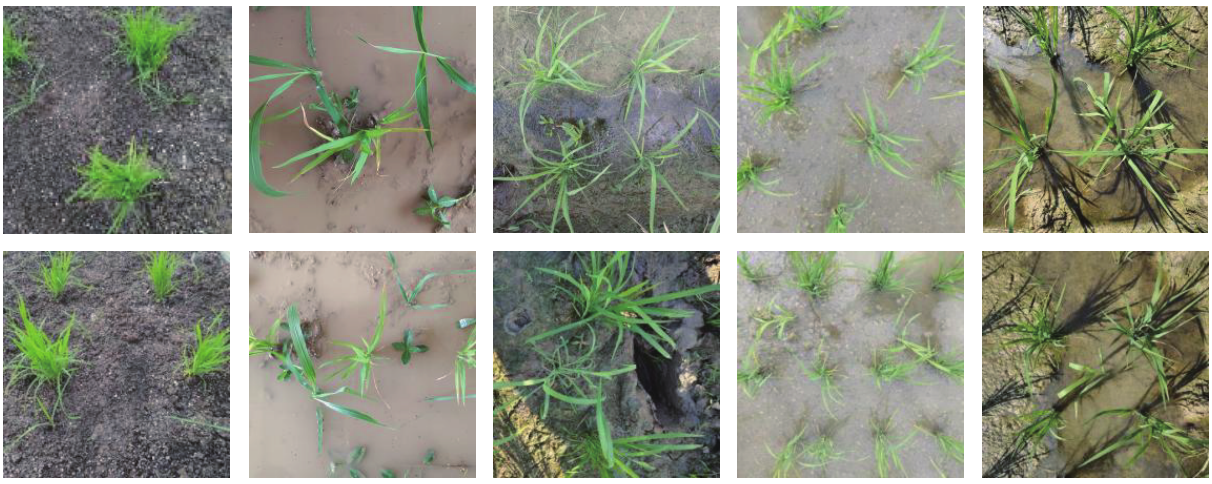


Figure 1 Partial images of rice seedling and weed in the field

2.1.2 Image annotation

The software roLabelImg was used to annotate the images, where each rice seedling and weed was demarcated with a rotated rectangular bounding box. As illustrated in Figure 2, this annotation format provides a more precise spatial enclosure for slender plant targets compared to horizontal bounding boxes, effectively minimizing inter-leaf overlap and reducing background interference. The annotated images were used to train the model with the training datasets and calculate the performance with the test datasets. The data samples included RGB images and annotated images were divided into the training datasets, the validation datasets, and the test datasets in a ratio of 7:2:1, respectively.

2.2 Rice seedling and weed detection model

To effectively reduce model size and computational complexity while maintaining high detection performance, this study uses YOLOv5[20] as the basic framework and proposes a lightweight YOLOv5 model for rice seedling and weed detection. The original backbone network is replaced with MobilenetV3, which integrates residual structures, linear bottlenecks, adjustable network widths, global average pooling, depth-wise separable convolutions, and the Squeeze-and-Excitation (SE) attention mechanism, to construct a more lightweight backbone network while maintaining detection precision. Furthermore, in the neck network, the standard convolutional modules are substituted with spatial and channel

reconstruction convolution (SCConv), and the CSPX_2 (Cross Stage Partial 2 with X Bottleneck Blocks) modules are replaced with a single-shot feature aggregation module (SCCSP). These modifications collectively streamline the network, reducing spatial and channel redundancies while preserving representational capacity. The overall architecture of the proposed model is depicted in Figure 3.

2.2.1 The lightweight backbone network

The YOLOv5 architecture utilizes a backbone network composed of CBS (Convolution BatchNorm SiLU) and CSP (Cross Stage Partial) modules to efficiently extract salient features from input images[21]. Nevertheless, this architecture introduces a considerable number of model parameters, leading to high computational overhead and posing challenges for deployment on embedded systems or devices with limited computational resources. Given the necessity for real-time performance in rice seedling and weed detection, the adoption of a lightweight backbone network can enhance the model's inference speed, enabling prompt responses and object detection in real-time scenarios. MobileNetV3, as the latest iteration of the MobileNet series, represents a lightweight convolutional neural network architecture characterized by its compact parameter size, high accuracy, and efficient inference capabilities[22]. While both the Large and Small versions of MobileNetV3 share a similar structural blueprint, the Large variant incorporates more Bneck (Bottleneck) modules, which increases model capacity and accuracy at the cost of higher computational load and slower inference[23]. Therefore, to better balance detection performance with operational efficiency, this study adopts MobileNetV3-Small as the backbone network for the proposed MAL-YOLOv5 model.

a. Oriented bounding boxes annotation

b. Horizontal bounding boxes annotation

Figure 2 Comparison of the visual effects of horizontal bounding boxes and oriented bounding boxes

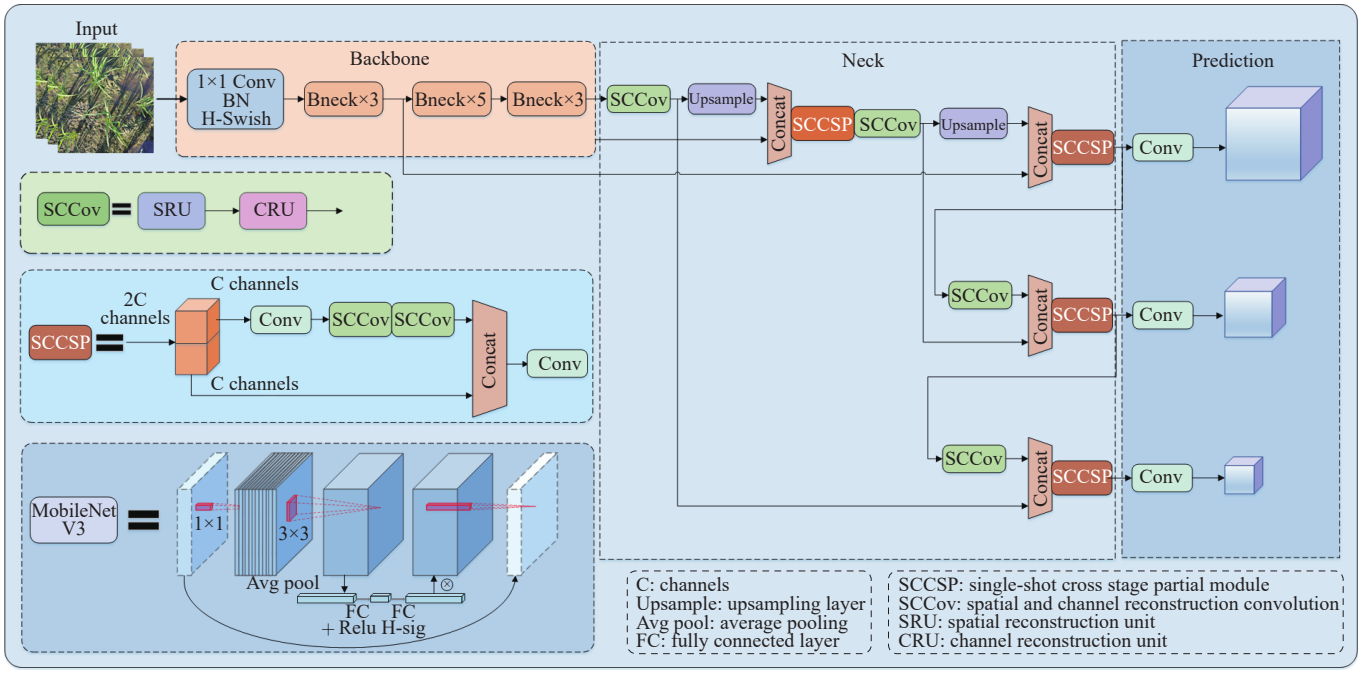


Figure 3 The overall architecture of the MAL-YOLOv5 model

In the MobileNetV3, the core structure is Bneck module, depicted in Figure 4, which employs an inverted residual structure with a linear bottleneck. As shown in Figure 4, the feature map dimension is firstly enhanced by a 1×1 convolutional layer. A depth-wise separable convolution is then executed on the feature map, followed by a reduced-dimensional 1×1 convolution. The SE attention module is integrated between these two convolution layers to enhance feature representation. A residual shortcut connection is also incorporated to facilitate cross-layer feature propagation. The key to the lightweight design of the Bneck module lies in the use of depth-wise separable convolution[24], which consists of depthwise convolution and pointwise convolution. The depthwise convolution applies a separate small convolutional kernel to each input channel, extracting spatial features independently per channel without inter-channel communication. This focuses solely on the spatial information in the input feature map, without exchanging information between channels, which maintains effective feature extraction while reducing computational complexity. The subsequent pointwise convolution, implemented via a 1×1 kernel, performs a linear combination of the output feature maps from the depthwise convolution. This step enables cross-channel information fusion while allowing flexible adjustment of the output dimension, further minimizing model parameters. To illustrate the parameter efficiency, an input feature map of size 5×5 with 3 channels is

considered. As shown in Figure 5a, a standard convolutional layer producing four 3×3 output feature maps would require 4×3×3×3 = 108 parameters. In contrast, as depicted in Figure 5b, the depthwise separable convolution achieves the same output using only 3×3 depth-wise convolution followed by 1×1 point-wise convolution. The number of parameters for the depth-wise separable convolution is only 3×3×3 + 1×1×3×4 = 39, resulting in a parameter reduction of over 60%.

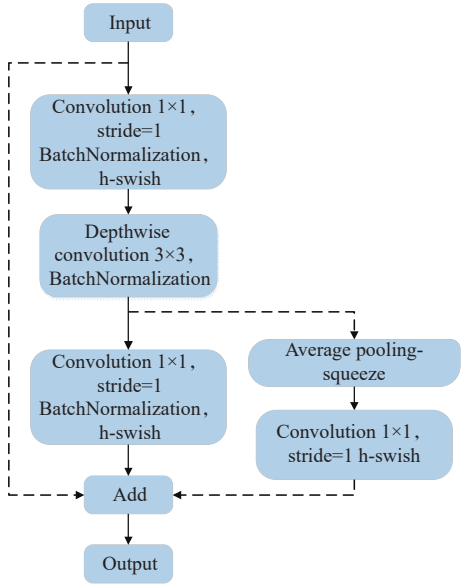
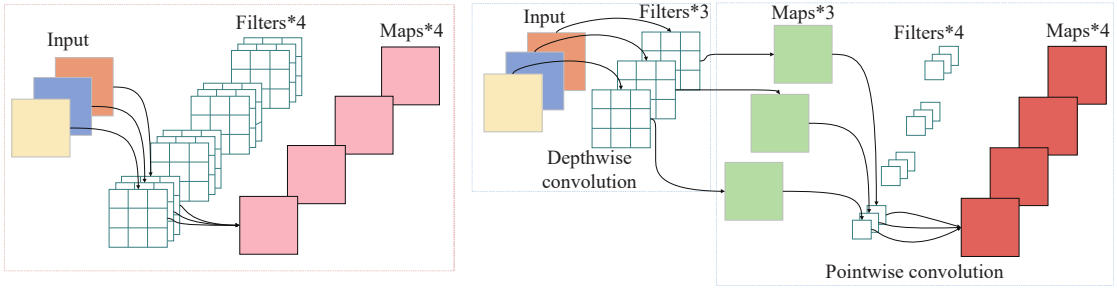


Figure 4 Bneck block structure

As mentioned above, while the depthwise separable convolution reduces the number of parameters and computational complexity compared to the standard convolution, it separates the channel information of the input feature map without considering the correlation mapping between the channel information. Since different channels in a feature map typically encode distinct types of feature information, this independence can lead to suboptimal feature representation. Therefore, the SE lightweight channel attention mechanism[25] is placed after the depthwise separable convolution layer of the Bneck module to dynamically adjust the weight of each channel, effectively modeling relationships across feature channels. As illustrated in Figure 6, the SE module operates through squeeze, excitation, and scale steps. The squeeze operation applies global average pooling to each channel, compressing the two-dimensional feature maps into a compact channel-wise descriptor vector of length *C*, which captures the global contextual information per channel. The excitation operation process then learns a set of adaptive weights for these channels. It is implemented by 1×1 convolution layer, ReLU layer, 1×1 convolution layer, and Sigmoid activations layer in order. The learned weight coefficients are then used to reweight each channel of the input feature map and multiply with the input feature map in the scale step. The introduction of the SE module captures the dependencies between different channels to enhance feature representation, improving the model's attention to the target while suppressing the weight of irrelevant information such as noise, thus effectively improving the model accuracy.

To achieve a lightweight backbone network, we replace the original Conv and C3 modules in YOLOv5 with the Bneck structure from MobileNetV3-Small. The detailed configuration is provided in Table 1. Exp_size denotes the number of output channels after the first 1×1 convolution layer in the Bneck module, which performs dimensionality expansion. Output indicates the number of output channels following the last 1×1 convolution layer within the Bneck. SE signifies whether the channel attention mechanism is employed. NL represents the type of activation function, which includes HS (H-Swish) and RE (ReLU). Stride indicates the stride value.



a. Standard convolution b. Depthwise separable convolution

Figure 5 Standard convolution and depthwise separable convolution

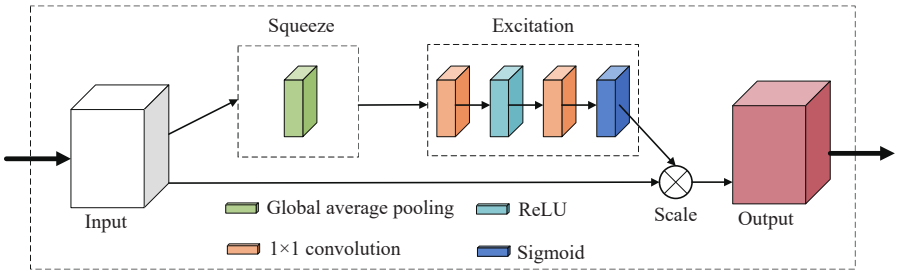


Figure 6 The structure of SE module

2.2.2 Improvement of neck network

In the original YOLOv5 architecture, the neck network employs a substantial number of standard convolutional operations, which does not involve feature redundancy reduction. When feature maps contain significant redundant information, this redundancy contributes to unnecessary computational load during convolution calculations, slowing down inference speed. Furthermore, redundant information can obscure crucial feature details, hindering the model's

ability to accurately extract key features and ultimately impacting recognition accuracy. Therefore, this study introduces the spatial and channel reconstruction convolution (SCConv)[26] into the neck network, aiming to reduce both spatial and channel redundancies in convolutional features, thereby compressing the model and enhancing feature representation. The SCConv module consists of two units: the spatial reconstruction unit (SRU) and the channel reconstruction unit (CRU), where the SRU employs a separate-reconstruct strategy to minimize spatial redundancy, while the CRU utilizes a split-transform-merge approach to reduce channel redundancy. The architecture of SCConv module is presented in Figure 7, which shows that the input feature is spatially refined through the SRU operation, followed by channel refinement with the CRU operation.

Table 1 Backbone network convolutional layer structure

Input	Operator	Exp_size	Output	SE	NL	Stride
640²×3	Conv2d	-	16	-	HS	2
320²×16	Bneck, 3×3	16	16	-	RE	2
160²×16	Bneck, 3×3	72	24	√	RE	2
80²×24	Bneck, 3×3	88	24	-	RE	1
80²×24	Bneck, 5×5	96	40	√	HS	2
40²×40	Bneck, 5×5	240	40	√	HS	1
40²×40	Bneck, 5×5	240	40	√	HS	1
40²×40	Bneck, 5×5	120	48	√	HS	1
40²×48	Bneck, 5×5	144	48	√	HS	1
40²×48	Bneck, 5×5	288	96	√	HS	2
20²×96	Bneck, 5×5	576	96	√	HS	1
20²×96	Bneck, 5×5	576	96	√	HS	1

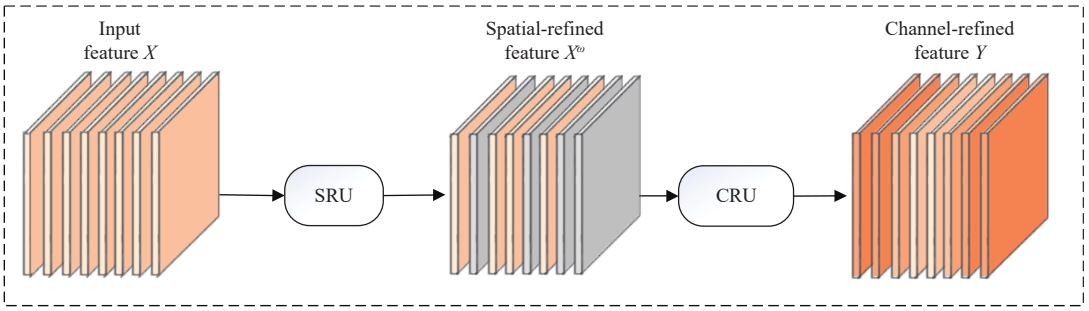


Figure 7 Spatial and channel reconstruction convolution

The structure of SRU is illustrated in Figure 8. The separation operation first employs the scaling factors from group normalization (GN) layer to evaluate the information content in different feature maps. Through a gating mechanism, the input features are subsequently partitioned into information-rich feature maps and information-poor feature maps. Then, the reconstruction operation follows, implementing a cross-reconstruction procedure that strategically reintegrates these separated feature maps. This integration enhances information flow between the two groups while optimizing spatial efficiency and ultimately producing features with enriched representational capacity. The mathematical formulations for both the separation and reconstruction operations of the SRU are detailed in Equation (1) and Equation (2), respectively.

$$\begin{cases} X_{out} = GN(X) = \gamma \dfrac{X-\mu}{\sqrt{\sigma^2+\varepsilon}} + \beta \\ W_\gamma = \{\omega_i\} = \dfrac{\gamma_i}{\sum_{j=1}^{C}\gamma_j}, \ i、\ j = 1,2,\ldots,C \\ W = Gate(Sigmoid(W_\gamma(GN(X)))) \end{cases} \tag{1}$$

$$\begin{cases} X_1^\omega = W_1 \otimes X \\ X_2^\omega = W_2 \otimes X \\ X^{\omega 1} = X_{11}^\omega \oplus X_{22}^\omega \\ X^{\omega 2} = X_{21}^\omega \oplus X_{12}^\omega \\ X^\omega = X^{\omega 1} \cup X^{\omega 2} \end{cases} \tag{2}$$

where, μ and σ represent the mean and standard deviation of X, respectively, while γ and β are trainable variables (affine transformations); X_{out} is the standardized input feature X; W_γ is the normalized correlation weight; W is the weight values of feature maps reweighted by W_γ mapped to the range (0, 1) by the sigmoid function and gated by a threshold; $X^{\omega 2}$ is a small constant used for numerical stability. A larger σ indicates greater variations among pixels, signifying richer spatial information; W_1 is the weight with rice information; and W_2 is the weight with poor information; $X^{\omega 1}$ and $X^{\omega 2}$ are two weighted features of distinct information profiles; X_1^ω and X_2^ω are the informative weighted feature and less informative weighted feature; X^ω is the resulting spatial refinement feature map; $\otimes$ represents element-wise multiplication; $\oplus$ denotes element-wise summation, and $\cup$ signifies concatenation.

Following spatial refinement by the SRU, CRU is used for channel redundancy reduction. The CRU architecture, depicted in Figure 9, operates through three sequential stages: split, transform, and fuse. In the split stage, the input features X^ω are partitioned along the channel dimension into two segments with channel counts of $\alpha \times C$ and $(1-\alpha) \times C$ respectively, where α is a hyper-parameter, and $0 \le \alpha \le 1$ represents a given condition or constraint. The feature map's channels are then compressed by using 1×1 convolution, resulting in X_{up} and X_{low} respectively. Subsequently, the transform operation utilizes global weighted convolution and point-wise weight convolution to extract features of X_{up} and X_{low}, and uses element-wise addition and concatenation to generate two sets of feature maps with different information richness, namely Y_1 and Y_2. The fusion stage employs global average pooling to capture spatial context from both Y_1 and Y_2, followed by channel-wise concatenation to produce the final channel-refined output feature Y. The mathematical formulation of CRU is defined in Equations (3)-(7).

$$Y_1 = M^G X_{up} + M^{P_1} X_{up} \tag{3}$$

$$Y_2 = M^{P_2} X_{low} \cup X_{low} \tag{4}$$

$$S_m = Pooling(Y_m) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} Y_c(i,j), \ m = 1,2 \tag{5}$$

$$\beta_1 = \frac{e^{s1}}{e^{s1}+e^{s2}},\ \beta_2 = \frac{e^{s2}}{e^{s1}+e^{s2}},\ \beta_1+\beta_2 = 1 \tag{6}$$

$$Y = \beta_1 Y_1 + \beta_2 Y_2 \tag{7}$$

where, M^G, M^{P_1}, and M^{P_2} are learnable weight matrices in convolution operations, and β_1 and β_2 are feature importance vectors. S_m is the channel-wise global feature vector obtained by applying global average pooling (GAP) to the transformed feature Y_m in the Channel Reconstruction Unit (CRU) module.

The original CSP2_X module in YOLOv5 employs the cross stage partial network (CSPNet) architecture[27], where partial features bypass convolutional layers to retain original information while the remaining features undergo deeper transformation through multiple convolutional blocks. Both paths are then fused via concatenation, followed by batch normalization and ReLU activation. However, the CSP2_X module still exhibits considerable high computational overhead in practical applications, especially when dealing with high-resolution images or in deep network configurations. Moreover, its multiple convolutional layers and residual connections introduce parameter redundancy that complicates training and increases overfitting risks. This design preserves multi-scale features with varying receptive fields while reducing aggregation frequency, enabling effective multi-scale visual information capture with maintained accuracy and reduced complexity.

To address these limitations, leveraging the feature aggregation approach based on SCConv[28], we incorporate a one-shot aggregation strategy to construct the single-shot feature aggregation module (SCCSP). The SCCSP module is applied to replace the CSP2_X module in the neck network. As illustrated in Figure 10, the SCCSP processes a single feature stream by splitting it into two branches: a main branch containing sequential convolutional layers and a residual branch that preserves the original features through an identity connection. The outputs of both branches are integrated via element-wise summation. This architecture reduces parameter redundancy and computational cost while maintaining feature integrity, enabling effective fusion of multi-scale features with varying receptive fields. Consequently, this design enables effective multi-scale visual information capture with maintained accuracy and reduced complexity.

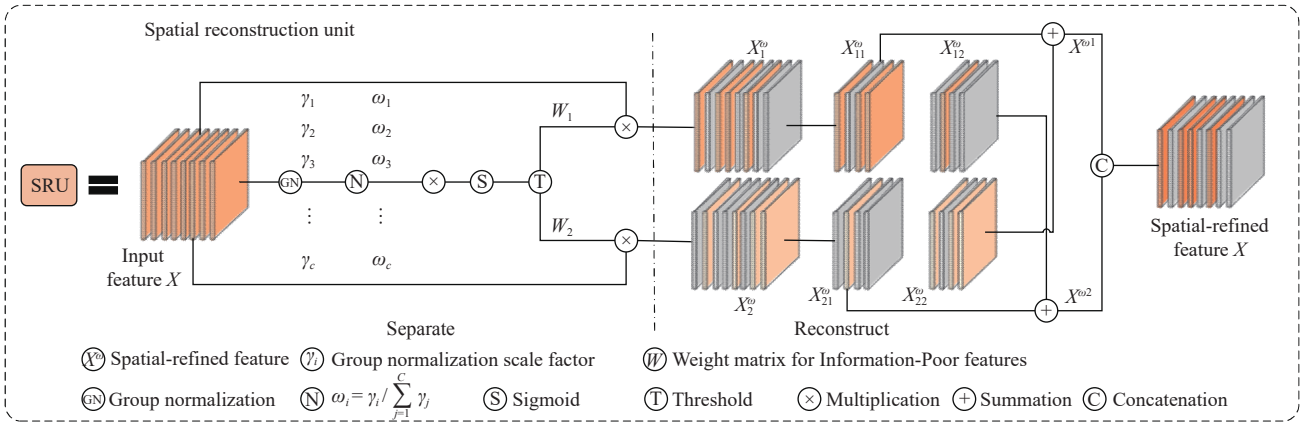


Figure 8 Spatial reconstruction unit (SRU)

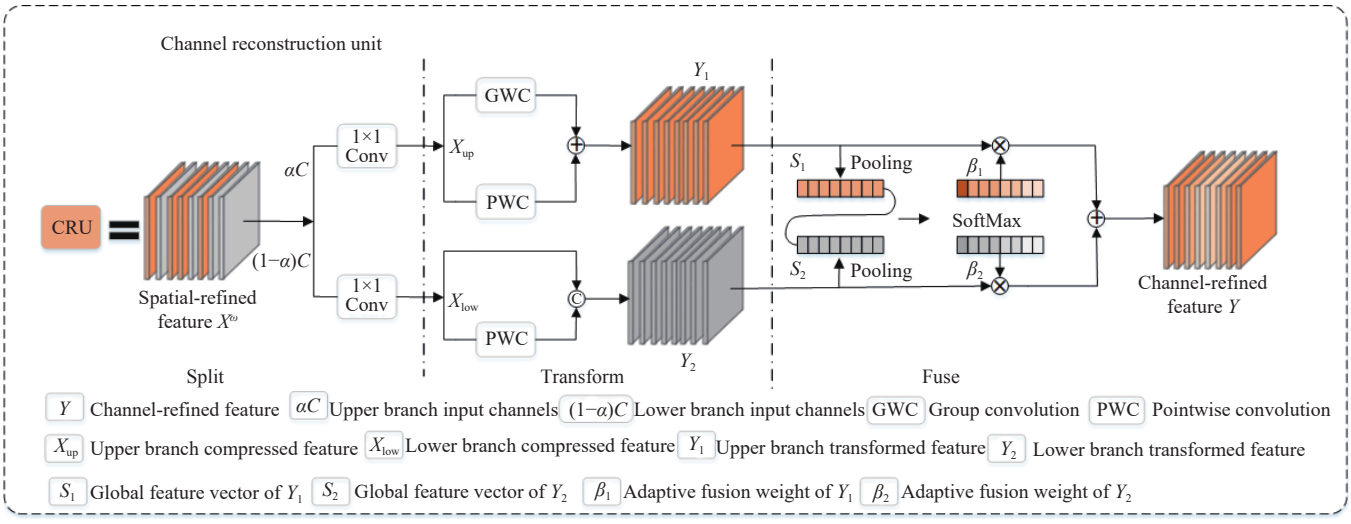


Figure 9 Channel reconstruction unit (CRU)

2.2.3 Adding angle prediction

The rice seedling and weed in images exhibit overlapping and intersecting regions. To minimize the spatial discrepancy between bounding boxes and actual plant morphology while reducing interference from background elements, this study introduces rotated bounding boxes for object detection, and an angle prediction is added into the model. Common representation methods for rotated bounding boxes include the OpenCV five-parameter format, long-edge representation, and eight-parameter representation. Among them, the long-edge representation requires fewer parameters, making it more computationally efficient for rotated bounding box regression. As illustrated in Figure 11, the long-edge representation method defines rotated bounding boxes using five parameters (x, y, w, h, θ), where (x, y) denotes the center coordinates

of the rotated bounding box, and w and h represent the short and long sides of the rectangle, respectively. The angle θ is defined as the inclination between the long side of the bounding box and the x-axis, with a positive value assigned to clockwise rotation within the range of [–90°, 90°). However, as shown in Figure 11, the angular boundary values of −90° and 90° differ by 180° numerically, despite representing the same physical orientation. This results in a periodic discontinuity in the angular parameter, which leads to non-differentiability in the loss function during model training and consequently compromises learning stability. To resolve this boundary discontinuity issue, the circular smooth label (CSL)[29] is introduced in this study, as shown in Figure 12. CSL provides an effective solution for rotated object detection by transforming angle regression into a classification task with smooth label encoding, thereby resolving boundary discontinuity and periodicity issues in angle prediction. As shown in Figure 12, CSL treats the target angle as a categorical label, and the number of angle categories depends on the range of the angle. Given the angle range defined as [–90°, 90°), if each degree corresponds to an angle category, the network model will have 180 angle categories. Alternatively, if each 2 degrees corresponds to an angle category, there will be 90 angle categories. If the angular interval is set as τ, the precision error introduced by the classification interval for angle classification can be expressed as shown in Equations (8) and (9).

$$Max_{loss} = \frac{\tau}{2} \tag{8}$$

$$E_{loss} = \int x \cdot P(x) = \int_0^{\frac{\tau}{2}} x \cdot \frac{1}{\frac{\tau}{2} - 0} \mathrm{d}x = \frac{\tau}{4} \tag{9}$$

where, Max_{loss} represents the maximum precision error; E_{loss} denotes the desired precision error, x signifies the angular loss; and $P(x)$ refers to the probability density function. Assuming a uniform distribution x, if the angular interval τ is set to 1, the maximum precision errors would be 0.5 and 0.25, respectively, which can be considered negligible. This approach can effectively resolve the issue of abrupt angular jumps caused by the continuity at angle boundaries. The expression for CSL is given by Equation (10).

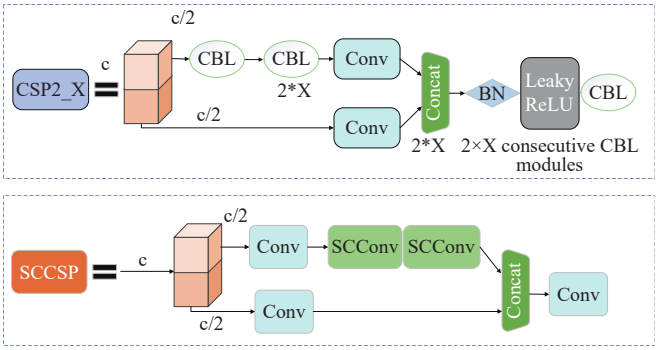


Figure 10 Structures of CSP2_X and SCCSP

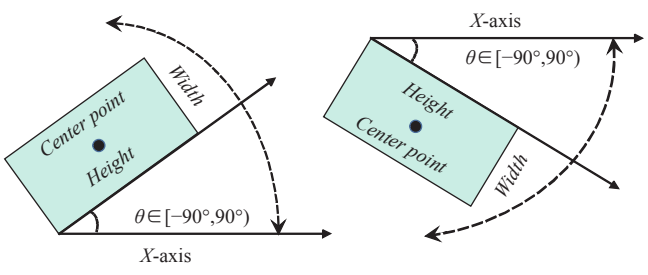


Figure 11 Long-edge representation

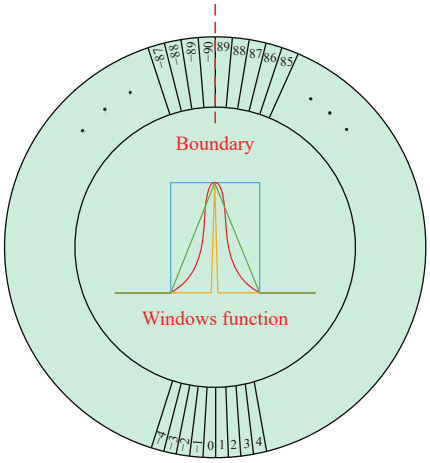


Figure 12 Circular smooth label

$$CSL = \begin{cases} g(x), & \theta - r < x < \theta + r \\ 0, & otherwise \end{cases} \tag{10}$$

where, $g(x)$ represents the window function, and the window radius is controlled by r.

To achieve angle prediction and rotated box representation, the angular classification loss is added to the original YOLOv5 loss function, which is calculated as follows:

$$Loss = \zeta_{box} \times L_{box} + \zeta_{cls} \times L_{cls} + \zeta_{obj} \times L_{obj} + \zeta_{ang} \times L_{ang} \tag{11}$$

where, L_{box}, L_{cls}, L_{obj}, L_{ang} are bounding box loss, classification loss, confidence loss, and angle classification loss, respectively; ζ_{box},ζ_{cls}, ζ_{obj},ζ_{ang} correspond to different loss weights, used to quantify the degree of overlap between predicted and ground truth bounding boxes in object detection. The L_{box} loss is calculated based on CIoU Loss, as shown in Equation (12).

$$L_{box} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \vartheta v \tag{12}$$

where, IoU represents the intersection ratio between predicted and ground truth bounding boxes in object detection; c represents the shortest diagonal length of the smallest bounding box covering a predicted box and a ground truth box; b and b^{gt} denote the center points of the predicted and ground truth bounding boxes, respectively; ρ is the Euclidean distance between b and b^{gt}; ϑ and v are used to measure the discrepancy of the width-to-height ratio of a predicted box and a ground truth box, which are calculated as follows:

$$\vartheta = \frac{v}{(1 - IoU) + v} \tag{13}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{14}$$

where, w^{gt} and h^{gt} represent the width and height of the ground truth bounding box, respectively, while w and h represent the width and height of the predicted bounding box, respectively. In addition, L_{cls} and L_{obj} are computed using the binary cross-entropy loss function, as shown in Equation (15).

$$L_{cross-entropy} = -\sum_{n=1}^{N} y_i^* \ln(y_i) + (1 - y_i^*)\ln(1 - y_i) \tag{15}$$

where, N represents the total number of sample classes; y_i stands for the probability of the identified class after being processed by an activation function; and y_i^* is the ground truth value for the current class. The angle classification loss L_{ang} is calculated based on the

binary cross-entropy loss function, as shown in Equation (16).

$$\begin{cases} L_\theta = mean\{l_0, \ldots, l_{N-1}\} \\ L_n = sum\left\{L_{n,0}, L_{n,1} \ldots, L_{n,179}\right\} \\ y_i^* = CS\,L(x) \end{cases} \tag{16}$$

where n represents the number of samples; i stands for the angle category, and $i \in [0°, 180°)$; y_i^* is the true label value for the i-th angle category. The predicted label y_i and the true label y_i^* are then input into a binary cross-entropy loss function, which is computed separately for each sample to obtain $l_{n,i}$. These individual losses are summed to get l_n, which represents the total angle loss across all samples. Finally, the average of these losses is taken over the N samples to derive the average angle classification loss.

3 Experiments and analysis

3.1 Experimental setup and evaluation metrics

The experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4080 GPU (32 GB VRAM), an Intel Core i7-11700K CPU, and the Windows 11 operating system. The software environment included CUDA 12.1, Python 3.8, and PyTorch 1.13.0 framework. During training, we used a batch size of 8 and trained all models for 200 epochs. To comprehensively evaluate model performance, we assessed both computational efficiency and detection accuracy using the following metrics: number of parameters, floating-point operations (FLOPs), precision (P), recall (R), and mean average precision (mAP).

3.2 Visualization of rotated bounding box and horizontal bounding box detection

To visually compare the effectiveness of rotated and horizontal bounding boxes in detecting seedling and weed, we employed Grad-CAM[30] to generate heatmaps for both detection models based on the YOLOv5s framework, as shown in Figure 13. In these heatmaps, regions highlighted in dark red indicate high model attention, while light blue areas represent lower relevance. From the figure, it can be seen that when the YOLOv5 model with horizontal bounding box is used for weed detection, the background regions between adjacent heat peaks exhibit higher heat intensity. This observation indicates that the weed features learned by the network are not sufficiently concentrated, resulting in a less distinct separation between the background and the weeds. In contrast, when using the YOLOv5 model with rotated bounding box (named as YOLOv5s_obb) for detection, the brightness of the surrounding background is lower. By aligning more precisely with target morphology and orientation, the rotated bounding boxes reduce background interference and help the model concentrate attention on the actual targets, resulting in more focused and discriminative feature activation patterns.

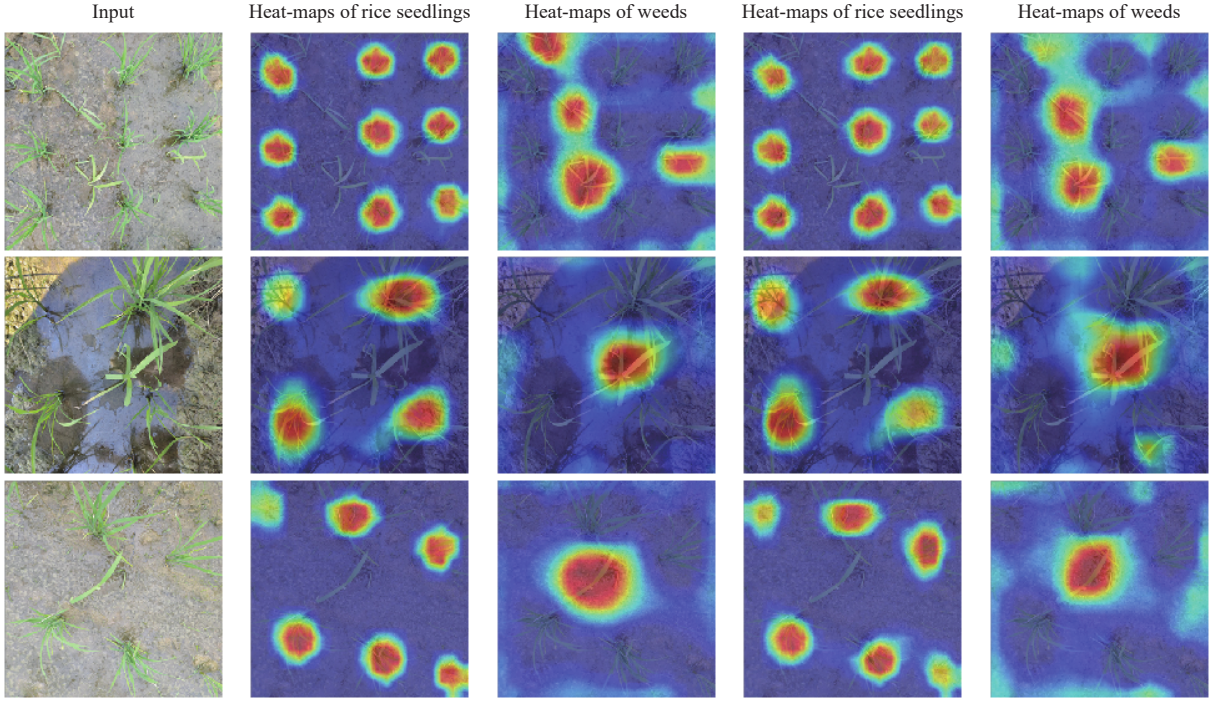


Figure 13 Comparisons of rotated box and horizontal box heatmaps

3.3 Comparison of different lightweight backbone networks

To evaluate whether MobileNetV3 achieves an optimal balance between lightweight design and detection performance, comparative experiments were conducted using three alternative lightweight networks, including ShuffleNetV2, GhostNet, and PP-LCNet, to replace the backbone network of YOLOv5. Table 2 lists the performance of the model using different backbone networks on the self-constructed dataset, where P represents the average recognition precision for seedling and weed, R denotes the average recall rate for seedling and weed, Parameter signifies the number of neural network parameters, GFLOPs indicates the computational complexity, and FPS is the number of frames per second, reflecting the detection speed of the compared model. In terms of precision and recall, the original YOLOv5 model demonstrates superior performance than the other four models. It also exhibited the largest parameter count and computational complexity. Replacing the backbone with ShuffleNetV2 resulted in the most lightweight model, reducing parameters and GFLOPs by 85.47% and 84.97%, respectively, compared to the original YOLOv5. However, this came at the cost of a significant decline in detection accuracy, which is undesirable for precision agriculture applications requiring reliable recognition. In contrast, models using MobileNetV3, GhostNet, and PP-LCNet as backbones maintained competitive accuracy with only marginal decreases compared to the original

YOLOv5. Although, the backbone model with MobileNetV3 showed a slight reduction in mAP of 0.5% and 0.4% lower than GhostNet and PP-LCNet, respectively, it required only one-third of the parameters of the GhostNet-based model and significantly fewer than the PP-LCNet version. Moreover, the MobileNetV3 backbone achieved the fastest inference speed among all lightweight variants. These results demonstrate that the MobileNetV3-enhanced YOLOv5 attains the most favorable trade-off between model performance and complexity, with only a limited compromise in detection accuracy.

Table 2 Comparison of performance metrics for different lightweight backbone models

Backbone	*P*/%	*R*/%	mAP/%	Parameter/ M	GFLOPs	FPS
YOLOv5	94.8	92.6	95.2	7.50	17.3	39.84
YOLOv5+ShuffleNetV2	88.0	86.4	89.6	1.09	2.6	154.31
YOLOv5+GhostNet	92.2	89.3	90.4	4.99	8.6	95.64
YOLOv5+PP-LCNet	92.1	90.7	92.6	3.85	6.3	112.20
YOLOv5+MobileNetV3	91.7	88.9	92.3	1.60	3.4	128.90

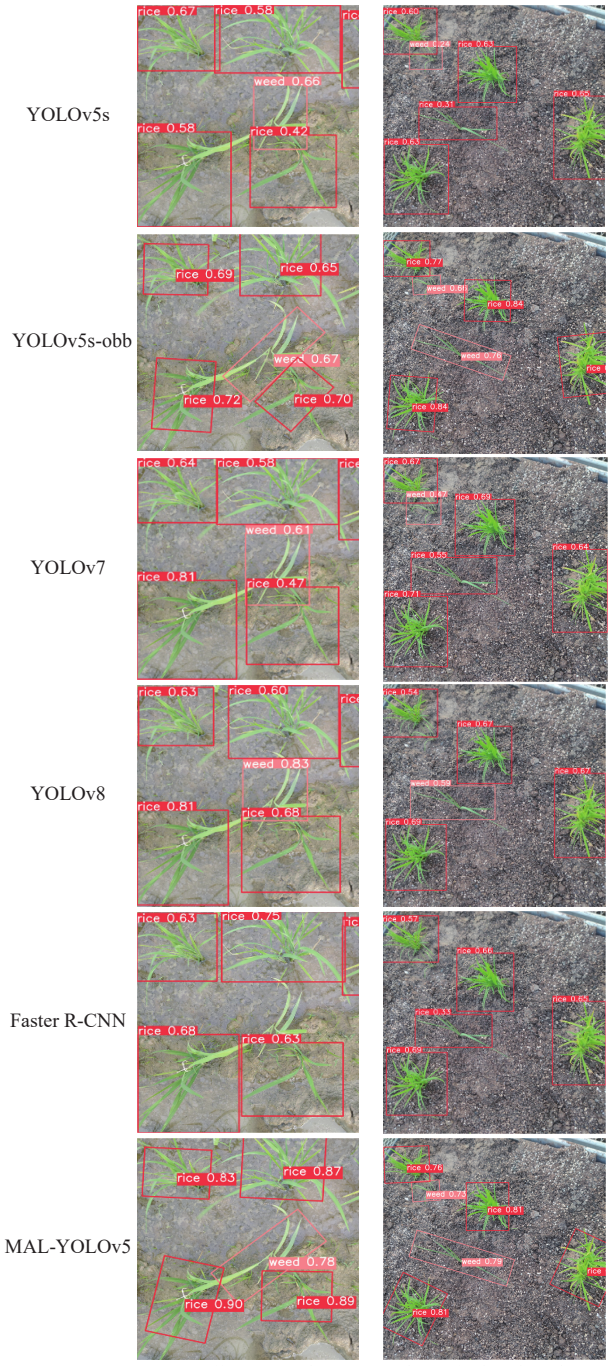
3.4 Performance comparison of different detection models

To validate the performance of the proposed model for rice seedling and weed detection, we compared it to the Faster R-CNN, YOLOv5s, YOLOv7, YOLOv8, and YOLOv5s_obb. The results are listed in Table 3. As shown in Table 3, Faster R-CNN, as a two-stage detector, showed the lowest detection efficiency. The proposed MAL-YOLOv5 achieved remarkable lightweight characteristics while maintaining competitive accuracy. Compared to YOLOv5s_obb, the MAL-YOLOv5 reduced the number of parameters by 80.1% and the GFLOPs by 81.5%. Furthermore, the MAL-YOLOv5 achieved a 238% increase in FPS over YOLOv5s_obb, reflecting significantly accelerated detection capability. This enhanced efficiency enables more effective utilization of computational resources and facilitates practical deployment on computationally constrained embedded platforms. Compared to YOLOv7 and YOLOv8, although YOLOv8 achieved a relatively high detection speed (55.69 FPS), its computational cost (28.2 GFLOPs) remains 8.8 times higher than ours, and YOLOv7 achieved high detection rates but suffered from excessive parameters (36.65 M). These results confirm that MAL-YOLOv5 exhibits superior cost-effectiveness, achieving an optimal balance between detection performance and computational efficiency for agricultural vision tasks.

Table 3 Comparison of performance metrics among different models

Model	*P*/%	*R*/%	mAP/%	Parameter/ M	GFLOPs	FPS
YOLOv5s	93.3	89.1	93.5	7.02	15.8	46.67
Faster R-CNN	91.8	90.2	91.2	13.70	37.0	17.20
YOLOv7	93.7	91.9	92.5	36.65	104.1	28.31
YOLOv8	94.2	92.4	92.7	11.06	28.2	55.69
YOLOv5s_obb	94.8	92.6	95.2	7.50	17.3	39.84
MAL-YOLOv5	93.1	91.9	93.4	1.49	3.2	134.78

Figure 14 presents a comparative visualization of detection results between the proposed MAL-YOLOv5 and baseline models on the test set. As shown in Figure 14b, MAL-YOLOv5 accurately identifies weeds that are misclassified as rice seedlings by YOLOv7, and Faster R-CNN, demonstrating superior discriminative capability in distinguishing similar morphological features. Figure 14a further reveals that our model achieves more accurate rice seedling detection compared to YOLOv8. YOLOv5s produces imprecise localizations with excessive background inclusion and overlapping bounding boxes, as visible in Figure 14, which illustrates missed detections when seedlings and weeds are in close proximity, indicating that horizontal bounding boxes struggle with accurate target representation in dense arrangements. Most notably, Figure14b shows misclassification where weeds are incorrectly identified as seedlings, suggesting that overlapping bounding boxes introduce feature contamination that impedes effective feature learning. In contrast, MAL-YOLOv5 with rotated bounding boxes effectively mitigates these issues. The angular annotations better accommodate plant morphology, reduce inter-leaf overlap, and minimize background interference, resulting in more precise localization and classification.



a. Quanjiao county base b. Agricultural garden

Figure 14 Comparison results of rice seeding and weed detection using different models

3.5 Ablation experiment results

To validate the effectiveness of each module in constructing a lightweight network model, we conducted comprehensive ablation tests with six different model configurations on the rice seedling datasets. The experimental results are presented in Table 4, where "√" indicates model adoption, and "-" indicates its omission. As shown in Table 4, when MobileNetV3 alone was adopted as the backbone network, a significant reduction in model size was achieved, with parameters decreasing from 7.50 M to 1.60 M and GFLOPs dropping from 17.3 to 3.4. This configuration also substantially improved inference speed, raising FPS from 39.84 to 128.90. However, these efficiency gains came at the cost of reduced precision and recall, illustrating the characteristic trade-off between model complexity and detection accuracy in lightweight design. The incorporation of SCConv alone in the neck network improved precision and recall by 1.1% and 1.6%, respectively, while simultaneously reducing both parameters and computational requirements. This confirms SCConv's effectiveness in minimizing spatial and channel redundancies, thereby enhancing feature learning efficiency. When exclusively employing the SCCSP module, the model maintained comparable accuracy to the baseline while reducing parameters by 0.09 M and GFLOPs by 0.3. The combination of SCConv and SCCSP into a Slim-neck structure yielded comprehensive improvements, demonstrating synergistic effects in redundancy reduction and performance enhancement. Most notably, the integration of MobileNetV3 backbone with the Slim-neck structure achieved the optimal balance between efficiency and accuracy. This configuration reduced model parameters by 80.1% and GFLOPs by 81.5% compared to the original YOLOv5, while the obtained network model exhibited a 1.7% decrease in precision and a 0.7% decrease in recall. This significant reduction in the network model size, achieved with minimal loss in accuracy, demonstrates the effectiveness of the lightweight approach in optimizing computational and storage resources.

Table 4 The results of ablation experiments

Model	MobileNetV3	SCConv	SCCSP	*P*/%	*R*/%	Parameter/ M	GFLOPs	FPS
YOLOv5	-	-	-	94.8	92.6	7.50	17.3	39.84
	√	-	-	91.7	88.9	1.60	3.4	128.90
	-	√	-	95.9	94.2	7.48	17.2	40.16
	-	-	√	94.7	92.8	7.41	17.0	42.04
	-	√	√	95.6	94.4	7.37	16.9	42.42
	√	√	√	93.1	91.9	1.49	3.2	134.78

4 Conclusions

This study presents MAL-YOLOv5, a multi-angle lightweight object identification model for rice seedling and weed detection in field environments. To address the computational challenges inherent in deep learning applications, MobileNetV3 serves as the lightweight backbone, incorporating inverted residual structures, linear bottlenecks, depthwise separable convolutions, and SE attention mechanisms to enable efficient feature extraction under resource constraints. In addition, the combination of SCConv and SCCSP lightweight neck network is constructed, which can reduce spatial and channel redundancies among features in convolutional neural networks, further compressing the network model size and enhancing network performance. Furthermore, rotating bounding boxes are adopted for annotating seedlings and weeds, providing precise morphological alignment with plant targets, thereby minimizing background interference and improving feature learning. Comprehensive evaluations demonstrate that MAL-YOLOv5 achieves exceptional cost-effectiveness compared to mainstream object detection models, such as YOLOv5s, YOLOv7, YOLOv8, and Faster R-CNN, significantly reducing network parameters and computational requirements with minimal loss in accuracy. Future work will focus on hardware-aware optimization for embedded platforms and extending the approach to additional crop-weed systems, facilitating practical implementation in intelligent agricultural machinery.

Acknowledgements

The research was funded by the National Science Foundation of China (Grant No. 32201665), the National Natural Science Foundation of China (Grant No. 32572210), and the Agricultural Sciences Academy of Anhui Province Talent Project (Grant No. QNYC-202504).

[References]

[1] Li W B, He Z F, Wu L P, Liu S J, Luo L C, Ye X X, et al. Impacts of co-culture of rice and aquatic animals on rice yield and quality: A meta-analysis of field trials. Field Crops Research, 2022; 280: 108468.

[2] Shao Y Y, Guan X L, Xuan G T, Gao F R, Feng W J, Gao G L, et al. GTCBS-YOLOv5s: A lightweight model for weed species identification in paddy fields. Computers and Electronics in Agriculture, 2023; 215: 108461.

[3] Rao A N, Singh R G, Mahajan G, Wani S P. Weed research issues, challenges, and opportunities in India. Crop Protection, 2020; 134: 104451.

[4] Vijayakumar V, Ampatzidis Y, Schueller J K, Burks T. Smart spraying technologies for precision weed management: A review. Smart Agricultural Technology, 2023; 6: 100337.

[5] Ghatrehsamani S, Jha G, Dutta W, Molaei F, Nazrul F, Fortin M, et al. Artificial intelligence tools and techniques to combat herbicide resistant weeds-A review. Sustainability, 2023; 15(3): 1843.

[6] Tudi M, Ruan H D, Wang L, Lyu J, Sadler R, Connell D, et al. Agriculture development, pesticide application and its impact on the environment. International Journal of Environmental Research and Public Health, 2021; 18(3): 1112.

[7] Li Y, Guo Z Q, Shuang F, Zhang M, Li X H. Key technologies of machine vision for weeding robots: A review and benchmark. Computers and Electronics in Agriculture, 2022; 196: 106880.

[8] Zheng J Q, Xu Y L. Development of plant protection methods and advances in pesticide application technology in agro-forestry production. Agriculture, 2023; 13(11): 2165.

[9] Liao J, Chen M H, Zhang K, Zhou H Y, Zou Y, Xiong W, et al. SC-Net: A new strip convolutional network model for rice seedling and weed segmentation in paddy field. Computers and Electronics in Agriculture, 2024; 220: 108862.

[10] Singh V, Singh D. Development of an approach for early weed detection with UAV imagery. In: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, 2022; pp. 4879–4882.

[11] Abouzahir S, Sadik M, Sabir E. Bag-of-visual-words-augmented histogram of oriented gradients for efficient weed detection. Biosystems Engineering, 2021; 202: 179–194.

[12] Dadashzadeh M, Abbaspour-Gilandeh Y, Mesri-Gundoshmian T, Sabzi S, Hernández-Hernández J L, Hernández-Hernández M, et al. Weed classification for site-specific weed management using an automated stereo computer-vision machine-learning system in rice fields. Plants, 2020; 9(5): 559.

[13] Chen Y J, Wu Z N, Zhao B, Fan C X, Shi S W. Weed and corn seedling detection in field based on multi feature fusion and support vector machine. Sensors, 2021; 21(1): 212.

[14] Naik N S, Chaubey H K. Weed detection and classification in sesame crops using region-based convolution neural networks. Neural Computing and Applications, 2024; 36(30): 18961–18977.

[15] Daşkın Z D, Alam M S, Khan M U. Ensemble transfer learning using MaizeSet: A dataset for weed and maize crop recognition at different

growth stages. Crop Protection, 2024; 184: 106849.
[16] Jiang H H, Zhang C Y, Qiao Y L, Zhang Z, Zhang W J, Song C Q. CNN feature based graph convolutional network for weed and crop recognition in smart farming. Computers and Electronics in Agriculture, 2020; 174: 105450.
[17] Li J Y, Zhang W, Zhou H, Yu C T, Li Q D. Weed detection in soybean fields using improved YOLOv7 and evaluating herbicide reduction efficacy. Frontiers in Plant Science, 2024; 14: 1284338.
[18] Chen J, Wang H, Zhang H, Luo T, Wei D, Li T, et al. Weed detection in sesame fields using a YOLO model with an enhanced attention mechanism and feature fusion. Computers and Electronics in Agriculture, 2022; 202: 107412.
[19] Wang Y. Wang F, Li K, Feng X, Hou W, Liu L, et al. Low-light wheat image enhancement using an explicit inter-channel sparse transformer. Computers and Electronics in Agriculture, 2024; 224: 109169.
[20] Li J, Qiao Y, Liu S, Zhang J, Yang Z, Wang M. An improved YOLOv5-based vegetable disease detection method. Computers and Electronics in Agriculture, 2022; 202: 107345.
[21] Zhang Y, Zhang H, Huang Q, Han Y, Zhao M. DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. Expert Systems with Applications, 2024; 241: 122669.
[22] Howard A, Sandler M, Chu G, Chen L C, Chen B, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; pp. 1314–1324.
[23] Zhou G, Liu W, Zhu Q, Lu Y, Liu Y. ECA-MobileNetV3(Large)+SegNet model for binary sugarcane classification of remotely sensed images. IEEE Transactions on Geoscience and Remote Sensing, 2022; 60: 1–15.
[24] Gennari M, Fawcett R, Prisacariu V A. DSConv: Efficient convolution operator. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; pp. 5148–5157.
[25] Wang H, Bhaskara V, Levinshtein A, Tsogkas S, Jepson A. Efficient super-resolution using MobileNetV3. In: Proceedings of the European Conference on Computer Vision, 2020; pp. 87–102.
[26] Li K, Wang J, Jalil H, Wang H. A fast and lightweight detection algorithm for passion fruit pests based on improved YOLOv5. Computers and Electronics in Agriculture, 2023; 204: 107534.
[27] Li H, Xiong P, Fan H, Sun J. DFANet: Deep feature aggregation for real-time semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019; pp. 9522–9531.
[28] Yu X, Lin M, Lu J, Ou L. Oriented object detection in aerial images based on area ratio of parallelogram. Journal of Applied Remote Sensing, 2022; 16(3): 034510.
[29] Yang X, Yan J. Arbitrary-oriented object detection with circular smooth label. In: Proceedings of the European Conference on Computer Vision, 2020; pp. 671–687.
[30] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision, 2020; 128: 336–359.