An Improved YOLOv8-based Method for Real-time Detection of Harmful Tea Leaves in Complex Backgrounds

Xin Leng
Northeast Forestry University
Jiakai Chen
Northeast Forestry University
Jianping Huang
jphuang@nefu.edu.cn
Northeast Forestry University
Lei Zhang
Northeast Forestry University
Zongxuan Li
Northeast Forestry University

Research Article

Keywords: Harmful tea leaves, YOLO-DBD, Focal-CIoU Loss, Dynamic Head, Bi-Level Routing Attention

Posted Date: May 3rd, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4286404/v1

Additional Declarations: No competing interests reported.

An Improved YOLOv8-based Method for Real-time Detection of Harmful Tea Leaves in Complex Backgrounds

Xin Leng[1†], Jiakai Chen[1†], Jianping Huang[1*], Lei Zhang[1†], Zongxuan Li[1†]

[1*]School of Computer and Control Engineering, Northeast Forestry University, He Xing Road, Harbin, 150006, Heilongjiang, China.

*Corresponding author(s). E-mail(s): jphuang@nefu.edu.cn;
Contributing authors: lengxin@nefu.edu.cn; 1067290105@nefu.edu.cn;
†These authors contributed equally to this work.

Abstract

Tea, a globally cultivated crop renowned for its unique flavor profile and health-promoting properties, ranks among the most favored functional beverages worldwide. However, pests and diseases severely jeopardize the production and quality of tea leaves, leading to significant economic losses.While early and accurate identification coupled with the removal of infected leaves can mitigate widespread infection, manual leaves removal remains time-consuming and expensive. To address this challenge, this paper introduces the YOLO-DBD network model for detecting of harmful tea leaves. The model excels in efficiently identifying harmful tea leaves with various poses in complex backgrounds, providing crucial guidance for the posture and obstacle avoidance of a robotic arm during the pruning process.The improvements proposed in this study encompass the C2f-DCN module, Bi-Level Routing Attention, Dynamic Head, and Focal-CIoU Loss function, enhancing the model's feature extraction, computation allocation, and perception capabilities. Comparative analysis with the YOLOv8s model demonstrates a 6% improvement in mAP and a reduction of 3.3G FLOPs in the YOLO-DBD model.

Keywords: Harmful tea leaves,YOLO-DBD, Focal-CIoU Loss,Dynamic Head, Bi-Level Routing Attention

1 Introduction

In the process of tea cultivation and growth, tea diseases are an important factor affecting yield and quality, and severe tea diseases can cause huge economic losses(Ahmed et al. 2018).If tea diseases are accurately and quickly identified in the early stages of the disease, and infected leaves are promptly removed or insecticides are sprayed, it can effectively block the source of the disease, prevent the spread and outbreak of the disease.

In recent years, many researchers have primarily focused on precision pesticide spraying. However, the use of pesticides not only impacts the natural ecological environment but also has implications for the health of consumers(Damalas and Eleftherohorinos. 2011).In contrast, although manual pruning of diseased leaves is safer(Toscano et al. 2015), it requires a lot of labor and is not as efficient as using pesticide, resulting in higher costs.The adoption of robotic pruning presents a solution to these challenges(Arad et al. 2020). Yet, the success of such robots hinges on their ability to accurately detect targets, which is crucial to avoid crop damage and potential collisions(Wu et al. 2021). Therefore, the main challenge addressed in this paper is how to rapidly and accurately

detect harmful tea leaves in complex environments, providing effective guidance for subsequent robot cutting of these harmful tea leaves.

With the advancement of deep learning technology, its application in the field of agriculture has been steadily increasing, particularly in tasks related to object detection. Deep learning enables researchers to tackle more complex image analysis tasks and effectively identify and detect diseases in crops.Image detection networks can be categorized into two classes based on the detection stage: two-stage and one-stage object detection networks.Two-stage object detection networks, exemplified by Faster R-CNN(Fu et al. 2020), operate in two steps: first, they generate candidate regions, and then they classify and perform bounding box regression on these regions. Faster R-CNN stands out as a representative model in the realm of two-stage detection networks.Zhou ea al. proposed a rice disease detection algorithm based on the fusion of Faster R-CNN and FCM-KM, achieving satisfactory performance(Zhou et al. 2019).Wang et al. improved Faster R-CNN and VGG16, enhancing the accuracy of tea leaves detection(Wang et al. 2023) .While these networks exhibit excellent accuracy in detection, their processing speed is relatively slow, rendering them less suitable for real-time applications. In contrast, one-stage networks, such as You Only Look Once(YOLO)(Redmon et al. 2016) and Single Shot MultiBox Detector(SSD)(Liu et al. 2016), directly predict both bounding boxes and class labels on images. The YOLO series, known for its balance of efficiency and accuracy, has been extensive used in agriculture. Xue et al. presented the YOLO-Tea method, utilizing YOLOv5 for detecting tea leaves diseases and achieving an mAP of 79.7%(Xue et al. 2023). Lin et al.introduced the TSBA-YOLO model, focused on detecting withering disease in tea leaves, with an improved mAP of 85.35%(Lin et al. 2023). Dai et al. developed a method for detecting crop leaves diseases based on YOLOv5. However, this method exhibits lower accuracy in complex environments(Dai and Fan. 2022). Additionally, Roy et al. made enhancements to YOLOv4, proposing a high-performance real-time fine-grained object detection framework capable of addressing challenges such as dense distribution and irregular shapes(Roy et al. 2022). Sun et.al. introduced a novel concept, utilizing the YOLO-v4 deep learning network collaboratively for ITC segmentation and refining the segmentation results involving overlapping tree crowns using computer graphics algorithms(Sun et al. 2022). Furthermore, Du et.al. proposed a tomato pose detection algorithm based on YOLOv5, aiming to detect the three-dimensional pose of individual tomato fruits, achieving a mAP of 93.4%(Du et al. 2023). Nan et al.developed WGB-YOLO based on YOLOv7 for harvesting ripe dragon fruits, further classifying them based on different growth poses, achieving a mAP of 86.0%(Nan et al. 2023). Therefore, in practical applications, simply classifying harmful tea leaves into one category cannot fully address the issues of robotic harvesting strategy and occlusion. For a cluster of leaves with only one harmful leaves, the whole bundle cutting is not rational; for clustered growing harmful leaves, cutting them one by one would significantly reduce the efficiency of the operation; for occluded harmful leaves, the robotic arm needs to rotate to another angle before cutting.

Current research on harmful leaves detection primarily focuses on simple environments, neglecting the complexities encountered in real-world robotic applications. When pruning harmful leaves, robots must navigate intricate environmental elements, pruning strategies, and occlusion challenges. In densely planted tea gardens, the target of harvesting harmful leaves is challenging to detect accurately due to mutual contact with other leaves and obstruction by tree branches. In real-time field operations, robots need to ensure high FPS and accuracy while also requiring the model's FLOPs to be low. Compared to RCNN and SSD, YOLO exhibits higher efficiency and performance. Therefore, this study chooses YOLO as the benchmark model.To further refine YOLO's performance, we introduce the YOLO-DBD algorithm.

The important contributions of this paper were as follows: (1) Integrating C2f and DCNv2 into the backbone network can better capture target shapes and positions, as well as adaptively altering feature weights to reduce noise during context semantic fusion. (2) Implementing a Bi-Level Routing Attention mechanism that employs

dynamic sparse attention with double-layer routing, ensuring flexible, content-aware computation while reducing computational demands.(3) Adding a dynamic head detection framework at the network's end to enhance the perception of spatial position, scale, and task regions.(4) To address the issue of imbalance between samples and distinguishing between easy and hard samples to reduce the impact of easily separable samples on network training, this paper incorporates Focal on top of CIOU.

2 Materials and Methods

2.1 Image Acquisition

This study concentrated on images of tea leaves. The source of these images was the Dayuan Tea Garden in Jinyun County, Zhejiang Province. The images were taken using a ONEPLUS8T on July 11, 2023, and captured in RGB format. These images, with a resolution of 640×640 pixels, totaled 1,108, were divided into training and validation sets at an 8:2 ratio. This allocation resulted in 887 images for training and 221 for validation. The annotation of images was carried out using labelimg software.

2.2 Classification Criteria

In practical applications, merely classifying harmful tea leaves into a single category falls short of addressing the challenges associated with robotic harvesting strategies and occlusion. For clusters with a single harmful leaf, cutting the entire bundle is not rational; for clusters of growing harmful leaves, cutting them individually would significantly reduce operational efficiency; and for occluded harmful leaves, the robotic arm must rotate to a different angle before cutting.Consequently, harmful leaves are classified into three primary categories. The first category is individual harmful leaves(referred to as IHL), as illustrated in Fig1 (a). The second category is clustered harmful leaves(referred to as CHL), shown in Fig1 (b). The third category is obscured harmful leaves(referred to as OHL), depicted in Fig1 (c). We obtained a total of 1,108 images of harmful tea leaves, comprising 595 images of IHL, 161 images of CHL, and 681 images of OHL.

2.3 Data Augmentation

To enhance the model's robustness and prevent overfitting, image and bounding box augmentation are performed separately. Image-level augmentation includes rotation, noise addition, cutting and splicing as four data augmentation methods, as shown in Fig2. Rotation data augmentation involves clockwise rotations of 90°, 180°, and 270° on the original image. Noise addition data augmentation applies noise to the entire image, with additions of 5% and 10%. Mask data augmentation involves randomly cropping a 10% portion of the image to simulate occlusion. Splicing data augmentation randomly selects four images, performs random cropping, blends the four images, and then blends the resulting mixture with a new image. Bounding box augmentation is mainly achieved by adding noise to the targets, with additions of 5% and 10%.

3 The Proposed YOLOv8-DBD Netwrok

The YOLO series of deep learning detection algorithms, especially YOLOv8, is widely acclaimed in the field of object detection for its outstanding performance in detecting large-scale and sparse targets. Nevertheless, when applied in complex tea garden environments with numerous interfering factors such as complex image backgrounds, overlapping and occlusion of leaves, and small and dense defects in harmful leaves sizes, the performance of YOLOv8s is not satisfactory. To address this issue, this paper proposes an improved YOLOv8s network framework, named YOLO-DBD. Extensive experiments demonstrate that this model significantly enhances the overall performance of detecting harmful leaves in complex environments, including accuracy and efficiency.

The network architecture of YOLO-DBD is illustrated in Fig3. The enhancements encompass four key areas. First, the integration of C2f and DCNv2 into a new C2f-DCNv2 module, coupled with a residual structure, bolsters the network's ability to discern target shapes and positions and refines feature accuracy. Second, the Bi-Level Routing Attention mechanism, following the SPPF module, employs dynamic sparse attention for more efficient computational resource

Fig. 1 Categories of Harmful Leaves: (a) Individual Harmful Leaves(IHL), (b) Clustered Harmful Leaves(CHL), (c) Obscured Harmful Leaves(OHL).

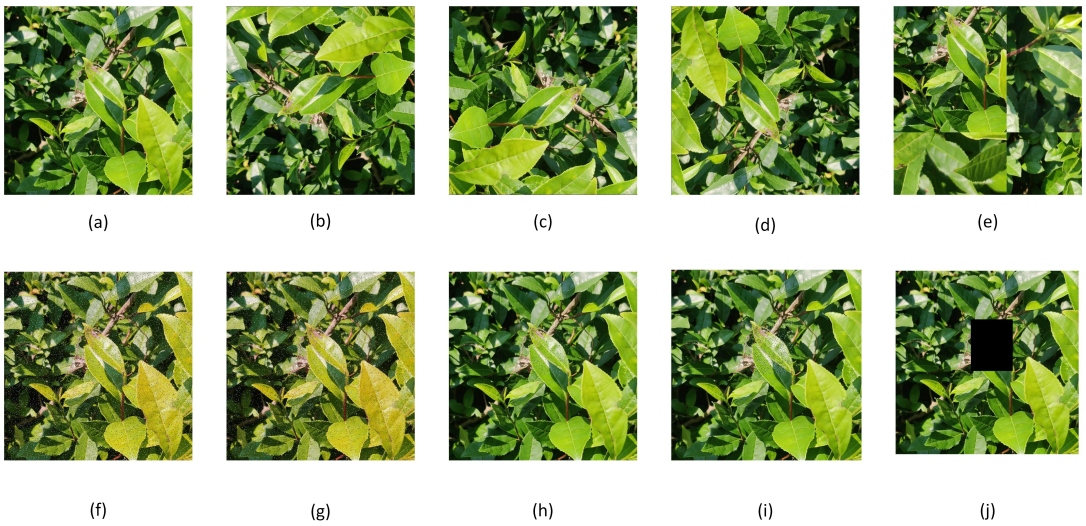

Fig. 2 Different image augmentation methods: (a) Original image, (b) 90° rotation, (c) 180° rotation, (d) 270° rotation, (e) Stitching, (f)5% noise was added to the entire image,(g) 10% noise was added to the entire image,(h) 5% noise was added to the targets,(i) 10% noise was added to the targets,(j) Masking.

allocation, thus augmenting feature extraction while reducing computational demands. Third, the incorporation of the DyHead framework at the network's head merges object detection with self-attention mechanisms, significantly improving detection head's perception and expression capabilities. Lastly, addressing sample imbalance, the CIOU loss function in YOLOv8 is combined with Focal Loss to form Focal-CIOU. These advancements collectively not only elevate YOLO-DBD's accuracy in detecting targets in challenging environments.

3.1 C2f-DCN

Deformable Convolutional Networks (DCNv1)is a Convolutional Neural Network (CNN) architecture that effectively handles geometric transformations(Dai et al. 2017).In traditional CNN modules, the sampling positions of convolutional kernels on input feature maps are fixed, restricting the network's adaptability to geometric transformations.Pooling layers also reduce spatial resolution by a fixed ratio, and Region of Interest Pooling divides the region of interest into a fixed number of spatial units, both of which are unable to flexibly handle geometric changes in targets.Deformable Convolutional Networks enhance convolutional operations by introducing learnable

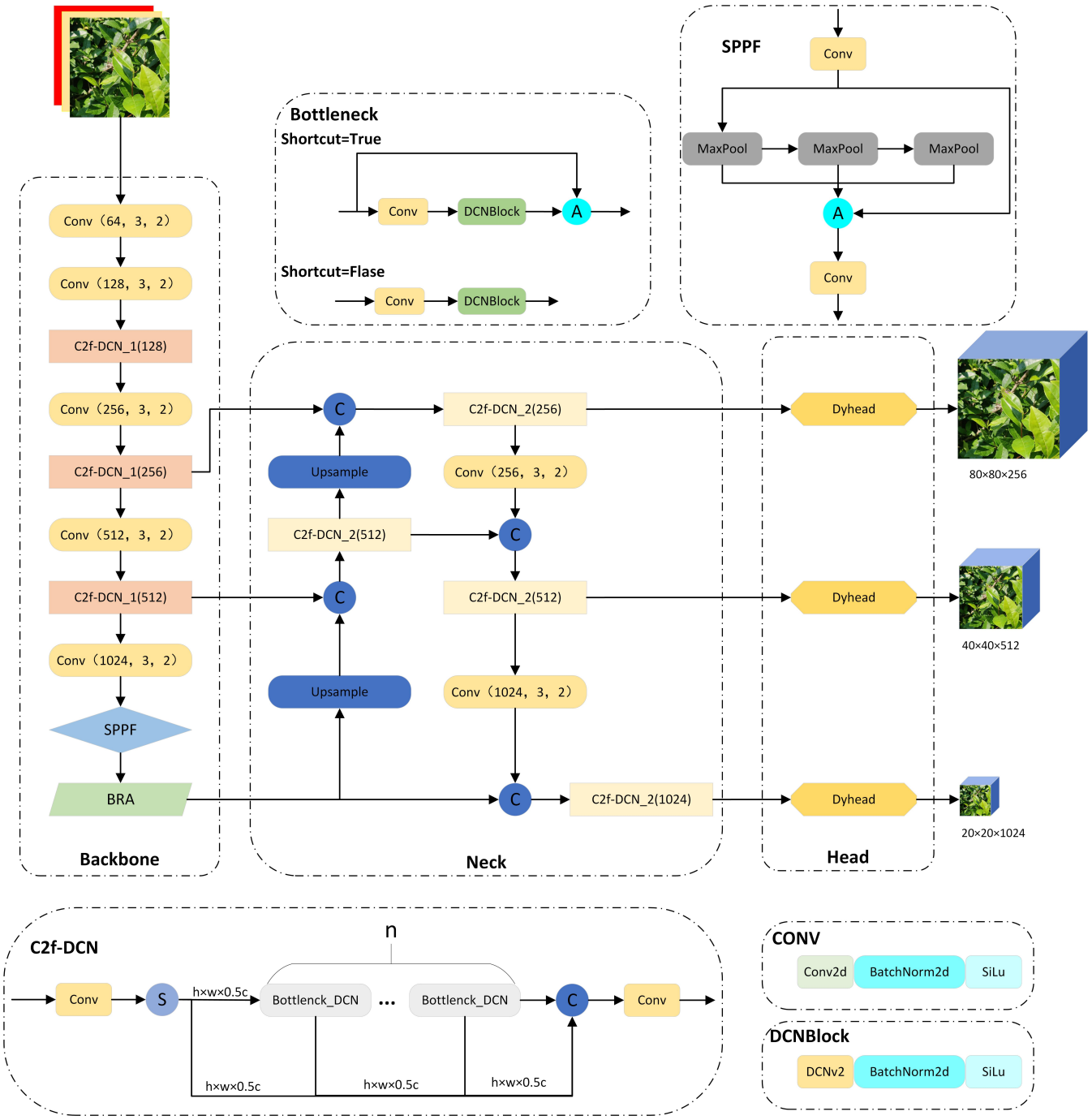

Fig. 3 YOLO-DBD network structure diagram.

offsets, allowing the sampling positions of convolutional kernels to dynamically adjust based on input features.This implies that the network can adaptively adjust the size and shape of its receptive field, thus better capturing the geometric changes in targets.DCNv2 improves upon this by adding extra convolutional layers to predict offsets for each convolutional kernel position, and these offsets are then used to guide the sampling positions in the main convolutional layer(Wang et al.

2021).This design allows DCNv2 to partially overcome the limitations of traditional CNN modules in handling geometric transformations, thereby improving the model's capability and accuracy in recognizing objects with diverse shapes.The architecture of DCNv2 is depicted in Fig4.

Additionally, in DCNv2 incorporates weight coefficients alongside offsets at each sampling point, effectively discerning relevant from irrelevant areas. These coefficients assign a weight to every sampling point, ensuring that points in irrelevant areas (those not of interest) receive a weight of zero, effectively excluding them from feature extraction. The computation formula is as shown in Equation1:

$$y(p_0) = \sum_{p_n \in \mathbb{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \quad (1)$$

In our study, we have fused the C2f module with the modulation mechanism of DCNv2, adopting a multi-branch and residual structure to more effectively capture the target's shape and positional features, as illustrated in Fig5. The DCNv2 modulation mechanism plays a pivotal role, as it learns the offsets and feature amplitudes of samples to precisely control sample space and impact. This enhances the model's accuracy in locating targets and its adaptability to varying target shapes. Consequently, the model demonstrates high precision and robustness, even in complex environments. The C2f-DCN module's multi-branch structure follows efficient aggregation network principles, enriching the model's gradient flow through added layers and connections. This structure, with consistent input-output channels across branches and concat operations for feature concatenation, ensures a stable and effective gradient path, maintaining network efficiency at deeper levels. To prevent the problem of gradient vanishing as the network depth increases, the C2f-DCN module introduces a residual structure in the backbone network. This design not only ensures that the network can extract finer-grained features but also prevents the degradation of network performance, ensuring training stability and effectiveness.

3.2 Bi-Level Routing Attention

YOLOv8 is a convolutional neural network model primarily focused on local processing, making it challenging to capture complex relationships among global features. On the other hand, the complexity of the background often makes it difficult for detection models like YOLOv8 to accurately identify targets, leading to excessive reliance on background information and neglect of crucial details. In contrast, Transformers, with their unique attention mechanism, can capture global correlations among data, constructing a more powerful and flexible data-driven model.However, while Transformers excel in enhancing a model's ability to handle complex data, their high computational complexity and large memory footprint pose challenges, especially when dealing with large-scale data. Directly integrating traditional attention mechanisms into the model consumes significant computational and storage resources, thereby reducing inference efficiency. To alleviate this burden, researchers suggest adopting a sparse query approach, focusing only on specific key-value pairs to reduce resource usage. Following this line of thinking, a series of related research outcomes have emerged, such as local attention, deformable attention, and expansive attention, all of which employ manually set static patterns and content-unrelated sparsity.To overcome these limitations, Zhu et al.proposed a new dynamic sparse attention mechanism(Zhu et al. 2023) : Bi-Level Routing Attention.This novel approach, illustrated in Fig6, offers a more efficient and effective method for attention-based processing.

The primary function of BRA is to enhance the model's attention to key information while suppressing dependence on background information. BRA can adaptively filter out the least relevant key-value pairs in the coarse-grained region of the input feature map. This means it can concentrate the model's computational resources on the information most relevant to the target, thereby more effectively extracting crucial features. In this way, BRA ensures that the model maintains high sensitivity and accurate recognition ability to the target even in complex backgrounds. BRA enhances the model's detection performance with a minimal increase in computational cost. In summary, BRA plays a crucial role in the YOLO-DBD network by

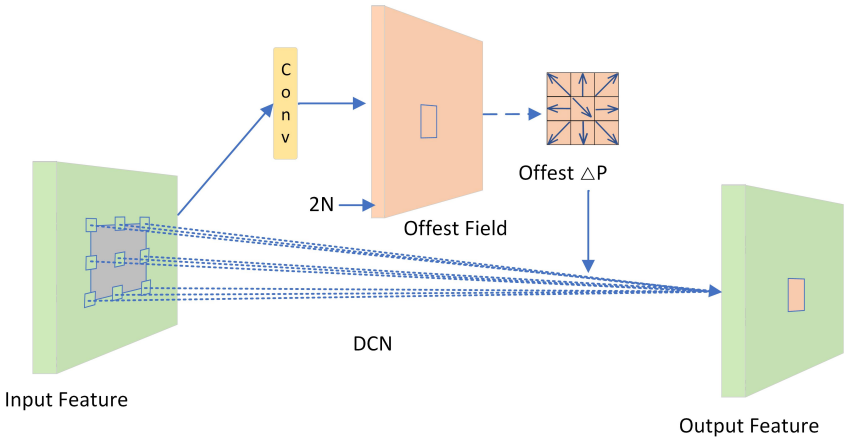

Fig. 4 DCNv1 network diagram.

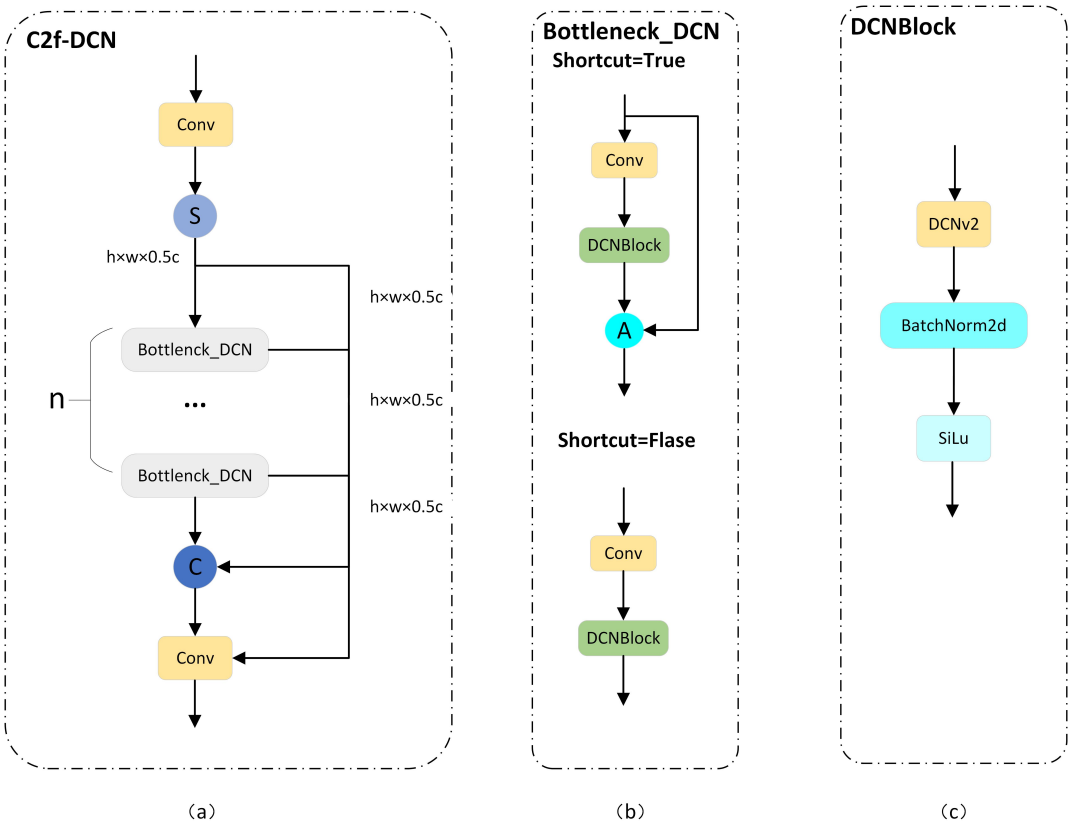

Fig. 5 (a) C2f-DCN network diagram, (b) Bottleneck_DCN network diagram, (c)DCNBlock network diagram.

improving sensitivity to key information through precise information filtering and attention calculation, reducing reliance on complex background information. This makes the YOLO-DBD network perform exceptionally well, more efficiently, and accurately in handling object detection tasks in complex backgrounds.

The process begins with the input feature map $x \in \mathbb{R}^{H \times W \times \mathbb{C}}$, which is divided into $\mathbb{S} \times \mathbb{S}$ sub-blocks, each comprising $\frac{HW}{S^2}$ feature vectors.

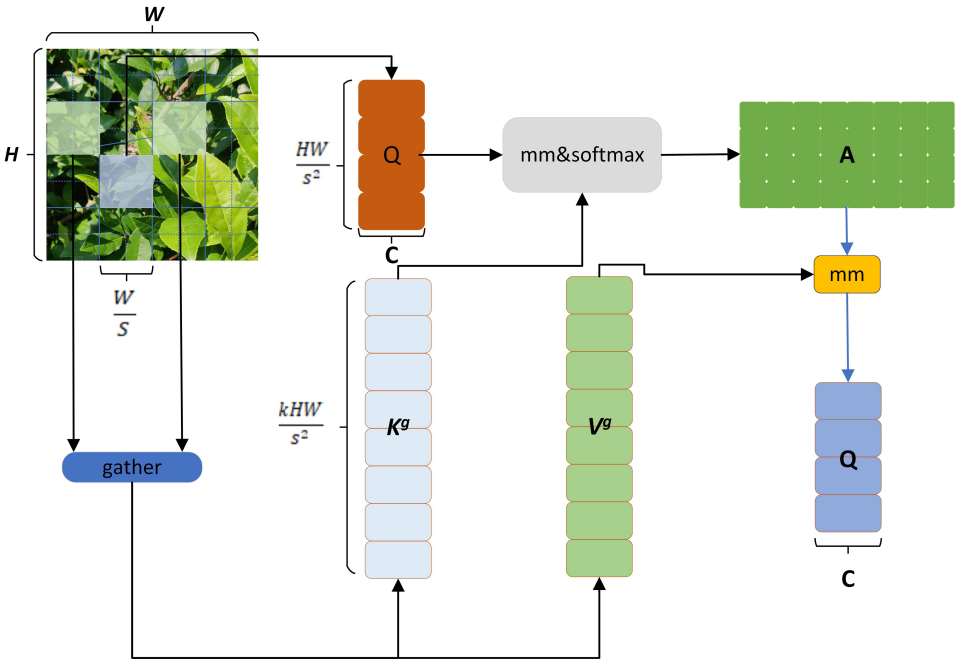

Fig. 6 Bi-Level Routing Attention.

Subsequently,by adjusting the dimensions of X,a new matrix with the shape $X^r \in R^{S^2 \times \frac{HW}{S^2} \times C}$ is obtained. Following this, a linear transformation operation on these vectors produces three matrices: Q, K, and V. The respective formulas for these transformations are detailed in Equations2, 3, and 4:

$$Q = X^r W^Q \tag{2}$$

$$K = X^r W^k \tag{3}$$

$$V = X^r W^V \tag{4}$$

Subsequently, a directed graph is constructed to locate the related areas within a given region, thereby determining the attention relationships between different regions. The specific steps are as follows: Firstly, the Q and V vectors within each region are averaged, resulting in the Q^r and K^r vectors representing the entire region, with dimensions $R^{S2 \times S2}$. Subsequently, by computing the dot product of Q^r andK^r, an adjacency matrix $A_r \in \mathbb{R}^{S^2 \times S^2}$is obtained. This matrix quantifies the relationships between different regions. The computation formula is as shown in Equation5:

$$A^r = Q^r (K^r)^T \tag{5}$$

Then, the adjacency matrix A^r is pruned. At a coarser granularity level, the less correlated tokens in A^r are filtered out, retaining only the top k regions with the strongest relevance, resulting in a routing index matrix $I^r \in \mathbb{R}^{S^2 \times k}$.The calculation formula for this processing step has been clearly demonstrated. Through such processing, we can more efficiently focus on important regions, thereby reducing computational load while maintaining key information. The computation formula is shown in Equation6:

$$I^r = topkIndex(A^r) \tag{6}$$

Next, token-to-token attention is applied at a finer granularity level. For the queries located in region i, attention is only paid to the k regions specified by the routing index matrix I^r.Specifically, the k indices $I^r_{(i,1)}, I^r_{(i,2)}, \cdots, I^r_{(i,k)}$ are used to retrieve and aggregate all the K and V tensors within these regions, thus forming K^g and V^g.The specific computational method for this process can be referred to in the given formula. In this way, we can capture the dependencies between tokens at a finer level, achieving more precise attention allocation. The computation formulas are shown in Equations7 and

8:

$$K^g = gather(K, I^r) \tag{7}$$

$$V^g = gather(V, I^r) \tag{8}$$

Finally, the aggregated K^g and V^g undergo attention processing, and a Local Context Enhancement term LCE(V)is introduced. Through this operation, the output tensor O is obtained. This process can be detailed through the provided formula. The introduction of the LCE(V) is intended to capture and reinforce the local dependencies between input features. This ensures that while maintaining the advantages of the global attention mechanism, the importance of local features is also fully considered, ensuring the comprehensiveness and accuracy of the model's output. The computation formula is shown in Equation9:

$$O = Attention(Q, K^g, V^g) + LCE(V) \tag{9}$$

3.3 Dynamic Head

In order to enhance the perceptual capabilities of the model, this paper introduces a new dynamic detection framework called Dynamic Head (DyHead)(Dai et al. 2021), DyHead ingeniously integrates the spatial scale, spatial position information, and task awareness of objects, and utilizes a multi-head self-attention mechanism to enhance the expressive power of the model. This not only allows DyHead to adapt more flexibly to defects of different shapes but also significantly improves the model's detection accuracy.Compared to the original head structure, DyHead demonstrates significantly enhanced expressive power and detection accuracy, showcasing its effectiveness and superiority in complex defect detection tasks.

DyHead enriches the target detection model's perception by incorporating a unique attention mechanism with three distinct components: spatial-aware attention, scale-aware attention, and task-aware attention. Each component plays a crucial role in enhancing the model's overall performance across different tasks, scales, and spatial locations:

(1) Spatial-Aware Attention: Using deformable convolution techniques, precise positional information of the targets is extracted from the feature map, ensuring the model accurately focuses on the spatial region where the targets are located.

(2) Scale-Aware Attention: By combining 1×1 convolution, ReLU activation function, and sigmoid activation function, features of different scales are merged, thereby obtaining spatial scale information of the target. This ensures the model's robustness to targets of varying sizes.

(3) Task-Aware Attention: Employing a fully connected network as a 'classifier' to expand and categorize input information, ensuring the model can extract the most relevant feature information according to different task requirements.

The application process of DyHead in the YOLOv8-DBD model is illustrated in Fig7: Firstly, feature extraction is performed on the input image as shown in Fig7(a), generating three different scales of feature maps temp1, temp2, and temp3. These feature maps correspond to targets of different sizes in the image, where temp1 corresponds to large-sized targets, temp2 to medium-sized targets, and temp3 to small-sized targets. Subsequently, to facilitate feature fusion, temp1, temp2, and temp3 undergo up/down-sampling operations to achieve a uniform scale. This step ensures that feature maps of different scales can be effectively merged at the same spatial dimension. In this way, a new three-dimensional tensor $F_1, F_2, F_3 \in \mathbb{R}^{L \times \bar{S} \times C}$ is generated, where L represents different feature levels, S the spatial dimension of the feature map, and C the number of feature channels.

Subsequently, these three feature tensors $\mathbb{F}_1$, $\mathbb{F}_2$, and $\mathbb{F}_3$ are passed through the spatial-aware attention, scale-aware attention, and task-aware attention mechanisms. The computation formula is as shown in Equation10:

$$W(F) = \pi(F) \cdot F \tag{10}$$

where $\pi(\cdot)$ is an attention function.

From the above expression, it is evident that significant enhancement in object detection tasks is achieved through in-depth learning of the tensor F across the three dimensions of scale, space, and task. In the scale dimension, by learning features across different levels, the model adapts to the diversity of target sizes, enhancing its robustness to scale variations. In the spatial dimension, learning features of different spatial positions enables the model to handle various geometric transformations of object shapes, maintaining detection accuracy and stability in complex scenes. In the

task dimension, cross-channel learning of tensor F allows the model to correlate features of different tasks and channels, automatically adjusting the importance of features to suit current task requirements.

By implementing the attention mechanism as fully connected layers, it is possible to perform complex attention operations on high-dimensional tensors, thereby capturing and enhancing the model's feature representation capabilities. However, this approach, when applied directly to high-dimensional tensors, incurs substantial computational costs, especially at higher dimensions, making it impractical in real-world applications. To address this issue, a decomposition strategy is employed, breaking down the overall attention function into three consecutive sub-attention operations, each focusing on a single dimension of the tensor. This decomposition strategy significantly reduces computational costs while still maintaining the effectiveness of the attention mechanism. The computation formula is shown in Equation11:

$$W(F) = \pi_{\mathbf{c}}(\pi_{\mathbf{s}}(\pi_{\mathbf{l}}(F) \cdot F) \cdot F) \cdot F \tag{11}$$

In the equation,$\pi_l(\cdot)$,$\pi_s(\cdot)$,$\pi_c(\cdot)$ respectively represent three distinct attention functions applied to the scale, spatial, and task dimensions.

The spatial-aware attention, as denoted by π_l in Fig7(b), is formulated as expressed in Equation12:

$$\pi_1(F) \cdot F = \sigma\left(f\left(\frac{1}{\text{sec}}\sum_{S,C} F\right)\right) \cdot F \tag{12}$$

Herein, $f(x)$ denotes a linear function akin to a (1×1) convolution layer; $\delta(x)$ represents the Hard Sigmoid function, as depicted in Equation13:

$$\sigma(x) = \max\left[0, \min\left(1, \frac{x+1}{2}\right)\right] \tag{13}$$

Scale-aware attention, as indicated by π_s in Fig7(b), is defined as per the formulation presented in Equation14:

$$\pi_S(F)\cdot F = \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} w_{l,k}\cdot F(1, p_k + \Delta p_k, c)\cdot \Delta m_k \tag{14}$$

Herein, k denotes the number of sparse sampling positions, and $p_k + \Delta p_k$ represents the position shifted by the self-learned spatial offset Δp_k, focusing on a discriminative region.Δm_k is a critical self-learned scalar at position p_k, Δp_k and Δm_k are learned from the input features at the median level of F.Task-aware attention, as shown by π_c in Fig7(b), is expressed as in Equation15:

$$\begin{aligned}\pi_c(F)\cdot F = \max\left(\alpha^1(F)\cdot_C + \beta^1(F), \alpha^2(F)\cdot\right.\\ \left. F_{\mathbf{C}} + \beta^2(F)\right)\end{aligned} \tag{15}$$

In the equation, $F_{\mathcal{C}}$ represents the feature segment of the c-th channel, and $\theta(i) = [\alpha^1, \alpha^2, \beta^1, \beta^2]$ is a hyper function employed for learning to control the activation thresholds in attention mechanisms.

The three attention mechanism modules are sequentially applied. The detailed structure of the dynamic sealing module is illustrated in Fig7. Initially, the input image is processed to extract features at different scales, F_1, F_2, and F_3, which are then optimized through scale-aware and spatial-aware attention mechanisms. Finally, the features are processed through an ROI pooling layer and further optimized by the task-aware attention module, resulting in the output image.

3.4 Focal-CIoU Loss Function

YOLOv8 employs the CIOU loss function to estimate the dissimilarity between the predicted values of the network model and the actual values(Zheng et al. 2020). The CIOU loss function not only considers the overlap area between the predicted and the actual bounding boxes but also takes into account the distance between their centers and the similarity in aspect ratio. This provides a more comprehensive and precise evaluation method for object box regression. The computational formulae are presented in Equations16 , 17, 18, 19, and 20:

$$IOU = \frac{A \cap B}{A \cup B} \tag{16}$$

$$R_{cIOU} = \frac{\rho^2(b, b^{\text{gt}})}{c^2} + \alpha v \tag{17}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{18}$$

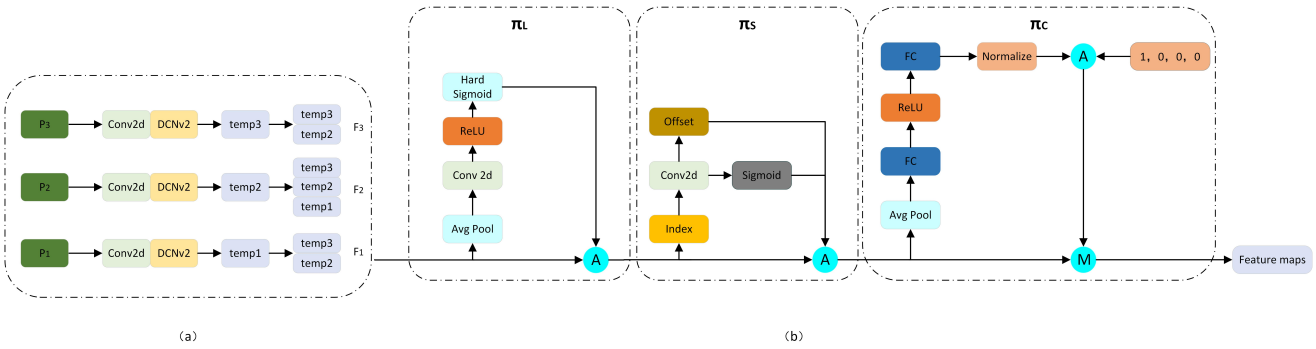

Fig. 7 DyHead detection framework structure diagram.

$$\alpha = \frac{v}{(1 - IOU) + v} \tag{19}$$

$$L_{CIOU} = 1 - loU + \frac{\rho^2(b, b^{\mathrm{gt}})}{c^2} + \alpha v \tag{20}$$

In Equation16, A and B represent the areas of the predicted and the actual bounding boxes, respectively, and are crucial parameters for calculating the IoU. IoU is a commonly used metric in object detection tasks to measure the overlap between the predicted and actual boxes. In Equation17, bdenotes the center coordinates of the predicted box, while b^{gt} represents those of the actual box W and w^{gt} are the widths of the predicted and actual boxes, respectively, h and h^{gt} are their heights. $\rho(b, b^{gt})$ calculates the Euclidean distance between the center points b of the predicted box and b^{gt} of the actual box. c denotes the diagonal distance of the smallest enclosing area containing both the predicted and actual boxes, a parameter in the CIOU loss used to measure the distance between centers. The parameter v quantifies the consistency of the aspect ratio, indicating the shape similarity between the predicted and actual boxes. α is a weighting parameter used to adjust the impact of v on the final loss value. Based on Equations 18 and 19, v and α are derived and jointly contribute to Equation20 to calculate the final loss function.

However, the process of predicting target bounding box regression often faces the issue of imbalanced training samples. Specifically, in an image, the number of high-quality anchor boxes with small regression errors is typically far less than that of low-quality anchor boxes with large errors. These low-quality anchor boxes, due to their significant regression errors, generate excessive gradients, adversely affecting the training process. Direct use of CIoU Loss often does not yield optimal results. To address this issue, this paper introduces Focal Loss, a loss function specifically designed to tackle sample imbalance problems. By combining CIoU Loss with Focal Loss(Lin et al. 2017), Focal-CIoU Loss is proposed. Focal-CIoU Loss approaches the issue from a gradient perspective, adjusting the form of the loss function to impose varying degrees of penalty on high and low-quality anchor boxes. Thus, even with a higher number of low-quality anchor boxes, their impact on the overall training process is effectively mitigated, while high-quality anchor boxes receive more attention and optimization. In this way, Focal-CIoU Loss better balances the impact of anchor boxes of varying qualities, thereby enhancing the performance of object detection. The formula is presented in Equation21:

$$L_{\mathrm{Focal-EIOU}} = IOU^{\gamma} L_{\mathrm{EIOL}} \tag{21}$$

In the formula, γ represents the weighting parameter, and it has been experimentally demonstrated that a γ value of 0.8 yields the best results.

4 Experimental Results and Analysis

4.1 Experimental Setup

The experimental setup for this study is as follows: Windows 10 operating system, CPU is AMD EPYC 7543 with 32 cores, turbo frequency is 3.7GHz, GPU is NVIDIA RTX 3090 (24GB), CUDA 11.3, training conducted in PyTorch 1.11 and Python 3.8 environment. Model training parameters include a batch size of 64 for each

training iteration, 200 epochs, 16 worker threads, and a learning rate of 0.001.

4.2 Comparative Experiments

To evaluate the performance of the YOLO-DBD model, the study selected 10 commonly used deep learning algorithms for multi-class harmful leaves detection experiments and compared their performance with that of YOLO-DBD. From the comparison results in 1, it can be seen that the YOLOv8-DBD network has achieved a significant improvement in mAP compared to the YOLOv8s network, indicating more accurate recognition of different classes of harmful leaves.Simultaneously, the FLOPs of YOLO-DBD, indicating the model's computational complexity, were also reduced, signifying increased efficiency. However, the detection speed of YOLO-DBD is lower, which may be due to a more complex model structure or the use of a more refined feature extraction mechanism. Additionally, there was a slight increase in the number of parameters of the model.

Compared to Faster-RCNN and SSD, the mAP of YOLO-DBD increased by 12.4% and 14.1%, respectively. The AP and FPS for IHL, CHL, and OHL showed significant improvements, while the number of parameters and FLOPs notably decreased. Compared with YOLOv4, the mAP of YOLO increased by 2.9

In comparison with YOLOv5s and YOLOv5l, the mAP of YOLO-DBD increased by 11.1% and 3.3%, respectively. The AP for IHL, CHL, and OHL significantly increased, but the detection speed was noticeably slower than these two models. Compared with YOLOv7-Tiny and YOLOv7, the mAP of YOLO-DBD increased by 10.3% and 4.5%, respectively. The AP for IHL, CHL, and OHL significantly increased, but FPS decreased. Against YOLOv8m, YOLOv8l, and YOLOv8x, the mAP of YOLO-DBD increased by 3.3%, 1.7%, and 2.8%, respectively. the AP of CHL significantly increased, the AP of OHL saw a slight rise, and compared to YOLOv8l and YOLOv8x, IHL decreased. The number of parameters and computational complexity significantly reduced, with an increase in FPS.

The YOLO-DBD network demonstrates significant performance advantages in the task of detecting multiple classes of harmful leaves, especially when compared with several popular deep learning models. Specifically, compared to Faster-RCNN and SSD, YOLO-DBD achieved significant improvements of 12.4% and 14.1% in mAP, respectively, along with increases in FPS, and notable reductions in parameters and FLOPs. In the comparison with YOLOv5s and YOLOv5l, YOLO-DBD showed mAP increases of 11.1% and 3.3%, respectively, but with slower detection speeds than these two models. When compared with YOLOv7-Tiny and YOLOv7, the mAP of YOLO-DBD improvements were 10.3% and 4.5%, respectively, while FPS decreased. Furthermore, compared to YOLOv8m, YOLOv8l, and YOLOv8x, YOLO-DBD exhibited mAP improvements of 3.3%, 1.7%, and 2.8%, respectively, and also showed advantages in computational complexity and FPS. This indicates that YOLO-DBD, while enhancing detection accuracy, effectively optimizes the model's efficiency and computational burden. Although there is a decrease in detection speed in certain scenarios, overall, it demonstrates significant competitiveness among many deep learning models.

Fig8 provides a visual comparison, illustrating the performance of YOLO-DBD alongside ten other deep learning models in detecting multiple classes of harmful leaves targets and cropping them accordingly. The results demonstrate that YOLO-DBD excelled in this task, accurately detecting all target, while other models exhibited varying degrees of detection errors. Specifically, models like SSD, YOLOv5s, YOLOv7, and YOLOv8m misclassified non-harmful leaves as harmful. Moreover, models such as Faster-RCNN, SDD, YOLOv5s, YOLOv5l, YOLOv7-Tiny, YOLOv7, YOLOv8s, YOLOv8m, and YOLOv8x failed to detect the target at least once. Additionally, models including Faster-RCNN, SDD, YOLOv5l, YOLOv7-Tiny, YOLOv8s, and YOLOv8m also made at least one target category detection error. These results fully reflect the advantages of YOLO-DBD in handling complex scene target detection tasks, especially in applications requiring accurate identification of different classes of targets. The high accuracy and robustness of YOLO-DBD make it a deep learning model with significant advantages in practical agricultural applications.

Table 1 The results of different deep learning models for detecting multi-class harmful leaves.

Model	Parmeter (M)	FLOPS(G)	P(%)	R(%)	AP(%)			mAP(%)	FPS
					IHL	CHL	OHL		
Faster-RCNN	43.1	60.5	75.6	72.8	76.2	77.8	65.5	74.4	11.3
SDD	26	-	71.5	74.7	76.5	73.7	68.3	72.7	15.4
YOLOv5s	6.8	16.0	74.2	73.8	77.1	78.2	66.2	75.7	58.3
YOLOv51	49.0	107.7	80.1	78.9	84.9	84.6	80.9	83.5	41.6
YOLOv7-Tiny	13.2	13.1	73.6	75.7	78.4	77.5	71.5	76.3	81.2
YOLOv7	36.5	105.2	84.6	78.7	83.4	83.5	77.5	82.3	60.4
YOLOv8s	10.6	28.4	82.0	73.3	82.5	83.0	77.2	80.8	64.5
YOLOv8m	24.6	78.7	81.3	77.1	84.68	87.1	78.7	83.5	30.8
YOLOv81	41.5	164.8	83.9	78.5	87.7	84.3	83.2	85.1	19.6
YOLOv8x	64.9	257.4	85.9	72.2	86.6	83.9	81.6	84.0	15.9
YOLO-DBD	11.6	25.1	80.8	83.2	85.9	91.5	83.2	86.8	36.2

Fig. 8 To detect harmful leaves using various deep learning models: (a) manually annotated image; (b) Faster-RCNN, (c) SSD; (d) YOLOv5s; (e) YOLOv5l; (f) YOLOv7-Tiny; (g) YOLOv7; (h) YOLOv8s; (i) YOLOv8m; (j) YOLOv8l; (k) YOLOv8x; (l) YOLOv8-DBD. Yellow boxes indicate undetected harmful leaves, blue boxes represent misclassified categories, and purple boxes indicate detection errors in the bounding box.

4.3 Ablation Study

To verify the effectiveness of the improvement modules proposed in this paper, a comparative analysis of the network performance with different improvement modules was conducted. The performance test results of different networks are shown in Table 2. As can be seen from Table 2, different improvement methods have certain impacts on network performance. An analysis of the performance after various network structural improvements was conducted, demonstrating the effects of different network structural improvements on the model's ability to detect harmful leaves.

By conducting experimental comparisons, improvement 1 added DCNv2 to the basic C2f model, resulting in a 1.4% increase in mAP compared to the initial model. This enhancement more effectively captures the shape and positional characteristics of targets, better adapting the network's perception capabilities in complex environments. Improvement 2 involved modifications to the detection head, replacing the original with Dyhead, which further enhanced the perception of spatial location, scale, and task regions in the head. This not only increased the mAP by 2.1% but also reduced FLOPs by 10.4%, making it more suitable for deployment on embedded mobile terminals. Users can choose a less expensive processor with minimal sacrifice in detection performance. Improvement 3 introduced a BRA mechanism at the end of the backbone network, slightly increasing computational demands but enhancing feature extraction capabilities, with a 0.9 increase in mAP and a 13.1% increase in detection speed over the original. Improvement 4 incorporated Focal Loss on top of CIOU, significantly addressing the issue of sample imbalance and resulting in a 1.6% increase in mAP.

The confusion matrices for YOLOv8-DBD and YOLOv8-s are shown in 9. Here, True represents the true distribution of categories, and Predict represents the predicted distribution of categories. The confusion matrix reveals that YOLOv8-DBD achieves a higher true positive rate for each category compared to YOLOv8s, indicating that YOLOv8-DBD performs better in terms of both precision and recall.

5 Conclusions

This paper proposes a method for detecting multiple classes of harmful leaves blades in target harvesting based on the YOLO-DBD network model. The YOLO-DBD network comprises C2f-DCN, BRA, SPP, PAN-FPN, and Dyhead. C2f-DCN effectively captures target features and enhances detail perception. BRA enhance feature extraction and reduces computational overhead. Additionally, Dyhead enhances spatial, scale, and task area perception, thereby improving detection accuracy. Finally, CIOU-Focal Loss effectively addresses sample imbalance issues, enhancing the model's accuracy and stability.

Experiments conducted in the complex and dense tea plantation environment indicate that the YOLO-DBD achieves an mAP value of 86.8% for the detection of multiple categories of harmful leaves blades. The mAP values for single harmful leaves blades, clustered harmful leaves blades, and occluded harmful leaves blades are 85.5%, 91.5%, and 83.2%, respectively. Compared to ten other commonly used deep learning networks, including Faster-RCNN, the YOLO-DBD model attains the highest mAP in detecting tea leaves while maintaining commendable detection speed.It can offer crucial guidance to the robotic arm during pruning processes related to posture and obstacle avoidance.

References

Ahmed S, Griffin T, Cash SB, et al (2018) Global climate change, ecological stress, and tea production. Stress physiology of tea in the face of climate change pp 1–23. Doi:10.1007/978-981-13-2140-5_1

Arad B, Balendonck J, Barth R, et al (2020) Development of a sweet pepper harvesting robot. Journal of Field Robotics 37(6):1027–1039. Doi:10.1002/rob.21937

Dai G, Fan J (2022) An industrial-grade solution for crop disease image detection tasks. Frontiers in Plant Science 13:921057. Doi:10.3389/fpls.2022.921057

Dai J, Qi H, Xiong Y, et al (2017) Deformable convolutional networks. In: Proceedings of the

Table 2 Results of ablation experiments.

C2f-DCN	Dyhead	BRA	Focal-CIOU	mAP	FLOPs(G)	FPS
				80.8	28.4	64.5
✓				82.2	28.0	64.1
✓	✓			84.3	25.1	35.8
✓	✓	✓		85.2	25.5	40.5
✓	✓	✓	✓	86.8	25.1	36.2

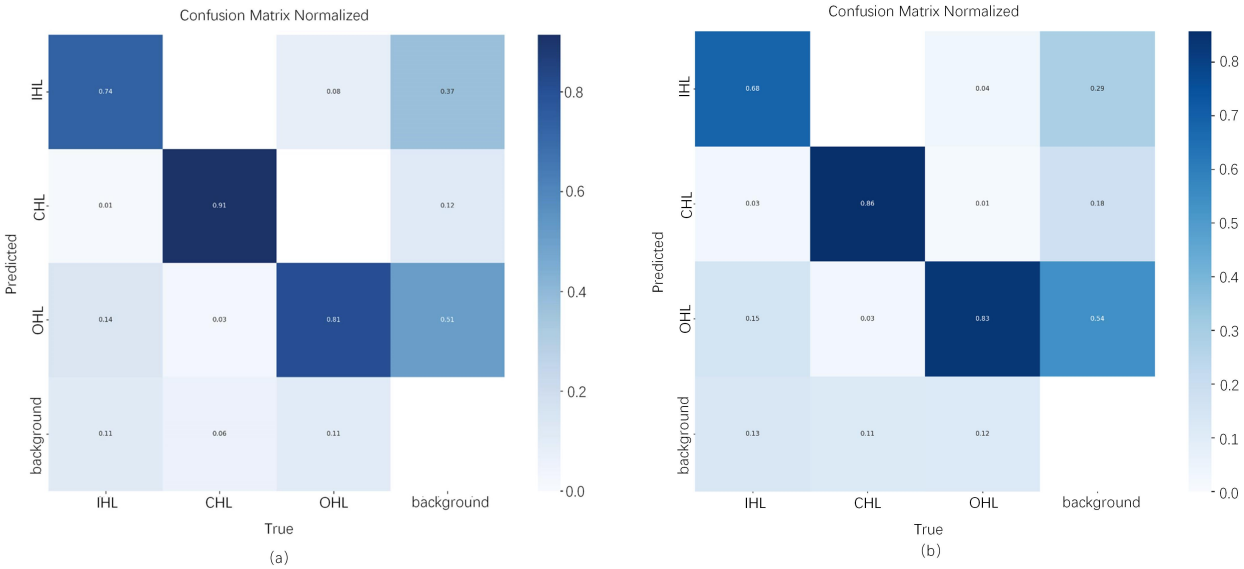

Fig. 9 (a) Confusion matrix of Yolov8-DBD; (b) Confusion matrix of Yolov8s.

IEEE international conference on computer vision, pp 764–773

Dai X, Chen Y, Xiao B, et al (2021) Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7373–7382

Damalas CA, Eleftherohorinos IG (2011) Pesticide exposure, safety issues, and risk assessment indicators. International journal of environmental research and public health 8(5):1402–1419. Doi:10.3390/ijerph8051402

Du X, Meng Z, Ma Z, et al (2023) Tomato 3d pose detection algorithm based on keypoint detection and point cloud processing. Computers and Electronics in Agriculture 212:108056. Doi:10.1016/j.compag.2023.108056

Fu L, Majeed Y, Zhang X, et al (2020) Faster r–cnn–based apple detection in dense-foliage fruiting-wall trees using rgb and depth features for robotic harvesting. Biosystems Engineering 197:245–256. Doi:10.1016/j.biosystemseng.2020.07.007

Lin J, Bai D, Xu R, et al (2023) Tsba-yolo: An improved tea diseases detection model based on attention mechanisms and feature fusion. forests 14 (3), 619. Doi:10.3390/f14030619

Lin TY, Goyal P, Girshick R, et al (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988

Liu W, Anguelov D, Erhan D, et al (2016) Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14,

2016, Proceedings, Part I 14, Springer, pp 21–37, doi:10.1007/978-3-319-46448-0_2

Nan Y, Zhang H, Zeng Y, et al (2023) Intelligent detection of multi-class pitaya fruits in target picking row based on wgb-yolo network. Computers and Electronics in Agriculture 208:107780. Doi:10.1016/j.compag.2023.107780

Redmon J, Divvala S, Girshick R, et al (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788

Roy AM, Bose R, Bhaduri J (2022) A fast accurate fine-grain object detection model based on yolov4 deep neural network. Neural Computing and Applications pp 1–27. Doi:10.1007/s00521-021-06651-x

Sun C, Huang C, Zhang H, et al (2022) Individual tree crown segmentation and crown width extraction from a heightmap derived from aerial laser scanning data using a deep learning framework. Frontiers in plant science 13:914974. Doi:10.3389/fpls.2022.914974

Toscano P, Iannotta N, Scalercio S (2015) Botanical and agricultural aspects: agronomic techniques and orchard management. Agricultural and Food Biotechnologies of Olea europaea and Stone Fruits Bentham pp 3–73

Wang R, Shivanna R, Cheng D, et al (2021) Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In: Proceedings of the web conference 2021, pp 1785–1797, doi:10.1145/3442381.3450078

Wang S, Wu D, Zheng X (2023) Tbc-yolov7: a refined yolov7-based algorithm for tea bud grading detection. Frontiers in Plant Science 14:1223410. Doi:10.3389/fpls.2023.122341

Wu Z, Chen Y, Zhao B, et al (2021) Review of weed detection methods based on computer vision. Sensors 21(11):3647. Doi:10.3390/s21113647

Xue Z, Xu R, Bai D, et al (2023) Yolo-tea: A tea disease detection model improved by yolov5. Forests 14(2):415. Doi:10.3390/f14020415

Zheng Z, Wang P, Liu W, et al (2020) Distance-iou loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence, pp 12993–13000, doi:10.1609/aaai.v34i07.6999

Zhou G, Zhang W, Chen A, et al (2019) Rapid detection of rice disease based on fcm-km and faster r-cnn fusion. IEEE access 7:143190–143206. Doi:10.1109/ACCESS.2019.2943454

Zhu L, Wang X, Ke Z, et al (2023) Biformer: Vision transformer with bi-level routing attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 10323–10333