electronics

Article

A Lightweight and Dynamic Feature Aggregation Method for Cotton Field Weed Detection Based on Enhanced YOLOv8

Doudou Ren [1], Wenzhong Yang [1,2,*], Zhifeng Lu [3,*], Danny Chen [1], Wenxuan Su [1] and Yihang Li [1]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830017, China; 107552201396@stu.xju.edu.cn (D.R.); kabuoxygen@stu.xju.edu.cn (D.C.); 107552304135@stu.xju.edu.cn (W.S.); 107552201392@stu.xju.edu.cn (Y.L.)
[2] Xinjiang Key Laboratory of Multilingual Information Technology, Xinjiang University, Urumqi 830017, China
[3] School of Information Science and Technology, Xinjiang Teacher's College, Urumqi 830043, China
* Correspondence: yangwenzhong@xju.edu.cn (W.Y.); xjdxsylb@xju.edu.cn (Z.L.)

Abstract: Weed detection is closely related to agricultural production, but often faces the problems of leaf shading and limited computational resources. Therefore, this study proposes an improved weed detection algorithm based on YOLOv8. Firstly, the Dilated Feature Integration Block is designed to improve the feature extraction in the backbone network by introducing large kernel convolution and multi-scale dilation convolution, which utilizes information from different scales and levels. Secondly, to solve the problem of a large number of parameters in the feature fusion process of the Path Aggregation Feature Pyramid Network, a new feature fusion architecture multi-scale feature interaction network is designed, which achieves the high-level semantic information to guide the low-level semantic information through the attention mechanism. Finally, we propose a Dynamic Feature Aggregation Head to solve the problem that the YOLOv8 detection head cannot dynamically focus on important features. Comprehensive experiments on two publicly accessible datasets show that the proposed model outperforms the benchmark model, with mAP_{50} and mAP_{75} improving by 4.7% and 5.0%, and 5.3% and 3.3%, respectively, whereas the number of model parameters is only 6.62 M. This study illustrates the utility potential of the algorithm for weed detection in cotton fields, marking a significant advancement of artificial intelligence in agriculture.

Keywords: weed detection; lightweight; YOLOv8; attention

check for updates

Citation: Ren, D.; Yang, W.; Lu, Z.; Chen, D.; Su, W.; Li, Y. A Lightweight and Dynamic Feature Aggregation Method for Cotton Field Weed Detection Based on Enhanced YOLOv8. *Electronics* **2024**, *13*, 2105. https://doi.org/10.3390/electronics13112105

Academic Editor: José Carlos Castillo

Received: 29 April 2024
Revised: 20 May 2024
Accepted: 27 May 2024
Published: 29 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Cotton, a popular natural fiber, has a wide range of applications in textiles and other fields. As urbanization accelerates, the area of land available for cultivation is becoming less and less, which highlights the importance of smart agriculture in the contemporary context [1]. Precision agriculture [2] is the integration of information technology with agricultural practices to achieve precise management of crop production and provide decision-making assistance [3]. Weed detection in cotton fields is a key application in the field of precision agriculture, which is of great significance and impact on cotton production.

Weeds absorb nutrients from the crop, resulting in cotton not receiving sufficient nutrients, which reduces crop yield [4]. Currently, common weed control methods used in agricultural management include chemical and mechanical weed control. Although chemical weed control is efficient and widely applied, its residual chemicals may cause damage to the crop itself and affect the soil environment. Mechanical weeding is more costly to maintain. In addition, there is a risk of weed regrowth after mechanical weeding, thus requiring multiple rounds of weeding [5]. To address these challenges, deep learning methods can be used to accurately identify weeds in the field and carry out targeted weeding operations with agricultural intelligent robots [6].

In the field of weed detection, researchers face a variety of difficulties and challenges. For the challenge of target scale diversity, this scale variation means that detection models

designed only for a single scale cannot effectively identify weeds at a variety of scales, so effective models that can adapt to multiple scale variations need to be developed. Secondly, the similarity between weeds and crops is also an important challenge. Their leaf shapes and colors may be very similar at different growth stages, which can affect the effectiveness of the detection model and lead to false or missed detections. Finally, dense occlusion during plant growth is another common challenge in object detection, which may obscure the edges of the target and reduce the accuracy and robustness of the weed detection process.

In order to solve the above problems, researchers have used various advanced techniques, including deep learning incorporating color index features [7], semi-supervised learning [8], multimodal information fusion [9], data augmentation [10], and attention mechanism [11]. In this study, we propose a lightweight dynamic feature aggregation method (EY8-LDFA) based on YOLOV8 improved for weed detection in cotton fields. Firstly, inspired by the idea of the large convolutional kernel [12], we designed a Dilated Feature Integration Block (DFIB). DFIB integrates feature information at different scales and levels through multipath convolutional operations and channel attention mechanism, and achieves the effective fusion and expression of global features and local features, to better handle features in complex contexts. To further enhance the feature extraction capability of the backbone feature network, we designed C2F-DFIB, which enhances the model's ability to understand global and local information through sophisticated multi-scale feature extraction and deep feature fusion techniques. After a careful study of the Path Aggregation Feature Pyramid Network (PA-FPN) in YOLOv8, we found that the fusion process of features at different scales involves multiple feature map upsampling, downsampling, and splicing operations, which increases the computational complexity and the number of parameters of the model. To reduce the number of parameters and computation of the feature fusion network, we propose a lightweight feature fusion architecture called a multi-scale feature interaction network (MFIN). It first enhances and fuses different scale features through the Attentive Channel Integration Block (ACIB), uses channel attention mechanism and jump connection techniques to achieve the guidance of high-level semantic information to low-level semantic information, and at the same time, uses transposed convolution to dynamically adjust the upsampling method according to the content of the input feature maps to improve the performance of the target detection task. Finally, the detection head of YOLOv8 lacks a dynamic perception mechanism to dynamically focus on important features, which reduces the perception capability of the target detection head. Inspired by [13], we propose Dynamic Feature Aggregation Head (DFAH), which utilizes techniques such as DCNv3 to achieve adaptive sampling and dynamic weighting of feature maps. By learning spatial offsets and feature channel weights, we capture distance-dependent information and feature alignment, which enhances the perception of the detection head. The key contributions of this study are as follows:

- We have designed a model for weed detection in cotton fields, called EY8-LDFA, which is lightweight while ensuring high accuracy and is particularly suitable for resource-constrained environments.
- To efficiently fuse the information of different levels and scales in the image, we designed DFIB (Dilated Feature Integration Block). Meanwhile, to enhance the feature extraction capability of the backbone network, we designed C2f-DFIB, which is a module that achieves the combined use of global and local information of image features by fusing the multi-scale null convolution and large-scale convolution.
- To reduce the computational effort as well as the complexity of the feature fusion network, we design MFIN (multi-scale feature interaction network), which reduces the number of parameters and improves the detection performance by augmenting and fusing multi-scale features and using high-level semantic information to guide the low-level semantic information.
- To allow the detection head to dynamically focus on important features, we designed the DFAH (Dynamic Feature Aggregation Head), which is a module that achieves adaptive sampling and dynamic weighting of the feature map by utilizing the DCNv3 (de-

formable convolution net v3). By learning spatial offsets and feature channel weights, it can capture long-range dependent information and feature alignment to efficiently capture feature information at a variety of scales and location feature information.

2. Related Work

2.1. Weed Detection

With the rapid development of artificial intelligence technology, a large number of algorithms based on traditional machine learning and deep learning have emerged in the field of weed recognition. Weed detection using traditional machine vision techniques usually relies on methods such as image filtering, and extraction of color and spectral features, which involves the use of thresholding and classification algorithms to identify and predict the location of weeds. Deep learning methods for weed detection, on the other hand, mainly focus on segmentation, classification, and identification of weed locations. Nicolas et al. [14] combined the weed detection task with zero-sample learning to enhance the adaptability of the weed detection model to climate change by constructing a semantic space and projection function, etc. Syamasudha et al. [15] first segmented the image using PSPNet, then used an image enhancement framework to generate the training samples and used a Coot political optimization algorithm trained deep quantum neural network to classify weeds and calculate weed density to assess the number of plants per unit area in the farmland. However, since the algorithm does not consider soil properties, it has limitations in assessing the distribution of weed species and their densities in different soil areas. Ju et al. [16] proposed an algorithm called MW-YOLOv5s for rice field weed identification, which combines YOLOv5's backbone and MobileViTv3 network for enhanced backbone feature extraction. Although this approach improves the detection accuracy, it increases the computational complexity due to the introduction of an additional feature extraction network, which leads to an increase in the network layers and parameters. The feature fusion structure in EY8-LDFA, on the other hand, does not have a lot of upsampling, downsampling, and splicing operations, and achieves lightweight and efficient networks by combining attention with high-level semantic information to guide low-level semantic information. Zhu et al. [17] proposed an advanced YOLOx maize weed detection model, which introduces an inverse convolutional layer in the residual module to improve the ability to handle small-sized features. Han et al. [18] proposed a lightweight crop detection model for detecting maize seedlings in complex field environments and combined with the distribution characteristics of weeds, the ExG index algorithm and the optimized Otsu method were used to quickly obtain the weed information. Liao et al. [19] proposed a banded convolutional network model called SC-Net, and the banded multiscale convolution effectively extends the convolutional layer's receptive field, and its attention-based feature fusion method effectively aggregates low- and high-level features to identify rice and weeds efficiently.

2.2. YOLO Algorithm

The YOLO family of algorithms has been widely adopted in a variety of industrial application scenarios, such as weed detection tasks, due to its superior performance, lightweight design, and real-time processing capabilities. The core idea of the YOLO algorithms is to perform a series of pre-processing on the input raw image, such as cropping and data augmentation, and then perform multiple downsampling through the backbone network to generate feature maps at three different scales. These feature maps are then subjected to multi-scale feature fusion in the Path Aggregation Feature Pyramid Network (PAFPN) of the neck network to generate the feature maps used to detect the head for bounding box regression and target classification. The output information includes the confidence level, coordinates, and category labels for each bounding box. YOLOv1 [20], as a representative single-stage detection algorithm, omits the candidate region generation step and performs target classification and bounding box localization directly on the image, which dramatically improves the detection efficiency despite the compromise in accuracy. YOLOv2 [21] dramatically increases the types of objects that can be recognized by the model

through an innovative training method. YOLOv3 [22] introduces a multi-scale prediction mechanism, which enables the model to perform simultaneous regression prediction for feature maps of different sizes. YOLOv4 [23] and YOLOv5 [24] were introduced successively, which significantly improved the detection effect by introducing techniques such as the CSP network module and adaptive anchor frame computation. Li et al. [25] introduced YOLOv6, which achieves the target detection effect by introducing an efficient decoupling header and a parameterizable backbone network, which achieves a further improvement in target detection performance. Alexey et al. [26] proposed YOLOv7, a model that uses novel data enhancement techniques and fuses the Neck and Head layers into a single Head layer, which significantly improves the accuracy of target detection. YOLOv8 [27] introduces a new C2f structure and a classification and detection Head separation, and these improvements significantly enhanced the accuracy of the model in target detection. YOLOv9 [28] used reversible function-based CNN architecture analysis, designed PGI and auxiliary reversible branches to improve the parameter usage of the model, and maintained a faster inference speed by optimizing the model structure and computation to meet the demand of real-time target detection.

2.3. Attention

Originally inspired by the human visual system, the attention mechanism aims to enable models to pay more attention to key regions in the feature map by assigning different weights to them. Different tasks require different attention mechanisms, and how to reduce the computational complexity of attention mechanisms has been a hot research topic. To solve the problem of slow inference of visual transformer (ViT) when capturing both global and contextual information, You et al. [29] proposed a new linear angle attention mechanism, which decomposes the angle kernel into linear terms and higher-order residuals and retains only the linear terms, thus avoiding extra computational overheads and reducing computational costs during inference. To reduce the computational burden when capturing long-distance dependencies, Zhu et al. [30] proposed dynamic sparse attention via two-layer routing, a mechanism that allocates attention in a content-aware manner and exploits sparsity to reduce computational and memory consumption while maintaining good performance. To address the computational redundancy caused by high head similarity in feature maps, Liu et al. [31] proposed a cascaded group attention module that provides different segmentations of the complete feature for the attention head as a way to save computational cost. Hassani et al. [32] proposed a visual sliding-window attention mechanism, which operates pixel-by-pixel to make the self-attention focus on the nearest neighboring pixels, thus achieving localized attention to the feature map. This approach effectively captures the interrelationships and information between neighboring pixels and significantly reduces the computational time complexity and space complexity. Since existing localized attention mechanisms rely on inefficient Im2Col functions or specific CUDA kernels, Pan et al. [33] proposed Slide Attention, which uses deep convolution and reparameterization techniques to create a new localized attention paradigm. To allow 3D point cloud Transformer models to run on resource-constrained devices, Liu et al. [34] proposed FlatFormer, which extracts local features by grouping them and applying self-attention within the group, and exchanging features across the group by moving the window, thus significantly reducing latency while improving accuracy. Since existing deep unfolding networks improve performance by increasing the number of parameters, Song et al. [35] proposed a cross-attention algorithm that designs a novel dual cross-attention sub-module, where the inertia-providing cross-attention block introduces multichannel inertial forces, and the projection-guided cross-attention block implements information augmentation, thereby significantly reducing the training complexity while improving performance. Woo et al. [36] proposed the CBAM, which adaptively adapts and enhances intermediate layer features by imposing an attention mechanism of channels and spatial dimensions on the feature map. CBAM can achieve good results with a slight increase in overhead and is therefore used by MFIN to enhance the feature representation.

3. Proposed Method

In Section 3, we first present the overall architecture of the model in Section 3.1, followed by the Dilated Feature Integration Block and C2f-DFIB module in Section 3.2, then the multi-scale feature interaction network in Section 3.3, and finally the Dynamic Feature Aggregation Head in Section 3.4.

3.1. Model

As one of the current high-profile target detection models, YOLOv8 has been widely used in many scenarios that require real-time response and lightweight deployment, including weed detection. Due to the small size of the YOLOv8 model, it is well suited for use in resource-limited devices such as micro drones and farmland inspection robots. Although YOLOv8 performs well on many datasets, there is still a large potential to improve its performance in weed detection tasks. Considering these factors, we chose YOLOv8 as our baseline model and developed a novel network model called EY8-LDFA for a specific cotton field weed detection scenario to cope with the demand for lightweight, real-time processing, and high efficiency of the model.

As shown in Figure 1, EY8-LDFA consists of three core components: Backbone, Neck, and Head, which, for an original input image, Backbone downsamples it several times to extract rich high-level semantic features from the image and obtains three different scales of feature maps. Then, these three feature maps enter into the Neck network for multi-scale feature fusion. By combining different scales of feature maps, the model can better understand the local and global structure of the image and learn a richer feature representation. The Head is responsible for processing the feature map coming out of Neck, which contains three different scales of features. By combining different scales of feature maps, the model can better understand the local and global structure of the image and learn richer feature representations. The head is responsible for processing the feature maps from Neck, which contains three different detection head paths, each corresponding to a scale of feature maps, and finally performs bounding-box regression and target classification on the targets and outputs coordinate information and confidence of the detected targets, i.e., coordinate information and confidence level.Firstly, to improve the feature extraction capability of Backbone network to cope with the problems of weed and crop similarity and dense occlusion in the detection process, we propose the C2f-DFIB module, which extracts the feature maps in a multi-scale and omnidirectional way by using multiple large kernel convolutions of different sizes and cavity convolutions with different sizes, and this combination not only covers a wider sensory field range, but also better captures the details and local structures in the image, and the combined use of global and local information can well improve the expressive power of the features. Since the third and fourth C2f modules of Backbone are responsible for extracting deep semantic information, we replace them with C2f-DFIB to enhance the network's ability to capture high-level semantic features of the target object. By optimizing these modules, we can improve the network's ability to understand and model complex scenarios, which in turn improves the model's accuracy and robustness in detecting tiny targets such as weeds in cotton fields. Second, the Neck part of YOLOv8 uses a Path Aggregation Feature Pyramid Network; however, there are many obvious problems with this feature fusion approach. In the Path Aggregation Feature Pyramid Network, the top-down feature fusion network requires cascade connections and lateral connections, which leads to a larger computational volume of the network, especially when dealing with high-resolution images, which increases the computational complexity and the number of parameters. Moreover, when dealing with images with occlusion and complex scenes, the Path Aggregation Feature Pyramid Network may result in incomplete feature fusion due to insufficiently direct information transfer, which affects the model's accurate detection of targets. Therefore, we propose a simpler and lighter model structure, MFIN, which realizes the effective use and combination of high-level and low-level features through the channel attention mechanism of the ACIB module and the feature fusion of the AAFI(Attention-augmented feature integration), thus reducing redundant information

and improving the efficiency and performance of feature representation, which in turn reduces the number of parameters and the computational effort of the model. Finally, since the detection head of YOLov8 does not have a dynamic sensing mechanism and cannot dynamically focus on important features, the weed detection head therein cannot suppress background noise and cannot efficiently capture feature information at various scales and locations, which reduces the characterization ability of the target detection head and leads to poor weed detection. Therefore, we propose DFAH, which utilizes techniques such as DCNv3 to achieve adaptive sampling and dynamic weighting of feature maps, capturing long-range dependent information and feature alignment by learning spatial offsets and feature channel weights to enhance the weed detection performance of the detection head. Three detection heads detect feature maps of different scales, and the same target in different feature maps will generate multiple detection boxes. Finally, for each target, the Non-Maximum Suppression algorithm (NMS) can eliminate the detection boxes with lower confidence to ensure that each target ultimately has only one detection box.

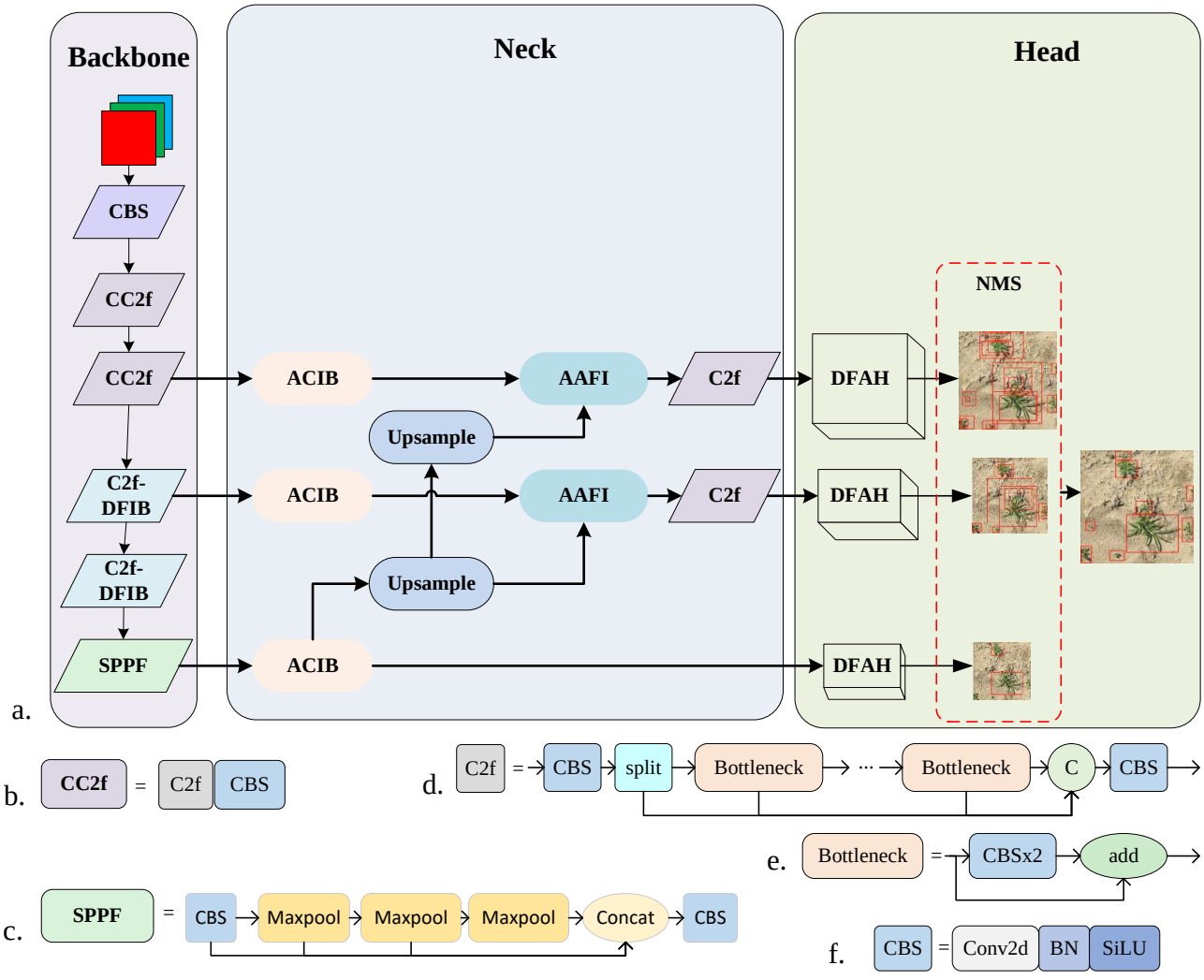


Figure 1. Schematic of the EY8-LDFA model. Among them, (**a**) is the overall structure diagram of the model, (**b**) is the CC2f module in Backbone, (**c**) is the SPPF structure, (**d**) is the C2f structure, (**e**) is the Bottleneck structure in the C2f module, and (**f**) is the CBS structure diagram in the model.

3.2. Dilated Feature Integration Block and C2f-DFIB

In object detection tasks, features at different scales are crucial for detecting targets of different sizes and shapes. DFIB achieves multi-scale feature extraction and fusion by

combining a large-scale convolution kernel and cavity convolution at different scales. With the large-scale convolution kernel and cavity convolution, the module can capture the global information of the feature map, whereas the 3×3 convolution kernel is used to extract the local details. This fusion of global and local information helps the model to understand the image content more comprehensively and improve the detection performance. Meanwhile, the null convolution allows the model to reduce the number of parameters and the amount of computation while maintaining the same receptive field, which is especially important for resource-constrained platforms. Therefore, the proposed DFIB and C2f-DFIB modules aim to improve the feature representation, generalization, and computational efficiency of the models through multi-scale feature extraction and fusion.

3.2.1. Dilated Feature Integration Block

Recent studies have shown that large-scale convolutional kernels help to capture global contextual information of an image, whereas different scales of null convolution can capture more detailed local features. The 3×3 convolutional kernels further enhance the extraction of local information. By combining this information at different scales, the model can understand the image content more comprehensively. Null convolution allows the model to reduce the number of parameters and the amount of computation while maintaining the same receptive field. By using different scales of null convolution, the model can learn different levels of feature information, which improves the detection of targets of different sizes. Combining these advantages, we propose the DFIB to enhance the feature representation.

Specifically, for an original input feature map, we first perform a convolution operation to halve the count of output channels and then execute batch normalization and SiLU actuation functions following the convolution operation. This approach not only effectively balances the computational intricacy along with the expressive power of the model, but also improves the nonlinearity and feature extraction capabilities of the network. Next, we start from three paths and use different convolution kernel sizes and varied dilation factors to extract details across various levels and scales in the image. Due to the influence of large kernel convolution, we suddenly came up with the idea of using two large kernel convolutions to obtain a larger receptive field. A larger convolution kernel and expansion rate can cover a larger receptive field, thus acquiring a wider array of semantic cues, which aids in understanding the overall structure and contextual details in the image. Smaller convolution kernels and expansion rates can better capture local details, thereby enhancing the detailed expression ability of features. Therefore, to obtain finer local features of the feature map, we choose to use a 3×3 ordinary convolution to extract local features within a smaller receptive field range. For the other two paths, we adopt the approach of large convolution kernels to better capture global information and complex features. For the branch path on the left in Figure 2, we used a convolutional kernel with a maximum size of 7×7 to achieve a more profound comprehension of the image's contextual aspects, improve the ability to recognize and locate weeds, and at the same time, we used convolutional kernels with different expansion rates to extract features in different receptive field ranges. This can capture information at different scales, helping the model to have an in-depth comprehension of the semantics and structure of the input data. For the branch path on the right in Figure 2, we used a convolutional kernel with a maximum size of 5×5 to achieve the same goal. Meanwhile, convolution kernels with different expansion rates can introduce varying degrees of nonlinear transformations, thereby enhancing the network's non-linear fitting ability. This helps to enhance the model's capacity for complex data distributions. After a series of large kernel convolutions and dilated convolutions, we use batch normalization to maintain the distribution of input data within a relatively stable range, which helps reduce the problem of gradient vanishing or exploding in deep networks. After batch normalization, the feature information extracted by multiple convolution operations is added element by element, achieving an effective combination of localized and worldwide characteristics. This allows the model to capture

different aspects of the image, such as texture, shape, color, etc., thus obtaining a richer and more comprehensive feature representation. Next, we merge the feature maps derived from the three paths in the channel dimension. Due to the use of the SiLU in YOLOv8, to adapt DFIB to this nonlinear distribution, we performed convolution on the concatenated feature maps in order to halve the channel count and used layer normalization and SiLU activation function for subsequent processing to better handle the distribution of features. Ultimately, the network's representational power and the model's generalizability are bolstered by integrating shallow features directly with deeper features through the use of skip connections. This enables higher-level features to leverage information from lower-level features. The implementation of the DFIB module is shown in Formulas (1)–(6):

$$X_1 = CBS(X) \tag{1}$$

$$Y[i,j,k] = BN\left(\sum_{u,v} X_1[i + r \cdot u, j + r \cdot v, l] \times K[u,v,l,k]\right) \tag{2}$$

$$Y_1 = \sum_{i=1}^{n} Y_i \tag{3}$$

$$Y_2 = \sum_{i=1}^{\mathrm{m}} Y_i \tag{4}$$

$$Y_3 = CBS(X_1) \tag{5}$$

$$Y = Add\big(CBS\big(Concat(Y_1, Y_2, Y_3)\big), X\big) \tag{6}$$

Among them, X is the original input feature map, the $CBS(\cdot)$ function represents 3×3 convolution, batch normalization, SiLU, K represents the convolution kernel, i, j, k, respectively, represent the spatial position of the output feature map; u, v, respectively, represent the spatial position pertaining to the convolution kernel, r serves as porosity, l acts as the index for the input feature map's channels, and Y_i denotes the resultant value after operations with multiple distinct convolution kernels and varying porosity, $BN(\cdot)$ represents batch normalization operation, X_1, Y_1, Y_2, Y_3 represent intermediate features, $Concat(\cdot)$ represents feature concatenation along the channel dimension, and $Add(\cdot)$ represents element by element addition operation.

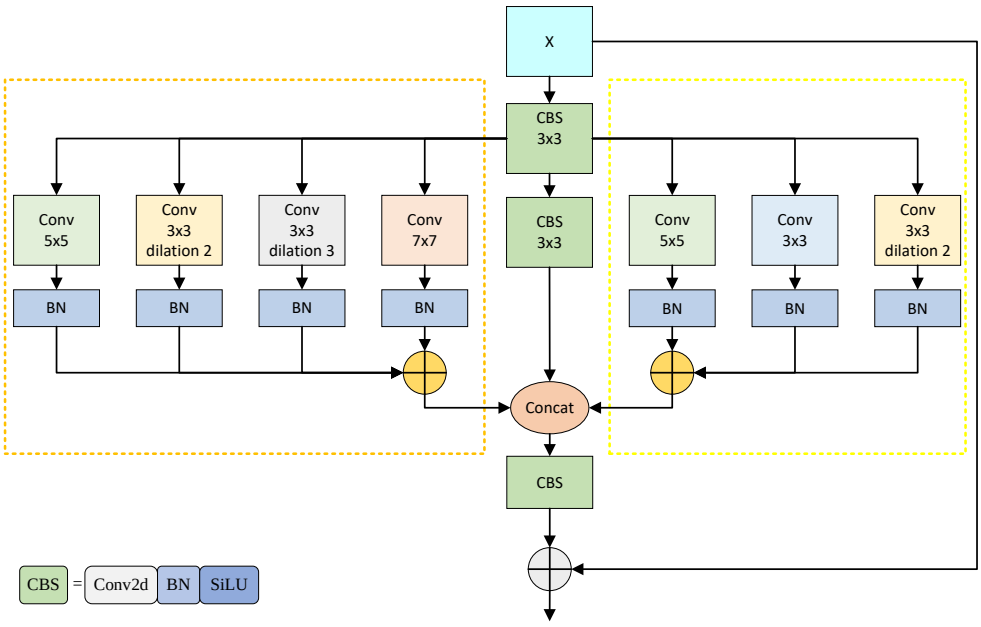


Figure 2. Schematic of the DFIB.

3.2.2. C2f-DFIB

The current research designed a C2f-DFIB module based on the newly developed DFIB module, aiming to substitute the C2f module in the original neural network structure to boost the extraction ability of deep features, as shown in Figure 3. The DFIB module combines multiple large-scale convolution kernels with multiple dilated convolutions for the purpose of extraction of profound features. Large-scale convolution kernels and convolutions with multi-level dilation assist in capturing the global information of feature maps, effectively covering a wider receptive field range. At the same time, 3×3 convolution is used for local information extraction, which can better capture the details and local structure in the image. This helps the model to more accurately identify and locate targets or specific areas in the image. The comprehensive utilization of information from both global and local details enables an enhancement in the expression ability of features, making the model more adaptable and generalizable when dealing with complex scenes. In the C2f-DFIB module, we first perform 3×3 convolutional processing on the original features and then process them through layer normalization and SiLU activation function. Then, we divide the output into two parts along the channel, with one part of the feature set to the DFIB module for multi-scale feature extraction and the other part used for subsequent skip connection operations. Within the paper, this work employs one DFIB module for the process of obtaining advanced features. Next, we concatenate the channel dimensions of the features generated at each layer and finally derive the desired features through the convolution module. Due to the introduction of multi-scale feature extraction in the C2f-DFIB module, further integration of local and global information is achieved through the operation of using C2f-DFIB instead of the bottleneck within the C2f module.

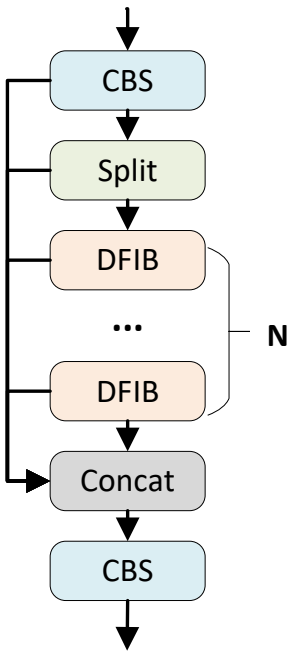


Figure 3. Schematic of the C2f-DFIB.

3.3. Multi-Scale Feature Interaction Network

During the procedure of integrating features across scales, Path Aggregation Feature Pyramid Network usually requires multiple upsampling, downsampling, and concatenation operations of feature maps, along with weighting and aggregation of features at various scales, which increases computational complexity, especially when processing high-resolution images. How to effectively fuse features at different scales to fully exploit the granular details of high-resolution features and the overall data from low-resolution features, while also decreasing the computational and algorithmic intricacy of multi-tiered

feature consolidation, poses a challenge to be addressed. Therefore, as shown in Figure 4, we present an MFIN architecture that markedly diminishes the parameter count and computational intricacy of feature integration, concurrently enhancing model performance.

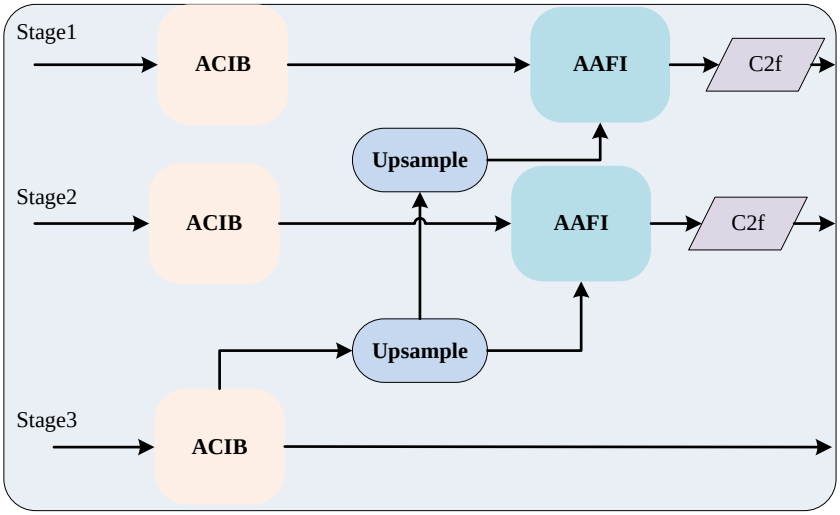


Figure 4. Schematic of the multi-scale feature interaction network.

Specifically, MFIN consists of ACIB, upsampling, AAFI, and C2f. Firstly, Backbone will output feature maps of advanced, intermediate, and shallow feature maps to the neck for multi-scale feature fusion. These three feature maps with different scales are first enhanced through the ACIB module for feature representation, whereas the features in Stage 3 are directly processed for object detection and classification after enhancement. At the same time, the feature maps in Stage 3 are then increased in image resolution through the upsampling module and fused with intermediate semantic features at scale. The fused semantic information is further extracted through the C2f module and then sent to the detection head for Stage 2 detection and classification. At the same time, advanced semantic information is increased in image resolution through two upsampling modules and fused with low-level semantic feature maps for Stage 1 detection and classification. It is important to mention that we implement transposed convolution for feature map upsampling here because transposed convolution is capable of making adaptive changes to the upsampling method according to the input feature map, suit diverse data inputs and task specifications, and bolster the model's generalization ability.

In Figure 5, the core idea of the ACIB module is to enhance the feature representation ability by utilizing the channel attention mechanism. The ACIB module first uses average pooling and maximum pooling to extract global statistical details and local peak data from the input feature map, respectively. Next, two convolutional layers serve to process the features after average pooling and maximum pooling. The first convolutional layer is applied for extracting features, and the second convolutional layer is designed for generating attention weights. By adding up the attention weights obtained from average pooling and maximum pooling, the module can integrate these two different types of information. The module uses the ReLU activation function to increase nonlinearity, enabling the model to capture more sophisticated feature representations. The module uses the Sigmoid to scale attention weights within the range of [0,1] and then multiplies them pixel-wise to the input feature map to enhance the information of important channels while suppressing the information of unimportant channels. This design enables the module to dynamically determine the significance of individual channels and retain the pertinent details of the original features by multiplying them on a per-element basis with the initial features. Ultimately, 1×1 convolution is utilized to modify the channel dimensions of the features to align with the anticipated input specifications of subsequent layers, guaranteeing

continuity and reliability within the framework of the network. The expression defining ACIB is presented below:

$$Y = Conv_{1\times1}\left(\sigma\left(MLP\left(X^c_{avg}\right) + MLP\left(X^c_{max}\right)\right) \otimes X\right) \tag{7}$$

Herein, X denotes the initial input feature map, X^c_{avg} and X^c_{max} denote the global average and maximum pooling applied to X across the channel axis. $MLP(\cdot)$ represents a sequence involving 1×1 convolution, the RELU, and another 1×1 convolution operation, where 1×1 convolution serves to modify the channel count.$\sigma(\cdot)$ denotes Sigmoid. $\otimes$ is utilized for pixel-wise multiplication, and finally, 1×1 convolution is used to adjust channel dimensions to match the expected input dimension of subsequent layers.

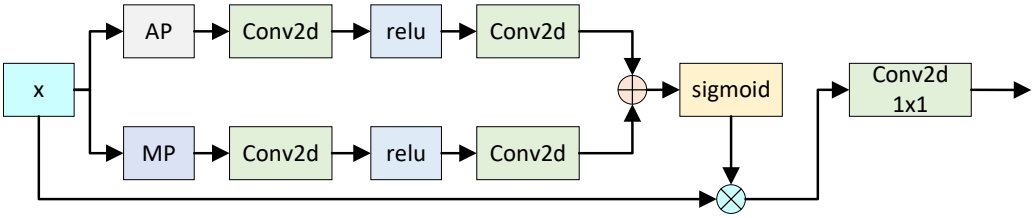


Figure 5. Attentive channel integration block.

As illustrated in Figure 6, the AAFI module feature fusion aims to effectively combine high-level and low-level features through the use of channel attention and skip connection technology, thus enhancing the model's performance and robustness for specific tasks. Specifically, first, the advanced feature X_1 generates an attention weight map through the channel attention module [36]. This attention weight map reflects the importance of different channels in X_1, that is, channels that have more critical and useful features.Next, we will multiply this attention weight map element by element with low-level feature X_2. The purpose of doing so is to utilize the high-level conceptual information in X_1 to guide the low-level conceptual details in X_2 so that low-level features can be better guided and corrected by high-level features. By multiplying each element, we can weight X_2 based on the significance of each channel of X_1, thereby enhancing the feature representation corresponding to the important channels in X_1. Finally, the multiplied result is concatenated and linked to the original high-level feature X_1, which successfully transfers the fine-grained details from low-tier features along with the contextual data from high-tier features, thereby achieving a more comprehensive and precise feature representation. Using this method, we acquire the ultimate feature representation that merges advanced contextual information to steer the efficacy of basic semantic information.

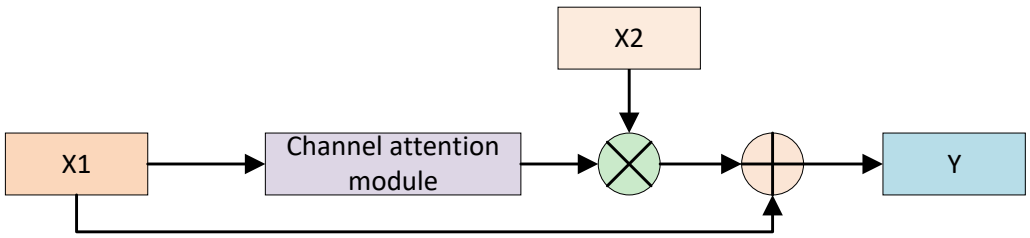


Figure 6. Attention-augmented feature integration.

3.4. Dynamic Feature Aggregation Head

We have noticed that the detection head of YOLOv8 does not have a dynamic perception mechanism and cannot dynamically focus on important features. The weed detection head cannot suppress background noise and efficiently capture feature information of various scales and positions, thereby reducing the representation ability of the target detection head and resulting in poor weed detection performance. As shown in Figure 7, we propose

DFAH to address this issue, where DCNv3 [37] enhances the model's expressive power by organizing the spatial aggregation process, allowing the model to acquire knowledge across various representational domains. By learning spatial offset and feature channel weights, DFAH can capture long-range dependency information and weigh the features of different channels based on the channel weights of the feature map to suppress the influence of unimportant features. Through these designs, DFAH can adaptively capture long-range information from feature maps, thereby enhancing the model's capacity for detailed representation and improving the performance of object detection. This method is more flexible and effective in handling features of different scales and positions, thereby bolstering the model's flexibility and robustness across intricate environments.

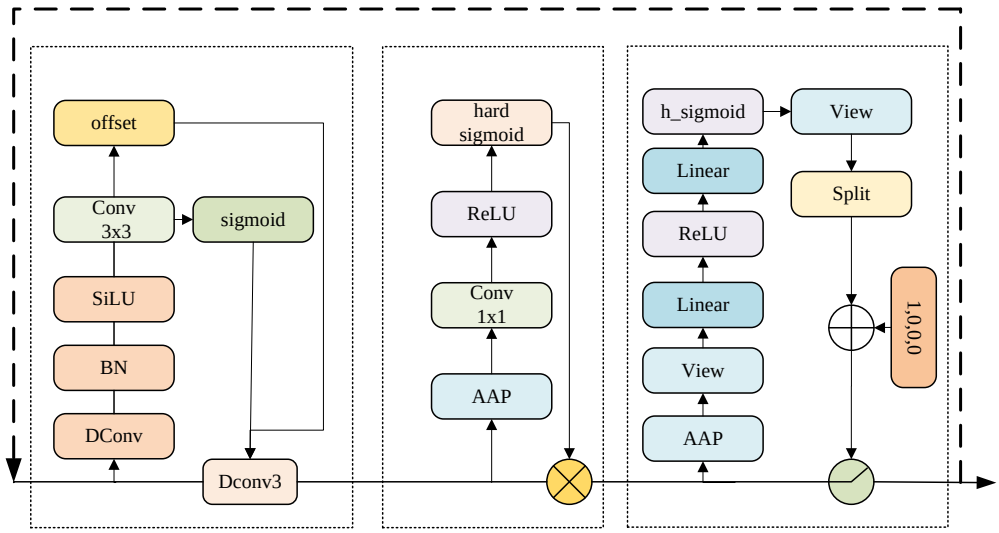


Figure 7. Overall structure diagram of dynamic feature aggregation head.

Specifically, for an input feature, we first use deep convolution to elicit spatial information contained within the feature map, dividing the input channel into several groups for convolution, effectively reducing parameter and computational complexity. Concurrently, channels between different groups can independently acquire distinct feature representations, which helps improve the model's expressive power. Next, we apply batch normalization layers and SiLU activation functions to alleviate the gradient explosion problem and help the model converge faster. The use of 3×3 convolutional layers here is to determine the displacement at every location in the feature map. Offset is usually used to modify the sampling locations throughout the target feature map, to achieve spatial feature alignment or deformation. In DFAH, offset is used to adjust the sampling points of intermediate feature maps to align feature maps of different scales. By learning the sampling offset, DCNv3 is capable of dynamically resizing the receptive field to capture long-range dependency information and thus improve the model's expressive power. Next, for deformable convolutional feature maps, we first perform adaptive average pooling, compressing the spatial dimension of the feature maps to 1×1 so that the feature values on each channel are integrated into a scalar. Next, a 1×1 convolutional layer serves to diminish the channel count to 1 for subsequent channel attention weighting. The ReLU activation function serves to activate the convolutional eigenvalues, increasing the model's non-linear representation ability. To reduce computational complexity, the hard sigmoid is used here, and then the processed feature map is element-wise multiplied with the deformable convolutional feature map to achieve dynamic weighting of features. The objective of this process is to assign weights to the features of different channels, predicated on the channel weights of the feature map, thereby enhancing the prominence of significant features, curbing the impact of unimportant features, and, as a result, bolstering the performance of the model. Although SiLU activation functions and hard sigmoid activation functions have

been used, they both have certain nonlinear characteristics, whereas DyReLU activation functions [36] have higher flexibility and adaptability. The DyReLU activation function [38] can dynamically modify the parameters of the activation function in response to input data, which means it can adaptively adjust its nonlinear characteristics as per different input data distributions and tasks, thereby better capturing the complex features of the data. When facing different tasks and types of data, this adaptability helps to additionally enhance the model's expressive ability, generalization ability, and anti-interference ability, thereby improving overall performance.

4. Experiment

In this section, we provided a detailed explanation of the dataset relied upon by the research and the specific steps implemented and conducted ablation experiments on our proposed EY8-LDFA model to demonstrate its performance. Additionally, we contrasted EY8-LDFA with a range of alternative approaches and engaged in an exhaustive and thorough examination of the experimental outcomes. Finally, we conducted an in-depth analysis of the detection results using visual analysis technology, thereby substantiating the superior efficacy of EY8-LDFA in executing weed identification task.

4.1. Implementation

All of our experiments were conducted only on NVIDIA A40 for training, validation, and testing. The NVIDIA A40 is manufactured by the American multinational technology company NVIDIA and is headquartered in Santa Clara, California. The detailed data regarding the system version, GPU model, Python version, and additional configurations on the experimental platform are presented in Table 1. Table 2 shows some parameter information used by our model during the training process. For each training round, our batch size is configured for 16, which means that the model will process 16 samples in each iteration. Epochs represent the training round of the model, lr0 is the original learning rate of the model, lrf is the final learning rate of the model, momentum controls the weight of historical gradient information, weight_decay is the weight attenuation coefficient of the model, and the optimizer uses SGD, workers denote the number of threads loading the data, patience denotes that model training is stopped if the training loss value remains unchanged for 50 consecutive times, amp denotes that half-precision training is turned on, iou denotes the threshold of the generated detection frame, and warmip_epochs denotes the number of iterations in the warm-up phase. The sample data are reshaped to 640 $\times$ 640 before being sent to the network. At the same time, we use a mixed precision training method to accelerate the model training speed. Finally, an early stop mechanism was used to prevent overfitting.

Table 1. Platform settings.

Attribute	Version
OS	Ubuntu 18.04.6
GPU	NVIDIA A40
Python	Python 3.8.19
Torchvision	0.15.2
Torch	2.0.1
DCNv3	1.1

Table 2. Parameter settings.

Attribute	Value
Epochs	300
Patience	50
Batchsize	16
lr0	0.01
lrf	0.01
Momentum	0.937
Weight_decay	0.0005
workers	8
patience	50
amp	true
iou	0.7
warmup_epochs	3.0
Optimizer	SGD

4.2. Datasets

The CottonWeedDet3 dataset [39] consists of color images taken with mobile phone cameras and digital cameras. All images are saved in .jpg format. They were taken by photographers at various locations in several cotton-producing states in the United States, with a primary focus on North Carolina and Mississippi. The images were captured during the planting season from 2020 to 2021. The image shooting work is completed in a natural light environment in the field, ensuring that weeds are recorded from various angles during each stage of their growth cycle. Our research mainly focuses on three main weed categories, namely carpetweed, morningglory, and palmer_amaranth. The average resolution of these images is 4442 × 4335 pixels, and the entire dataset consists of 848 images with annotations of over 1500 bounding boxes; these annotations identify three common weeds within the cotton plantations of the southern U.S.

Following that, we further utilized the cotton-weed dataset [40] to assess the model's ability to generalize across various scenarios. The cotton-weed consists of 570 images, encompassing a myriad of different types of weeds, such as Artemisia annua, Fructus Kochia, Cichli coriander, Castor bean, Matang, and false sea horse tooth ensuring the range of the dataset. These images were mostly taken in the natural lighting environment of cotton fields in India, using smartphones or digital cameras, and were saved in .jpg format. There are significant differences in visual attributes, shape, soil background, and photographic variables among images of the same weed category. This variety helps to construct models that are resilient to changes in photographic variables. During model training, we divided the dataset into a 60% training set, 20% validation set, and 20% test set, facilitating a nuanced assessment of the model's performance and generalization capabilities.

The aerial weeds dataset [41] was collected by DJI Phantom 4 Pro drone over different agricultural fields in North Dakota, USA. It covers the period from May 2021 to September 2022 to obtain data from different seasons and environments. The resolution of the acquired images was 5472 × 3648 pixels. A total of 551 high-resolution images was acquired. Weeds in the images were manually labeled and a total of 2390 weed instances were labeled. Five common weeds including Kochia, Common ragweed, Horseweed, Redroot pigweed, and Waterhemp were covered. The dataset considered real farmland scenarios such as different soil backgrounds, crop shading, unknown weeds, objects, shadows, and image blurring to improve the usefulness and robustness of the dataset. Finally, a ratio of 60%:20%:20% was used to divide the data into training, validation, and test sets.

4.3. Ablation Experiment

To confirm the efficacy of the proposed method, we conducted ablation experiments to analyze the contributions of each constituent part and the interactions between separate parts and compared them against the baseline model. We applied C2F-DFIB, MFIN, and DFAH to YOLOv8s to evaluate their impact on model performance. In Tables 3 and 4,

the nonmaximum inhibition threshold of IoU for the experiment is defined as 0.7. We use mean average precision (mAP), accuracy (P), recall (R), F1 value, count of parameters (Params), and computational complexity (FLOPs) in the experiment to assess the general performance of the model. It is worth noting that once a satisfactory level of accuracy is achieved, a model with less extensive parameters and simpler computational requirements is increasingly valuable, particularly for edge devices with constrained resources.

Table 3. Ablation experiment of C2f-DFIB, MFIN, and DFAH on CottonWeedDet.

YOLOv8s	C2f-DFIB	MFIN	DFAH	P	R	F1	mAP_{50}	mAP_{75}	Params (M)	FLOPs (G)
✓				81.1	75.5	78.1	77.2	66.8	11.12	28.4
✓	✓			82.4	74.8	78.4	78.4	67.2	10.45	27.4
✓		✓		83.3	72.1	77.2	80.4	67.1	7.13	23.9
✓			✓	84.0	74.5	78.9	79.8	68.3	10.46	29.2
✓	✓	✓		78.9	77.8	78.3	80.7	67.2	6.46	22.8
✓	✓		✓	87.0	72.6	79.1	81.4	67.8	9.79	28.1
✓		✓	✓	84.3	72.4	77.8	80.7	69.0	7.29	25.8
✓	✓	✓	✓	83.8	73.9	78.5	81.9	71.8	6.62	24.7

Table 4. Ablation experiment of C2f-DFIB, MFIN, and DFAH on cotton-weed.

YOLOv8s	C2f-DFIB	MFIN	DFAH	P	R	F1	mAP_{50}	mAP_{75}	Params	FLOPs
✓				84.1	67.2	74.7	76.3	53.4	11.12	28.4
✓	✓			81.3	73.9	77.4	76.4	51.7	10.45	27.4
✓		✓		79.3	75.9	77.5	77.5	54.9	7.13	23.9
✓			✓	84.5	71.5	77.4	78.1	55.5	10.46	29.2
✓	✓	✓		85.7	72.0	78.2	79.2	54.0	6.46	22.8
✓	✓		✓	76.2	73.6	74.8	75.0	55.4	9.79	28.1
✓		✓	✓	78.7	72.8	75.6	76.5	55.1	7.28	25.8
✓	✓	✓	✓	79.9	76.9	78.3	81.6	56.7	6.62	24.7

After introducing the C2f-DFIB module, the data delineated in Table 3 show that F1 increased by 0.3%, mAP_{50} increased by 1.2%, and mAP_{75} increased by 0.4%. The data delineated in Table 4 demonstrate that F1 increased by 2.7%, mAP_{50} increased by 0.1%, and mAP_{75} decreased by 1.7%. It is apparent that the module enhanced the comprehensive performance; however, mAP_{75} decreased, which we speculate is influenced by the distribution of the dataset. During the weed detection process, challenges frequently arise due to the varied sizes of targets and the occlusion that occurs between crops and weeds throughout their growth cycle. Therefore, improving the feature extraction ability of the backbone is extremely important. The C2F-DFIB module can capture the global information of feature maps through large-scale convolution kernels and dilated convolutions, whereas ordinary convolutions are used to extract local details. This fusion of global and local information helps the model to understand image content more comprehensively, thereby enhancing the network's ability to distill core features. After integrating the MFIN architecture for multi-scale feature fusion, the findings depicted in Table 3 demonstrate that F1 decreased by 0.9%, mAP_{50} increased by 3.2%, and mAP_{75} increased by 0.3%. The experimental results in Table 4 show that F1 increased by 0.8%, mAP_{50} increased by 1.2%, and mAP_{75} increased by 1.5%. Surprisingly, with the addition of the MFIN architecture, the parameter number of the model decreased by 3.99 M and the computational complexity decreased by 4.5 G. This optimization is particularly important for resource-constrained platforms such as mobile devices, IoT devices, and other embedded systems, as it allows the model to run on these platforms without causing excessive burden on hardware resources. The MFIN architecture utilizes a channel attention mechanism to enhance feature representation capability while using transposed convolution for upsampling feature maps to accommodate diverse data and task specifications. Finally, using the channel attention mechanism and skip connection

technology, high-level semantic information is guided to low-level semantic information, effectively reducing parameter and computational complexity. After replacing the detection head in YOLOv8 with DFAH, Table 3 shows that F1 increased by 0.8%, mAP_{50} increased by 2.6%, and mAP_{75} increased by 1.5%. Table 4 indicates that F1 increased by 2.7%, mAP_{50} increased by 1.8%, and mAP_{75} increased by 1.1%. Moreover, it is observed that the detection mechanism we developed cut the parameter count by 0.66 M compared to the detection head of YOLOv8 but increased the computational load by 0.8 G. We speculate that using DCNv3 in DFAH may reduce the parameter count but increase the computational load. Through spatial aggregation, DCNv3 empowers the model to assimilate data from diverse descriptive realms, enhancing the model's expressive power. This method may have higher computational complexity than traditional convolution operations, as DCNv3 requires processing more parameters (such as sampling offset and scaling coefficient) and computing more complex operations (such as dynamically adjusting receptive fields).

When C2f-DFIB and MFIN work together on the benchmark model, they show some positive effects. The experiment in Table 3 shows that its mAP_{50} index is 2.3% and 0.9% higher than when C2f-DFIB and MFIN act alone, respectively. Its mAP_{75} index has the same accuracy as when C2f-DFIB acts alone but is 0.1% higher than when MFIN acts alone. The experimental results in Table 4 show that when C2f-DFIB and MFIN act together on the benchmark model, their mAP_{50} index is 3.46% and 1.7% higher than when they act alone, respectively. However, the mAP_{75} index is reduced by 0.9% compared to when MFIN acts alone on the benchmark. We speculate that this result is due to the influence of the dataset distribution. When C2f-DFIB and DFAH work together on the benchmark model, as shown in Table 3, it is apparent from the data that the accuracy of both F1 and mAP_{50} indicators is higher when they work together than when they work alone. However, the accuracy of mAP_{75} decreased compared to when DFAH acts alone, indicating that there is a mutual promotion effect between the two modules in detecting small targets, and the performance in detecting large targets still needs to be improved. The results in Table 4 show that when the two modules work together on the baseline model, the parameter count of the model is only 9.79, which is even smaller than the baseline parameter count. This is because both C2f-DFIB and DFAH have the effect of reducing the model parameter count. When MFIN and DFAH jointly act on the model, Table 3 experiments show that the positive effects between the two modules are displayed on the CottonWeedDet3 dataset. The mAP_{50} indicator is 0.3% and 0.9% higher than when they act alone, and the mAP_{75} indicator is 1.9% and 0.7% higher than when they act alone. However, on the Cotton Weed dataset, when the two work together on the baseline model, modules show a certain degree of mutual assistance, showing a positive side. In summary, the comparative experiments between these modules indicate that the main effect between modules is mutual promotion.

Ultimately, the EY8-LDFA model, which we introduced, realized a significant enhancement in performance metrics when juxtaposed with the benchmark YOLOv8s model. The F1, mAP_{50}, and mAP_{75} indicators improved by 0.4%, 4.7%, and 5.0%, and 3.6%, 5.3%, and 3.3%, respectively, on two publicly available datasets. The computational complexity of the model also decreased by 3.7 G compared to the baseline, to 24.7 G. What we are most proud of is that the parameter quantity of EY8-LDFA is 4.5 M lower than the baseline model, only 6.62 M, facilitating our model implementation on more resource-constrained platforms, particularly within edge computing and Internet of things devices.

4.4. Compared with Other Methods

To comprehensively evaluate the efficacy of the model proposed in this study, we compared EY8-LDFA with some other state-of-the-art models, including representative models based on two-stage anchors, single-stage anchors, and anchor-free models. Our main evaluation indicators include F1, mAP_{50}, mAP_{75}, parameter count, and computational complexity. The accuracy comparison is shown in Tables 5 and 6, and the performance comparison is shown in Tables 7 and 8.

Table 5. Accuracy comparison of various models on CottonWeedDet3.

Method	P	R	F1	mAP_{50}	mAP_{75}
Faster_Rcnn [42]	67.7	79.2	72.9	78.1	63.3
Tridentnet [43]	56.1	78.6	65.4	74.6	63.5
Fcos [44]	75.2	56.4	64.4	65.6	42.2
YOLOv3 [22]	78.7	68.6	73.3	75.3	63.4
YOLOv5s [24]	82.5	70.6	76.1	77.2	61.3
YOLOv6s [25]	76.2	72.6	74.3	77.6	61.2
YOLOv7-tiny [26]	76.9	68.5	72.4	77.4	55.4
YOLOv8s [27]	81.1	75.5	78.1	77.2	66.8
YOLOv9 [28]	79.9	75.2	77.4	79.7	64.9
Gelan [28]	81.9	72.6	76.9	77.6	62.9
RTDETR-L [45]	86.9	75.3	80.6	78.8	68.2
RTDETR-R50 [45]	79.6	71.5	75.3	73.6	59.9
RTDETR-R101 [45]	82.4	70.4	75.9	74.1	59.2
EY8-LDFA (ours)	83.8	73.9	78.5	81.9	71.8

Table 6. Accuracy comparison of various models on cotton-weed.

	P	R	F	mAP_{50}	mAP_{75}
Faster_Rcnn [42]	54.0	77.2	63.5	72.9	52.2
Tridentnet [43]	54.2	81.1	64.9	74.5	53.7
Fcos [44]	87.0	71.8	78.6	75.3	50.9
YOLOv3 [22]	83.2	71.5	76.9	78.0	54.5
YOLOv5s [24]	80.3	71.0	75.3	79.2	51.4
YOLOv6s [25]	85.3	69.2	76.4	77.8	55.1
YOLOv7-tiny [26]	77.7	65.0	70.7	72.6	40.7
YOLOv8s [27]	84.1	67.2	74.7	76.3	53.4
YOLOv9 [28]	82.0	63.0	71.2	72.3	52.7
Gelan [28]	78.5	66.2	71.8	75.0	54.5
RTDETR-L [45]	86.2	74.0	79.6	77.8	57.0
RTDETR-R50 [45]	91.7	62.6	74.4	75.5	53.7
RTDETR-R101 [45]	85.2	65.1	73.8	73.0	53.1
EY8-LDFA (ours)	87.8	71.1	78.5	77.6	55.6

Table 7. Performance comparison of various models on CottonWeedDet3.

Method	Size	Params (M)	FLOPs (G)
Faster_Rcnn [42]	640 × 640	41.36	69.32
Tridentnet [43]	640 × 640	33.07	763
Fcos [44]	640 × 640	32.11	56.58
YOLOv3 [22]	640 × 640	103.66	282.2
YOLOv5s [24]	640 × 640	9.11	23.8
YOLOv6s [25]	640 × 640	16.29	44.0
YOLOv7-tiny [26]	640 × 640	6.01	13.0
YOLOv8s [27]	640 × 640	11.12	28.4
YOLOv9 [28]	640 × 640	60.5	263.9
Gelan [28]	640 × 640	31.19	116.8
RTDETR-L [45]	640 × 640	31.99	103.4
RTDETR-R50 [45]	640 × 640	41.96	129.6
RTDETR-R101 [45]	640 × 640	74.66	247.1
EY8-LDFA (ours)	640 × 640	6.62	24.7

Table 8. Performance comparison of various models on cotton-weed.

Method	Size	Params (M)	FLOPs (G)
Faster_Rcnn [42]	640 × 640	41.36	90.91
Tridentnet [43]	640 × 640	33.07	774
Fcos [44]	640 × 640	32.11	78.59
YOLOv3 [22]	640 × 640	103.66	282.2
YOLOv5s [24]	640 × 640	9.11	23.8
YOLOv6s [25]	640 × 640	16.29	44.0
YOLOv7-tiny [26]	640 × 640	6.01	13.0
YOLOv8s [27]	640 × 640	11.12	28.4
YOLOv9 [28]	640 × 640	60.4	263.9
Gelan [28]	640 × 640	31.1	116.8
RTDETR-L [45]	640 × 640	31.98	103.4
RTDETR-R50 [45]	640 × 640	41.95	129.5
RTDETR-R101 [45]	640 × 640	74.65	247.1
EY8-LDFA (ours)	640 × 640	6.62	24.7

As shown in Tables 5 and 6, our proposed EY8-LDFA has shown significant improvements in F1, mAP_{50}, and mAP_{75} metrics compared to two-stage detection models such as Faster-Rcnn and Tridentnet. With respect to parameter count and computational complexity, we can observe the advantages of EY8-LDFA in Tables 7 and 8. Among them, Faster-Rcnn has approximately 6.2 times the parameter count of EY8-LDFA, and Tridentnet has approximately 5 times the parameter count of EY8-LDFA. This enhancement qualifies our model for effective implementation on lightweight devices for detection in some real-time tasks. From Tables 5 and 6, it can be seen that for the anchor-free box model Fcos, the mAP_{50} and mAP_{75} of EY8-LDFA perform much better than the Fcos model. However, in Table 6, the recall and F1 of Fcos are better than those of EY8-LDFA. Our analysis suggests that disparate datasets exhibit unique characteristics, including variations in target distribution and object scale. The design of FCOS is potentially better aligned with the specific attributes of the dataset in question, which may account for its elevated recall performance. In our comparative analysis of the YOLO series models, the model selection criteria were primarily centered on models that have a parameter count commensurate with that of YOLOv8 across different versions. Currently, the source code of YOLOv9s is not publicly available, so we chose the YOLOv9 model. From Table 5, it can be shown that the indicators of EY8-LDFA comprehensively suppress the various indicators of YOLOv3. On another dataset, as shown in Table 6, we can see that EY8-LDFA sacrifices 0.4% accuracy on the mAP_{50} indicator. However, from the comparative analysis of parameter and computational costs, this is acceptable because YOLOv3 has 15.6 times and 11.4 times the parameter and computational costs of EY8-LDFA, respectively. The huge computational and parameter costs have not brought corresponding effects, which is something we do not want to see. Upon careful observation, it can be observed that in Table 5, EY8-LDFA outperforms all previous YOLO versions in F1, mAP_{50}, and mAP_{75} metrics. In Table 6, it becomes clear that almost the same effect was achieved on another dataset. However, it is noteworthy that the mAP_{50} of YOLOv6s is 0.2% higher than that of EY8-LDFA. By comparing the parameter and computational costs in Table 8, we can see that YOLOv6s has 2.4 times and 1.7 times the parameter and computational costs of EY8-LDFA, respectively. Such a high cost for such low accuracy is a result that we are unwilling to see. Therefore, for the YOLO series lightweight models, considering the comprehensive accuracy, parameter quantity, and computational complexity, it is still more cost-effective for EY8-LDFA. Finally, we also compared some Transformer series models. Here, we selected the RTDETR model with high real-time performance. Here, we selected RTDETR models with different sizes and backbone feature extraction. From Tables 5 and 6, it can be seen that RTDETR-L has the potential to surpass EY8-LDFA in terms of accuracy in various indicators, whereas the parameter size of RTDETR-L is only 31.99 M. Through certain model architecture improvements, it is expected that the Transformer model can be deployed on edge devices with high real-time requirements and lightweight, which also

provides a new idea for future weed detection-related tasks. However, it cannot be denied that the current RTDETR-L model has a computing power of 103.4, which is 4.18 times that of EY8-LDFA. The huge computing power makes it difficult to deploy on resource-limited platforms. In summary, our proposed EY8-LDFA performs very well on the mAP_{50} and mAP_{75} indicators, reaching the ideal level. What we are most proud of is that the model has only 6.62 M parameters and 24.7 G computing power, which meets the requirements of real time and efficiency, and can be deployed on more resource-limited hardware platform devices at a lower cost.

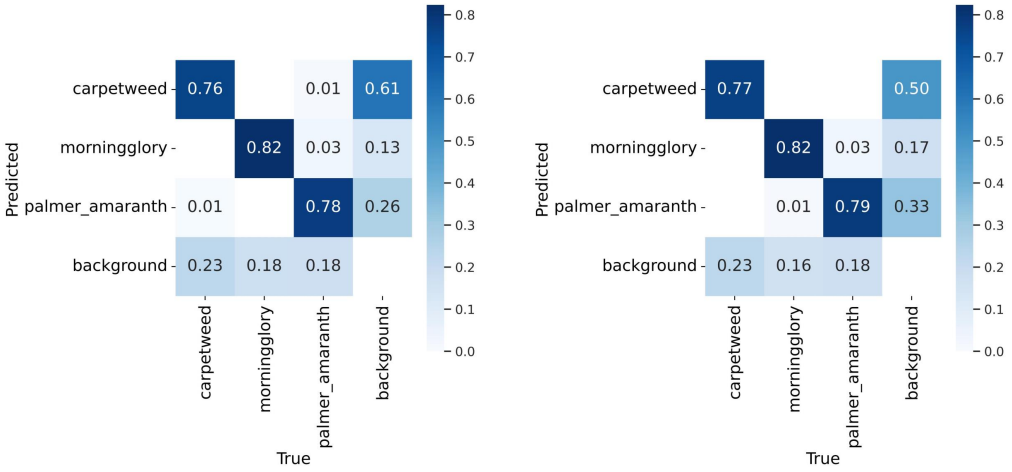
Finally, to further validate the generalization of the model, we conducted a large number of comparative experiments on the Aerial weeds dataset to verify the excellent results of our model, as shown in Table 9. It can be seen that in the F1 metric, EY8-LDFA is only 0.76% and 0.19% less accurate than the YOLOv3 and YOLOv9 models, respectively, in the mAP_{50} metric, EY8-LDFA is only 0.4% and 0.6% less accurate than the YOLOv3 and YOLOv9 models, respectively, and in the mAP_{75} metric, EY8-LDFA is more accurate than the YOLOv3 and YOLOv9 model accuracies are 1.4% and 0.8% lower, respectively. As shown in Table 8, although there is a small decrease in the accuracy of EY8-LDFA compared to YOLOv3 and YOLOv9, the number of parameters of YOLOv3 and YOLOv9 is 15.65 and 9.12 times higher than that of EY8-LDFA, and the amount of computation is 11.42 times higher than that of EY8-LDFA, respectively, 10.68 times. Such high computing resources are exchanged for such low performance. This is a result we do not want to see. The experimental results in the Aerial weeds dataset also prove the efficiency of our model.

Table 9. Accuracy comparison of various models on aerial weeds dataset.

	P	**R**	**F**	**mAP_{50}**	**mAP_{75}**
Faster_Rcnn [42]	47.6	40.6	43.8	42.5	12.5
Tridentnet [43]	44.3	51.1	47.46	42.5	8.3
Fcos [44]	58.6	29.9	39.6	37.8	10.1
YOLOv3 [22]	56.8	60.8	58.7	58.0	33.1
YOLOv5s [24]	48.0	57.4	52.2	50.5	28.9
YOLOv6s [25]	61.1	52.0	56.1	55.3	32.1
YOLOv7-tiny [26]	64.0	26.3	37.2	23.4	10.5
YOLOv8s [27]	51.3	51.5	51.4	50.1	27.7
YOLOv9 [28]	58.1	58.2	58.1	58.2	32.5
Gelan [28]	55.4	57.4	56.3	55.5	32.3
RTDETR-L [45]	47.1	45.3	46.1	39.0	22.2
RTDETR-R50 [45]	46.7	37.0	41.2	35.3	20.2
RTDETR-R101 [45]	30.5	32.5	31.4	23.7	12.1
EY8-LDFA (ours)	61.4	54.9	57.9	57.6	31.7

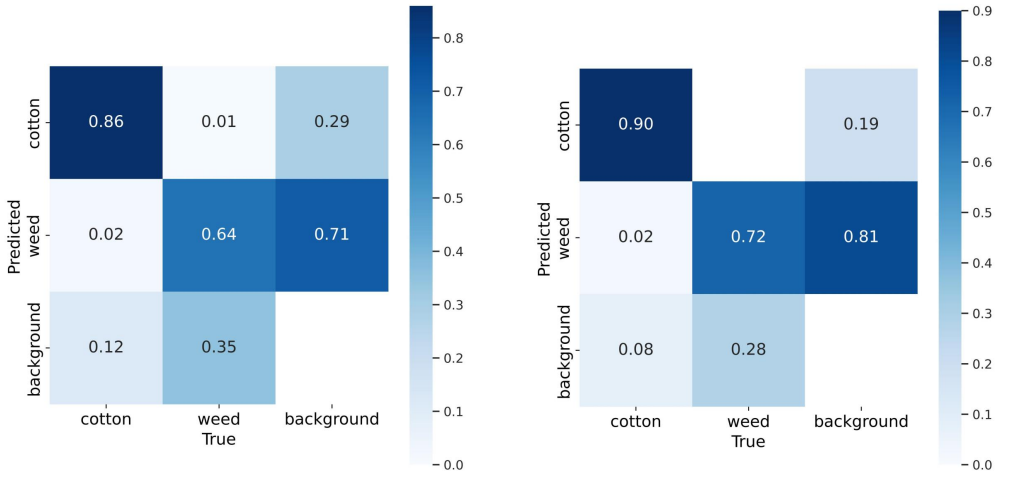
4.5. Visualization

To visually demonstrate the efficacy of the EY8-LDFA model in weed detection tasks, we used the confusion matrix as a tool to present its classification accuracy on different categories. Considering that there may be differences in sample size between different categories, we normalized the confusion matrix, which facilitates a more precise assessment of the classification performance for the model across all categories. In the normalized confusion matrix, rows indicate the model's prediction labels, whereas columns correspond to the true labels. From Figures 8 and 9, it is evident that the normalized comprehensive confusion matrix outcomes indicate that, when contrasted with the baseline model YOLOv8, on the CottonWeedDet3 dataset, although the prediction accuracy of EY8-LDFA for each target category has not changed, it reduces the probability of false detection as carpetweed for the palmer_amaranth category of weeds. On the Cotton Weed dataset, EY8-LDFA showed significant improvement in prediction for both categories, with accuracy rates increased by 4% and 8%, respectively. The presented findings robustly substantiate the superior performance of the EY8-LDFA model, particularly within the domain of weed detection tasks.



YOLOv8 EY8-LDFA

Figure 8. Normalized confusion matrix on CottonWeedDet3.



YOLOv8 EY8-LDFA

Figure 9. Normalized confusion matrix on cotton-weed.

In actual agricultural settings, with changes in growth and development stages, weeds of different scales often appear, especially those in the seedling stage. YOLOv8 is adept at identifying larger targets within agricultural scenes; however, it encounters challenges when attempting to discern smaller weeds. In contrast, the EY8-LDFA model, which has been optimized for the extraction of nuanced backbone features, exhibits a heightened capacity for detecting minor targets, as evidenced by the comparative analysis between groups A and C illustrated in Figure 10. When there is a weed target in the shadow of the leaves, EY8-LDFA can also detect the target well, as shown in Group B in Figure 10. This enhancement is attributed to the proposed C2F-DFIB module, which adeptly integrates global and local information to augment the discriminative capacity of feature representations, making the model more adaptable and generalizable when dealing with complex scenes. In the renderings of groups B, C, and D in Figure 11, it can be seen that when cotton grows vigorously, YOLOv8 will mistakenly detect it as multiple plants, but, in fact, it only has one crop. This is because the model overly relies on local features during the detection process and fails to fully consider global features. DFAH can capture long-range dependency information and weigh the features of different channels based on the channel weights of the feature map to enhance the representation of important features and suppress the influence of unimportant features. As shown in Group A in Figure 11, YOLOv8

failed to recognize the entire plant, but only partially recognized it. However, EY8-LDFA enhanced the model's cognitive ability to input data by fusing local and global information, this enhancement bolsters the backbone network's proficiency in both the extraction and nuanced representation of features.



(**a**) Missed detection. (**b**) Shadow occlusion. (**c**) Missed detection. (**d**) Misdetection.

Figure 10. Visualization of YOLOv8 (**top**) and EY8-LDFA (**bottom**) on CottonWeedDet3. The orange box represents palmer_amaranth, the red box represents carpetweed, and the pink box represents morningglory. (**a**,**c**) Show the situation of YOLOv8 missing detection for small targets; (**b**) demonstrates that EY8-LDFA still has good performance under occlusion conditions; (**d**) showcases YOLOv8's ability to identify a single plant as multiple plants.



(**a**) Partial detection. (**b**) Misdetection. (**c**) Misdetection. (**d**) Misdetection.

Figure 11. Visualization of YOLOv8 (**top**) and EY8-LDFA (**bottom**) on cotton weeds. Among them, the pink box represents weed information, and the red box represents cotton crops. (**a**) Shows the situation where YOLOv8 cannot detect whole plants of weeds; (**b**) shows the situation where YOLOv8 mistakenly detects plants as cotton; (**c**,**d**) show the situation where YOLOv8 mistakenly detects one cotton plant as multiple plants.

5. Conclusions and Future Work

5.1. Conclusions

This study proposes an improved EY8-LDFA model based on the YOLOv8 model to address the issues of target scale diversification and occlusion during the growth cycle of crops and weeds in cotton field weed detection. Firstly, we designed the DFIB module, which

uses large-scale convolution kernels to capture global contextual information of images, whereas dilated convolutions at different scales can capture more detailed local features. By combining these different scales of information, the model can more comprehensively understand image content. Secondly, we designed the C2F-DFIB module, which adds the function of multi-scale feature extraction. This operation enhances the model's cognitive ability to input data by integrating local and global information, thereby augmenting the backbone network's proficiency in the extraction and the nuanced articulation of features. Subsequently, our observations indicate that the Path Aggregation Feature Pyramid Network, through its iterative application of downsampling and skip connections throughout the feature fusion procedure, repetitively processes numerous features, culminating in a heightened computational load and increased parameter complexity. Consequently, we introduce a novel lightweight architecture for multi-scale feature fusion, termed MFIN, which is designed to augment and integrate features across various scales, and utilizes channel attention mechanism and skip connection technology to guide low-level semantic information with high-level semantic information, effectively reducing the number of parameters. Finally, we propose the DFAH module to address the issue of YOLOv8's detection head not being able to dynamically focus on important features. This module captures long-range dependency information and feature alignment by adaptive sampling and dynamically weighting feature maps to enhance the weed detection performance of the detection head. The experiment shows that the improved model has significantly improved performance on two public datasets. In comparison with other prevalent algorithms, the EY8-LDFA demonstrates superior performance, particularly in terms of its lightweight nature and high accuracy. It should be emphasized that EY8-LDFA only has 6.62 M parameter quantities, which allows the model to be deployed on more resource-limited platforms such as micro drones and intelligent inspection robots, providing a new technical approach for the deployment of lightweight models for weed detection.

5.2. Future Work

Although we have made some progress, this research still has some limitations and needs further improvement. Firstly, our model cannot accurately adapt to all situations when dealing with complex and diverse cotton field environments or weeds in other crop fields, which includes factors such as growth characteristics of different plants, light variations, etc., and there are potential data discrepancies in image data captured by different devices. In the future, we plan to introduce multimodal information, such as infrared images, hyperspectral images, etc., to be fused with visible light images to enhance the model's ability to adapt to different environments. Among other things, multimodal information can provide richer feature representations, which can help to solve problems such as changes in lighting conditions and differences in vegetation. Secondly, we noticed that after using the DFAH detection head, although the number of parameters of the model decreased by 0.66 M, the computation increased by 0.8 G, which is because DCNV3 needs to dynamically compute the weights and offsets for each sampling point, which may be more complicated than the traditional 3×3 convolution operation. In the future, we consider a low-rank approximation of the weights and offsets to reduce the number of parameters and computational complexity of the model. In this way, complex weights and offsets can be approximated with fewer parameters, thus reducing the computational effort. Finally, we will continue to develop the weed identification and detection system for cotton fields for more efficient deployment and application on edge devices with limited computational resources.

Author Contributions: Conceptualization, D.R.; methodology, D.R.; validation, D.R.; writing—original draft preparation, D.R.; writing—review and editing, D.R., W.Y., D.C. and W.S.; supervision, D.C., Z.L. and Y.L.; funding acquisition, W.Y.

Funding: This research was funded by the National Key R&D Program of China (Grant No. 2022ZD011 5802), the Key Research and Development Program of the Autonomous Region (Grant No. 2022B01008),

the National Natural Science Foundation of China (Grant No. 62262065), the Tianshan Science and Technology Innovation Leading talent Project of the Autonomous Region (Grant No. 2022TSYCLJ0037).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kwaghtyo, D.K.; Eke, C.I. Smart farming prediction models for precision agriculture: A comprehensive survey. *Artif. Intell. Rev.* **2023**, *56*, 5729–5772. [CrossRef]
2. Phang, S.K.; Chiang, T.H.A.; Happonen, A.; Chang, M.M.L. From Satellite to UAV-based Remote Sensing: A Review on Precision Agriculture. *IEEE Access* **2023**, *11*, 127057–157076. [CrossRef]
3. Zhou, R.; Yin, Y. Digital agriculture: Mapping knowledge structure and trends. *IEEE Access* **2023**, *11*, 103863–103880. [CrossRef]
4. Iqbal, N.; Manalil, S.; Chauhan, B.S.; Adkins, S.W. Investigation of alternate herbicides for effective weed management in glyphosate-tolerant cotton. *Arch. Agron. Soil Sci.* **2019**, *65*, 1885–1899. [CrossRef]
5. Liu, B.; Bruch, R. Weed detection for selective spraying: A review. *Curr. Robot. Rep.* **2020**, *1*, 19–26. [CrossRef]
6. Raja, R.; Slaughter, D.C.; Fennimore, S.A.; Siemens, M.C. Real-time control of high-resolution micro-jet sprayer integrated with machine vision for precision weed control. *Biosyst. Eng.* **2023**, *228*, 31–48. [CrossRef]
7. Xu, B.; Fan, J.; Chao, J.; Arsenijevic, N.; Werle, R.; Zhang, Z. Instance segmentation method for weed detection using UAV imagery in soybean fields. *Comput. Electron. Agric.* **2023**, *211*, 107994. [CrossRef]
8. Liu, T.; Jin, X.; Zhang, L.; Wang, J.; Chen, Y.; Hu, C.; Yu, J. Semi-supervised learning and attention mechanism for weed detection in wheat. *Crop. Prot.* **2023**, *174*, 106389. [CrossRef]
9. Louargant, M.; Jones, G.; Faroux, R.; Paoli, J.N.; Maillot, T.; Gée, C.; Villette, S. Unsupervised classification algorithm for early weed detection in row-crops by combining spatial and spectral information. *Remote Sens.* **2018**, *10*, 761. [CrossRef]
10. Thomas, L.F.; Änäkkälä, M.; Lajunen, A. Weakly Supervised Perennial Weed Detection in a Barley Field. *Remote Sens.* **2023**, *15*, 2877. [CrossRef]
11. Fan, X.; Chai, X.; Zhou, J.; Sun, T. Deep learning based weed detection and target spraying robot system at seedling stage of cotton field. *Comput. Electron. Agric.* **2023**, *214*, 108317. [CrossRef]
12. Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; Shan, Y. Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. *arXiv* **2023**, arXiv:2311.15599.
13. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7373–7382.
14. Belissent, N.; Peña, J.M.; Mesías-Ruiz, G.A.; Shawe-Taylor, J.; Pérez-Ortiz, M. Transfer and zero-shot learning for scalable weed detection and classification in UAV images. *Knowl.-Based Syst.* **2024**, *292*, 111586. [CrossRef]
15. Veeragandham, S.; Santhi, H. Optimization enabled Deep Quantum Neural Network for weed classification and density estimation. *Expert Syst. Appl.* **2024**, *243*, 122679. [CrossRef]
16. Ju, J.; Chen, G.; Lv, Z.; Zhao, M.; Sun, L.; Wang, Z.; Wang, J. Design and experiment of an adaptive cruise weeding robot for paddy fields based on improved YOLOv5. *Comput. Electron. Agric.* **2024**, *219*, 108824. [CrossRef]
17. Zhu, H.; Zhang, Y.; Mu, D.; Bai, L.; Wu, X.; Zhuang, H.; Li, H. Research on improved YOLOx weed detection based on lightweight attention module. *Crop. Prot.* **2024**, *177*, 106563. [CrossRef]
18. Han, X.; Wang, H.; Yuan, T.; Zou, K.; Liao, Q.; Deng, K.; Zhang, Z.; Zhang, C.; Li, W. A rapid segmentation method for weed based on CDM and ExG index. *Crop. Prot.* **2023**, *172*, 106321. [CrossRef]
19. Liao, J.; Chen, M.; Zhang, K.; Zhou, H.; Zou, Y.; Xiong, W.; Zhang, S.; Kuang, F.; Zhu, D. SC-Net: A new strip convolutional network model for rice seedling and weed segmentation in paddy field. *Comput. Electron. Agric.* **2024**, *220*, 108862. [CrossRef]
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
21. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
22. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
23. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
24. Jocher, G. Ultralytics YOLOv5. 2020. Available online: https://zenodo.org/records/7347926 (accessed on 23 April 2024).
25. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
26. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.

27. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 23 April 2024).
28. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616v2.
29. You, H.; Xiong, Y.; Dai, X.; Wu, B.; Zhang, P.; Fan, H.; Vajda, P.; Lin, Y.C. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14431–14442.
30. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R.W. Biformer: Vision transformer with bi-level routing attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 10323–10333.
31. Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; Yuan, Y. Efficientvit: Memory efficient vision transformer with cascaded group attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14420–14430.
32. Hassani, A.; Walton, S.; Li, J.; Li, S.; Shi, H. Neighborhood attention transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6185–6194.
33. Pan, X.; Ye, T.; Xia, Z.; Song, S.; Huang, G. Slide-transformer: Hierarchical vision transformer with local self-attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2082–2091.
34. Liu, Z.; Yang, X.; Tang, H.; Yang, S.; Han, S. Flatformer: Flattened window attention for efficient point cloud transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1200–1211.
35. Song, J.; Mou, C.; Wang, S.; Ma, S.; Zhang, J. Optimization-inspired cross-attention transformer for compressive sensing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6174–6184.
36. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
37. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 14408–14419.
38. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic relu. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 351–367.
39. Rahman, A.; Lu, Y.; Wang, H. Deep Neural Networks for Weed Detections Towards Precision Weeding. In Proceedings of the 2022 ASABE Annual International Meeting. American Society of Agricultural and Biological Engineers, Houston, TX, USA, 17–20 July 2022; p. 1.
40. Deepak Kumaran, T. Cotton-Weed Dataset. 2021. Available online: https://universe.roboflow.com/deepak-kumaran-t/cotton-weed (accessed on 23 April 2024).
41. Rai, N.; Mahecha, M.V.; Christensen, A.; Quanbeck, J.; Zhang, Y.; Howatt, K.; Ostlie, M.; Sun, X. Multi-format open-source weed image dataset for real-time weed identification in precision agriculture. *Data Brief* **2023**, *51*, 109691. [CrossRef] [PubMed]
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
43. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6054–6063.
44. Tian, Z.; Chu, X.; Wang, X.; Wei, X.; Shen, C. Fully convolutional one-stage 3d object detection on lidar range images. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 34899–34911.
45. Lv, W.; Xu, S.; Zhao, Y.; Wang, G.; Wei, J.; Cui, C.; Du, Y.; Dang, Q.; Liu, Y. Detrs beat yolos on real-time object detection. *arXiv* **2023**, arXiv:2304.08069.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.