agriculture

MDPI

Article

WED-YOLO: A Detection Model for Safflower Under Complex Unstructured Environment

Zhenguo Zhang [1,2,*], Yunze Wang [1], Peng Xu [1,2], Ruimeng Shi [1], Zhenyu Xing [3] and Junye Li [1]

[1] College of Mechanical and Electrical Engineering, Xinjiang Agricultural University, Urumqi 830052, China; wangyunze_531@163.com (Y.W.); xupeng9018@163.com (P.X.); lijunyexjnydx@163.com (J.L.)
[2] Key Laboratory of Xinjiang Intelligent Agricultural Equipment, Xinjiang Agricultural University, Urumqi 830052, China
[3] College of Agriculture, Nanjing Agricultural University, Nanjing 210095, China; xingzhenyu@stu.njau.edu.cn
* Correspondence: zhangzhenguo@xjau.edu.cn; Tel.: +86-15099092586

Abstract: Accurate safflower recognition is a critical research challenge in the field of automated safflower harvesting. The growing environment of safflowers, including factors such as variable weather conditions in unstructured environments, shooting distances, and diverse morphological characteristics, presents significant difficulties for detection. To address these challenges and enable precise safflower target recognition in complex environments, this study proposes an improved safflower detection model, WED-YOLO, based on YOLOv8n. Firstly, the original bounding box loss function is replaced with the dynamic non-monotonic focusing mechanism Wise Intersection over Union (WIoU), which enhances the model's bounding box fitting ability and accelerates network convergence. Then, the upsampling module in the network's neck is substituted with the more efficient and versatile dynamic upsampling module, DySample, to improve the precision of feature map upsampling. Meanwhile, the EMA attention mechanism is integrated into the C2f module of the backbone network to strengthen the model's feature extraction capabilities. Finally, a small-target detection layer is incorporated into the detection head, enabling the model to focus on small safflower targets. The model is trained and validated using a custom-built safflower dataset. The experimental results demonstrate that the improved model achieves Precision (*P*), Recall (*R*), mean Average Precision (*mAP*), and F_1 score values of 93.15%, 86.71%, 95.03%, and 89.64%, respectively. These results represent improvements of 2.9%, 6.69%, 4.5%, and 6.22% over the baseline model. Compared with Faster R-CNN, YOLOv5, YOLOv7, and YOLOv10, the WED-YOLO achieved the highest *mAP* value. It outperforms the module mentioned by 13.06%, 4.85%, 4.86%, and 4.82%, respectively. The enhanced model exhibits superior precision and lower miss detection rates in safflower recognition tasks, providing a robust algorithmic foundation for the intelligent harvesting of safflowers.

Keywords: safflower filaments; object detection; improved YOLOv8 algorithm; deep learning

Academic Editor: Roberto Alves Braga Júnior

Received: 24 December 2024
Revised: 9 January 2025
Accepted: 16 January 2025
Published: 18 January 2025

Citation: Zhang, Z.; Wang, Y.; Xu, P.; Shi, R.; Xing, Z.; Li, J. WED-YOLO: A Detection Model for Safflower Under Complex Unstructured Environment. *Agriculture* **2025**, *15*, 205. https://doi.org/10.3390/agriculture15020205

1. Introduction

Safflower is an important specialty crop. The natural pigment safflower yellow, which is extracted from the safflower filaments, is widely used in medicine due to its high medicinal value [1,2]. China is one of the major producers of safflower [3]. The harvesting of safflowers is characterized by highly seasonal and labor-intensive. Currently, safflower harvesting primarily relies on manual labor, which is inefficient and costly. Furthermore, due to the delay in harvesting, some safflower filaments wilt, leading to a reduction in

yield. As such, the intelligent harvesting of safflower has become a key area of research. Safflowers bloom in multiple batches, with each batch going through flower opening and shedding periods [4,5]. The safflowers are small, numerous, compact, and vary in height, which makes it difficult to effectively extract the features of small safflower targets. Additionally, the complex background in unstructured environments introduces significant noise, further affecting the feature extraction process. This ultimately reduces the accuracy of safflower recognition and hampers the development of automated safflower harvesting. Therefore, there is an urgent need to develop algorithms that can effectively extract small safflower features in complex environments, minimize background noise, and enable accurate recognition across different flowering periods. Such algorithms could provide valuable references for the intelligent harvesting of safflowers.

Numerous deep learning techniques have been utilized for identifying and detecting crop flowers and fruits within challenging environments [6–8]. Object detection algorithms are categorized into two-stage and one-stage detection methods. Two-stage detection algorithms operate by first generating a series of candidate bounding boxes, which are then classified using convolutional neural networks. Farjon et al. [9] used Faster R-CNN [10] to detect apple blossoms, achieving a *mAP* value of 68%. Zhang et al. [11] improved the backbone feature extraction network, anchor box generation structure, and feature mapping mechanism of the Faster R-CNN network. The improved algorithm achieved a *mAP* of 91.49% for safflower detection. Bhattarai et al. [12] used the Mask R-CNN algorithm, for instance, to segment apple blossoms, conducting experiments during training to optimize hyperparameters of the deep learning network, achieving an average precision (*AP*) of 86%. Tian et al. [13] proposed a Mask Scoring R-CNN model with a U-Net backbone network for detecting and segmenting different forms of apple blossoms, including buds, semi-open, and fully open flowers.

In contrast, one-stage detection algorithms bypass the need for generating candidate boxes, formulating the task of object localization as a regression problem instead. This fundamental difference in design gives two-stage methods a slight edge in detection accuracy and localization precision, while one-stage methods excel in processing speed. However, with the continuous evolution of the YOLO series, one-stage detection algorithms have achieved significant improvements in accuracy while maintaining their advantage in speed, narrowing the performance gap with two-stage methods. Zhang et al. [14] proposed GSC-YOLOv3 based on YOLOv3 algorithm to detect safflower filaments, achieving a *mAP* of 91.89%. Guo et al. [15] introduced a safflower pistil detection and localization algorithm based on YOLOv5m, incorporating the CBAM attention mechanism, which led to an *AP* of 95.2%. Wang et al. [16] introduced a safflower recognition approach for complex environments, leveraging an enhanced YOLOv7 algorithm. The optimization involved integrating the Swin Transformer attention mechanism and refining the Focal Loss function, resulting in an *AP* of 88.5%, marking a 7% improvement compared to the original model. Wang et al. [17] presented a YOLOv5-based object detection algorithm for real-time detection of apple stems and sepals, achieving a *mAP* of 88%. Bhattarai et al. [18] proposed a deep regression-based network, AgRegNet, capable of estimating the density, quantity, and location of flowers and fruits in tree crowns. In an unstructured orchard environment, their model achieved a *mAP* of 81% for apple blossom detection. Dias et al. [19] proposed an improved end-to-end residual convolutional neural network to enhance color sensitivity. Using the Deeplab + RGR method, they achieved accurate recognition of flowers with both *Recall* and *Precision* rates exceeding 90%. Qi et al. [20] introduced a TC-YOLO algorithm that uses an improved backbone network to extract features and detect chrysanthemums in complex environments. Bai et al. [21] developed an enhanced YOLOv7 algorithm tailored for the recognition of flowers and fruits in greenhouse strawberry seedlings. They classified

the flowers into bud and flower periods, achieving a *mAP* value of 94.7% for flowers and 89.5% for fruits. Zhao et al. [22] proposed a tomato flower period recognition and detection method based on a cascade convolutional neural network, with a *Precision* of 76.67% in a glass greenhouse environment. While these studies have achieved promising results in flower and fruit recognition and detection, they often overlook the challenges posed by complex backgrounds, blurred images, and low-quality samples during actual harvesting. These factors result in fuzzy detection target features at varying distances, leading to lower detection accuracy. To address these challenges, this study collected safflower images under different weather conditions and shooting distances to create a dataset suitable for model training and evaluation. Based on the YOLOv8n model, we propose improvements and optimizations for safflower filament detection. The primary contributions of this study can be summarized as follows:

(1) Based on the significant variations in safflower size and shape as the shooting distance changes, the loss function was replaced with the WIoU, better suited for handling these variations.
(2) To retain more detailed safflower feature information during upsampling, the upsample module in the neck network was replaced with the DySample module, enhancing the model's upsampling capabilities.
(3) To better capture safflower features, the EMA module was integrated into the C2f module of the backbone network, improving the backbone's feature extraction ability for safflower targets.
(4) A small target detection layer was incorporated to improve the model's capability in extracting and identifying small safflower target features within the dataset, providing a precise detection method for safflower automation.

The remainder of this study is structured as follows: Section 2 provides an overview of the relevant materials and details the recognition and detection methods employed in this research; Section 3 outlines the experimental setup, presents the evaluation metrics, and reports the experimental results; finally, Section 4 offers a discussion of the findings, presents the conclusions, and highlights potential directions for future work.

2. Materials and Methods

2.1. Material

2.1.1. Acquisition of Image Data

Safflower is the experimental subject in this study. The safflower structure is composed of three primary parts arranged from top to bottom: the filament, the fruit ball, and the stem. It goes through two main flowering stages: the opening period and the shedding period. The safflower dataset was gathered from the safflower cultivation base located at 81°8′ E and 43°33′ N in Qapaqal Xibe Autonomous County, Yili, Xinjiang, China. The imaging devices include a Canon digital camera (E700D) (Canon Inc., Tokyo, Japan) and a Huawei P50 (Huawei Technologies Co., Ltd., Shenzhen, China). To ensure the versatility and operational range of the safflower intelligent harvesting machinery, the dataset was designed to be diverse. It includes safflower samples captured under different weather conditions and at varying distances. Additionally, each image contains multiple safflower plants at different flowering periods. The weather conditions during data collection were categorized as sunny, cloudy, and overcast. Since a safflower plant typically consists of multiple branches and has a certain spread, with height variations, the shooting distance was selected between 0.4 and 1 m to match the operational range of the equipment. The dataset was divided into two categories based on shooting distance: close (0.4~0.7 m) and far (0.7~1 m). This range ensures that the height variations of the safflower plants are kept within a manageable range, maximizing the visibility of all flowers in the frame. The

dataset includes images captured in various weather conditions, including sunny, cloudy, overcast, and at both close and far distances. This combination allows for flexibility in selective harvesting tasks. A total of 1126 RGB images were collected for model training and testing, with the categories and quantities detailed in Table 1. Some of the images are shown in Figure 1.

Table 1. Dataset and their quantities under different conditions.

Weather Condition	Image Quantities Under Different Shooting Distance	
	Close Distance (0.4~0.7 m)	Far Distance (0.7~1 m)
Sunny	204	169
Overcast	223	192
Cloudy	172	166

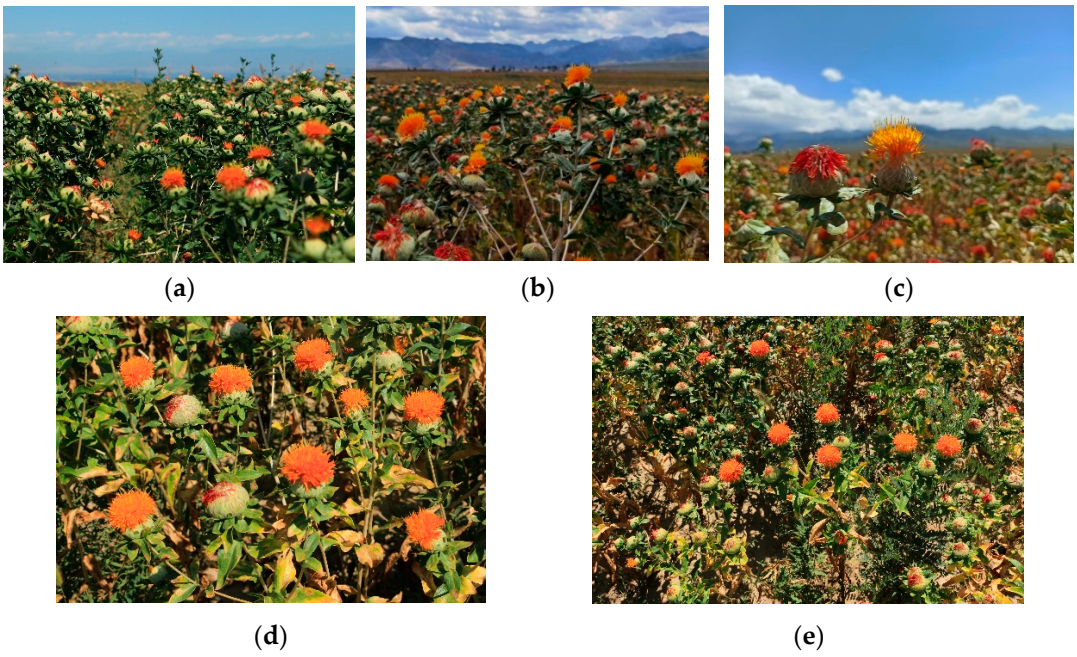


Figure 1. Example of safflower image dataset in complex environment: (**a**) Sunny, (**b**) overcast, (**c**) cloudy, (**d**) close distance (0.4~0.7 m), (**e**) far distance (0.7~1 m).

2.1.2. Augmentation and Processing of the Safflower Filaments Image Dataset

To mitigate overfitting during training, various data augmentation techniques were implemented on the original dataset. The applied techniques included rotations (90°, 180°, and 270°), noise injection, horizontal and vertical mirroring, as well as brightness modifications. Such augmentations effectively expanded the dataset, highlighting both global and local features of the safflower samples and increasing the contrast between different flowering stages. This approach contributes to enhancing the learning capacity of deep neural networks. Expanding the dataset's variability greatly enhanced the model's robustness and generalization capabilities. Examples of the augmented images are shown in Figure 2.

The dataset was captured under different weather conditions and at varying shooting distances to ensure its diversity. To guarantee the accuracy of the data parameters, the dataset was manually annotated using the LabelImg tool before model training. Annotations followed the YOLO format, with safflower in the blooming phase labeled as *Opening_period* and safflower in the wilting phase labeled as *Flower_shedding_period*. During annotation, the minimum enclosing rectangle around the filaments of the safflower was

used as the ground truth bounding box, minimizing the inclusion of irrelevant background pixels. Once annotated, the corresponding .txt annotation files were generated. After data augmentation, a total of 3940 safflower images were obtained. These were split into training, validation, and test sets at a ratio of 8:1:1. The final distribution consisted of 3152 images in the training set, 394 images in the validation set, and 394 images in the test set. The training and validation sets were used for model training and intermediate evaluation during each training cycle, while the test set was reserved for the final evaluation of the model's detection performance.

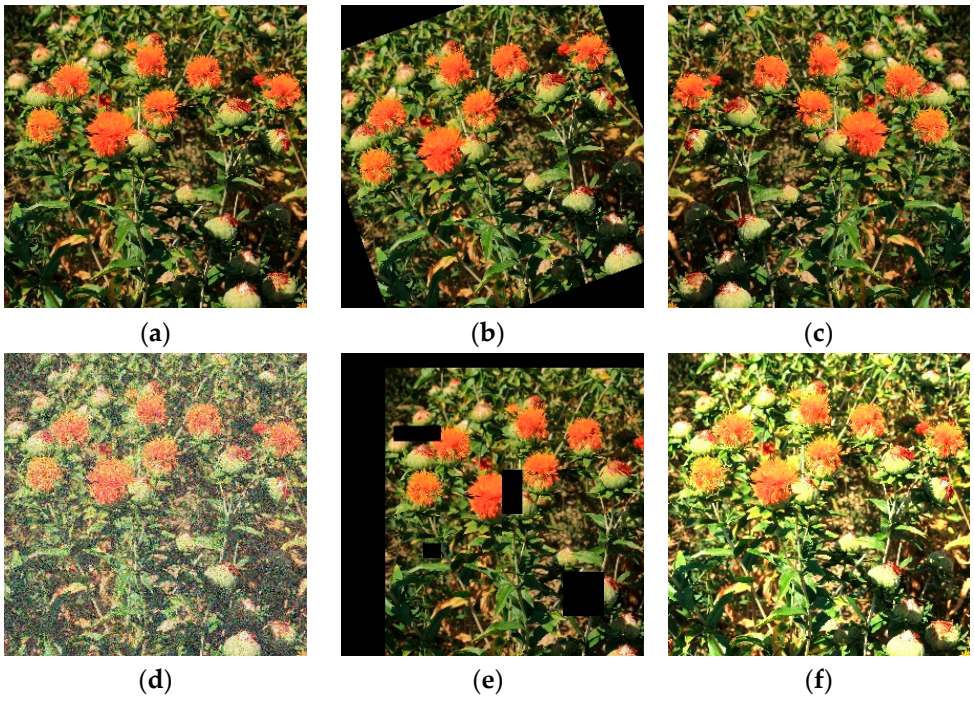


Figure 2. Data augmentation: (**a**) Original image, (**b**) rotation, (**c**) horizontal mirroring, (**d**) Gaussian noise, (**e**) random cropping, (**f**) adjusting brightness.

2.2. Methods

2.2.1. YOLOv8 Model

The You Only Look Once (YOLO) algorithm series comprises single-stage detection models that achieve an effective balance between detection speed and accuracy, particularly in detecting small objects. These models enable rapid classification and detection directly from images [23–25]. YOLOv8 is an anchor-free, single-stage object detection algorithm that has emerged as one of the leading models in the state-of-the-art (SOTA) category. The YOLOv8 architecture largely follows the YOLOv5 model while introducing new features and improvements to further enhance its performance and flexibility. In YOLOv8, the C2f structure replaces the C3 structure used in YOLOv5. The C2f module is designed by drawing inspiration from both the C3 module and the ELAN [26] concept. It divides the input feature map into two parts along one dimension, reducing computational complexity while enriching gradient flow, which improves the model's ability to represent non-linear features. Additionally, YOLOv8 incorporates a Decoupled-Head as the final detection head, which separates the classification and regression tasks, allowing for independent predictions. This approach reduces conflicts between the two tasks and introduces an anchor-free method, simplifying the object detection pipeline and improving model performance. The structural diagram of the model is shown in Figure 3.

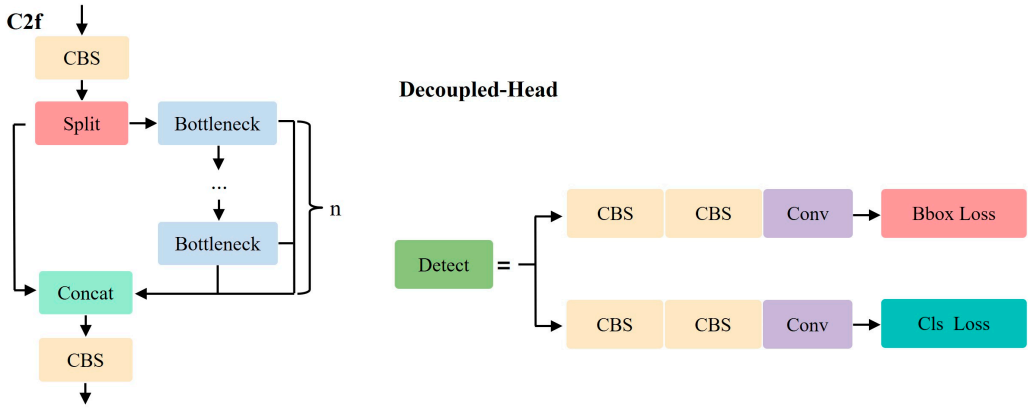


Figure 3. Partial diagram of the model architecture. (**a**) C2f module structure and (**b**) Decoupled-Head structure.

2.2.2. Improved YOLOv8 Model for Small Target Safflower Filaments Detection

Owing to the safflower filaments features small size and large number, accurately identifying safflower filament features, generating target bounding boxes, as well as identifying the different flowering periods of safflower filaments in natural environments are the research focuses of this paper. The main improvements include:

(1) Replace the CIoU of the original model with the WIoU loss function. WIoU assigns different weights to different size targets of safflower samples, which could improve the bounding box fitting ability and speed up the convergence of the network.
(2) In the neck network, the Dysample module is introduced to replace the Upsample module of the original model. Dysample could enhance the ability of upsampling for safflower targets in low-resolution images or far-field views and then reduce the loss of safflower features during the sampling process.
(3) Incorporate an efficient multi-scale attention mechanism in the C2f module in the backbone network to improve the extraction capability of the backbone network for safflower features.
(4) A small object prediction head with a size of 160 × 160 × 32 is added, intending to enhance the ability to detect small target safflower samples. The structure of the YOLOv8-WED model is shown in Figure 4.

2.2.3. WIoU Loss Function

At different distances within the field of view, the actual size and shape of the red flowers undergo significant changes. The clarity and quality of the images captured by the camera are also affected by the varying field-of-view sizes. Additionally, when the dataset contains a large number of low-clarity safflower samples, merely improving the loss function's ability to fit the bounding box does not significantly enhance its localization performance and may even degrade it. However, as the original bounding box regression loss function in YOLOv8, CIoU exclusively focuses on the fitting accuracy of the bounding box. When the predicted box and the ground truth box have the same aspect ratio but differ in width and height, CIoU may hinder further optimization of the model. To address this issue, this paper introduces Wise-IOU (WIoU) [27] to improve bounding box localization performance in safflower samples with varying levels of clarity across different fields of view. The WIoU loss function utilizes a dynamic, non-monotonic focusing mechanism that replaces IoU with outlier degree to assess anchor box quality. It also incorporates a gradient gain allocation strategy, which enhances the focus on high-quality anchor boxes while

mitigating the negative impact of low-quality samples in the dataset. The computation method is shown in Figure 5, and the formula for calculating the loss value is as follows:

$$L_{IOU} = 1 - \frac{W_i H_i}{S_u} \tag{1}$$

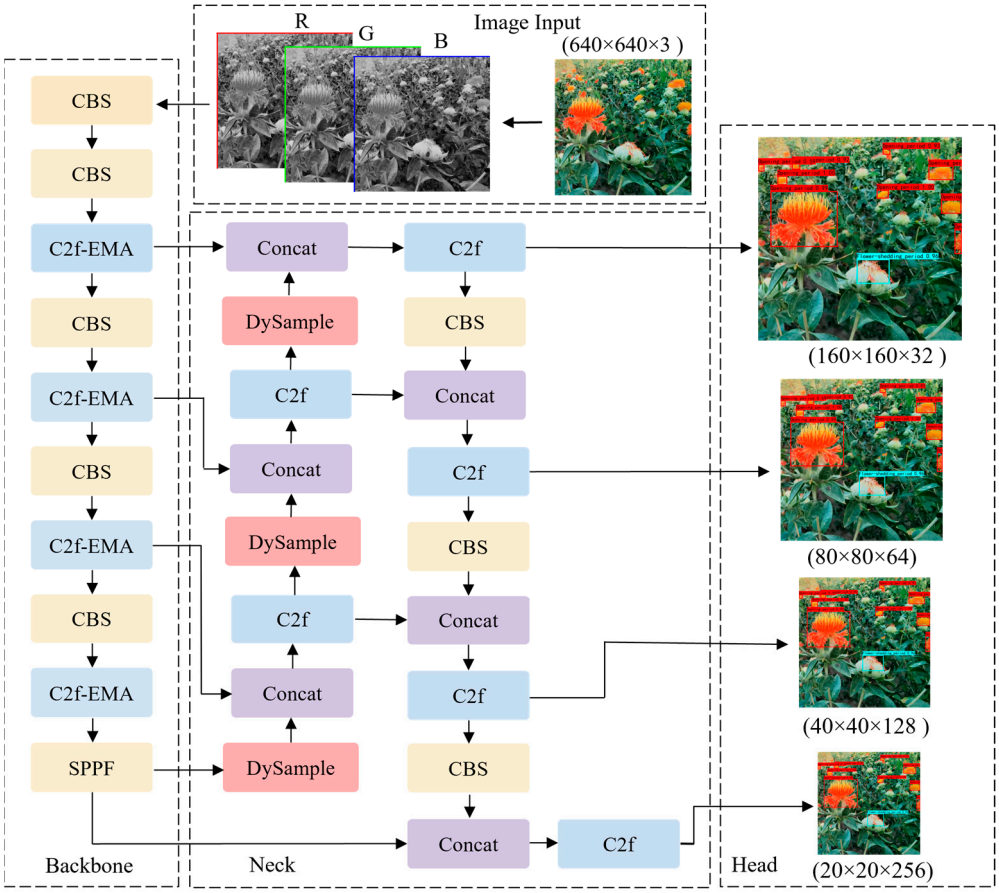


Figure 4. Network structure of the improved YOLOv8. C2f- EMA refers to the C2f module interacted the efficient multi-scale attention mechanism, Concat to the feature concatenation module, Dysample to the dynamic upsample mechanism.

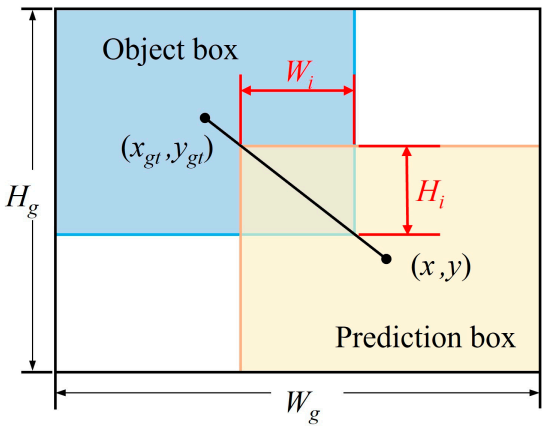


Figure 5. Illustration of the calculation method for the WIoU loss function. (x_{gt}, y_{gt}) represents the center point coordinates of the ground truth object bounding box; (x, y) represents the center point coordinates of the predicted bounding box; W_g represents the width of the minimum enclosing box for the predicted bounding box; H_g represents the height of the minimum enclosing box for the ground truth bounding box.

S_u represents the area of the intersection, and the calculation formula is as follows:

$$S_u = wh + w_{gt}h_{gt} - W_iH_i \tag{2}$$

where w and h represent the width and height of the predicted bounding box, respectively; w_{gt} and h_{gt} represent the width and height of the ground truth object bounding box; W_i and H_i represent the width and height of the intersection area between the ground truth and the predicted bounding boxes.

In the WIoU loss function, we adopted the more performant WIoUv3 [28], with its formula expressed as follows:

$$L_{WIoUv1} = R_{WIoU}L_{IoU} \tag{3}$$

$$R_{WIoU} = exp(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}) \tag{4}$$

When the predicted bounding box overlaps with the object bounding box, the LIoU loss weakens the penalty for geometric factors and shifts the focus to the center distance. The non-monotonic focus coefficient R_{WIoU} in Formula (4) helps balance gradient contributions, ensuring that low-quality samples do not adversely affect the training process. Furthermore, within the aggregation factor r, an outlier value β is introduced to assess the anomaly level of the anchor frame, expressed as:

$$\beta = \frac{L_{IoU}}{\overline{L_{IoU}}} \in [0, +\infty) \tag{5}$$

In Formula (5), $\overline{L_{IoU}}$ represents the moving average of momentum m, where m is typically set to $1 - \sqrt[tn]{0.05}$, t denotes the training epoch, and n represents the batch size. Consequently, focus factor r can be defined as:

$$L_{WIoUv3} = rL_{WIoUv1}, r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{6}$$

where α and δ are hyperparameters. When the outlier degree of an anchor frame reaches a certain constant, the anchor frame achieves the highest gradient gain. The dynamic characteristics of L_{IoU} also render the quality classification criterion β for anchor frames dynamic. This enables WIoUv3 to dynamically determine the most appropriate gradient gain allocation strategy for the given scenario, allowing the model to concentrate more on medium-quality anchor boxes of safflower samples, which in turn enhances overall detection performance. Additionally, the scale invariance provided by L_{WIoUv3} in Formula (6) ensures that the model consistently delivers stable performance across safflower samples of different sizes. This effectively addresses significant variations in safflower samples caused by distance and environmental changes in real harvesting scenarios. Consequently, the detection model becomes more robust and precise, especially under challenging weather conditions, thereby improving its generalization capability in complex real-world harvesting scenarios.

2.2.4. Dysample

In the real-world safflower image samples, the size of the safflowers varies with distance, and safflower samples in the background may experience pixel distortion. This distortion leads to the loss of fine-grained details in the filaments, making it difficult for the model to learn the true features of safflower samples during recognition tasks. YOLOv8 utilizes an upsampling layer, Upsample, in its neck feature enhancement network,

which employs the Nearest Neighbor interpolation algorithm. This method relies on the nearest neighboring pixels and fails to effectively capture subtle changes and dense semantic information in safflower features, resulting in feature loss. Additionally, this method incurs significant computational and parameter overhead. To address this issue, this paper introduced a dynamic point sampling-based Dysample [29] operator, which aims to improve the network's upsampling ability for safflower targets in low-resolution or distant scenes. This module reduces the loss of safflower features during the sampling process. Furthermore, it does not require additional CUDA packages, offering a significant reduction in computational and parameter overhead compared to the original module during operation.

The structure of DySample is shown in Figure 6. First, given an input feature map X with dimensions $C \times H \times W$, an upsampling factor s, and a static range factor of 0.25 to limit the sampling position's movement. A linear layer with specified input and output channel sizes of C and $2s^2$, respectively, is used to generate feature offsets of size $H \times W \times 2s^2$. These offsets are reshaped into the size $2g \times sH \times sW$ through a Pixel Shuffle operation to obtain the final dynamic offsets O. Finally, the offsets O are added to the original sampling grid G, resulting in the sampling set S. The set S is then used to perform a grid sampling operation with the original input X, producing an upsampled feature map X' of size $C \times sH \times sW$. This dynamic point sampling method effectively upsamples low-resolution feature maps to higher resolutions, which is crucial for safflower recognition. At higher resolutions, finer details of the safflower can be better preserved and presented, ultimately enhancing recognition accuracy.

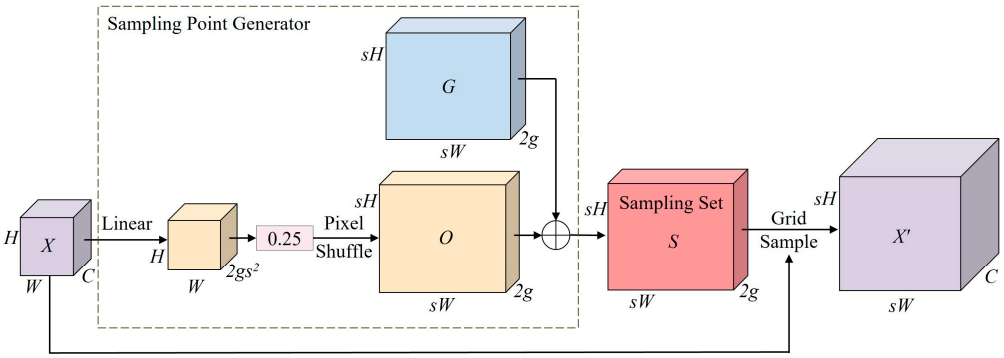


Figure 6. Structure of the Dysample. X, X', O, G denote the input feature map, upsampled feature map, generated offsets, and original grid, respectively.

2.2.5. C2f-EMA Module

The dataset contains safflowers at multiple scales; meanwhile, the background features numerous objects such as branches, leaves, and weeds, which are similar to the color of the safflower. These elements significantly interfere with safflower detection. Therefore, it is crucial to extract safflower features effectively from complex backgrounds for achieving accurate detection results. To address this challenge, this paper introduces an efficient multi-scale attention module (EMA) [30], which is integrated into the C2f module of the backbone network to form the C2f-EMA structure. This module reshapes certain channels to the batch dimension and groups the channel dimensions into multiple subfeatures. This allows it to retain key safflower features in each channel while reducing the interference caused by the complex background. As a result, the feature representation capability of the input image is enhanced, improving the overall safflower feature extraction performance.

The structure of the EMA module is shown in Figure 7. The input is first grouped and then processed through three parallel branches. The input is first grouped and then processed through three parallel branches. In the two 1×1 branches, one-dimensional

global average pooling is applied to encode the channels in both spatial directions. The channel-wise attention maps within each group are then combined using basic multiplication operations, facilitating cross-channel interactions between the two parallel paths. In the 3×3 branch, a single 3×3 convolution is stacked to expand the feature space, enabling the capture of multi-scale features. The two feature sets obtained from the different branches are processed through a cross-space learning module and a Sigmoid function, resulting in the final output feature map.

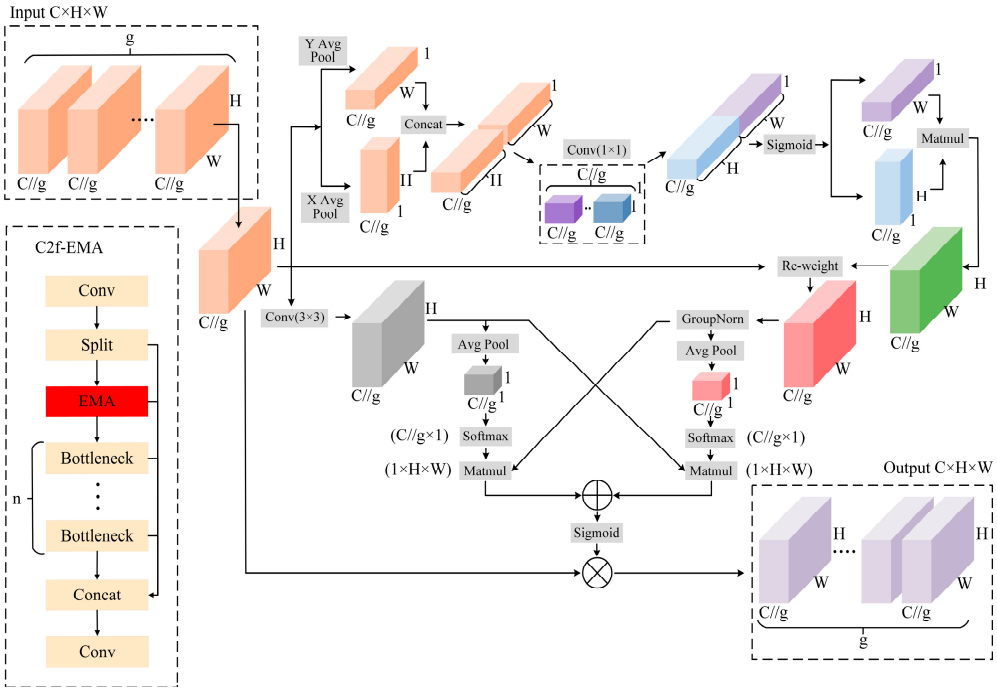


Figure 7. EMA module structure diagram. *g* represents the number of groups into which the input channels are divided. X Avg Pool and Y Avg Pool denote the one-dimensional global average pooling operations applied in the horizontal and vertical directions, respectively. Sigmoid is the activation function, while Softmax refers to the normalized exponential function.

2.2.6. Small-Target Detection Layer

In the process of recognizing small target safflower images, inputting excessively large images leads to scaling, which reduces the pixel coverage of small targets, thereby hindering the accurate recognition of safflowers. To address this challenge and enhance the precision of small target detection, the YOLOv8 network is optimized by incorporating an additional prediction head, which is specifically designed to process small targets. This modification enables the network to focus more effectively on safflower samples with smaller pixel sizes, thus improving the recognition accuracy of small targets [31]. Specifically, the network is enhanced by fusing the output feature maps from the second-layer C2f-EMA module and the dynamic upsampling module. This fusion integrates information across multiple network depths, effectively bridging high-level semantic features with low-level details. Such combination significantly improves the model's ability to accurately recognize small safflower targets. The fused feature maps are combined to generate a newly constructed high-resolution detection layer, measuring $160 \times 160 \times 32$. This layer is specifically tailored to improve the identification and detection capabilities for small safflower targets.

3. Results

3.1. Experimental Setup

3.1.1. Test Platform

The experiment was conducted on a system running Windows 11, with an Intel(R) Core (TM) i9-10700K CPU operating at 3.80 GHz (3.79 GHz base clock) and an NVIDIA GeForce GTX 4070 graphics card. The general parallel computing architecture utilized was CUDA 11.0, with the deep neural network acceleration library CUDNN V8.0.5.39. The deep learning framework employed was PyTorch 1.7.1, and the programming environment was PyCharm, using Python 3.9 as the programming language.

3.1.2. Evaluation Indicators

In this study, the metrics chosen to evaluate the performance of the safflower recognition model include the Precision rate (P), Recall rate (R), F_1 score, Average Precision (AP), and Mean Average Precision (mAP). AP is defined as the area under the precision-recall curve, where a higher AP value reflects superior model performance. The mAP is the mean value of AP across all classes, calculated at a given Intersection over Union (IoU) threshold. The formulas for the evaluation metrics are as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{7}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

$$F_1 = \frac{2PR}{P + R} \times 100\% \tag{9}$$

$$AP = \int_0^1 P(R)dR \tag{10}$$

$$mAP = \frac{\sum_{n-1}^{2} AP(n)}{2} \times 100\% \tag{11}$$

In the equation, TP denotes the number of correctly detected target objects, FP represents the number of falsely detected target objects, and FN refers to the number of target objects that were not detected. Precision (P) is defined as the ratio of correctly detected filaments to the total number of detected filaments, expressed as a percentage. Recall (R) is the ratio of correctly detected filaments bounding boxes to the total number of ground truth filaments bounding boxes, also expressed as a percentage. The F_1 score is the harmonic mean of Precision and Recall. AP represents the Average Precision of a specific class of safflower filaments, expressed as a percentage. Finally, mAP denotes the Mean Average Precision, which is the Average Precision across both the opening and flower-shedding period of safflower filaments, expressed as a percentage.

3.2. Experimental Result

3.2.1. Model Training

The training was conducted using the default hyperparameters of YOLOv8n. The initial image size was set to 640 × 640 pixels, with a batch size of 16. A learning rate of 0.01 was used, along with a weight decay factor of 0.0005 and a momentum factor of 0.937. The model was trained for a total of 1000 iterations, and the optimal results were analyzed. As shown in Figure 8, the loss function curve and mAP curve during the training process illustrate that as the number of iterations increases, the total loss decreases. As shown in Figure 8a, when the number of training iterations reaches 900, the loss curve begins to plateau and eventually stabilizes at a minimum value of 0.080. Simultaneously, the mAP

curve steadily increases with each training iteration and eventually reaches stability. As shown in Figure 8b, the *mAP* (*IoU* = 0.5) achieves a value of 95.03%. The trends of the curves throughout the training process effectively reflect the model's learning progress and the training's overall effectiveness.

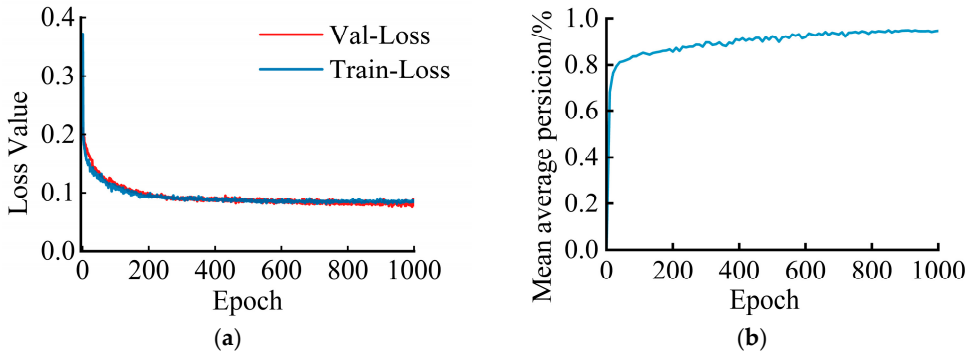


Figure 8. Training results of (**a**) loss curve and (**b**) *mAP*.

3.2.2. Ablation Experiment

To evaluate the contribution of the proposed improvements to the overall model performance, ablation experiments were conducted, and the results are presented in Table 1. As shown in the table, replacing the WIoU loss function led to improvements in Precision, Recall, and *mAP*, demonstrating that the WIoU loss function enhances the model's fitting ability and recognition accuracy. Additionally, when the upsampling module in the neck network was replaced with the dynamic upsampling module, the model achieved increases of 0.54%, 1.65%, 1.15%, and 0.77% in Precision, Recall, F_1 score, and *mAP*, respectively, compared to Test NO. 2. Meanwhile, the inference time of the images was reduced by 3 ms. This improvement is attributed to the DySample module. It does not require additional CUDA packages, which makes a difference from the traditional upsampling methods. As a result, it significantly reduces computational overhead and parameter costs, thereby enhancing overall computational efficiency. These results indicate that the DySample module effectively enhances the model's capability to detect low-resolution and small-target safflowers. Furthermore, when the C2F module in the backbone network was replaced with the C2f-EMA module, the model showed substantial improvements in Precision, Recall, F_1 score, and *mAP* over Test NO. 3. The improvements were 1.04%, 1.87%, 1.66%, and 1.72%, respectively. These findings suggest that incorporating the attention mechanism significantly increases the model's focus on small-target safflowers. Finally, the improved YOLOv8n model proposed in this study demonstrated slightly longer inference time compared to Test NO. 4 but achieved the best performance across all other evaluation metrics. It outperformed Test NO. 4 in terms of Precision, Recall, F_1 score, and mAP by 1.22%, 1.86%, 1.4%, and 1.85%, respectively. With an image inference time of 13.9 ms, it is capable of meeting the requirements of practical harvesting scenarios. This indicates that the newly introduced small-target detection layer effectively improves the model's ability to detect small safflower targets, thereby enhancing the overall detection performance.

4. Discussion

4.1. Comparison of Different Object Detection Methods

To further validate the advantages of the improved YOLOv8 algorithm in red flower recognition, a comparative experiment was conducted using Faster R-CNN, YOLOv5n, YOLOv7, YOLOv8n, and YOLOv10n object detection models. The experiments were performed using constructed red flower dataset for training, followed by evaluation on the same test set for the six detection networks. The experimental results are shown in Table 2.

Table 2. Ablation experiment.

Test NO.	Base Model	WIoU	DySample	C2f-EMA	S-T D Layer	Precision /%	Recall /%	F_1/%	*mAP*/%	Inference Time/ms
1	√	-	-	-	-	90.25	80.01	84.82	90.53	15.2
2	√	√	-	-	-	90.35	81.33	85.43	90.69	15.2
3	√	√	√	-	-	90.89	82.98	86.58	91.46	12.2
4	√	√	√	√	-	91.93	84.85	88.24	93.18	12.6
5	√	√	√	√	√	93.15	86.71	89.64	95.03	13.9

Note: In the table, the "Base Model" represents the original YOLOv8n model. S-T D Layer represents the small-target detection layer. A "√" indicates that the module has been added, while a "-" indicates that the module has not been added.

As shown in Table 3, the two-stage detection algorithm Faster R-CNN exhibits the lowest detection accuracy, the slowest inference speed, and the largest model size, making it unsuitable for small safflower target detection. Among the one-stage detection algorithms, YOLOv5n, YOLOv7, YOLOv8n, and YOLOv10 show similar performance in terms of *Precision*, *Recall*, *mAP*, and F_1 score. However, YOLOv7 performs poorly in inference speed and model size, with a model size of 74.8 MB, significantly larger than the other three algorithms. The improved YOLOv8n model demonstrates the best performance in *Precision*, *Recall*, *mAP*, and F_1 score. The reason lies in the fact that the filaments occupy only a small number of pixels in the images, leading to a decline in the feature quality of safflower filaments. This makes it challenging for other algorithms with weaker small-object detection capabilities to achieve satisfactory results. In contrast, the WED-YOLO algorithm significantly enhances small-object detection performance and the ability to extract feature information by the improvement mentioned in Section 2.2.2, with its *mAP* value substantially surpassing the other algorithms. Specifically, its *Precision* exceeds that of Faster R-CNN, YOLOv5n, YOLOv7n, YOLOv8n, and YOLOv10 by 14.19%, 3.23%, 3.26%, 2.9%, and 3.17%, respectively; *Recall* exceeds by 4.38%, 9.84%, 6.76%, 6.69%, and 6.66%; *mAP* exceeds by 13.06%, 4.85%, 4.86%, 4.5%, and 4.82%; F_1 score exceeds by 13.14%, 7.14%, 6.31%, 4.82%, and 6.22%, respectively. The inference speed of the improved YOLOv8n is 72 frames per second (FPS), which is slightly lower than the speed YOLOv10n but has a higher *mAP* value. Among these models, the proposed algorithm demonstrates the highest *mAP* in small safflower detection, with a moderate model weight, making it suitable for precise detection of safflowers during both the flower-opening and flower-shedding periods.

Table 3. Comparison of experimental results of different models.

Model	*P*/%	*R*/%	F_1/%	*AP*/%		mAP/ %	FPS/ Frame·s^{-1}	Model Size/MB
				Opening Period	Flower-Shedding Period			
Faster R-CNN	78.96	82.32	76.50	82.92	77.34	81.97	23	322
YOLOv5n	89.92	76.86	82.50	90.17	90.19	90.18	69	4.1
YOLOv7	89.89	79.94	83.33	90.11	90.23	90.17	47	74.8
YOLOv8n	90.25	80.01	84.82	90.40	90.66	90.53	66	6.2
YOLOv10n	89.98	80.04	83.42	90.13	90.28	90.21	80	3.5
YOLOv8-WED	93.15	86.7	89.64	95.25	94.82	95.03	72	6.4

To evaluate the detection performance under different weather conditions, we selected one image from each weather condition—sunny, cloudy, and overcast—taken at different distances (far and close). The detection results of the six models are shown. The red detection box labeled "Flower-opening period" indicates the detection of safflowers in the blooming period, while the blue detection box labeled "Flower-shedding period" indicates safflowers in the shedding period. The numbers following the labels represent the detection confidence values. As shown in Figure 9, due to the two-stage detection algorithm's candidate region generation network being highly dependent on large-scale feature maps

to generate candidate boxes, Faster R-CNN struggles to capture multi-scale features in the scene, leading to poor detection of small safflower targets. This results in significant misdetection, with many small safflower targets not being correctly generated or recognized by the candidate region network. YOLOv5n, YOLOv7, YOLOv8n, and YOLOv10 exhibit similar detection performance, with only a few missed detections. For instance, in overcast images, there are some distant safflower samples, making safflower features with a low quality in the images. When processing these low-quality safflower images, YOLOv5n and YOLOv7 detected 9 safflowers in the flower-openning and 12 safflowers in the flower-shedding period. Meanwhile, YOLOv8n and YOLOv10n detect 8 safflowers in the flower-openning and 15 safflowers in the flower-shedding period. All these models outperformed Faster R-CNN and exhibited commendable performance. However, some safflowers with low-quality features were still misidentified by all models. Only the improved YOLOv8n model achieved the best detection performance. In overcast images, it detected 9 safflowers in the flower-opening period and 16 safflowers in the flower-shedding period, outperforming all other models. Similarly, in cloudy images, there were also some safflower samples with blurry outer contours and low-quality features. All models missed detections in these cases, but the improved YOLOv8n model performed the best. This superior performance can be attributed to the improved model's enhanced ability to extract features of small safflower targets, enabling it to effectively recognize safflowers in low-quality images.

4.2. Comparison of Different Loss Functions

The original YOLOv8 object detection model employs CIoU as its bounding box regression loss function. During training, this loss function exhibits strong fitting ability; however, since the aspect ratio of the predicted box is a relative value, there is inherent uncertainty in the calculation. Additionally, both high-quality and low-quality mispredictions negatively affect the regression loss. This study examines the convergence behavior of the improved YOLOv8 model with different loss functions, including WIoU, CIoU, EIoU [32], and SIoU [33]. The experimental results are illustrated in the loss comparison chart.

As shown in Figure 10, when using EIoU, the convergence speed is the slowest, and the final loss value is the highest. The convergence speed of SIoU and CIoU is slightly faster than that of EIoU, with final loss values marginally lower than EIoU. However, when using WIoU, the model exhibits the fastest gradient descent during training, converging rapidly within the first 300 epochs and achieving a final loss value significantly lower than that of the other three loss functions. Additionally, as indicated in Table 1 of the ablation study, the use of WIoU results in improvements across various evaluation metrics, including *Precision*, *Recall*, and *mAP*.

4.3. Visualization of Heatmap

To visually showcase the optimization effect of the proposed WED-YOLOv8 model on the Red Flower dataset, the Gradient-weighted Class Activation Mapping (Grad-CAM) [34] algorithm is utilized for visualization. In the heatmap, different colors indicate the importance of each position in the image for the red flower detection task. Darker colors indicate regions where the model focuses more attention. As shown in Figure 11, YOLOv8n is susceptible to noise from the background area during detection, leading the model to focus on irrelevant background regions. In contrast, the improved model significantly reduces attention to noise from the background, allowing it to accurately focus on the red flower region. This demonstrates that the improved model is better at extracting red flower features in complex environments, thereby enhancing the model's accuracy in recognition and detection tasks in challenging background settings.

Figure 9. Detection effects of different models on safflower images at different weather and shooting distance: (**a**) Sunny, (**b**) overcast, (**c**) cloudy, (**d**) far distance, and (**e**) close distance. The red boxes indicate the filaments in the opening period, the blue boxes indicate the filaments in the shedding period, and the pink boxes indicate the filaments that were missed during detection.

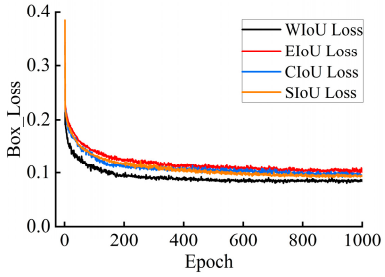
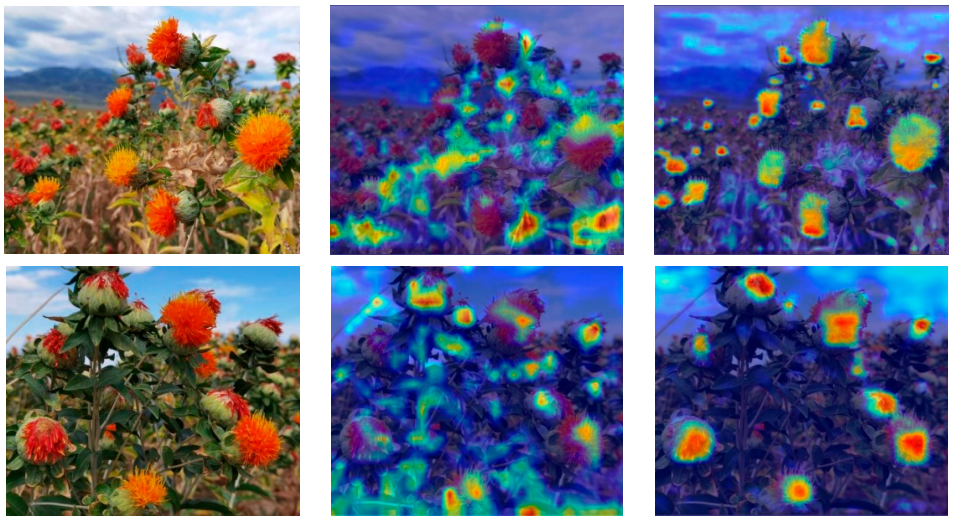


Figure 10. Convergence of different bounding box loss functions. CIoU Loss indicates that the model uses CIoU as the bounding box loss function. EIoU Loss indicates that the model uses EIoU as the bounding box loss function. SIoU Loss indicates that the model uses SIoU as the bounding box loss function. WIoU Loss indicates that the model uses WIoU as the bounding box loss function.



(**a**) (**b**) (**c**)

Figure 11. Heatmap visualization results. (**a**) Original image, (**b**) YOLOv8, and (**c**) WED-YOLO. The colors represent a scalar of one order of magnitude, with warm colors (e.g., red or yellow) indicating high activity or importance regions and cool colors (e.g., blue or green) indicating low activity or importance regions.

5. Conclusions

To achieve precise safflower recognition in complex and unstructured environments, a novel detection methodology, WED-YOLO, has been proposed. First, safflower samples from both the opening and flower-shedding periods were selected as detection targets to create a safflower dataset. A series of robust data augmentation techniques, including random occlusion and brightness enhancement, were implemented to enhance the dataset's diversity. Next, the original CIoU loss function of the model was replaced with WIoU, mitigating the adverse effects of low-quality samples in the safflower dataset, enhancing the model's fitting performance, and accelerating network convergence. Subsequently, the upsampling mechanism in the neck network, UpSample, was replaced with the dynamic upsampling mechanism, DySample, which further enhances the model's detection capability for small safflower targets in low-resolution images or distant views, improving the model's learning ability and robustness. Building on this, the EMA attention mechanism was integrated into the C2f module of the backbone network, strengthening the model's feature extraction ability for safflowers. Finally, a small target detection layer was added to enable the network to focus more on smaller safflower samples, thereby improving the accuracy of small target safflower recognition. When compared to other detection algorithms, including Faster R-CNN, YOLOv5n, YOLOv7, YOLOv8n, and YOLOv10n, the proposed WED-YOLO model demonstrated superior performance, achieving the highest *mAP* value (*mAP* = 95.03%). It outperforms the module mentioned by 13.06%, 4.85%, 4.86%, 4.5%, and 4.82%, respectively. The performance is also more balanced in terms of detection accuracy and detection speed, with a moderate model size. Experimental results demonstrate that the WED-YOLO model is effective for safflower recognition in complex, unstructured environments. The precision meets the requirements of safflower-harvesting robots, providing an algorithmic reference for the intelligent harvesting of safflowers.

In future studies, it will be essential to gather a wider variety of safflower images to enhance the dataset's diversity. At the same time, the YOLOv8 model will undergo further improvements and optimizations to adapt more effectively to detection tasks in diverse scenarios, facilitating the extension of the YOLOv8-WED model. Moreover, if applied to real-time tasks such as dynamic safflower counting, the computational efficiency of the

approach should be further refined. These challenges will remain the focus of ongoing exploration and investigation in future research efforts.

Author Contributions: Conceptualization, Z.Z., Y.W. and P.X.; methodology, Y.W. and P.X.; software, Y.W.; validation, P.X. and R.S.; formal analysis, Z.Z.; investigation, Y.W.; resources, Z.X.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, Z.Z., Y.W. and J.L.; visualization, R.S.; supervision, Z.Z.; project administration, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Central Guidance for Local Science and Technology Development Funding Projects under Grant No.ZYYD2025ZY11, and the National Natural Science Foundation of China under Grant 32460449 and 52265041. The authors also acknowledge the Key Laboratory of Xinjiang Intelligent Agricultural Equipment, China, for their assistance in conducting field experiments.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Kassa, B.A.; Mekbib, F.; Assefa, K. Effects of plant hormones and genotypes on anther culture response of safflower (*Carthamus tinctorius* L.). *Sci. Afr.* **2024**, *26*, e02367. [CrossRef]
2. Chen, Y.; Li, M.; Wen, J.; Pan, X.; Deng, Z.; Chen, J.; Chen, G.; Yu, L.; Tang, Y.; Li, G.; et al. Pharmacological activities of safflower yellow and its clinical applications. *Evid.-Based Complement. Altern. Med.* **2022**, *2022*, 2108557. [CrossRef] [PubMed]
3. Gongal, A.; Amatya, S.; Karkee, M.; Zhang, Q.; Lewis, K. Sensors and systems for fruit detection and localization: A review. *Comput. Electron. Agric.* **2015**, *116*, 8–19. [CrossRef]
4. Zhang, Z.; Guo, J.; Yang, S.; Zhang, X.; Niu, Q.; Han, C.; Lv, Q. Feasibility of high-precision numerical simulation technology for improving the harvesting mechanization level of safflower filaments: A review. *Int. Agric. Eng. J.* **2020**, *29*, 139–150.
5. Zhang, Z.; Xing, Z.; Yang, S.; Feng, N.; Liang, R.; Zhao, M. Design and experiments of the circular arc progressive type harvester for the safflower filaments. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 10–21. [CrossRef]
6. Fan, X.; Zhou, J.; Xu, Y.; Li, K.; Wen, D. Identification and localization of weeds based on optimized faster R-CNN in cotton seedling stage. *Trans. Chin. Soc. Agric. Mach.* **2021**, *52*, 26–34. [CrossRef]
7. Lin, G.; Tang, Y.; Zou, X.; Li, J.; Xiong, J. In-field citrus detection and localisation based on RGB-D image analysis. *Biosyst. Eng.* **2019**, *186*, 34–44. [CrossRef]
8. Deng, X.; Zhou, B.; Hou, Y. A reliability test method for agricultural paddy field intelligent robot. *NMATEH-Agric. Eng.* **2021**, *3*, 271–280. [CrossRef]
9. Farjon, G.; Krikeb, O.; Hillel, A.B.; Alchanatis, V. Detection and counting of flowers on apple trees for better chemical thinning decisions. *Precis Agric.* **2020**, *21*, 503–521. [CrossRef]
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
11. Zhang, Z.; Shi, R.; Xing, Z.; Guo, Q.; Zeng, C. Improved Faster Region-Based Convolutional Neural Networks (R-CNN) model based on split attention for the detection of safflower filaments in natural environments. *Agronomy* **2023**, *13*, 2596. [CrossRef]
12. Bhattarai, U.; Bhusal, S.; Majeed, Y.; Karkee, M. Automatic blossom detection in apple trees using deep learning. *IFAC-PapersOnLine* **2020**, *53*, 15810–15815. [CrossRef]
13. Tian, Y.; Yang, G.; Wang, Z.; Li, E.; Liang, Z. Instance segmentation of apple flowers using the improved mask R–CNN model. *Biosyst. Eng.* **2020**, *193*, 264–278. [CrossRef]
14. Zhang, Z.; Xing, Z.; Zhao, M.; Yang, S.; Guo, Q.; Shi, R.; Zeng, C. Detecting safflower filaments using an improved YOLOv3 under complex environments. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 162–170. [CrossRef]
15. Guo, H.; Chen, H.; Gao, G.; Zhou, W.; Wu, T.; Qiu, Z. Safflower corolla object detection and spatial positioning methods based onYOLO v5m. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 272–281. [CrossRef]
16. WANG, X.; XU, Y.; ZHOU, J.; CHEN, J. Safflower picking recognition in complex environments based on an improved YOLOv7. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 169–176. [CrossRef]

17. Wang, Z.; Jin, L.; Wang, S.; Xu, H. Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. *Postharvest Biol. Technol.* **2022**, *185*, 111808. [CrossRef]
18. Bhattarai, U.; Bhusal, S.; Zhang, Q.; Karkee, M. AgRegNet: A deep regression network for flower and fruit density estimation, localization, and counting in orchards. *Comput. Electron. Agric.* **2024**, *227*, 109534. [CrossRef]
19. Dias, P.A.; Tabb, A.; Medeiros, H. Multispecies fruit flower detection using a refined semantic segmentation network. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3003–3010. [CrossRef]
20. Qi, C.; Gao, J.; Pearson, S.; Harman, H.; Chen, K.; Shu, L. Tea chrysanthemum detection under unstructured environments using the TC-YOLO model. *Expert Syst. Appl.* **2022**, *193*, 116473. [CrossRef]
21. Bai, Y.; Yu, J.; Yang, S.; Ning, J. An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosyst. Eng.* **2024**, *237*, 1–12. [CrossRef]
22. Zhao, C.; Wen, C.; Lin, S.; Guo, W.; Long, J. Tomato florescence recognition and detection method based on cascaded neural network. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 143–152. [CrossRef]
23. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-adaptive YOLO for object detection in adverse weather conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; Volume 36, pp. 1792–1800. [CrossRef]
24. Jiang, F.; Zhang, H.; Feng, C.; Zhu, C. A Closed-Loop Detection Algorithm for Indoor Simultaneous Localization and Mapping Based on You Only Look Once v3. *Traitement du Signal* **2022**, *39*, 109–117. [CrossRef]
25. Rahim, U.F.; Mineno, H. Data augmentation method for strawberry flower detection in non-structured environment using convolutional object detection networks. *J. Agric. Crop Res.* **2020**, *8*, 260–271. [CrossRef]
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. [CrossRef]
27. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051. [CrossRef]
28. Xu, X.; Zhang, G.; Zheng, W.; Zhao, A.; Zhong, Y.; Wang, H. High-precision detection algorithm for metal workpiece defects based on deep learning. *Machines* **2023**, *11*, 834. [CrossRef]
29. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 6027–6037.
30. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5. [CrossRef]
31. Pan, Y.; Xiao, X.; Hu, K.; Kang, H.; Jin, Y.; Chen, Y.; Zou, X. Odn-pro: An improved model based on yolov8 for enhanced instance detection in orchard point clouds. *Agronomy* **2024**, *14*, 697. [CrossRef]
32. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
33. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740. [CrossRef]
34. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [CrossRef]