agronomy

MDPI

Article

Light-FC-YOLO: A Lightweight Method for Flower Counting Based on Enhanced Feature Fusion with a New Efficient Detection Head

Xiaomei Yi [1,†], Hanyu Chen [1,†], Peng Wu [1,*], Guoying Wang [1], Lufeng Mo [1], Bowei Wu [1], Yutong Yi [2], Xinyun Fu [1] and Pengxiang Qian [1]

[1] College of Mathematics and Computer Science, Zhejiang A & F University, Hangzhou 311300, China; yxm@zafu.edu.cn (X.Y.); 2021611011075@stu.zafu.edu.cn (H.C.); molufeng@zafu.edu.cn (L.M.); wbw@stu.zafu.edu.cn (B.W.); fxy111@stu.zafu.edu.cn (X.F.); 202224070120@stu.zafu.edu.cn (P.Q.)

[2] College of Humanities and Development Studies, China Agricultural University, Beijing 100091, China; 2020317010220@cau.edu.cn

[*] Correspondence: wp@zafu.edu.cn

[†] These authors contributed equally to this work.

Abstract: Fast and accurate counting and positioning of flowers is the foundation of automated flower cultivation production. However, it remains a challenge to complete the counting and positioning of high-density flowers against a complex background. Therefore, this paper proposes a lightweight flower counting and positioning model, Light-FC-YOLO, based on YOLOv8s. By integrating lightweight convolution, the model is more portable and deployable. At the same time, a new efficient detection head, Efficient head, and the integration of the LSKA large kernel attention mechanism are proposed to enhance the model's feature detail extraction capability and change the weight ratio of the shallow edge and key point information in the network. Finally, the SIoU loss function with target angle deviation calculation is introduced to improve the model's detection accuracy and target positioning ability. Experimental results show that Light-FC-YOLO, with a model size reduction of 27.2% and a parameter reduction of 39.0%, has a Mean Average Precision (mAP) and recall that are 0.8% and 1.4% higher than YOLOv8s, respectively. In the counting comparison experiment, the coefficient of determination (R^2) and Root Mean Squared Error (RMSE) of Light-FC-YOLO reached 0.9577 and 8.69, respectively, both superior to lightweight models such as YOLOv8s. The lightweight flower detection method proposed in this paper can efficiently complete flower positioning and counting tasks, providing technical support and reference solutions for automated flower production management.

Keywords: deep learning; target detection; multi-objective flower counting; yield estimation; lightweighting

check for updates

Citation: Yi, X.; Chen, H.; Wu, P.; Wang, G.; Mo, L.; Wu, B.; Yi, Y.; Fu, X.; Qian, P. Light-FC-YOLO: A Lightweight Method for Flower Counting Based on Enhanced Feature Fusion with a New Efficient Detection Head. *Agronomy* **2024**, *14*, 1285. https://doi.org/10.3390/agronomy14061285

Academic Editor: Gniewko Niedbała

Received: 15 May 2024
Revised: 10 June 2024
Accepted: 13 June 2024
Published: 14 June 2024

1. Introduction

Flowers, as crops with ornamental, edible, and medicinal value, are widely loved by the public [1]. These values make flowers one of the most important economic crops in the world. According to statistics from the China Flower Association [2], China has become the largest flower production base in the world, and the market size of China's flower industry reached a retail scale of CNY 229.1 billion in 2022. To meet the demand of the flower market, the scale of flower cultivation is gradually expanding. Therefore, the traditional manual cultivation management mode can no longer meet the production needs of large-scale flower cultivation bases. In actual horticultural cultivation, real-time monitoring of the specific flowering conditions of flowers in the nursery, intelligent quantity statistics, and positioning can better obtain information on flower yield, distribution, and growth conditions, and thereby lead to taking corresponding management measures to improve agricultural planting quality and production efficiency [3].

In recent years, with the improvement of agricultural machinery technology and the promotion of agricultural automation operations, target detection has gradually become a focus in crop counting [4]. In the field of flower counting, target detection methods based on machine learning have begun to be applied to flower counting research. Prabira Kumar Sethy et al. [5] used the transformation of HSV color blocks and the Circular Hough Transform (CHT) method to accurately locate and count the flowers of marigolds. Chao Li et al. [6] applied SVM to the segmentation of lily cut flower images, and in response to the problems of flower bud adhesion and leaf occlusion, they adopted the method of ellipse fitting to more accurately locate the lily buds. Although traditional target detection techniques based on machine learning can complete detection and counting tasks, they have poor generalization capabilities in the face of more complex detection environments and cannot meet the needs of multi-class variety recognition and other integrated functions.

With the significant improvement in computer computational performance, target detection algorithms based on deep learning, characterized by high generalization and robustness, have gradually replaced traditional target detection algorithms and are widely used in the field of detection counting [7]. Li Sun et al. [8] proposed an improved peach blossom counting model based on YOLOv5s [9], adding a combination of a CAM [10] module and FSM [11] module to enhance the model's ability to locate small targets, and introduced K-means++ [12] to regenerate suitable candidate box sizes. P. Lin et al. [13] proposed an automatic strawberry flower detection system in the field for outdoor strawberry yield estimation, using Faster R-CNN [14] to detect strawberry flowers in the field, and adopted an improved VGG19 [15] structure for extracting multi-scale features of strawberry flower images. Daniel Petti et al. [16] used a weakly supervised method based on a CNN network to automatically complete the counting task of cotton flowers on images collected by drones, and adopted the Multi-Instance Learning (MIL) [17] method to train the model, improving the model's processing performance and recognition accuracy. Although deep learning methods have achieved higher detection accuracy and efficiency in flower counting and positioning than traditional image processing algorithms, their deeper networks bring higher computational costs and network scales, which are not conducive to their deployment on mobile and embedded devices. Therefore, lightweight detection algorithms are needed, which are conducive to the deployment of algorithms on devices for practical flower counting and positioning.

In practical applications, due to the deployment needs of detection algorithms on mobile and embedded devices, the development of lightweight and high-precision detection networks has gradually become a research focus. Niraj Tamrakar et al. [18] proposed a lightweight strawberry detection and counting algorithm YOLOv5s-CGhostnet based on YOLOv5s. By combining the Ghost module [19] with CBS and C3 modules, the model size and computation are significantly reduced, and the CBAM [20] attention mechanism is introduced to enhance the model's ability to extract strawberry features. Li Shuo et al. [21] in response to the slow recognition speed of high-density bayberries under complex backgrounds, designed a lightweight bayberry counting model YOLOv7-CS based on YOLOv7 [22]. They proposed the CNxP module to replace the E-Elan module in the backbone, achieving model lightweight while improving the model's detection accuracy and positioning ability. In combination with the Wise-IoU loss function [23], the model's ability to recognize occluded objects is enhanced. Jie Chen et al. [24] used FasterNet [25] as the basic feature extraction network and designed a lightweight wheat counting model, Wheat-FasterYOLO, significantly reducing the model's parameter quantity. They introduced deformable convolution [26] and a dynamic sparse attention mechanism [27] in the network, enhancing the model's ability to extract wheat features and improving the accuracy of wheat ear counting. The YOLO single-stage algorithm, due to its fast positioning, high precision, and small size, is widely used in crop counting.

Existing lightweight YOLO deep neural networks have shown good performance in the field of multi-object counting. However, these studies mainly focus on the detection of flowers and fruits of crops, primarily applied to crop yield prediction. Guy Farjon et al. [28] have constructed an apple flower detection system based on Faster-RCNN, which counts the number of open apple flowers, but there is still room for improvement in its detection accuracy. Yifan Bai et al. [29] have improved the YOLOv7 network to count strawberry flowers and fruits separately, but the targets in their detection images are relatively scattered, and the target features are significant. Due to the high-density growth of flowers in the natural environment, there are various factors such as mutual occlusion, leaf occlusion, and a large proportion of background area, which cause a certain degree of detection difficulty for the model. Therefore, this paper proposes a lightweight model for the accurate detection and counting of flowers in actual environments and selects five representative common flowers to explore a new lightweight multi-target flower counting method under complex backgrounds. The main contributions of this paper are as follows:

(1) A method proposed for accurately counting high-density flowers in complex backgrounds.

(2) The integration of the C2f module with the Ghost module has resulted in a reduction in both the parameter and the size of the model. This combination has effectively streamlined the model, making it more efficient for practical applications.

(3) A new efficient detection head has been proposed, which enhances the model's ability to express complex functions and improves the feature extraction capabilities for the target. This advancement contributes to the overall performance and accuracy of the model.

(4) The introduction of the LSKA attention mechanism in the feature extraction module has amplified the role of shallow shape encoding information of the target within the network. This enhancement facilitates the fusion of spatial information across different scales, thereby improving the model's adaptability and performance.

(5) The incorporation of the SIoU loss function has enhanced the detection performance of the model and accelerated the convergence speed during training. This improvement has made the model more efficient and effective in its operations.

2. Materials and Methods

2.1. Flower Datasets

2.1.1. Data Acquisition

To enhance the robustness of the model, this study utilized three datasets throughout the entire experimental process. Two public datasets were respectively sourced from the flower recognition dataset [30] provided by the Kaggle website and the Oxford 102 flower dataset [31], provided by the University of Oxford. After image filtering, the flower recognition dataset ultimately yielded 2701 flower images, encompassing a total of 7653 flower targets. Similarly, after image filtering, the Oxford 102 flower dataset resulted in 517 usable images, containing 1464 flower targets.

The custom dataset used in this study, the East Lake Flower Dataset, was collected from the East Lake Campus of Zhejiang A&F University in Lin'an District, Hangzhou City, Zhejiang Province, China. The university is located in the western part of Hangzhou, with geographical coordinates ranging from 118°51′ E to 119°52′ E and 29°56′ N to 30°23′ N. Flower image collection was conducted in March and June 2022. Due to the quality of image collection being affected by the intensity of light, the field collection tasks were scheduled between 9:00 and 11:00 and 14:00 and 16:00. In different image collection areas, collection points were randomly selected for image collection work. The image collection device for the East Lake Flower Dataset was an iPhone 13, with a main camera parameter of 12 million pixels, an aperture of f/1.6, and shooting angles mostly forward horizontal, with a slight downward tilt of 20–30°. The resolution of the collected images was 4032 × 3024 pixels, and a total of 432 original flower images were collected from different collection points, containing 1224 flower targets.

The flower density of the aforementioned three datasets varies significantly. Therefore, this paper categorizes them into different scenarios based on flower density to enhance the generalization capability of the model. As shown in Figure 1, according to the density of flowers, they are divided into three density levels: low-density, medium-density, and high-density. The division of flower density levels mainly refers to the severity of the obstruction between flowers. In the flower images, if there are only a few targets and no stacking or adhesion, it is classified as low-density. If there are more than 10 targets, and only some stacking and adhesion occur, and the area of obstruction between flowers is 10–30%, it is classified as medium-density; if there are many targets, and the stacking and adhesion between targets are severe, and the area of obstruction between flowers is more than 30%, it is classified as high-density. These images clearly show that compared with the scene of low-density flowers, the counting task in dense scenes is more challenging. The reasons include the adhesion between flowers, the obstruction of branches and leaves, the complexity of the background, and the scale changes of different targets, etc. These factors will affect the model's ability to extract deep flower features, thereby reducing the recognition accuracy of the model. All details of the datasets can be seen in Table 1.

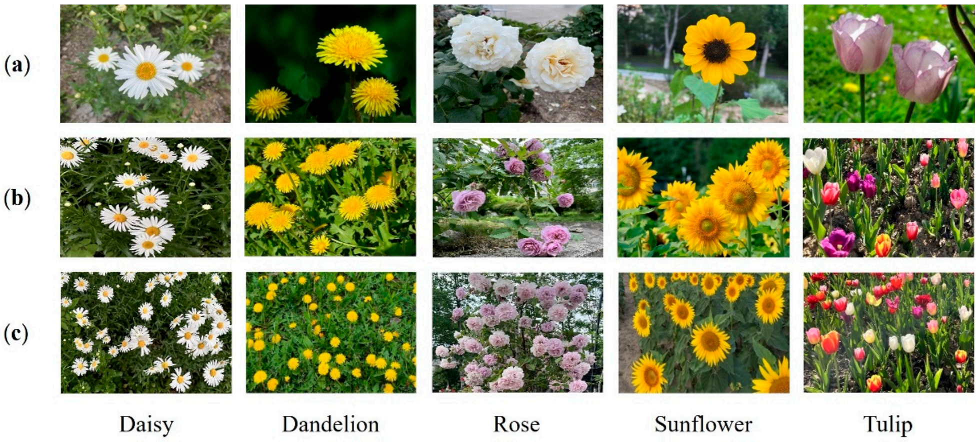

Figure 1. Flower datasets with different densities. (**a**) Low-density. (**b**) Medium-density. (**c**) High-density.

Table 1. Flower dataset information.

Dataset	Flower Type	Number of Images	Number of Flower Images with Different Levels of Densities			Total Images	Total Number of Targets
			Low Density	Medium Density	High Density		
Flowers Recognition	Daisy Dandelion Rose Sunflower Tulip	541 544 503 632 481	805	1007	889	2701	7653
Oxford 102	Daisy Dandelion Rose Sunflower Tulip	120 117 171 63 46	160	201	156	517	1464
Donghu flower	Daisy Dandelion Rose Sunflower Tulip	103 40 105 44 140	121	151	160	432	1224

2.1.2. Data Labeling

This study used LabelImg 1.8.6 to annotate 3218 images in the three datasets. After the datasets were divided, LabelImg was used manually to annotate each flower target in the images with a bounding rectangle, as shown in Figure 2. Depending on the different types of flowers, a corresponding label content was set, and finally, the format for saving annotated images was set to YOLO. The YOLO annotation format file mainly contains the following information: the category number of each target in the image; the center position (X, Y) information of each annotated target object in the image; and the width and height (W, H) information of each target object.

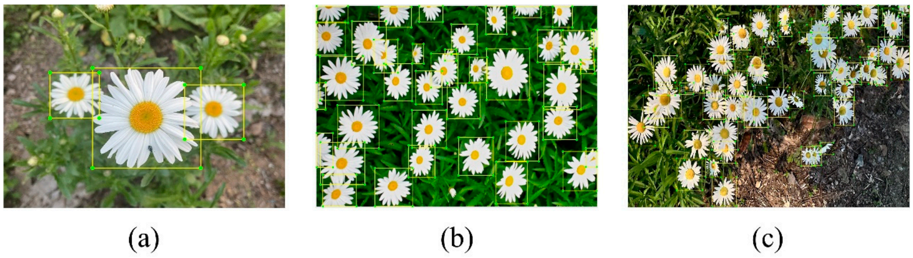

Figure 2. Schematic of LabelImg labeling for different densities. The yellow box represents the annotation box. (**a**) Low density. (**b**) Medium density. (**c**) High density.

2.1.3. Data Augmentation

To avoid overfitting due to the scarcity and high similarity of sample data, this study employs a series of data augmentation operations on the original data. As shown in Figure 3, this study adopts data augmentation methods such as flipping, rotating, random cropping, brightness adjustment, blurring, and noise addition to process the flower images. These augmentation operations expand the volume of the training set to twice that of the original training set. A total of 5942 images, including augmented images and original images, are used in this study's experiments. These images are divided into a training set (5286 images), a validation set (325 images), and a test set (331 images) at a ratio of 8:1:1. The test set only selects original images that have not undergone augmentation processing. The original images are more suitable for validating and explaining the performance of the model in this study, and for evaluating the detection effect of the model and the accuracy of flower counting.

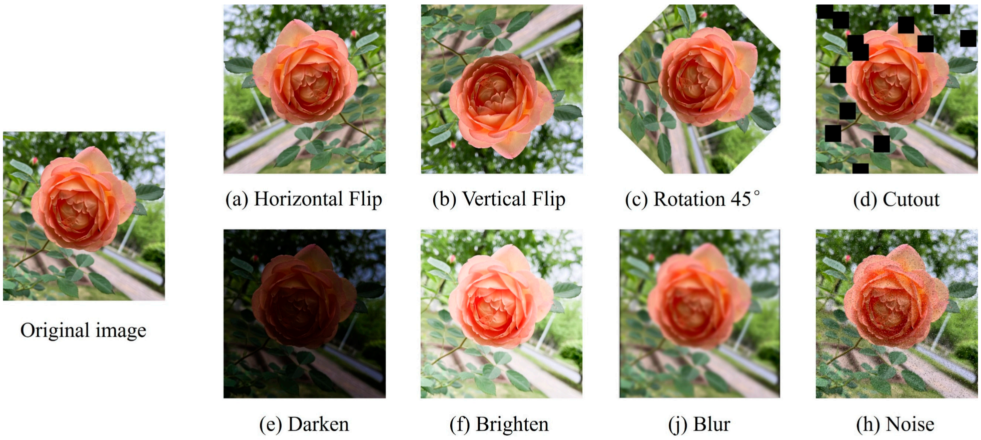

Figure 3. Example of dataset augmentation.

2.2. *Light-FC-YOLO*

This paper improves and applies the Light-FC-YOLO lightweight model based on YOLOv8s for accurate flower counting (as shown in Figure 4), providing technical support for the deployment of the model on embedded devices. The specific improvements are as follows: (1) A new efficient module is proposed, which replaces the 3 × 3 convolution layer in the original detection head, improving the deep feature extraction ability of the detection head. (2) The lightweight LSKA attention mechanism is embedded in the SPPF module, enhancing the model's ability to extract spatial information at different scales.

(3) In response to the problem that CIoU lacks consideration for the impact of target angles on the results, the SIoU loss function is introduced, improving the detection accuracy and training speed of the model.

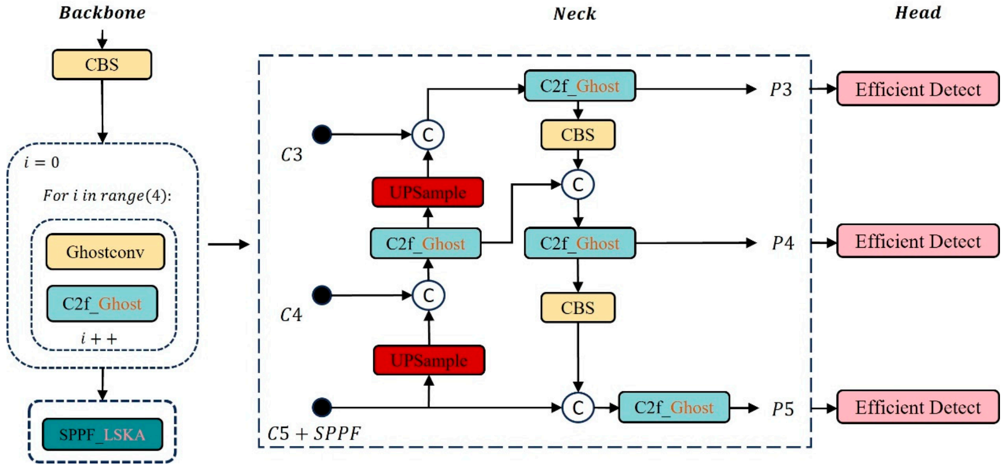

Figure 4. FC-YOLO model.

2.3. C2f_GhostNet

In convolutional neural networks, the redundancy of feature maps is considered an important characteristic of the network. In previous convolutional neural networks, after network feature extraction, many similar feature maps are generated. Moreover, for redundant feature maps, multiple convolution operations are often used to generate redundant feature maps one by one, which consumes a large amount of floating-point computation and parameters. The Ghost module generates more similar feature maps through linear transformation, avoiding the need to perform convolution operations on the redundant feature maps generated during the feature extraction process, thereby reducing computational cost and consumption. The Ghost module is shown in Figure 5.

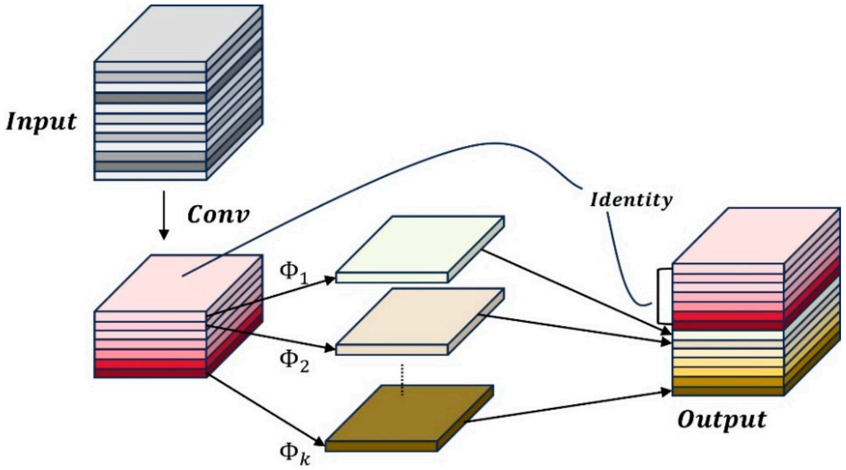

Figure 5. Ghost model.

In GhostNet, the Ghost BottleNeck is composed of Ghost modules. Ghost BottleNeck employs a structure similar to the basic residual block in the Residual Network (ResNet) [32], as shown in Figure 6. When the stride is 1, Ghost BottleNeck continuously stacks two Ghost modules. The first Ghost module expands the number of channels, and the second Ghost module reduces the dimensionality of the features, reduces the number of channels, and performs feature matching. When the stride is 2, a depthwise separable convolution (Deptwise Conv) with a stride of 2 is added between the two Ghost modules to compress the size of the feature map. Finally, an add alignment operation is performed on the input features and the processed features to obtain the output result.

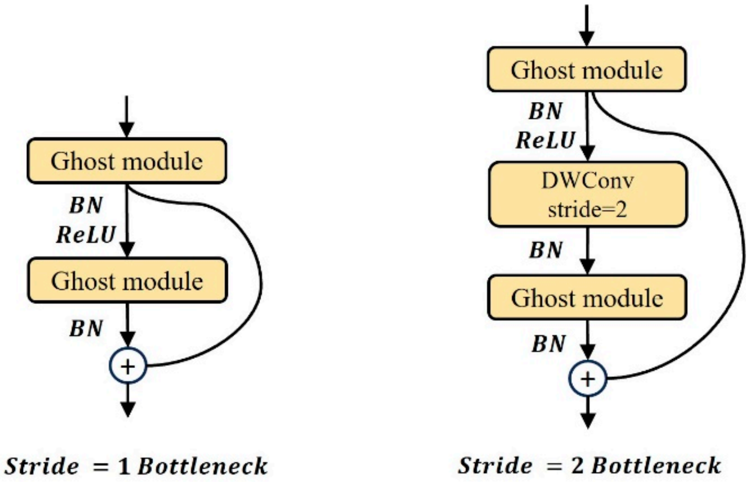

Figure 6. Ghost BottleNeck module.

Depthwise separable convolution, as an important component of GhostNet, is characterized by dividing the complete convolution calculation into two steps, as shown in Figure 7. Therefore, it can be mainly divided into two parts: Depthwise Conv and Pointwise Conv. Depthwise primarily performs per-channel convolution operations on each channel of the input feature map, does not change the original number of channels, and does not share feature information. Pointwise is a 1 × 1 convolution layer, which can change the number of output channels and perform channel fusion operations on the feature map output by Depthwise Conv.

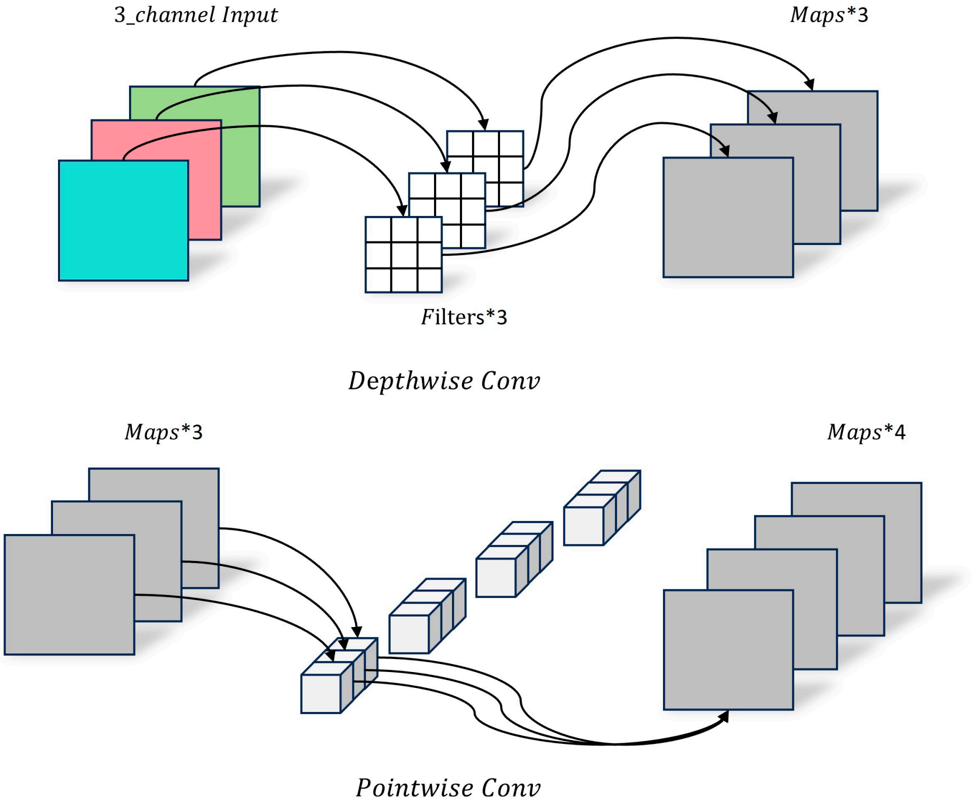

Figure 7. Schematic of depthwise separable convolution.

The GhostNet primarily consists of a series of Ghost BottleNecks with varying strides, overall following the architecture as shown in MobileNetv3 [33]. The first layer is composed of 16 filters, followed by a series of Ghost BottleNecks, which are divided into different stages based on the size of their input feature maps. The stride of the last Ghost BottleNeck in each stage is 2, while the stride of the remaining Ghost BottleNecks is 1. Subsequently, global average pooling and a 1 × 1 convolution layer are used to increase the dimensionality of the feature map to 1280. Finally, a fully connected layer is used to perform feature transformation and classification operations.

Therefore, this study utilizes the efficient GhostBottleNeck module to replace the BottleNeck module in the C2f module, constructing a new module, C2f_Ghost. This allows the model to maintain detection accuracy and speed while having less computational load. The C2f_Ghost module is shown in Figure 8.

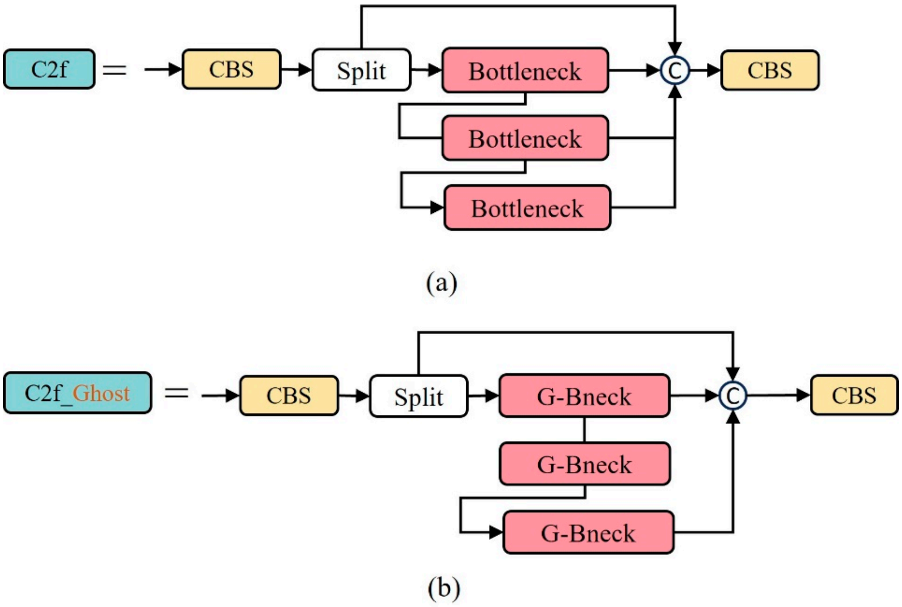

Figure 8. (**a**) C2f module. (**b**) C2f_Ghost module.

2.4. Efficient Detection Head

In target detection, the head structure is generally classified into two types. One is the fully connected head (FC-head), where in the fully connected layer network, the features extracted from each node are connected, making the fully connected head more spatially sensitive, but also leading to a generally higher parameter volume than the convolution head. The other is the convolution head (Conv-head), which has a simpler network structure and less computation compared to the fully connected head. Among them, the fully connected head performs better in classification tasks, while the convolution head is more suitable for positioning tasks [34]. Although the connection head may perform slightly better than the convolution head in terms of detection accuracy, it is not conducive to model lightweighting. The detection head part of YOLOv8s [35] adopts the convolution head, outputting classification and regression information. As can be seen from Figure 9, the original detection head adopts the decoupled head [36] structure and uses a parallel branch method, allowing the features to first pass through two 3 × 3 convolution layers, and then a standard convolution calculates the bounding box loss value and classification loss value, respectively. However, due to the need for lightweighting, the number of network layers in the backbone has been greatly reduced. Although this saves a lot of computational cost and consumption, the decline in detection accuracy is inevitable. Therefore, it is necessary to redesign the detection head part to obtain better feature detail extraction capability. Therefore, this paper proposes an efficient module to replace the 3 × 3 convolution layer in the detection head part of the original YOLOv8s model.

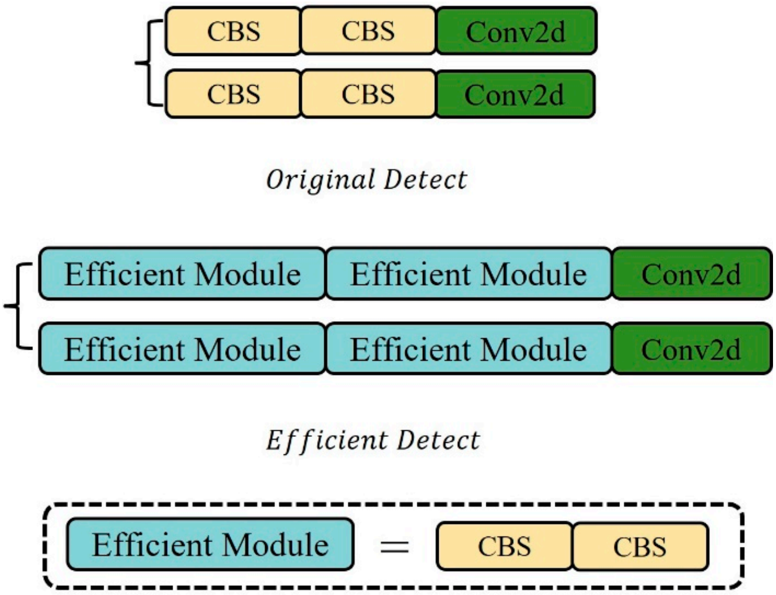

Figure 9. Original and efficient detection heads.

The improved efficient detection head still adopts the decoupled head structure and uses a parallel branch feature processing method. However, in each branch, two efficient modules are stacked to replace the two 3 × 3 convolution layers in the original detection head, and finally, the output is calculated by the standard convolution layer. The efficient module consists of two 3 × 3 convolution layers. Due to the small computational burden of 3 × 3 convolution, it can also enhance non-linearity, improving the model's expressiveness for complex functions. Therefore, on the premise of slightly increasing the network computation, it can extract deeper and richer image features, retain more image spatial information, and thus significantly improve the model's detection performance.

2.5. SPPF_LSKA Module

The multi-scale feature extraction module is an important part of the YOLO series of algorithms, typically located at the end of the backbone network, with a fixed output size. SPPF redesigns the structure based on the SPP module [37]. Its output purpose remains unchanged, but every time a feature passes through a pooling layer, the result is retained. After the feature undergoes three maximum pooling processes, all the results obtained will be concatenated. The advantage of this method is that it greatly reduces the computational load and parameter volume brought by the multi-scale feature extraction module, and greatly improves the running speed and efficiency of the model. However, its disadvantage is that its ability to extract spatial information at different scales is not as good as the SPP module. Therefore, this paper integrates the LSKA lightweight attention mechanism module into the SPPF module, and under the premise of almost no increase in model computation and parameter volume, improves its ability to extract spatial information at different scales.

Large Kernel Attention (LSA) serves as the prototype for the LSKA [38] module, and its performance has been validated in the Visual Attention Network. This is shown in Figure 10. The innovation of LSKA lies in decomposing the 2D convolution kernel $\left|\frac{k}{d}\right| \times \left|\frac{k}{d}\right|$ of the depthwise separable convolution into cascaded horizontal or vertical 1D kernels $(1 \times \left|\frac{k}{d}\right|, \left|\frac{k}{d}\right| \times 1)$, enabling it to adaptively capture long-range relationships. At the same time, it maintains a performance comparable to that before the improvement while consuming less computation and memory.

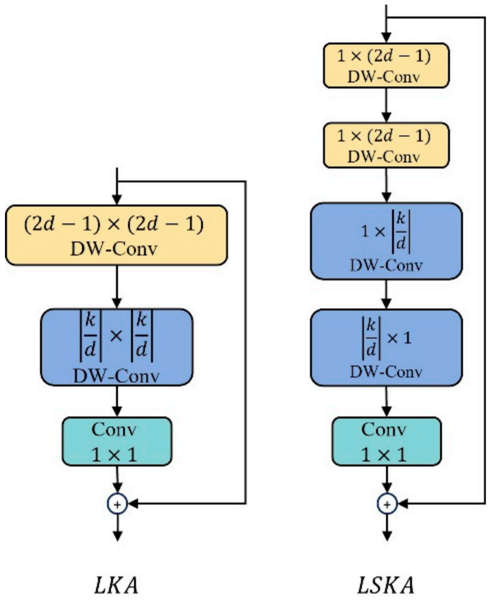

Figure 10. LKA and LSKA structure diagrams.

The improved SPPF module is shown in Figure 11, with the LSKA attention module embedded at the end of the module. After all feature mappings have completed the concatenation operation, they are re-input into the LSKA attention module. Based on the context-dependent relationship, the feature weights are adaptively recalibrated, which aids the SPPF module in enhancing its ability to extract multi-scale features.

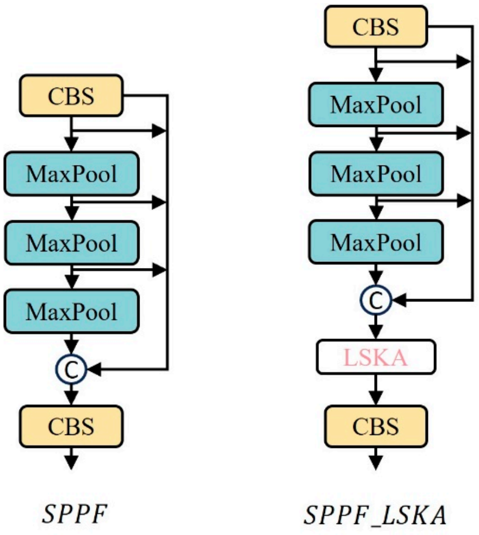

Figure 11. Comparison chart of SPPF before and after improvement.

2.6. SIoU Loss Function

Given that multi-object flower counting constitutes a dense-object detection task, the loss function for model detection and localization must consider not only overlap area, aspect ratio, and center point distance, but also incorporate occlusion relationship loss and scale loss. The Complete Intersection over Union (CIoU) loss function [39] utilized in YOLOv8s overly depends on bounding box regression metrics during target localization, neglecting the direction of mismatch between the prediction and ground truth boxes. The Smoothed Intersection over Union (SIoU) loss function [40] redefines the penalty measure, enabling the prediction box to move more rapidly towards the nearest axis during training.

Furthermore, it takes into account the vector angle between the prediction and ground truth boxes, thereby accelerating the convergence speed during the model training phase.

When calculating CIoU, not only the overlapping area and the distance between the two center points are considered, but also the aspect ratio is taken into account. Its formula is as follows:

$$\mathrm{CIoU} = \mathrm{IoU} - \frac{\rho^2\left(q, q^{gt}\right)}{C^2} - \alpha v \tag{1}$$

Herein, q represents the center point of the predicted box, and q^{gt} represents the center point of the actual box. $\rho^2\left(q, q^{gt}\right)$ is the square of the Euclidean distance between the two center points. C represents the length of the diagonal of the minimum bounding region (the smallest rectangular box area that can contain both the predicted box and the actual box). α is a trade-off parameter, and v is used to measure the aspect ratio. The formulas for α and v are as follows:

$$\alpha = \frac{v}{1 - IoU + v} \tag{2}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{3}$$

However, in the CIoU loss function, the aspect ratio is only used as an influencing factor. If the center points of the detection box and the prediction box are consistent with the original image, a situation may occur where the aspect ratios are the same but the values are different. The regression result obtained after CIoU calculation does not match the actual situation.

SIoU addresses this issue by incorporating the consideration of bounding box angle into the calculation of CIoU loss, making the penalty loss positively correlated with the angle cost, as shown in Figure 12. It also redefines the formula for center point distance cost based on the measurement of angle loss:

$$\Delta = \sum\nolimits_{t=x,y}\left(1 - e^{-\gamma\rho_t}\right) \tag{4}$$

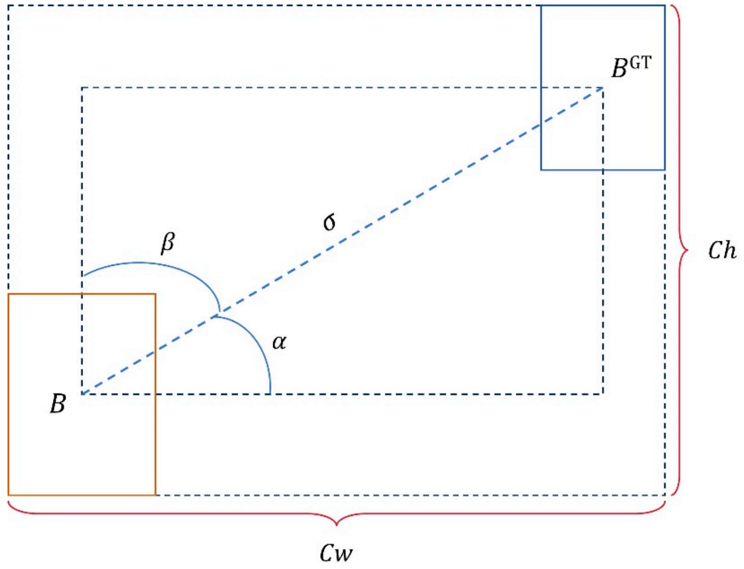

Figure 12. Schematic diagram of the SIoU function.

Its shape cost, i.e., the aspect ratio, calculates the difference in width between the predicted box and the actual box and the ratio of the width between the two boxes, so it is defined as:

$$\Omega = \sum_{t=w,h}\left(1 - e^{-w_t}\right)^{\theta} \tag{5}$$

The final definition of the SIoU formula is as follows:

$$L_{box} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{6}$$

2.7. Evaluation Metrics

This study uses a series of evaluation metrics to verify the performance and effectiveness of the FC-YOLO model. These metrics include Precision, Recall, Average Precision (AP), and Mean Average Precision (mAP). Precision refers to the probability of true positive results among all samples. Recall refers to the probability that actual positive samples are correctly predicted as positive, representing the prediction accuracy among all positive samples. Their calculation methods are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{AP} = \int_0^1 p(r)dr \tag{9}$$

$$\text{mAP} = \frac{1}{c}\sum_{i=1}^{c} AP_i \tag{10}$$

MAE, MAPE, RMSE, and R^2 are used as evaluation metrics to assess the model's specific performance in flower counting. MAE reflects the average difference between the actual number of flowers and the number predicted by the model. The smaller the MAPE value, the smaller the error between the result and the actual value. RMSE is based on MSE to measure the square deviation between the actual value and the predicted value, and it is often used as a performance evaluation metric in regression tasks. The closer the R^2 value is to 1, the better the model's performance. Their calculation formulas are as follows:

$$\text{MAE} == \frac{1}{\text{n}}\sum_{i=1}^{n}|(\hat{y}_i - y_i)| \tag{11}$$

$$\text{MAPE} == \frac{1}{\text{n}}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{12}$$

$$\text{RMSE} == \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{13}$$

$$\text{R}^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\overline{y}_i - y_i)^2} \tag{14}$$

where y_i, $\hat{y}_i$ and $\overline{y}_i$ represent the number of flowers in the i-th image in the file to be detected, the average actual flower count, and the predicted count of the i-th flower image, respectively. n is the total number of images to be detected.

3. Results

3.1. Implementation Details

The model proposed in this paper and the models used for comparison were all trained on a local GPU. Table 2 shows the specific configuration details. In the experiments of this chapter, all target detection algorithms used the SGD optimizer to optimize the learning rate during training, with the momentum parameter set to 0.937. The initial learning rate was set to 0.001, and the weight decay was 0.0005. At the beginning of training, a Warmup strategy was adopted, setting the warm-up Epoch to three. An EarlyStopping strategy was also adopted, automatically stopping model training to prevent overfitting when the loss of the test set no longer decreases. Throughout the overall training process, the batch_size for training was set to 16, and the number of training rounds was 200.

Table 2. Experimental hardware and software configuration information.

Project	Detail
CPU	AMD Ryzen 7800H (AMD, Santa Clara, CA, USA)
GPU	GeForce RTX 3060 6G (NVIDIA, Santa Clara, CA, USA)
RAM	16 GB
Operating system	64-bit Windows 11
PyTorch	1.11.0
CUDA	CUDA 11.3
Python	3.9

3.2. Analysis of Lightweighting Results

In this paper, based on YOLOv8s, various lightweighting improvement methods were attempted, including replacing its native backbone network with lightweight backbone networks such as Fasternet, EfficientVIT [41], and HGNetV2 [42], using SlimNeck [43] to replace the neck network of YOLOv8s, and using the Ghost Bottleneck module in Ghostnet to replace the Bottleneck module in the C2f module. Performance analysis was conducted on the models that incorporated different improvement methods, and appropriate lightweighting improvement strategies were selected to make the algorithm more easily portable to mobile or embedded devices.

Upon completion of the training of the improved model, it was validated on the Donghu Flower Dataset. The performance and complexity comparisons are shown in Tables 3 and 4. As indicated in the tables, Slimneck_YOLOv8s, after lightweight improvement, possesses higher accuracy than the original model. However, the rest of the lightweight models, mAP_{50} and $\mathrm{mAP}_{50\text{–}95}$, all show a decline compared to the original model. Ghost_YOLOv8s has a Recall value that is 1.0% higher than the original model. All models, after being made lightweight, do not perform as well as the original model in terms of frame rate, with HGNetV2_YOLOv8s having the FPS value closest to YOLOv8s. The lightweight backbone networks of EfficientVIT and HGNetV2, based on the Transformer architecture, when combined with YOLOv8s, show a significant decline in detection performance.

Table 3. Comparison of detection performance of different models.

Models	Recall/%	mAP/%	$\mathrm{mAP}_{50:95}$%	FPS
YOLOv8s	81.1%	87.0%	73.7%	90.2
EfficientVIT_YOLOv8s	76.9%	85.2%	71.0%	33.2
FasterNet_YOLOv8s	81.5%	85.9%	71.4%	75.2
Ghost_YOLOv8s	82.1%	86.2%	73.5%	76.8
HGNetV2_YOLOv8s	76.9%	86.2%	72.9%	86.8
Slimneck_YOLOv8s	80.6%	87.2%	74.3%	75.3

Table 4. Comparison of complexity of different models.

Models	$\mathrm{GFLOP}_\mathrm{S}$	Parameters/M	Model Size/MB
YOLOv8s	28.4	11.13	22.5
EfficientVIT_YOLOv8s	20.4	8.38	17.5
FasterNet_YOLOv8s	21.7	8.61	17.5
Ghost_YOLOv8s	16.1	5.92	12.2
HGNetV2_YOLOv8s	23.3	8.47	17.3
Slimneck_YOLOv8s	25.1	10.27	20.9

From the perspective of model complexity, the YOLOv8s models that have undergone lightweight improvements all perform better than the original YOLOv8s. Among them, the GFLOPs, parameter quantity, and model size of Ghost_YOLOv8s are 16.1, 5.92, and 12.2, respectively, which are 43.3%, 46.8%, and 45.8% lower than YOLOv8s, making it the most significantly lightweight model among all. Although Slimneck_YOLOv8s slightly outperforms YOLOv8s in terms of detection accuracy, its model complexity is closest to YOLOv8s, and the degree of lightweighting is not significant.

Considering the comprehensive comparison of model performance and complexity, Ghost_YOLOv8s is the optimal choice. Although its mAP$_{50}$ and mAP$_{50–95}$ values differ from YOLOv8s by 0.8% and 0.2%, respectively, its Recall is 1.0% higher than YOLOv8s, and it has the highest degree of lightweighting. The comparison of the training mAP of all models is shown in Figure 13. As can be seen from the figure, due to the presence of the early stopping mechanism, the convergence speed of EfficientVIT_YOLOv8s and Ghost_YOLOv8s models is slower than YOLOv8s, with YOLOv8s, triggering the early stopping mechanism and completing training around the 180th round. The FasterNet_YOLOv8s model has the fastest convergence speed, converging around the 110th round. The difference in mAP values among all models is not significant, but there is a large span in convergence speed.

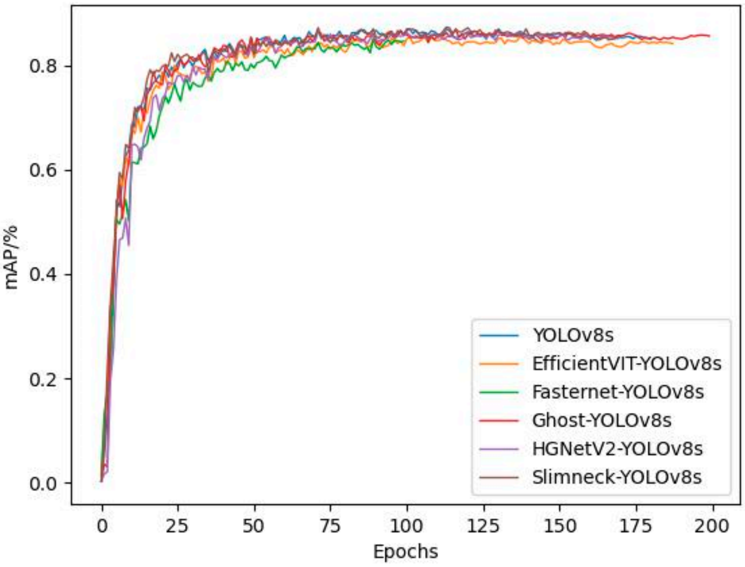

Figure 13. Variation curves of mAP during training for different models.

3.3. Analysis of Detection and Counting Results

3.3.1. Ablation Experiments

This paper has designed ablation experiments for the Light-FC-YOLO model to verify the effectiveness of the improvements. The ablation experiments are shown in Tables 5 and 6.

Table 5. The detection performance ablation experiments of Light-FC-YOLO model.

Improvement Points			Recall/%	mAP$_{50}$%	mAP$_{50:95}$%	FPS
Efficient Head	SPPF_LSKA	SIoU				
			82.1%	86.2%	73.5%	76.8
✓			82.0%	86.5%	73.8%	94.7
	✓		78.9%	86.6%	73.9%	86.3
		✓	79.8%	86.3%	73.5%	84.3
✓	✓	✓	82.5%	87.8%	73.6%	93.1

Table 6. The complexity ablation experiments of Light-FC-YOLO model.

Improvement Points			GFLOPs	Parameters/M	Model Size/MB
Efficient Head	SPPF_LSKA	SIoU			
			16.1	5.92	12.2
✓			17.5	9.06	16.5
	✓		16.9	6.99	14.4
		✓	16.4	5.92	12.2
✓	✓	✓	17.3	10.1	16.6

As can be seen from the table, by improving the detection head, the mAP_{50} and $mAP_{50–95}$ values increase by 0.3% and 0.3%, respectively, and the FPS increases from 76.8 to 94.7, indicating that the 3 × 3 convolution layer can effectively enhance the non-linear expression of features. However, the Efficient module slightly increases the complexity of the model, with the Parameters increasing from 5.92 to 10.1. In the SPPF feature extraction module, the lightweight attention LSKA was introduced, and the mAP_{50} and $mAP_{50–95}$ values increased by 0.4% and 0.4%, respectively. The decomposition operation of depthwise separable convolution effectively improves the model's ability to extract spatial information. The introduction of the loss function adds the calculation of angle loss for the target, which increases the mAP_{50} by 0.1%. Compared with Ghost-YOLOv8s, the model complexity of Light-FC-YOLO slightly increases, but the model performance significantly improves, with the mAP_{50}, $mAP_{50–95}$, Recall, and FPS values increasing by 1.6%, 0.1%, 0.4%, and 21.2%, respectively.

3.3.2. Comparison of Detection Performance of Lightweight Models

Given that the YOLO series of algorithms simultaneously take into account detection speed and accuracy, they are relatively balanced in terms of speed and accuracy. This section of the experiment compares the performance of different lightweight YOLO algorithms and Light-FC-YOLO on the Donghu Flower Dataset. The experimental results are shown in Table 7.

Table 7. Comparison of detection performance of different lightweighting models.

Model	Recall/%	mAP_{50}%	FPS	GFLOPs	Parameters/M	Model Size/MB
YOLOv4-tiny	80.3%	85.1%	61.2	8.7	7.14	14.2
Ghost-YOLOv5s	82.3%	85.8%	67.5	10.1	5.92	12.7
YOLOv5s	80.5%	86.3%	82.7	23.7	9.11	18.5
YOLOv7-tiny	81.9%	86.0%	74.6	13.2	6.02	11.3
YOLOv8s	81.1%	87.0%	85.9	28.4	11.1	22.8
Light-FC-YOLO	82.5%	87.8%	93.1	17.3	10.1	16.6

According to the experimental results, the Light-FC-YOLO model outperforms other models used for comparison in terms of Recall, mAP_{50}, and FPS. This indicates that the Light-FC-YOLO model has the best performance in detection, and runs at the fastest speed. From the perspective of model complexity, the GFLOPs, parameters, and model size of Light-FC-YOLO are 17.3, 10.1 M, and 16.6 MB, respectively, showing a better lightweight effect than YOLOv5s and YOLOv8s. Although the model complexity of Light-FC-YOLO is slightly higher than YOLOv4-tiny [44], Ghost-YOLOv5s, and YOLOv7-tiny, the latter perform poorly in detection accuracy, especially in Recall, which directly relates to the counting performance of the lightweight model.

3.3.3. Comparison of Counting Performance of Lightweight Models

In this section, trained lightweight models were selected for a counting experiment on the Donghu Flower Dataset. Table 8 shows the counting metrics of different models. As can be seen from the table, all the lightweight models used for comparison experiments do not show significant differences in counting metrics. Among them, the R^2 of the YOLOv8s model reached 0.9490, while the R^2 of Light-FC-YOLO was 0.9577. Compared with the YOLOv8s model, the MAE, MAPE, and RMSE of Light-FC-YOLO decreased by 0.8, 1.25%, and 0.13, respectively. Light-FC-YOLO has higher counting accuracy in counting multiple target flowers, which validates the effectiveness of the improvement strategies proposed in this chapter. In addition, Ghost-YOLOv5s, due to its good Recall in detection performance, has the smallest difference in counting performance compared to the method proposed in this chapter.

Table 8. Comparison of counting performance of different models.

Model	R^2	MAE	MAPE	RMSE
YOLOv4-tiny	0.9389	5.97	13.05%	11.43
Ghost-YOLOv5s	0.9508	4.81	10.78%	8.82
YOLOv5s	0.9482	5.54	12.57%	10.93
YOLOv7-tiny	0.9447	5.64	12.25%	10.71
YOLOv8s	0.9490	5.33	11.97%	9.82
Light-FC-YOLO	0.9577	4.53	10.62%	8.69

The performance of different lightweight models in flower counting is shown in Figure 14. As can be seen from the figure, although YOLOv7-tiny has an advantage in terms of lightweighting, its counting performance is not satisfactory. The counting performance of Ghost-YOLOv5s shows a good combination of detection speed and accuracy, and it can also have a good counting performance while being lightweight. Light-FC-YOLO's counting performance in all flower categories is superior to other lightweight models. Although there are slight omissions, there are almost no false detections. It successfully identifies most of the targets in the multi-target flower images and has the best counting performance. Compared with YOLOv8s, Light-FC-YOLO achieves higher counting accuracy under the premise of being more lightweight. In addition, due to the introduction of the LSKA attention mechanism, the shape encoding information of the target is more focused on in the network. In this group of tulip, sunflower, and rose test images, there are many areas with poor lighting conditions and slight reflections on the target surface, but Light-FC-YOLO can overcome the impact in the counting operation and complete the recognition well.

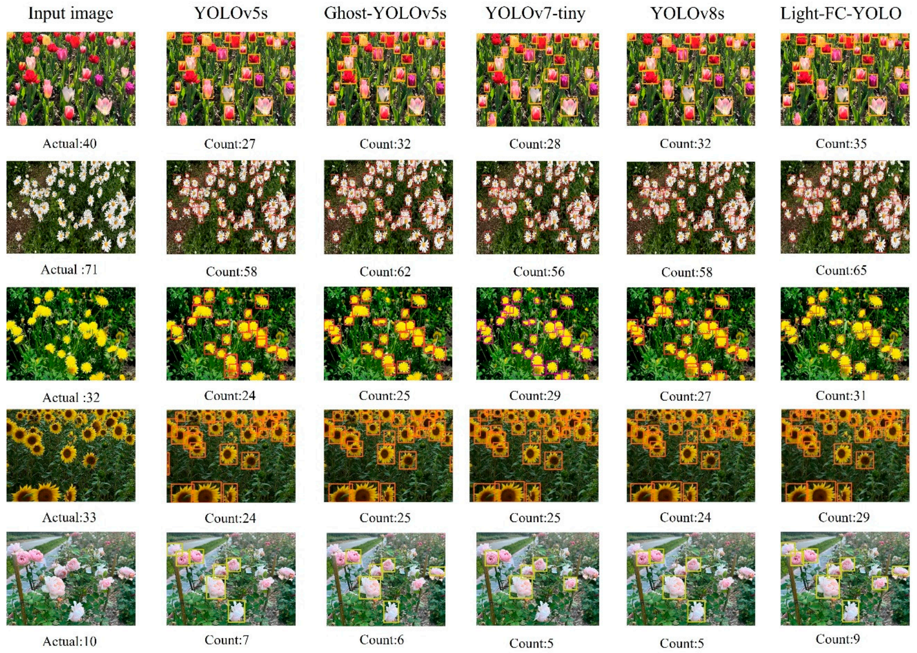

Figure 14. Visual comparison of different model counting performances.

4. Discussion

Traditional manual flower counting methods suffer from low efficiency, difficulty in ensuring accuracy, and over-reliance on subjective judgment. The density of flowers, as well as their shape, texture, and color, are key factors affecting the high-precision detection and accurate localization of the model. Therefore, rapid and accurate counting of multiple target flowers in natural scenes remains a challenge. In practical applications, due to situations such as dense flower growth, mutual occlusion between flowers, and a large proportion of background area, the feature information of the target is easily partially lost during the feature extraction process. To address this detection difficulty, this paper introduces a lightweight flower counting method based on multi-scale feature fusion. The Light-FC-YOLO model outperforms other lightweight models in multi-target flower counting under complex backgrounds. While achieving the purpose of model lightweight and improving deployment efficiency, it also improves counting accuracy to a certain extent, reduces the error rate, and provides a theoretical reference for the intelligent counting and positioning of flowers. Currently, research on lightweight detection of ornamental flowers is still very limited. Xie et al. [45] based on the improved YOLOv4 lightweight model, recognized multi-target flower images, achieving 79.63% mAP on the Oxford 102 and flower recognition datasets, but its detection performance was not as good as the method proposed in this paper.

The lightweight counting method used in this paper can meet the current demand for flower counting in flower quantity prediction, but there are still counting errors, and its detection accuracy and localization capabilities still have room for improvement. In the future, we will consider adding more multi-target flower images under more complex conditions and improving the model to further enhance accuracy. In order to delve deeper into the technical details influencing decision-making, we will employ interpretable artificial intelligence methods to further analyze the interactive features and learning patterns that Light-FC-YOLO has acquired. At present, this paper only uses some common types of flowers as research objects. In practical applications, it may be necessary to collect images of more types of flowers, study the impact of their differences on model detection, and make the model more adaptable to the detection of different types of flowers. The collection methods and shooting equipment for multi-target flower images also have room for improvement, further optimizing the shooting angle and using polarizers to reduce the impact of reflection on detection accuracy.

5. Conclusions

With the construction and development of smart agriculture, the estimation of flower quantity is transitioning from traditional manual evaluation methods to intelligent detection methods. To improve the model's ability to extract and locate flower features under high-density flower cultivation, this paper proposes a lightweight multi-objective flower counting model, Light-FC-YOLO, based on the YOLO framework. In this model, the C2f_Ghost module helps the model achieve its lightweight purpose. By utilizing the SPPF_LSKA module and Efficient head, the model's feature extraction ability is enhanced, strengthening the role of shallow shape encoding information in the network. Through a deeper fusion of deep and shallow flower features, the model can more accurately detect and locate targets. The introduction of the SIoU loss function, by considering the angle loss of the target, accelerates the convergence speed during model training. Overall, the method proposed in this paper improves the multi-objective flower detection situation in actual environments, while also enhancing its localization ability. The mAP, Recall, R^2, MAE, MAPE, and RMSE of Light-FC-YOLO reached 87.8%, 82.5%, 89.2%, 0.9577, 4.53, 10.62%, and 8.69, respectively, achieving a balance between detection speed and accuracy, providing a theoretical basis and technical support for the deployment of the model on mobile or embedded devices. The focus of future research in this paper is to further improve the model's robustness to environmental interference factors such as changes in illumination, accelerate the integration of computer vision technology with actual application scenarios, and further improve the efficiency and quality of automated agricultural production.

Author Contributions: Methodology, supervision, writing—review and editing, X.Y.; software, methodology, writing—original draft, H.C.; conceptualization, investigation, writing—review and editing, P.W.; formal analysis, methodology, resources, G.W.; resources, methodology, L.M.; data curation, investigation, B.W.; data curation, visualization, Y.Y.; data curation, validation, X.F.; investigation, validation, P.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. De, L.Y.; Zhi, B.L.; Wang, H. Analysis of market demand of urban flower industry. *North. Hortic.* **2022**, *13*, 134–140.
2. Chun, Q.C.; Lin, L.; Zhen, Y.T. Development status and countermeasures of Wenzhou flower market. *Agric. Sci. Technol. Newsl.* **2023**, 15–16+117.
3. Ya, D.C.; Xin, L.C.; Hong, Y. Practical Exploration of Informatization Technology in Flower Industry. *Agric. Eng. Technol.* **2022**, *42*, 20–21. [CrossRef]
4. Chlingaryan, A.; Sukkarieh, S.; Whelan, B. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Comput. Electron. Agric.* **2018**, *151*, 61–69. [CrossRef]
5. Sethy, P.K.; Routray, B.; Behera, S.K. Detection and counting of marigold flower using image processing technique. In *Advances in Computer, Communication and Control: Proceedings of ETES 2018*; Springer: Singapore, 2019; pp. 87–93.
6. Li, C.; Song, Z.; Wang, Y.; Zhang, Y. Research on bud counting of cut lily flowers based on machine vision. *Multimed. Tools Appl.* **2023**, *82*, 2709–2730. [CrossRef]
7. Huang, Y.; Qian, Y.; Wei, H.; Lu, Y.; Ling, B.; Qin, Y. A survey of deep learning-based object detection methods in crop counting. *Comput. Electron. Agric.* **2023**, *215*, 108425. [CrossRef]
8. Sun, L.; Yao, J.; Cao, H.; Chen, H.; Teng, G. Improved YOLOv5 Network for Detection of Peach Blossom Quantity. *Agriculture* **2024**, *14*, 126. [CrossRef]
9. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [CrossRef]
10. Xiao, J.; Zhao, T.; Yao, Y.; Yu, Q.; Chen, Y. Context Augmentation and Feature Refinement Network for Tiny Object Detection. 2021. Available online: https://openreview.net/forum?id=q2ZaVU6bEsT (accessed on 12 June 2024).
11. Xu, B.; Liang, H.; Liang, R.; Chen, P. Locate globally, segment locally: A progressive architecture with knowledge review network for salient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 3004–3012.
12. Zhao, Z.; Wang, J.; Liu, Y. User electricity behavior analysis based on K-means plus clustering algorithm. In Proceedings of the 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 19–21 December 2017; pp. 484–487.
13. Lin, P.; Lee, W.S.; Chen, Y.M.; Peres, N.; Fraisse, C. A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field. *Precis. Agric.* **2020**, *21*, 387–402. [CrossRef]
14. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
16. Petti, D.; Li, C. Weakly-supervised learning to automatically count cotton flowers from aerial imagery. *Comput. Electron. Agric.* **2022**, *194*, 106734. [CrossRef]
17. Carbonneau, M.A.; Cheplygina, V.; Granger, E.; Gagnon, G. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognit.* **2018**, *77*, 329–353. [CrossRef]
18. Tamrakar, N.; Karki, S.; Kang, M.Y.; Deb, N.C.; Arulmozhi, E.; Kang, D.Y.; Kook, J.; Kim, H.T. Lightweight Improved YOLOv5s-CGhostnet for Detection of Strawberry Maturity Levels and Counting. *AgriEngineering* **2024**, *6*, 962–978. [CrossRef]
19. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
20. 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.
21. Li, S.; Tao, T.; Zhang, Y.; Li, M.; Qu, H. YOLO v7-CS: A YOLO v7-Based Model for Lightweight Bayberry Target Detection Count. *Agronomy* **2023**, *13*, 2952. [CrossRef]

22. 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.
23. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.
24. Chen, J.; Hu, X.; Lu, J.; Chen, Y.; Huang, X. Efficient and Lightweight Automatic Wheat Counting Method with Observation-Centric SORT for Real-Time Unmanned Aerial Vehicle Surveillance. *Agriculture* **2023**, *13*, 2110. [CrossRef]
25. Chen, J.; Kao, S.H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
26. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
27. 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.
28. 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]
29. 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]
30. Available online: https://www.kaggle.com/datasets/alxmamaev/flowers-recognition (accessed on 12 June 2024).
31. Available online: https://www.robots.ox.ac.uk/~vgg/data/flowers/102 (accessed on 12 June 2024).
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Adam, H. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1314–1324.
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Song, G.; Liu, Y.; Wang, X. Revisiting the sibling head in object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11563–11572.
35. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO (Version 8.0.0) [Computer Software]. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 June 2024).
36. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
38. Lau, K.W.; Po, L.M.; Rehman, Y.A.U. Large separable kernel attention: Rethinking the large kernel attention design in cnn. *Expert Syst. Appl.* **2024**, *236*, 121352. [CrossRef]
39. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef]
40. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
41. 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.
42. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Chen, J. Detrs beat yolos on real-time object detection. *arXiv* **2023**, arXiv:2304.08069.
43. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424.
44. Yu, J.; Zhang, W. Face mask wearing detection algorithm based on improved YOLO-v4. *Sensors* **2021**, *21*, 3263. [CrossRef] [PubMed]
45. Xie, Z.; Hu, Y. Multi-target recognition system of flowers based on YOLOv4. *J. Nanjing Agric. Univ.* **2022**, *45*, 818–827.

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.