IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 June 2023, accepted 19 June 2023, date of publication 22 June 2023, date of current version 28 June 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3288587*

RESEARCH ARTICLE

# An Improved Lightweight Parameters Network for Strawberry Flowers Detection

**BAO ZHOU, XUEYING LIN, JIE ZHOU, YUJIN WANG, AND FANGCHAO HU**
School of Mechanical Engineering, Chongqing University of Technology, Chongqing 400054, China

Corresponding author: Fangchao Hu (fangchaohu@cqut.edu.cn)

This work was supported in part by the Youth Project of Science and Technology Research Program of Chongqing Municipal Education Commission of China under Grant KJQN202101131 and Grant KJQN202201147; in part by the Cooperative Project Between China Academy Science and University in Chongqing under Grant HZ2021011; in part by the Scientific Research Foundation of Chongqing University of Technology under Grant 2020ZDZ011; in part by the Postdoctoral Science Foundation Program of Chongqing Science and Technology Bureau under Grant CSTB2022NSCQ-BHX0674; in part by the Chongqing Science and Technology Commission of China under Grant cstc2020jcyj-msxmX0242; in part by the Science and Technology Research Program of Banan District of Chongqing under Grant 2020TJZ009; in part by the Student Innovation and Entrepreneurship Program under Grant 2022CX158 and Grant 2022CX148; and in part by the Student Research Project of Chongqing University of Technology under Grant KLB22049, Grant KLB20018, and Grant KLB22045.

**ABSTRACT** Accurate and efficient detection for target crops is crucial to develop intelligent agriculture. A great deal of studies have been devoted to improving the accuracy and efficiency of detection algorithms, but the increasing requirement of computing power makes them particularly difficult to implement on embedded devices. Although some methods have been proposed to accelerate inference by lightening the weights of the algorithms after training, the huge computing power requirements of the algorithms are still a problem. In this paper, an improved lightweight parameters network with lightweight designed backbone and neck by grouped convolution is proposed, which also integrates convolutional (Conv) layers and Batch Normalization (BN) layers to accelerate inference. The experiments in this paper utilize the Strawberry Flower Detection dataset, Tomato dataset, Wind Turbine Detection dataset, and VOC2007 dataset to verify performances of the proposed network. And the results show that the computational cost, the number of parameters, memory footprint and inference time of the improved model are all reduced, while the mean Average Precision(mAP) is increased comparing with the baseline algorithm. Furthermore, the detection performances of the proposed algorithm implemented on Jetson Nano platform indicate it is suitable to be deployed in practical scenarios, especially for embedded platforms with limited computing power.

**INDEX TERMS** Lightweight, grouped convolution, real-time detection, embedded platforms.

## I. INTRODUCTION

Although computer vision technology has been applied everywhere in people's life, to decode image information as fast and accurately as person do is still a tricky problem [1]. Particularly, object detection is the most important and challenging part, which aims to classify and localize objects in images or videos [2], [3]. Fast and accurate object detection is essential for the smooth advancement of downstream tasks, such as using robots to pollinate strawberry flowers. Achieving the rapid and accurate detection of strawberry flowers is indispensable for yield estimation and development of pollination robots [4], [5].

From the VJ Det(Viola-Jones Face Detector) based manual features to the YOLO(You Only Look Once) series based deep learning, object detection continues to develop rapidly and deeply. And, detection algorithms with higher accuracy are constantly proposed by research institutions and universities [6]. However, the computing power demand is huge for both traditional algorithms and deep learning-based algorithms, which means that dedicated large computing devices are needed. And that is extremely unfriendly to UAVs (Unmanned Aerial Vehicles) or mobile robots with restricted load [7]. The computing devices equipped with

The associate editor coordinating the review of this manuscript and approving it for publication was Kah Phooi (Jasmine) Seng.

mobile devices are so lacking in computing power that cannot match the requiring of high-precision detection algorithms. In addition, overload computing shortens the lifetime of mobile devices significantly. Therefore, algorithms with high accuracy and low arithmetic power requirements are indispensable for mobile devices.

Although the problem of computing power demand in the object detection domain is still not completely solved, a large amount of outstanding works have made good progress. The field of traditional object detection based on manual features, from VJ Det to DPM (Deformable Parts Model), has seen an obvious improvement in detection speed and accuracy [8]. With the CNN (Convolutional Neural Network) making a splash in the field of computer vision, a lot of works have started to apply it to improve the efficiency of object detection. From RCNN to YOLO, the detection speed and accuracy have made a qualitative leap compared to traditional algorithms [9].

Despite the good progress made by a large amount of excellent works, the current field of object detection still has the following problems:

1) Some works only focused on improving the detection accuracy of the algorithm but ignored the algorithm's computing power requirement, which resulted in the algorithm not being successfully applied to embedded devices.
2) Some works ignored the algorithm's huge parameter that is the root cause of the huge computing power requirement. They only used pruning, quantization or other methods to lightweight the weight after training, which would decrease the detection accuracy.

In order to solve the above problems, our works aim to reduce the huge number of parameters brought by the bloated backbone of general object detection networks. A lightweight backbone with the VGG (Visual Geometry Group) paradigm is designed, which is simple enough to make the network lightweight and efficient [10]. In addition, to increase the sensitivity of the algorithm, an improved PAN (Path Aggregation Network) architecture is deployed as the neck of the detection network, which used skip-layer connections to transfer the strong localization information from the shallow layer to deep layer [11]. Both the backbone and neck use grouped convolution for calculation, which further reduced the parameters to accelerate the speed of training and inference [12]. After the network training with the above methods, the Conv layers and BN layers are further integrated to reduce the memory footprint of intermediate variables during the computation to accelerate the inference. The network is implemented on embedded device to verify the feasibility, and the experimental results demonstrate that the work of this paper have both highly accurate and efficient on embedded devices.

The contributions of our work are summarized as follows:

1) Under the premise of accuracy, a lightweight backbone with low parameters based on group convolution is designed, which ensures the running speed of the algorithm.
2) To improve the sensitivity of the algorithm to the object position information, the neck of the network based on PAN structures is improved through group convolution and skip-layer connection.
3) Conv layers and BN layers are integrated to reduce memory footprint and accelerate inference.
4) The algorithm is deployed on an embedded device and compared with state-of-the-art methods to verify the feasibility and practicability of our work.

## II. RELATED WORK

Due to the characteristics of contactless and noninvasive, computer vision is widely used for crop detection considering the advantage of protecting delicate plants, particularly fruits and flowers. In this section, a brief review of existing researches on crop detection based on traditional methods and deep learning methods is presented.

### A. TRADITIONAL METHODS

Lü et al. [13] used computer vision and support vector machine (SVM) to simultaneously segment the fruits and branches, and then acquired a recognition rate of 92.4% for citrus fruits. Kurtulmus et al. [14] detected immature peach fruits in natural environment using statistical classifiers and neural network, then 84.6%, 77.9% and 71.2% of the actual fruits were successfully detected using three different image scanning methods. Bulanon et al. [15] achieved an accuracy of 84.3% while monitor flowers using 20 hyperspectral aerial images which are sensitive to light. McCarthy et al. [16] identified the maize flowering status based on color segmentation and shape analysis using the images captured by infield low-cost fixed cameras. Zhou et al. [17] used four cameras to capture strawberry flowers illuminated with UVA light. According to the fluorescence of strawberry flower, they accomplished the flower detection from the captured images with an accuracy of 90% through the procedures of threshold segmentation, morphological operations and object size analysis. Nowadays, computer vision has become an indispensable technology in flower detection. However, due to the poor robustness resulting in a weak adaption of natural environments, the traditional computer vision technology is hard to provide effective information for downstream automated equipment, such as pollination robot and flower thinning robot.

### B. DEEP LEARNING METHODS

With the development of deep learning and its application in computer vision, the accuracy of flower detection has begun to be improved rapidly. Different region-based convolutional neural network (R-CNN), including the R-CNN, Fast R-CNN and Faster R-CNN, were used to detect strawberry flowers in outdoor field in the work of Lin et al. [18]. After trained by 400 strawberry flower images and tested by another 100 images, the networks acquired the detection accuracies

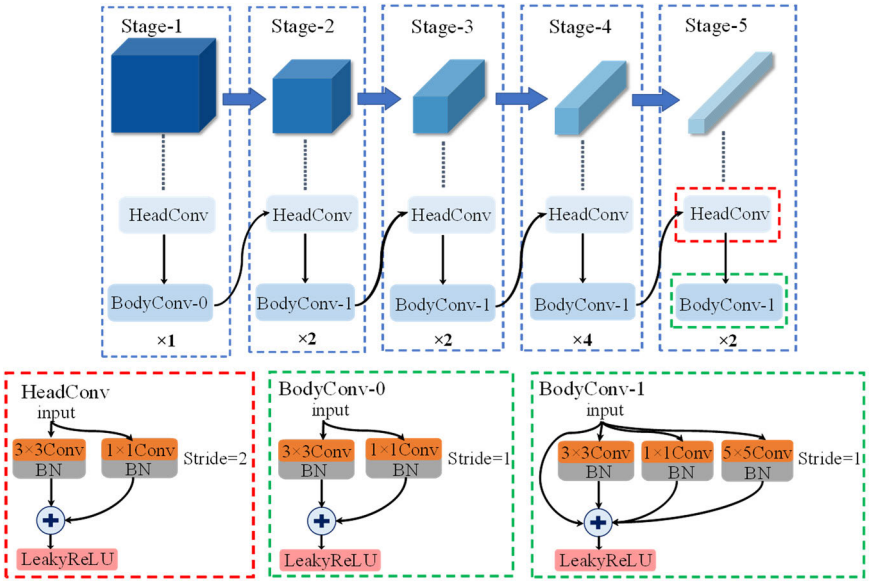



**FIGURE 1.** The architecture of improved backbone network.

of 63.4%, 76.7% and 86.1%, respectively for R-CNN, Fast R-CNN and Faster R-CNN. With the goal of detecting flowers and optimizing fruit production, Dias et al. [19], [20] proposed a CNN-based model which is robust to clutter and changes of illumination by combining both color and morphological information. Palacios et al. [21] developed a non-invasive method for grapevine flower counting under field conditions using a mobile sensing platform at a speed of 5 km/h to automatically capture RGB images. Tian et al. [22] proposed a Mask Scoring R-CNN with a U-Net backbone (MASU R-CNN) model to detect and segment apple flowers in different growth stages: bud, semi-open and fully open, resulting in the precision, recall and $F_1$-score are 96.43%, 95.37% and 95.90%, respectively. Li et al. [23] detected kiwi fruit flowers using the original YOLO v4 algorithm, and achieved a mean average precision (mAP) of 97.61%. In order to detect apple flowers accurately, Wu et al. [24] proposed a channel pruning-based YOLO v4 deep learning algorithm, which has an inference time of 0.046 second and a mAP of 97.31% after trained by apple flowers images collected manually in natural environments. The detection accuracy of flowers can be improved greatly through the methods of computer vision based on deep learning. However, the deep learning algorithms require huge computing power, resulting in a low calculation speed and dissociation from the real-time requirement when deployed in actual scene, especially for the automated pollination robot in precision agriculture [25].

In summary, most works did not consider algorithms implemented on embedded devices, which made it difficult to apply the algorithm in practical scenarios. Based on the mentioned above, an improved lightweight parameters network is proposed to implement on embedded devices.

## III. METHODOLOGY

In this section, an improved lightweight parameters network for strawberry flower detection is proposed. The state-of-the-art YOLO series are chosen as the baseline to compare the progress of our work in this paper.

### A. STEP 1: BACKBONE NETWORK LIGHTWEIGHT DESIGN

The usage of CSP structure [26] in the backbone network of YOLO v4(baseline) can greatly reduce the quantity of computation caused by the repetition of gradient information. And this not only enhances the learning ability of CNN network, but also eliminates the computational bottleneck and accelerates the inference speed [26]. The backbone network in baseline enhances the ability of object detection availably. However, the computing power required by baseline is still huge, which makes it difficult to be deployed on platforms with limited computing power.

In order to obtain a high performance of the algorithm, the Inception Network with multi-branch structure was proposed by Google firstly in 2015. It may significantly deepen the network and enable different convolution kernels to obtain different receptive fields, resulting in a better prediction accuracy of the algorithm. Subsequently, CSPDarknet53 also consists of multi-branch structures which can ensure a higher accuracy and a faster inference speed than that of Darknet53. Nevertheless, due to the preservation of the intermediate computing results in multi-branch structures, the memory footprint will increase significantly until the multi-channel fusion occurs. As a result, the backbone network based on CSPDarknet53 is unfavorable to be deployed on the platforms with limited computing power.

In view of the reasons mentioned above, the backbone network of the improved lightweight parameters network is

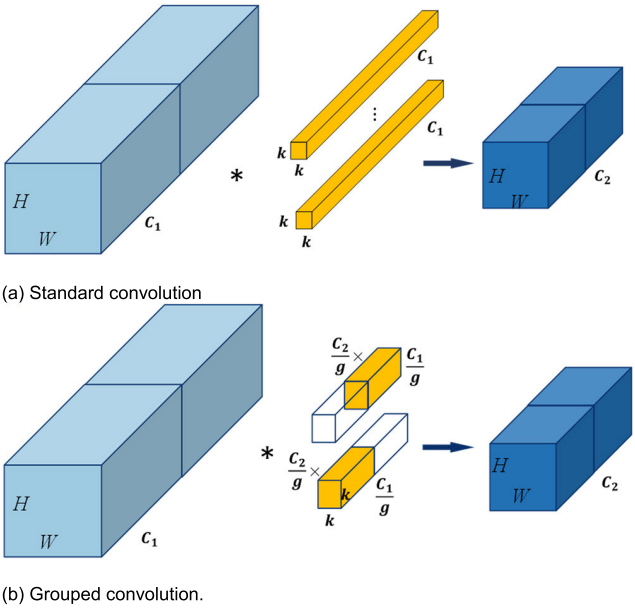



**FIGURE 2. Schematic diagrams of standard convolution and grouped convolution.**

lightweight designed to reduce the parameters in this paper, with the aim to obtain a more efficient object detection model. The lightweight design is mainly referred to the classic classification networks of VGG and re-parameterization VGG (Rep-VGG) [10]. The architecture of the backbone network is shown as Fig. 1.

The main improved strategies of the backbone network are detailed as below.

1) The topology of VGG is so concise and easy-to-use to be widely applied in industry and academia. Therefore, the main part of the backbone network in this paper is also designed based on VGG-style, in which the output of previous layer is simply input into the next layer without a large number of cross-layer branches. This topology could reduce the memory footprint and ensure the simplicity and efficiency of the network [27].

2) A convolution network with stride of 2 is used as the HeadConv. Through the subtraction of excess redundant information, it can provide different scales of feature maps for the multi-scale detection tasks of the downstream networks. It ensures the sensitivity of the algorithm to the objects with different sizes.

3) In order to increase the receptive field of the network and ensure the downstream networks to have a perfect detection accuracy, a 5∗5 convolution branch is added in the BodyConv on the basis of Rep-VGG network.

To reduce the quantity of computation, the standard convolution, as shown in Fig. 2(a), is replaced by grouped convolution in this paper [12], [28]. The comparison of them is shown as Fig. 2.

As shown in Figure 2(b), the input feature map is divided into $g$ groups according to the number of channels to perform the convolution calculation in the grouped convolution network, followed by the Concat operation. And the grouped convolution kernels are learned sparsely on the channels in a block-diagonal structure style. As a result, the convolution kernels with higher correlation are learned more structured, while the lower ones are no longer parameterized. The numbers of parameters and quantity of computation of the standard convolution and grouped convolution are shown in (1)-(2), respectively.

$$\begin{cases} \text{Standard\_params} = k^2 \times C_1 \times C_2 \\ \text{Standard\_FLOPs} = k^2 \times C_1 \times C_2 \times W \times H \end{cases} \tag{1}$$

$$\begin{cases} \text{Grouped\_params} = k^2 \times \frac{C_1}{g} \times \frac{C_2}{g} \times g \\ \text{Grouped\_FLOPs} = k^2 \times \frac{C_1}{g} \times \frac{C_2}{g} \times W \times H \times g \end{cases} \tag{2}$$

where, $k$ represents the size of the convolution kernel, $C_1$ and $C_2$ respectively represent the number of channels of the input feature map and output feature map, $W$ and $H$ respectively represent the width and height of the feature map, while $g$ represents the number of groups.

Comparing with standard convolution, the grouped convolution not only reduce the number of parameters and quantity of computation, but also make the convolution kernels learn more accurately and efficiently in the deep networks with less overfitting. From this perspective, the performance of the network with abundant groups will be more appropriate in a lightweight network. However, the large number of groups may lead to a significant increasement of the memory access cost (MAC) and a slow inference speed of the network. In order to ensure the detection accuracy of the network, the grouped convolution method is only deployed in the multi-branch layer part, namely, BodyConv.

### B. STEP 2: IMPROVED THE ARCHITECTURE OF PAN AS NECK

Learning the different scale features of objects is priority for object detection algorithms, because that the objects usually have different sizes in the images. The FPN (Feature Pyramid Network) with the top-down fusion strategy is used as the neck of YOLO v3 to obtain the feature map with the semantic information in the deep network layers and the texture information in the shallow network layers [29]. Furthermore, on the basis of FPN, the neck network of baseline is improved through the addition of PAN which has a bottom-up fusion strategy [11]. It makes the neck to be a two-way fusion network, and enhances the representation capability of the object detection algorithm.

For mobile platforms with limited computing power, it is important to have algorithms with low computing power requirements. The massive standard convolution operations of PAN may lead to a poor efficiency. Then, grouped convolutions are used to substitute them in the neck network apart from the sampling layer. This strategy may greatly decrease the number of parameters and calculation amount of the network. Additionally, the original PAN network focuses on the fusion of different scales but neglects the information

transformation from the shallow layer to the deep layer in the chain link [30]. In order to ensure the deep layers to acquire the spatial information existed in the shallow layers, thereupon, the shallow layers and the deep layers in the same chain link are skip-layer connected for three different scales links. The comparison of the original PAN network and the improved PAN network is shown in Fig. 3.

### C. STEP 3: INTEGRATION OF CONV LAYERS AND BN LAYERS

During the training stage of DNN, there is a notoriously phenomenon named internal covariate shift, which can greatly slow down the learning rate. Internal covariate shift refers to the fact that the distribution of each layer's inputs will change along with the variation of previous layers' parameters. And Awais et al. [31] solved this problem to accelerate the training of DNN by the method of Batch Normalization (BN), as shown in (3)-(5).

$$\tilde{X}_i = \gamma \frac{X_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta = \frac{\gamma}{\sqrt{\sigma^2 + \epsilon}} X_i + \left(\beta - \frac{\gamma \mu}{\sqrt{\sigma^2 + \epsilon}}\right) \quad (3)$$

$$\mu = \frac{1}{n} \sum X_i \quad (4)$$

$$\sigma^2 = \frac{1}{n} \sum (X_i - \mu)^2 \quad (5)$$

where, $\tilde{X}_i$ represents the feature map which is the output of the BN layer, $X_i$ represents the $i^{\text{th}}$ feature map of the batch acquired by convolution calculation of a certain layer and $1 < i \leq n$; $\mu$ and $\sigma^2$ represent the mean and variance of the batch, respectively; $\gamma$ and $\beta$ represent the scaling factor and translation factor, respectively; while $\epsilon$ represents a constant that is used to ensure a non-vanishing divisor.

As the $\mu$, $\sigma^2$, $\gamma$ and $\beta$ are fixed during the inference stage of the algorithm, we integrated the Conv layers and BN layers in the inference stage, in order to reduce the internal covariate shift and thus further improve the inference speed. And the integration method is shown in (6)-(9).

$$X_i = wX_{ii} + b \quad (6)$$

$$\tilde{X}_i = m\,(wX_i + b) + n \quad (7)$$

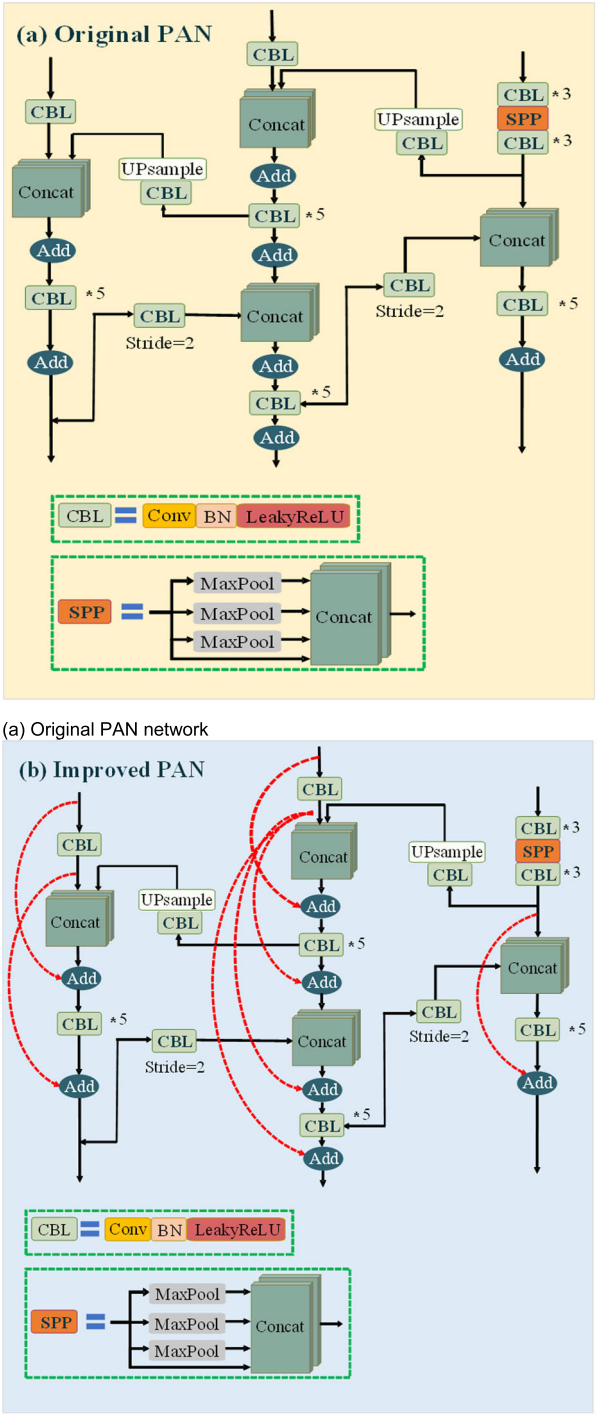
$$m = \gamma \Big/ \sqrt{\sigma^2 + \epsilon} \quad (8)$$

$$n = \beta - \gamma\mu \Big/ \sqrt{\sigma^2 + \epsilon} \quad (9)$$

where, $w$ represents the weight of the convolution kernel, $b$ represents the bias, $X_{ii}$ represents the feature map which is the output of the previous layer in the network.

After the improvement by the strategies mentioned in Step 1-Step 3, the improved lightweight parameters network is acquired and its framework could be illustrated as Fig. 4. Compared with YOLOv4, our work has the following differences:

1) Our work employs a simple VGG-style network as the backbone, rather than the usage of CPSDarknet53 backbone in YOLOv4. Considering the decrease of memory consumption by avoiding the heavy use of skip connections, our work is more suitable to be deployed on embedded devices.
2) YOLOv4 uses the PAN structure as its neck, while our work utilizes an Improved PAN. The Improved PAN is acquired by adding a small number of skip




(a) Original PAN network

(b) Improved PAN network in our works

**FIGURE 3.** Schematic diagrams of original PAN and improved PAN (red dotted lines represent skip-layer connections). Where, SPP stands for Spatial Pyramid Pooling; CBL stands for the sequential stack of a Convolutional layer, a Batch Normalization layer, and a LeakyReLU activation function.

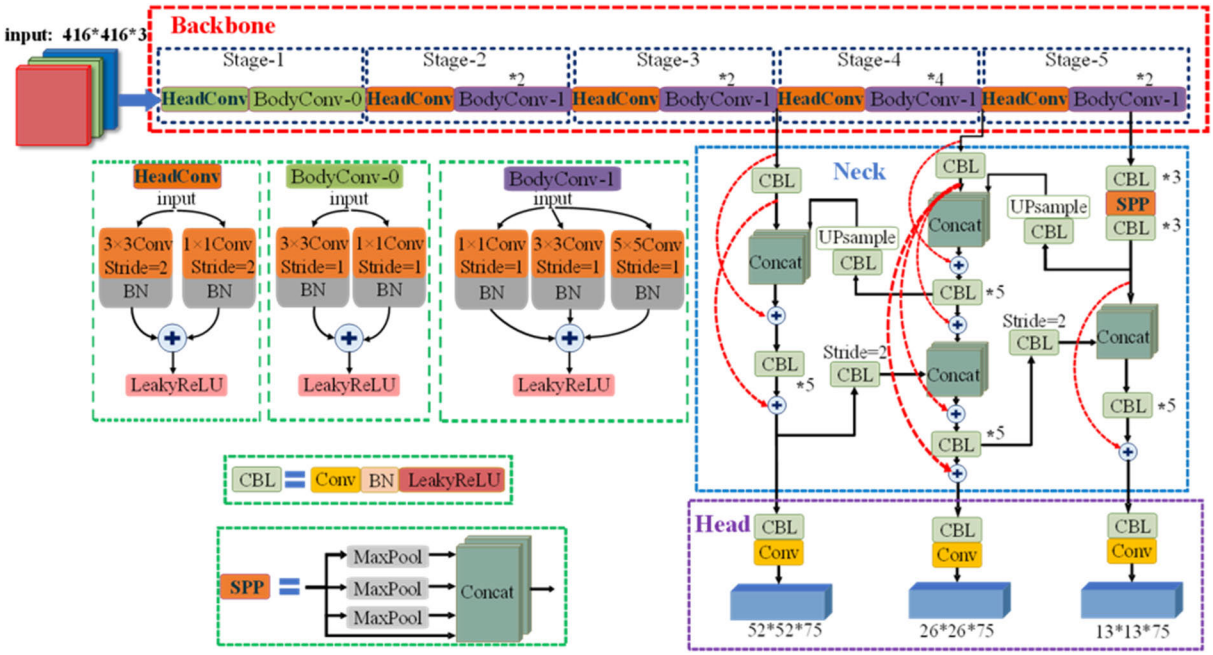



**FIGURE 4.** The framework of improved lightweight parameters network.

connections on the same branch of the PAN structure. And then, spatial information can effectively propagate from shallow layers to deeper ones. This approach is particularly effective in enhancing the performance of object detection algorithms on embedded devices.

3) The training and inference processes of YOLOv4 use the same network, while our work integrates the convolutional and BN layers of the network in the inference process to improve computational efficiency and reduce memory footprint.

## IV. EXPERIMENTS AND DISCUSSION

### A. MATERIALS

#### 1) DATASET GENERATION

The subjects used in this research include three varieties of strawberry flowers: Mengxiang, Redface and Ssanta. The images used in this study are collected in a strawberry plantation located in Jiulongpo District, Chongqing, China, using a simulated view from a mobile robot or UAVs. The images are photographed by a Xiaomi MI8 mobile phone (Xiaomi Technologies Co., Ltd, Beijing, China) with a resolution of 3024 pixels (horizontal) $\times$ 3024 pixels (vertical). Then, a total of 2424 images with detection objects are obtained from 2:00 pm to 5:30 pm in April 2022. And all images are collected in the natural environment of strawberry plantation, including natural illumination condition, natural growth orientation, natural shielding of leaves against illumination and flowers overlap.

Subsequently, LabelImg is used to manually label the strawberry flowers in these 2424 images, and the pistils of each flower are ensured to be located in the center of the bounding box when labeling. Then, the label files are stored as $*$.xml format. 81.00% (1962 images) of the prepared data set are used as training data for the training of the improved lightweight parameters network, while 9.00% (219 images) and 10.00% (243 images) are respectively used to verify and test the improved lightweight parameters network. The setup of dataset is shown as Table 1.

**TABLE 1.** Strawberry Flower Dataset.

| Datasets | Varieties | Number | Total |
|---|---|---|---|
| Training set | Mengxiang | 650 | 1962 |
| | Redface | 644 | |
| | Ssanta | 668 | |
| Validation set | Mengxiang | 78 | 219 |
| | Redface | 70 | |
| | Ssanta | 71 | |
| Test set | Mengxiang | 80 | 243 |
| | Redface | 78 | |
| | Ssanta | 85 | |

**TABLE 2.** Parameters of the simulation platform.

| Hardware | Parameters |
|---|---|
| Mainboard | ASUS WSX299 * 1 |
| CPU | Intel i9-10940X * 1 |
| RAM | KingstonDDR4 16GB * 4 |
| GPU | GEFORCE GTX2080 Ti 11GB * 1 |
| Hard disk | Kingston 500G * 4 |

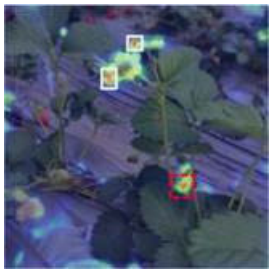
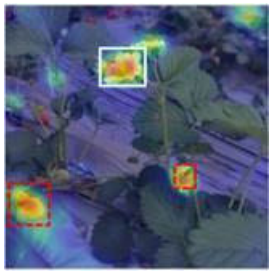
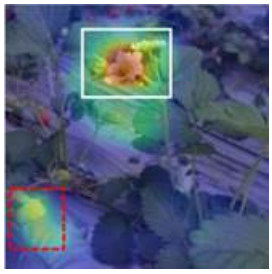
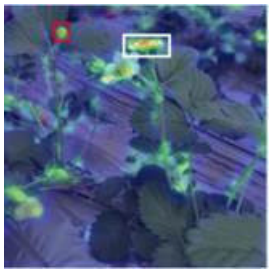
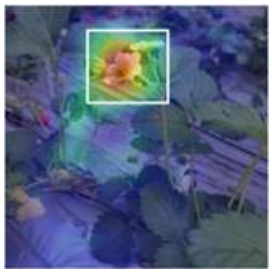
#### 2) SIMULATION PLATFORM

In order to verify the performance of the improved lightweight parameters network on the strawberry flower detection, we compare it with the baseline, as well as the other

**TABLE 3.** Performances comparison of baseline and the improved lightweight parameters network.

| Performances | Baseline | Our Work | | |
|---|---|---|---|---|
| | | Step 1 | Step 2 | Step 3 |
| FLOPs | 29.88 GMac | 16.53 GMac (↓44.68%) | 6.91 GMac (↓76.87%) | **6.83 GMac (↓77.14%)** |
| Params | 63.94 MB | 44.52 MB (↓30.37%) | 15.17 MB (↓76.27%) | **15.12 MB (↓76.35%)** |
| Memory | 1.12 GB | 1.00 GB (↓10.87%) | 0.89 GB (↓20.63%) | **0.57 GB (↓48.71%)** |
| Inference time | 12.44 ms | 8.06 ms (↓35.20%) | 8.34 ms (↓32.96%) | **7.62 ms (↓38.75%)** |

**TABLE 4.** The comparison of Grad-CAM heatmaps between original PAN and improved PAN.

| | Head 1 | Head 2 | Head 3 |
|---|---|---|---|
| Baseline with original PAN |  |  |  |
| Our work with improved PAN |  |  |  |

Note: the red dotted box and white box represent the mistake and correct features, respectively.

**TABLE 5.** Training parameter settings.

| Parameters | Value |
|---|---|
| Input image size | 416×416 |
| Maximum learning rate | 0.01 |
| Minimum learning rate | 0.0001 |
| Learning rate adjustment strategy | Cosine Annealing LR |
| Weight initialization method | Normal distribution |
| Batch_size | 8 |
| Epoch | 300 |

Note: the size of the input image is resized from 3024*3024 to 416*416.

object detection networks. The parameters of the simulation platform used for training and testing are shown in Table 2.

### B. LIGHTWEIGHT BENEFITS PRELIMINARY ASSESSMENT

The benefits of lightweighting are pre-evaluated on the simulation platform (as shown in Table 2), and the evaluation performances include the FLOPs, number of parameters, memory footprint, as well as inference time. In step with the process of improvement, the pre-evaluation results of the improved lightweight parameters network are shown in Table 3.

It can be found from Table 3 that the FLOPs, number of parameters, memory footprint and inference time of the improved lightweight parameters network are reduced by 77.14%, 76.35%, 48.71% and 38.75% after the three steps, respectively, which indicated that the proposed methods are effective. And it is noteworthy that the FLOPs and number of parameters decrease rapidly in Step 1 and Step 2, rather than Step 3. These decreases can be mainly attributed to the utilization of grouped convolution method in the two steps. Meanwhile, the memory footprint is also decreased rapidly in all steps, and the amounts of decrease are 10.87%, 9.76% and 28.08% comparing with the previous step. This could be mainly attributed to the grouped convolution method utilized both in Step 1, Step 2, and the integration of Conv layers and BN layers in Step 3. However, the decrease amount of memory footprint in Step 2 (9.76%) is lower than that of Step 1 (10.87%). In Step 1, the backbone network is lightweight design based on the concise VGG-style topology, which can decrease a large amount of memory footprint by reducing the preservation of intermediate computing results in multi-branch structures. Conversely, the complexity of the network is increased in Step 2 of neck architecture modification. Last but not least, the inference times are reduced by 35.20%, -2.24% and 5.79% comparing with the previous step. Based on the same reasons as the decrease of memory footprint, the inference time is greatly reduced in Step 1. Nevertheless, the inference time in Step 2 is slightly increased by

**TABLE 6.** Model training loss descent diagrams.

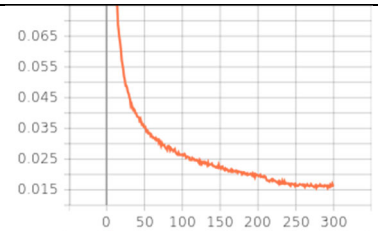
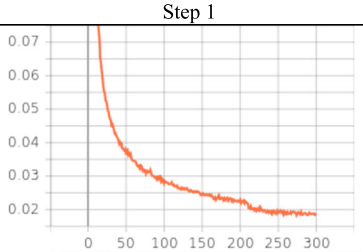
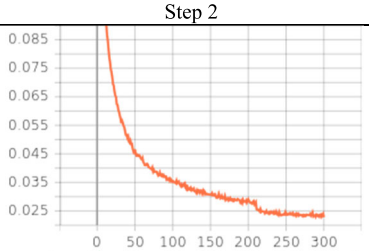
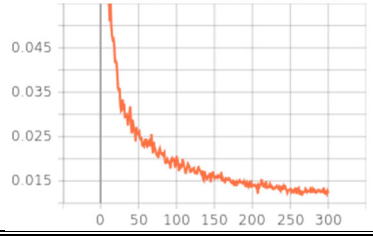
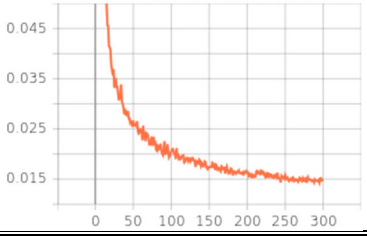
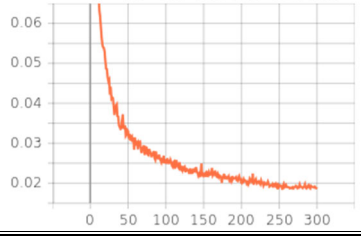
| | Baseline | Our Work | |
|---|---|---|---|
| | | Step 1 | Step 2 |
| Train-loss |  |  |  |
| Val-loss |  |  |  |

**TABLE 7.** Experimental results of proposed network for strawberry flower detection.

| Models | | precision (%) | recall (%) | F1 score | mPA (%) | inference time (ms) |
|---|---|---|---|---|---|---|
| Baseline | | 97.45% | **95.02%** | 0.96 | 97.77% | 12.44 |
| Our Work | Step 1 | 97.43% | 94.31% | 0.96 | 98.00% | 8.06 |
| | Step 2 | 97.79% | 94.31% | 0.96 | 98.13% | 8.34 |
| | Step 3 | **97.79%** | 94.31% | 0.96 | **98.13%** | **7.62** |

2.24% comparing with Step 1, caused by the skip-layers connection.

Neck is capable of generating feature maps with multi-scale information, which is crucial for improving the accuracy of object detection. Therefore, in addition to the direct numerical comparison mentioned above, our work also visualize Neck to compare the algorithm's attention to objects on feature maps of different scales. The heatmap of feature localization generated by Grad-CAM method is used to assess the effect of the improved PAN network [32]. The Grad-CAM heatmap provides a visualization method in a form of model gradients to highlight what is the DNN model focus on. The heatmaps of the outputs of different scales in baseline and our work are worked out by Grad-CAM method, as shown in Table 4.

It can be seen that the improved PAN in our work has a better ability to focus on the detection object than the original PAN. For the Head 1 of the baseline, the original PAN and improved PAN both mistakenly pay some attention to the background rather than the object features (the mistake feature is marked by red dotted box), but the correct object feature that the improved PAN focus on is more complete than that of original PAN (the correct feature is marked by white box). For the Head 2, the original PAN and improved PAN are able to pay attention to the object features, but there are more mistakes occurred in the original PAN that the backgrounds are identified as detection objects. For the Head 3, the original PAN and improved PAN are almost completely focused on the target features, but the original PAN still mistakenly pay a little attention to the background.

### C. PROPOSED NETWORK FOR STRAWBERRY FLOWER DETECTION

#### 1) TRAINING OF THE PROPOSED NETWORK

Subsequent work is the training of network using the images in the dataset. The training parameters are set as Table 5.

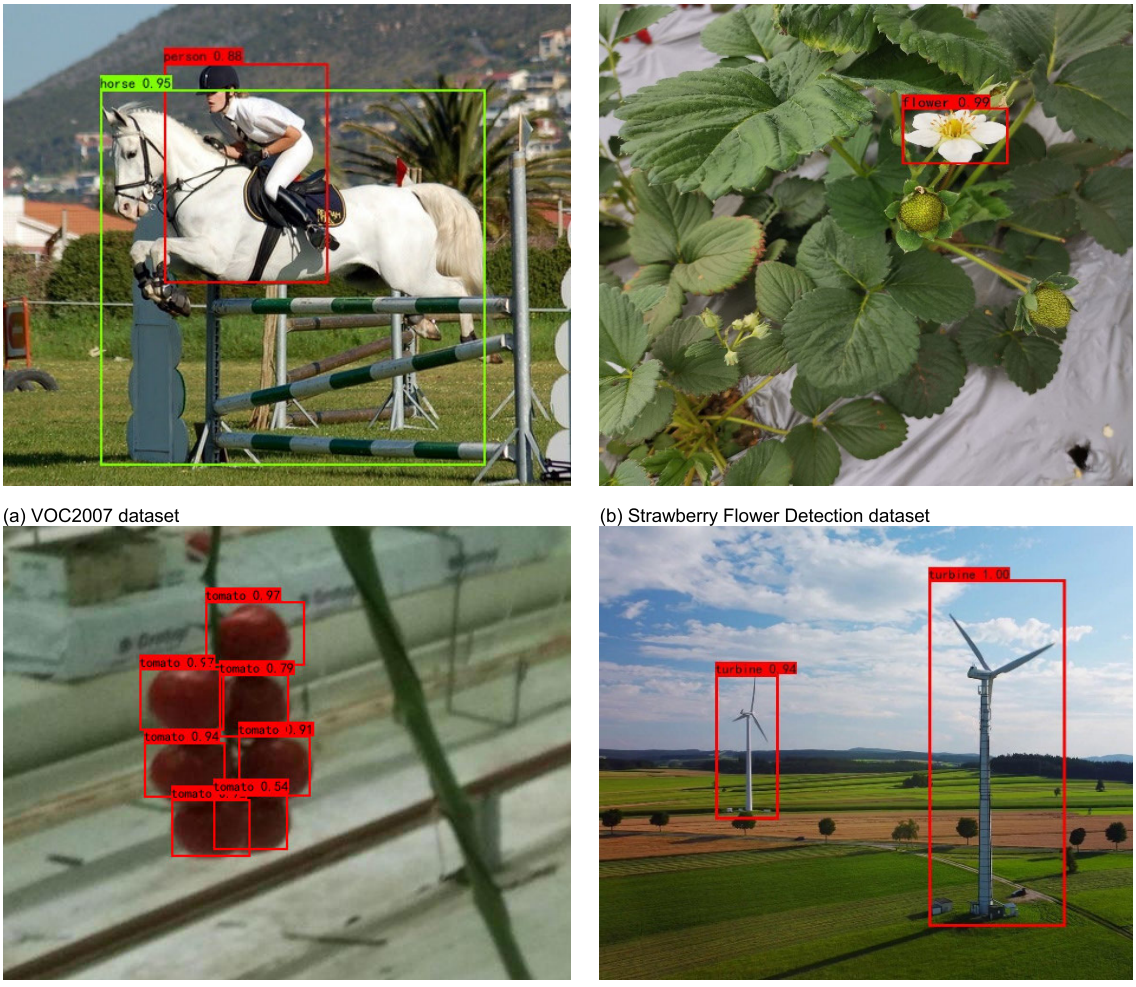
Because that only the integration of Conv layers and BN layers method is utilized in Step 3, while the modification of topological structure is uninvolved. Then, the weights of the improved lightweight parameters network after Step 3 are derived from the previous step, so there is no need to retrain. It means that the networks involved in Step 2 and Step 3 had the same training results. Therefore, the parameters of training process of baseline and parameters of the first two steps of the improved lightweight network are given in Table 6. It can be seen that the results of the three models tend to be flat in the later training stage, which means a convergent training.

#### 2) EXPERIMENTAL RESULTS OF STRAWBERRY FLOWER DETECTION

Five criteria including precision, recall, $F_1$ score, mAP and inference time are used to evaluate the performances of the algorithm for strawberry flower detection in this paper. The criteria mentioned above are used to evaluate the strategies of the improved lightweight parameters network stepwise.

**TABLE 8.** Experimental results for open source dataset.

| Dataset | | precision (%) | recall (%) | F1 score | mPA (%) |
|---|---|---|---|---|---|
| Tomato | Baseline | **87.93%** | 53.24% | 0.66 | 73.90% |
| | Our Work | 84.58% | **58.19%** | **0.69** | **75.57%** |
| Wind Turbine Detection | Baseline | **90.36%** | 78.79% | 0.84 | 85.44% |
| | Our Work | 89.84% | **82.78%** | **0.86** | **86.16%** |
| VOC2007 | Baseline | **76.46%** | 40.15% | 0.51 | 53.30% |
| | Our Work | 75.12% | **43.46%** | **0.54** | **54.49%** |




(a) VOC2007 dataset

(b) Strawberry Flower Detection dataset

(c) Tomato dataset

(d) Wind Turbine Detection dataset

**FIGURE 5.** The actual detection results on different dataset.

The parameters of simulation platform are detailed in Table 2. And the weighs of the networks used to test the performances are respectively assigned as the ones that showed the best property on the validation set, while the weights of Step 3 are derived from the previous step. The experimental results of the improved lightweight parameters network tested on the test set are shown in Table 7. As shown in Table 7, the inference time of the improved lightweight parameters network in this paper is greatly reduced comparing with the baseline, and other evaluation criteria are roughly equivalent. This means that the improved lightweight parameters network is stable and suitable to be deployed in actual scenarios, especially for the mobile platforms with limited computing power.

To verify the generality of proposed network, we have compared mAP on open source datasets, including Tomato

**TABLE 9.** Comparison of the proposed method with previous studies.

| Models | precision (%) | recall (%) | $F_1$ score | *mPA* (%) | inference time (ms) |
|---|---|---|---|---|---|
| Faster R-CNN | 77.78 | 97.15 | 0.86 | 96.94 | 82.34 |
| SSD | 95.82 | **97.86** | 0.97 | **98.34** | 11.46 |
| YOLOX-s | **97.84** | 96.80 | 0.97 | 98.05 | **5.85** |
| EfficientDet | 97.82 | 95.73 | **0.97** | 97.46 | 24.06 |
| Our Work | 97.79 | 94.31 | 0.96 | 98.13 | 7.62 |

**TABLE 10.** Hardware parameters of Jetson Nano.

| Hardware name | Parameters |
|---|---|
| GPU | NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores |
| CPU | Quad-core ARM Cortex-A57 MPCore processor |
| Memory | 4 GB 64-bit LPDDR4 |
| Storage | 128 GB |

**TABLE 11.** Inference speed comparison of the algorithms tested on Jetson Nano.

| Model | Faster R-CNN | SSD | YOLOX-s | EfficientDet | Baseline | Our work |
|---|---|---|---|---|---|---|
| Inference time | 57336 ms | 2081 ms | **909 ms** | 1284 ms | 2091 ms | 1628 ms (after Step 1)<br>1181 ms (after Step 2)<br>926 ms (after Step 3) |

**TABLE 12.** The complexity of the algorithms.

| Algorithms | Memory | FLOPs | Params |
|---|---|---|---|
| EfficientDet | 615.56 MB | **2.12 GMac** | **3.83 MB** |
| Faster RCNN | 5.19 GB | 454.29 GMac | 28.27 MB |
| YOLOX-s | **249.39 MB** | 5.63 GMac | 8.94 MB |
| SSD | 888.32 MB | 57.92 GMac | 23.75 MB |
| Our Work | 588.18 MB | 6.83 GMac | 15.12 MB |

dataset, Wind Turbine Detection dataset and VOC2007 dataset. As shown in Table 8, our work outperforms baseline in mAP, Recall and $F_1$-score.

The actual detection results of our work on different datasets are shown in Figure 5.

### D. COMPARISON OF THE PROPOSED NETWORK WITH PREVIOUS STUDIES

With the rapid development of CNN and its usage in the field of computer vision, numerous excellent object detection algorithms based on CNN were proposed by scholars. Then, four object detection algorithms with excellent performance in recent years, namely, Faster R- CNN [33], [34], SSD [35], YOLOX-s [36], [37] and EfficientDet [38], are selected to compare with our work. And the data of training set generated in Section IV-A1 are used to train the networks, while the weights of each network are respectively set as the ones that have the best property on the validation set. The precision, recall, $F_1$ score, mAP and inference time of the five object detection algorithms are shown in Table 9, respectively.

It can be seen that the SSD algorithm acquires the highest mAP of 98.34%, while the YOLOX-s algorithm acquires the minimum inference time of 5.85 ms. Our work is 0.21% lower than SSD algorithm in the mAP, while 1.77 ms greater than YOLOX-s algorithm in inference time. In other words, our work in this paper could not obtain the best scores on all criteria, especially for both the mAP and inference time. However, the overall performance of our work is better than that of the others, considering the high mAP and short inference time. Compared with other networks, our work is suitable for strawberry flower detection in natural environment.

### E. DISCUSSION

Followed by the experiments based on the platform with high computing power, the performance of the improved lightweight parameters network is further discussed on the platform with limited computing power. Therefore, the improved lightweight parameters network, baseline and also the other algorithms in previous studies, are transferred to Jetson Nano for the purpose of inference time comparison. And the hardware parameters of Jetson Nano are shown in Table 10, while the results of inference speed tested on Jetson Nano are shown in Table 11. Remarkably, the inference time of our work is only 0.44 times that of the baseline, demonstrating the effectiveness of our approach on low computing power platforms. It also can be seen that the inference time of our work is lower than Faster R-CNN, SSD and EfficientDet, but slightly higher than YOLOX-s.

The mAP of our work is higher than that of the YOLOX-s algorithm. However, the YOLOX-s algorithm has a faster inference speed than that of our work, not only for the platform with high computing power but also the embedded devices. In order to explore the impact factors of the inference speed, we choose memory footprint, FLOPs and number of parameters as indicators to analyze the complexity of algorithm which could provide a reference for the farther research of lightweight. And the comparing results of complexity are shown in Table 12.

As shown in Table 12, the comparion results of complexity are consistent with the inference speed tested on Jetson Nano. Our work has an intermediate complexity among the five algorithms, visualized as the three indicators. This may indirectly indicate that not only the accuracy but also the complexity should be taken into consideration in the lightweight design of a network. The trade-off between accuracy and complexity is essential to ensure the comprehensive performance of the network when run on platforms with limited computing power. In general, our work in this paper acquires an extremely high accuracy as well as a fast inference speed, not only for high computing power platforms but also the lightweight devices with limited computing power. Thus, that proves our work could support yield estimation for strawberry flower pollination robots or UAVs.

## V. CONCLUSION

Accurate and efficient detection of strawberry flowers is very important for yield estimation and the development of a pollination robot. Hereby, the improved lightweight parameters network is proposed in this paper. After the training and testing of the network on the strawberry flower dataset, we compared it with the baseline as well as the other algorithms in previous studies. Then the conclusions are carried out as below.

1) The improved lightweight parameters network includes backbone network lightweight design, neck architecture modification and also the integration of Conv layers and BN layers. As the results, the number of parameters, quantity of computation, memory footprint andinference time of the improved lightweight parameters network are reduced vastly while comparing with the baseline, respectively. The results indicate that the improved lightweight parameters network is suitable for the mobile pollination robots which had a high-speed requirement of strawberry flowers detection but with limited computing power.

2) The improved lightweight parameters network not only has a faster inference speed than YOLOv4, but also has a higher mAP than YOLOv4. Moreover, the improved lightweight parameters network also has a better overall performance than the other algorithms in previous studies. It shows that the improved lightweight parameters network in this paper makes the algorithm more accurate than the baseline, thus, could provide technical supports for the development of pollination robots and yield estimation of strawberry in natural environment.

## DATA AVAILABILITY STATEMENT

Wind Turbine Detection dataset could be found at:
*https://www.kaggle.com/datasets/saurabhshahane/wind-turbine-obj-detection*

VOC2007 dataset could be found at:
*http://host.robots.ox.ac.uk/pascal/VOC/voc2007/*

The Tomato dataset can be found at:
*https://www.kaggle.com/datasets/andrewmvd/tomato-detection*

The dataset of strawberry flowers can be found at:
*https://drive.google.com/drive/folders/1aT6ur3cLPp0xD0urIH6ex_mrFYkIAtm8?usp=sharing.*

Code is available at:
*https://github.com/huansu/An-Improved-Lightweight-Parameters-Network.git*

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## REFERENCES


[1] K. V. Sriram and R. H. Havaldar, "Analytical review and study on object detection techniques in the image," *Int. J. Model., Simul., Sci. Comput.*, vol. 12, no. 5, Oct. 2021, Art. no. 2150031.

[2] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020.

[3] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[4] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[5] C. Zheng, A. Abd-Elrahman, and V. Whitaker, "Remote sensing and machine learning in crop phenotyping and management, with an emphasis on applications in strawberry farming," *Remote Sens.*, vol. 13, no. 3, p. 531, Feb. 2021.

[6] S. Zhao, J. Liu, and S. Wu, "Multiple disease detection method for greenhouse-cultivated strawberry based on multiscale feature fusion faster R_CNN," *Comput. Electron. Agricult.*, vol. 199, Aug. 2022, Art. no. 107176.

[7] J. Deng, Z. Shi, and C. Zhuo, "Energy-efficient real-time UAV object detection on embedded platforms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 3123–3127, Oct. 2020.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," in *Proc. 36th Chin. Control Conf. (CCC)*, Dalian, China, Jul. 2017, pp. 11104–11109.

[10] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13728–13737.

[11] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8759–8768.

[12] Y. Liao, S. Lu, Z. Yang, and W. Liu, "Depthwise grouped convolution for object detection," *Mach. Vis. Appl.*, vol. 32, no. 6, pp. 1–13, Sep. 2021.

[13] Q. Lü, J. R. Cai, B. Liu, L. Deng, and Y. J. Zhang, "Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine," *Int. J. Agricult. Biol. Eng.*, vol. 7, no. 2, pp. 115–121, Nov. 2014.

[14] F. Kurtulmus, W. S. Lee, and A. Vardar, "Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network," *Precis. Agricult.*, vol. 15, no. 1, pp. 57–79, Feb. 2014.

[15] R. Horton, E. Cano, D. Bulanon, and E. Fallahi, "Peach flower monitoring using aerial multispectral imaging," *J. Imag.*, vol. 3, no. 1, p. 2, Jan. 2017.

[16] A. McCarthy and S. Raine, "Automated variety trial plot growth and flowering detection for maize and soybean using machine vision," *Comput Electron Agricult.*, vol. 194, Mar. 2022, Art. no. 106727, doi: 10.1016/j.compag.2022.106727.
[17] C. L. Zhou, W. S. Lee, and R. Lin, "Strawberry flower detection using fluorescence imaging," in *Proc. ASABE Annu. Int. Meeting*, St. Joseph, MI, USA, Jul. 2020, pp. 13–15.
[18] P. Lin, W. S. Lee, Y. M. Chen, N. Peres, and C. Fraisse, "A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field," *Precis. Agricult.*, vol. 21, no. 2, pp. 387–402, Apr. 2020.
[19] P. A. Dias, A. Tabb, and H. Medeiros, "Apple flower detection using deep convolutional networks," *Comput. Ind.*, vol. 99, pp. 17–28, Aug. 2018.
[20] P. A. Dias, A. Tabb, and H. Medeiros, "Multispecies fruit flower detection using a refined semantic segmentation network," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3003–3010, Oct. 2018.
[21] F. Palacios, G. Bueno, J. Salido, M. P. Diago, I. Hernández, and J. Tardaguila, "Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105796.
[22] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Instance segmentation of apple flowers using the improved mask R-CNN model," *Biosyst. Eng.*, vol. 193, pp. 264–278, May 2020.
[23] G. Li, R. Suo, G. Zhao, C. Gao, L. Fu, F. Shi, J. Dhupia, R. Li, and Y. Cui, "Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination," *Comput. Electron. Agricult.*, vol. 193, Feb. 2022, Art. no. 106641.
[24] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105742.
[25] Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, "Real-time strawberry detection using deep neural networks on embedded system (RTSD-Net): An edge AI application," *Comput. Electron. Agricult.*, vol. 192, Jan. 2022, Art. no. 106586.
[26] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, Jun. 2020, pp. 1571–1580.
[27] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856.
[28] T. Liu, S. Wang, Y. Liu, W. Quan, and L. Zhang, "A lightweight neural network framework using linear grouped convolution for human activity recognition on mobile devices," *J. Supercomput.*, vol. 78, no. 5, pp. 6696–6716, Oct. 2021.
[29] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.
[30] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, "Efficient densely connected convolutional neural networks," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107610.
[31] M. Awais, M. T. B. Iqbal, and S. Bae, "Revisiting internal covariate shift for batch normalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 5082–5092, Nov. 2021.
[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626.
[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[34] Y. Mu, R. Feng, R. Ni, J. Li, T. Luo, T. Liu, X. Li, H. Gong, Y. Guo, Y. Sun, Y. Bao, S. Li, Y. Wang, and T. Hu, "A faster R-CNN-based model for the identification of weed seedling," *Agronomy*, vol. 12, no. 11, p. 2867, Nov. 2022.
[35] H. Lu, C. Li, W. Chen, and Z. Jiang, "A single shot multibox detector based on welding operation method for biometrics recognition in smart cities," *Pattern Recognit. Lett.*, vol. 140, pp. 295–302, Dec. 2020.
[36] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[37] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, "Complete and accurate holly fruits counting using YOLOX object detection," *Comput. Electron. Agricult.*, vol. 198, Jul. 2022, Art. no. 107062.
[38] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 10778–10787.

**BAO ZHOU** received the bachelor's degree in robotics engineering from the Chongqing University of Technology, Chongqing, China. His current research interests include object detection, digital image processing, and agricultural robot.

**XUEYING LIN** is currently with the Department of Mechanical Engineering, Chongqing University of Technology, Chongqing, China. Her current research interests include digital image processing, object detection, and machine learning.

**JIE ZHOU** was born in 1986. He received the M.S. degree from Wuhan Textile University, in 2012, and the Ph.D. degree from the Huazhong University of Science and Technology, in 2018. He is currently with the Department of Mechanical Engineering, Chongqing University of Technology, Chongqing, China, where he is also an Associate Professor with the Department of Mechanical Engineering. His research interests include robotics and computer vision. His current research interests include object detection and machine learning.

**YUJIN WANG** was born in 1986. He received the M.S. degree from Wuhan Textile University, in 2012, and the Ph.D. degree from the Huazhong University of Science and Technology, in 2018. He is currently an Associate Professor with the Department of Mechanical Engineering, Chongqing University of Technology, China. His research interests include robotics and computer vision.

**FANGCHAO HU** received the B.S. and M.S. degrees in control science and engineering and the Ph.D. degree in computer science from the Chongqing University of Posts and Telecommunications, in 2011, 2014, and 2019, respectively. He is currently a Lecturer with the Department of Mechanical Engineering, Chongqing University of Technology, China. His research interests include computer vision on autonomous vehicle, vision odometry, 3D driving environment reconstruction, simultaneous localization, and mapping. He was funded by the China Scholarship Council as a co-training Ph.D. student to study with Purdue University, West Lafayette, IN, USA, from 2016 to 2017.