Article

SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode

Haiying Liu [1,*]**, Fengqian Sun** [1]**, Jason Gu** [2] **and Lixia Deng** [1]

1 School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
2 School of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3H 4R2, Canada
* Correspondence: haiyingliu2019@qlu.edu.cn

Abstract: In the research of computer vision, a very challenging problem is the detection of small objects. The existing detection algorithms often focus on detecting full-scale objects, without making proprietary optimization for detecting small-size objects. For small objects dense scenes, not only the accuracy is low, but also there is a certain waste of computing resources. An improved detection algorithm was proposed for small objects based on YOLOv5. By reasonably clipping the feature map output of the large object detection layer, the computing resources required by the model were significantly reduced and the model becomes more lightweight. An improved feature fusion method (PB-FPN) for small object detection based on PANet and BiFPN was proposed, which effectively increased the detection ability for small object of the algorithm. By introducing the spatial pyramid pooling (SPP) in the backbone network into the feature fusion network and connecting with the model prediction head, the performance of the algorithm was effectively enhanced. The experiments demonstrated that the improved algorithm has very good results in detection accuracy and real-time ability. Compared with the classical YOLOv5, the mAP@0.5 and mAP@0.5:0.95 of SF-YOLOv5 were increased by 1.6% and 0.8%, respectively, the number of parameters of the network were reduced by 68.2%, computational resources (FLOPs) were reduced by 12.7%, and the inferring time of the mode was reduced by 6.9%.

Keywords: object detection; visual tracking; small object; segmentation and categorization; YOLO

1. Introduction

Object detection aims to identify the location, size, spatial relationship, and classes of specific targets in images or video sequences. It is a cross domain of computer vision, digital image processing, and machine vision. It always had high research and application value in face recognition [1,2], industrial defect detection [3], UAV aviation detection [4,5], traffic and vehicle detection [6], pedestrian detection and counting [7], and other fields.

In recent years, with the update and iteration of high-performance GPUs, the continuous emergence and improvement of various large-scale datasets (such as DOTA [8], ImageNet [9], COCO [10]) and the strong rise of neural networks based on convolutional, object detection algorithms using deep learning techniques have become the mainstream of contemporary object detection technology.

The current common object detection algorithms can be divided into two-stage algorithm and one-stage detection algorithm according to whether candidate regions are generated or not. The representative algorithms of the former are: RCNN [11], SPP(Space Pyramid Pooling)-Net [12], Fast-RCNN [13], Faster-RCNN [14], Mask-RCNN [15], etc. The latter mainly includes YOLO (YOLOv1 [16], YOLOv2 [17], YOLOv3 [18], YOLOv4 [19], YOLOv5 [20], etc.) and SSD [21] algorithm. Although the two-stage algorithms can reach high accuracy, the prolonged time of detection makes it hard to satisfy the real-time requirement in daily object detection scenarios. Because it has the advantages of high precision

and fast speed, the one-stage detection algorithm has become the research focus in the study of object detection.

For early versions of object detection algorithms, for example, YOLOv1 [16], YOLOv2 [17], etc., its network structure is often the stacking of multiple convolution layers and the full-connection layer. Only the feature map of a fixed size is calculated, which greatly limited the multi-scale detection ability of the model. On this occasion, the detection accuracy of the algorithm for small targets is not ideal.

In the case of solving the above problems, an effective method is to adopt the image pyramid strategy [22], which can enhance the multi-scale detection ability of the model; however, the high computational cost and inference speed limit the development of the image pyramid strategy in an object detection algorithm.

Based on the aforementioned methods, FPN [23] up-samples the feature map and then fuses it with the corresponding output in the backbone network in this process, and generates multiple new feature maps for final detection. This modification improved the feature expression ability of the algorithm and further improved the effect of the algorithm based on real-time detection and multi-scale detection ability. Meanwhile, FPN adopted the top-down feature fusion path, which also strengthened the small object detection performance of the algorithm.

Due to the excellent performance of FPN, adding a feature fusion network after the backbone network of the model become the mainstream configuration of object detection algorithms. For example, YOLOv3 [18], YOLOv4 [19], and YOLOv5 [20], all use FPN as a feature fusion network. The subsequent NAS-FPN [24], ASFF [25], PANet [26], BiFPN [27], and so on have achieved better results through various feature fusion paths, but their basic ideas are the same as that of FPN.

Through the combination of well-designed backbones and feature fusion networks, the current object detection algorithms can ensure the detection accuracy and real-time capability, and have the ability to detect various scales objects; however, for the specific design, these algorithms often focus on various complex life scenarios. Their original purpose is to guarantee that the mathematical model has perfect generalization and detection ability for targets with different scales, but ignore the detection optimization in some specific scale scenes. In this context, designing a simple, efficient, lightweight, and easy-to-deploy algorithm for small object detection has important study significance.

YOLOv5 is the latest version of the YOLO series algorithms. Based on YOLOv5, this research has invented an improved YOLOv5 algorithm for small object detection with the name SF(Small-Fast)-YOLOv5, which can not only significantly reduce the number of parameters and calculated amount of the model, but also has better performance compared with the original version in the small object detection direction.

The major innovations and contributions of this research are as follows:

- In the default network structure of YOLOv5, the network layer that originally used to generate the large object detection feature map has been reasonably clipped. While realizing the lightweight model, the computing resources required by the model are effectively released and the speed of the model is greatly improved.
- Based on the PANet [26] and BiFPN [27], a new feature fusion method (PB-FPN) for small object detection is proposed, which improved the detection performance of the algorithm for small targets.
- The (SPP [12]) layer at the end of the backbone network is introduced into the feature fusion network and connected with multiple prediction heads to improve the feature expression ability of the final output feature map and further enhance the ability of the algorithm.

The other parts of this paper are arranged as follows: Section 2 introduces the basic idea and network structure of the classical algorithm YOLOv5; Section 3 illustrates the improvement strategy of SF-YOLOv5 in detail; Section 4 presents the experimental environment, dataset selection, experimental result analysis, ablation experiment, algo-

rithm comparison, and algorithm generalization ability test; Finally, Section 5 provides the conclusion, and introduces follow-up work and improvement direction.

2. Relevant Work

2.1. Some Classical Algorithms

The detection of small objects belongs to the problem of abnormal scale in object detection. Most of the classical algorithms are full-size detection algorithms, that is, the algorithm itself will give priority to ensuring the detection ability of objects of various sizes, and make various optimizations for small object detection on this basis. Common optimization strategies include the using of image pyramids [22] or FPN [23]. Adding more detection heads [4] can also attain a good effect, although it will greatly increase the calculation cost.

At present, common detection algorithms (such as [11–21], etc.) have been introduced. In addition, the latest YOLOv7 [28] (which is still under update) and various improved algorithms based on ResNet [29,30] also have excellent performance.

YOLOv5 is an algorithm with high reliability and stability, and it is easy to deploy and train. At the same time, it is also one of the one-stage detection algorithms with the highest accuracy at present; therefore, in this paper, we choose YOLOv5 for subsequent improvement.

2.2. Basic Idea of YOLOv5

YOLOv5 continues the consistent idea of the YOLO series in algorithm design: namely, the image to be detected was processed through a input layer (input) and sent to the backbone for feature extraction. The backbone obtains feature maps of different sizes, and then fuses these features through the feature fusion network (neck) to finally generate three feature maps P3, P4, and P5 (in the YOLOv5, the dimensions are expressed with the size of 80×80, 40×40 and 20×20) to detect small, medium, and large objects in the picture, respectively. After the three feature maps were sent to the prediction head (head), the confidence calculation and bounding-box regression were executed for each pixel in the feature map using the preset prior anchor, so as to obtain a multi-dimensional array (BBoxes) including object class, class confidence, box coordinates, width, and height information. By setting the corresponding thresholds (confthreshold, objthreshold) to filter the useless information in the array, and performing a non-maximum suppression (NMS [31]) process, the final detection information can be output. The process of converting the input picture into BBoxes is called the inference process, and the subsequent threshold and NMS operations are called post-processing. The post-processing does not involve the network structure. The default inference process of YOLOv5 can be represented by Figure 1.
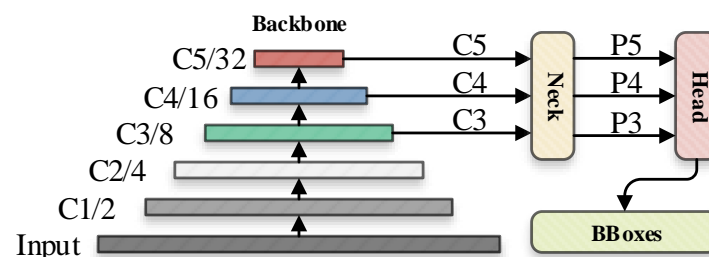

Figure 1. The default inference flowchart of YOLOv5.

In the details of bounding-box regression, different from the previous version of YOLO algorithm, the process of YOLOv5 can be explained by (1).

$$g_x = 2\sigma(s_x) - 0.5 + r_x$$
$$g_y = 2\sigma(s_y) - 0.5 + r_y$$
$$g_h = p_h(2\sigma(s_h))^2 \tag{1}$$
$$g_w = p_w(2\sigma(s_w))^2$$

In the above formula, set the coordinate value of the upper left corner of the feature map to (0, 0). r_x and r_y are the unadjusted coordinates of the predicted center point. g_x, g_y, g_w, g_h represents the information of the adjusted prediction box. p_w and p_h are for the information of the prior anchor. s_x and s_y represents the offset calculated by the model. The process of adjusting the center coordinate and size of the preset prior anchor to the center coordinate and size of the final prediction box is called bounding-box regression. There are five versions of YOLOv5, namely YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, and YOLOv5n. The performance of each version is shown in Table 1.

Table 1. Comprehensive performance of five versions of YOLOv5 on COCO dataset [20].

Model	Size (Pixels)	mAP @0.5:0.95	mAP @0.5	Time CPU b1 (ms)	Time V100 b1 (ms)	Time V100 b32 (ms)	Params (M)	FLOPS @640 (B)
YOLOv5n	640	28.0	45.7	45	6.6	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

In this paper, for the comprehensive consideration of computing resources, model parameters, detection accuracy, deployment ability, and algorithm practicability, we chose YOLOv5s as the basic algorithm and made subsequent improvements and contributions around it.

2.3. Network Structure of YOLOv5

Generally speaking, the network structure of YOLOv5 refers to the backbone and neck.

2.3.1. Backbone

The backbone of YOLOv5 is shown in Figure 2. The main structure is the stacking of multiple CBS (Conv + BatchNorm + SiLU) modules and C3 modules, and finally one SPPF module is connected. CBS module is used to assist C3 module in feature extraction, while SPPF module enhances the feature expression ability of the backbone.
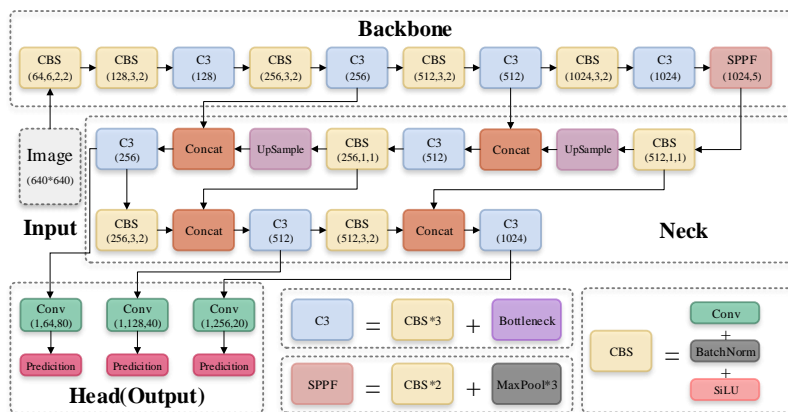

Figure 2. Default network structure of YOLOv5.

Therefore, in the backbone of YOLOv5, the most important layer is the C3 module. The basic idea of C3 comes from CSPNet (cross stage partial networks [32]). C3 can actually be regarded as the specific implementation of CSPNet. YOLOv5 uses the idea of CSPNet to build the C3 module, which not only ensures that the backbone has excellent feature extraction ability, but also curbs the problem of gradient information duplication in the backbone.

2.3.2. Neck

In the neck, YOLOv5 uses the methods of FPN [23] and PAN [26], as shown in Figure 3. The basic idea of FPN is to up-sampling the output feature map (C3, C4, and C5) generated by multiple convolution down sampling operations from the feature extraction network to generate multiple new feature maps (P3, P4, and P5) for detecting different scales targets.
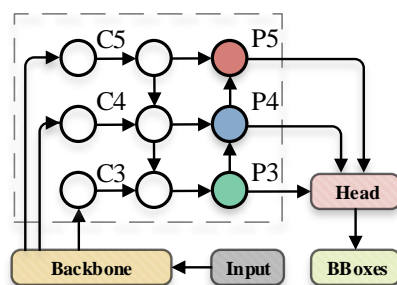

Figure 3. The dotted line in the figure is the default feature fusion path of YOLOv5.

The feature fusion path of FPN is top-down. On this basis, PAN reintroduces a new bottom-up feature fusion path, which further enhance the detection accuracy for different scales objects.

3. Approach

The performance of YOLOv5 algorithm is very excellent for object detection, but when the scenario is full of small targets, the detected results cannot obtain ideal results, so there is still has large room for improvement. In order to solve these issues, some questions were proposed:

- YOLOv5's feature extraction network set three different sizes of feature map output for detection scenes of various scales. The process of obtaining the feature map requires multiple convolution down sampling operations, which takes up a lot of computing resources and parameters; however, in the actual detection for small targets, is the down sampling operation and feature fusion process for high-level feature map necessary?
- The feature fusion network of YOLOv5 combined a top-down path (FPN) with a bottom-up path (PAN), which enhances the detection performance on various scale objects; however, for dense small object scenes, can we set a new feature fusion path to improve the feature expression potential of the output feature map? Can we fuse features horizontally to further enhance the detection performance of the algorithm?
- YOLOv5 adds an SPPF module at the end of the backbone, which improves the performance of the feature extraction network through multiple convolution and pooling operations with different sizes. Can we use this idea to further tap the feature expression potential of the feature map in the end of the neck?

Aiming at the above three questions, we proposed a novel small object detection algorithm: SF(Small-Fast)-YOLOv5. In the multiple down sampling process of the backbone, in order to bring about the lightweightness of the algorithm, we canceled the default C5 (fifth convolution down sampling operation) layer, and made corresponding adjustments in the feature fusion part, leaving only the C3 and C4 feature maps for feature fusion. In the part of the neck, based on PANet and BiFPN, we redesigned a new small object detection

feature fusion path (PB-FPN), which aims to strengthen the fusion effect of the model on the underlying features and strengthen the small object detection performance. Meanwhile, we introduce the SPPF module that originally existed only in the backbone at the connection between the neck and the head, which further tapped the feature expression potential of the output feature maps.

3.1. Feature Map Clipping

YOLOv5 outputed three sizes of feature maps by default, which are used to predict the large, medium, and small objects, respectively. The way to obtain these feature maps is to carry out continuous convolution down sampling operations and feature fusion processes. Table 2 shows the numerical visualization results of YOLOv5 default feature extraction network.

Table 2. Parameter of backbone in YOLOv5 network structure.

C	From	n	Params	Module	Arguments
0	−1	1	3520	CBS	[3, 32, 6, 2, 2]
1	−1	1	18,560	CBS	[32, 64, 3, 2]
2	−1	1	18,816	C3	[64, 64, 1]
3	−1	1	73,984	CBS	[64, 128, 3, 2]
4	−1	2	115,712	C3	[128, 128, 2]
5	−1	1	295,424	CBS	[128, 256, 3, 2]
6	−1	3	625,152	C3	[256, 256, 3]
7	−1	1	1,180,672	CBS	[256, 512, 3, 2]
8	−1	1	1,182,720	C3	[512, 512, 1]
9	−1	1	656,896	SPPF	[512, 512, 5]

As shown in Table 2, C represents the position of the module, From −1 indicates that the input of this layer comes from the previous layer, n indicates the amount of modules, params indicates the parameter usage of this layer, module indicates the name of the module, arguments represent the input channel value, output channel value, and convolution kernel attribute, respectively. The 7 and 8 layers indicates the C5 sampling layer in the feature extraction network; as can be seen, the sampling layer occupies enormous parameters.

In the proposed mathematical model, for the purpose of being lightweight, the default C5 feature map of the traditional YOLOv5 is deleted. The feature map clipping corresponding to the improved method in this section can be represented by the flowchart in Figure 4.
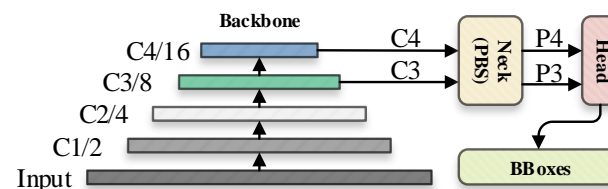

Figure 4. The inference flowchart of SF-YOLOv5. The backbone has been cut accordingly, corresponding to the improvement method proposed in Section 3.1, neck (PBS) represents a new feature fusion network with PB-FPN and SPPF prediction head, which corresponds to the methods described in Sections 3.2 and 3.3 of the improved algorithm in this paper.

3.2. Improvement for Feature Fusion Path (PB-FPN)

Early object detection algorithms usually used the high-level feature map that generated by multiple down sampling operations for detection, which will make the model unable to effectively predict objects of various scales. The introduction of FPN [23] solves this problem well, and its basic process is shown in Figure 5b.
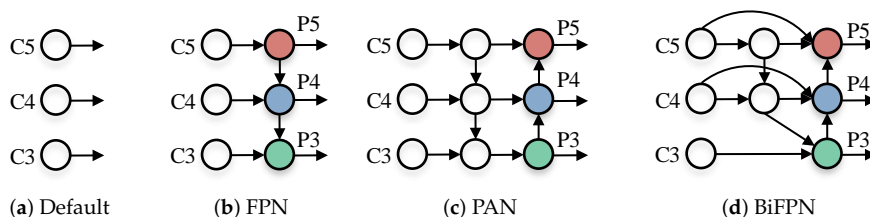
Figure 5. Some common feature fusion paths.

Taking the P4 node in Figure 5d as an example:

$$P_4^{td} = Conv\left(\frac{w_1 \cdot P_4^{in} + w_2 \cdot Resize\left(P_5^{in}\right)}{w_1 + w_2 + \varepsilon}\right) \tag{2}$$

$$P_4^{out} = Conv\left(\frac{w_1' \cdot P_4^{in} + w_2' \cdot P_4^{td} + w_3' \cdot Resize\left(P_3^{out}\right)}{w_1' + w_2' + w_3' + \varepsilon}\right) \tag{3}$$

In the (2) and (3), P_4^{td} represents the intermediate feature map between C4 and P4, P_4^{in} and P_4^{out} correspond to C4 and P4 respectively, *Conv* and *Resize* correspond to convolution and sampling operation respectively, w and ε represent weight and a preset small value to avoid numerical instability, which is usually set to 0.0001.

Based on the previous analysis, we designed an improved feature fusion path, as shown in Figure 6:
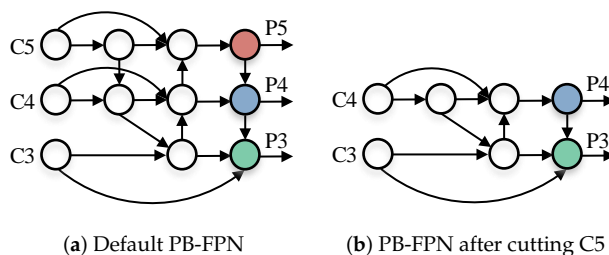

Figure 6. The feature fusion path of PB-FPN.

Compared with the classical algorithms PANet [26] and BiFPN [27], this section additionally introduces a new feature fusion path to further integrate the features from the high-level into the bottom. Meanwhile, new branches are set horizontally to participate in the fusion process of the bottom, which further enhances the detection effect of the algorithm for small targets.

3.3. The Improved Feature Fusion Network (SPPF)

The latest version of YOLOv5 used the SPPF (SPP-Fast) module. Compared with the original SPP module, its structure comparison is shown in Figure 7. By simplifying the pooling process, SPPF avoided the repeated operation of SPP and significantly improved the running speed of the module.
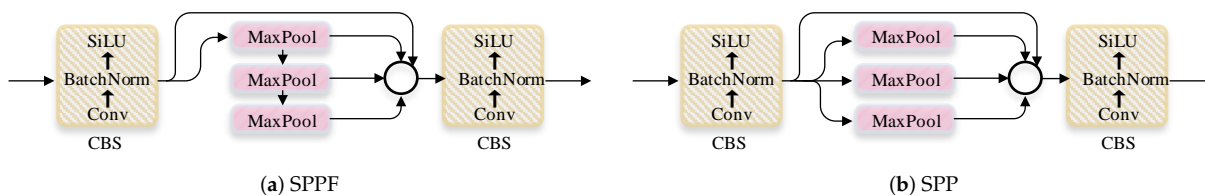

Figure 7. Structure comparison of SPPF and SPP.

In this paper, we introduced several SPPF modules at the connection between the feature fusion network and the model prediction head, in order to further tap the feature expression potential of the finally output feature map by the neck and sent to the head, and further enhance the performance of the model. The modified network in this section can be expressed in Figure 8.
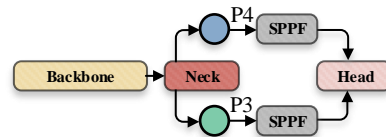

Figure 8. Structure diagram after adding SPPF at the junction of feature fusion network and prediction head.

4. Experiments

4.1. Experimental Environment

In this experiment, the workstation is win10, 16 GB, the CPU is used AMD-4800H, the GPU is used Nvidia-Rtx2060. The algorithm is based on PyTorch, using CUDA11.1 to operation acceleration. The training epoch is set to 300, and the mosaic data enhancement and a prior anchor adaptive adjustment strategy are used.

4.2. Dataset Introduction

In order to testify the performance of the improved algorithm (SF-YOLOv5) for the small object detection, this study adopted the WIDER FACE [33] dataset to train and verify the algorithm. This dataset contains the annotation information of 393,703 faces. As shown in Figure 9, the dataset is highly variable in scale, posture, angle, light, and occlusion. The dataset is complex and includes a great quantity of dense small targets.

Figure 9. The WIDER FACE dataset contains a variety of complex and rich scenes [33].

Considering that the SF-YOLOv5 mainly detects small objects, while the original WIDER FACE dataset contains not only small targets, but also large targets, which will affect the experimental results; therefore, we first screened according to the number and size of face images in a single image in the original dataset. In total, 4441 images were used to train the algorithm and 1123 images were used to verify the algorithm. The number proportion of the train and verify was 4:1, same as the original dataset. Figure 10 is an attribute visualization result of the new dataset.
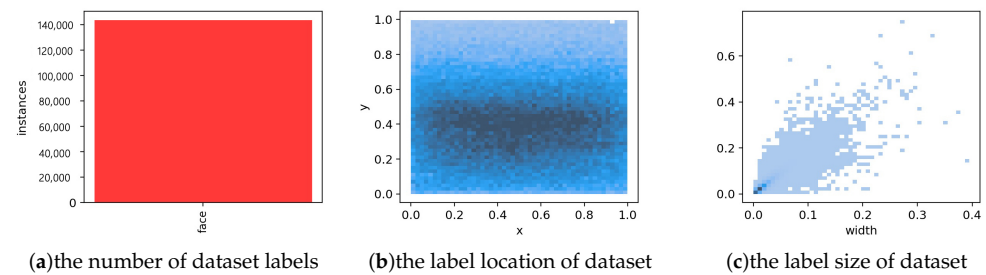
(**a**)the number of dataset labels　　　(**b**)the label location of dataset　　　(**c**)the label size of dataset

Figure 10. The attribute visualization results of the dataset used in this paper.

Figure 10a expressed the number of labels in the dataset. Figure 10b shows the central coordinate position of the object. Figure 10c shows the size of the object. It can be seen that this dataset is sufficient to ensure the training and verification of small object detection algorithm.

4.3. Evaluation Criterion

At present, the mainstream general indicators for evaluating the performance of object detection algorithms include precision, recall, AP (average precision), mAP (mean AP), parameters(the number of parameters in model), FLOPs (floating point operations per second), inference time, etc.

Considering that mAP can more reasonably reflect the accuracy of the algorithm than precision and recall, in this experiment, we choose mAP@0.5, mAP@0.5:0.95, parameters, FLOPs, and inference time to evaluate the algorithm.

4.4. Analysis of Experimental Results

The experimental performance comparison between YOLOv5s and SF-YOLOv5 is illustrated in Table 3. Compared with the traditional algorithm YOLOv5s, the mAP@0.5 and mAP@0.5:0.95 of SF-YOLOv5 has been increased by 1.6 and 0.8, respectively, which proved the improvement in comprehensive detection performance of SF-YOLOv5. At the same time, the parameters (M) value and FLOPs (G) value of SF-YOLOv5 are reduced by 68.2% and 12.7%, respectively, indicating that SF-YOLOv5 can further decrease the number of parameters and lessen computing power required for model operation under the condition of improving the detection accuracy, which enhances the deployment performance of the algorithm on various low-end workstations and small mobile devices. The reduction in inference time (ms) shows that the model can process more images and videos in the same time, which ensures the speed of SF-YOLOv5 in detecting small objects.

4.5. Ablation Experiment

Table 3 expressed the ablation experimental data of improved methods 1, 2, and 3 proposed in this paper. YOLOv5-N5 is a new algorithm obtained by cutting the feature layer with the default YOLOv5, which corresponds to the improved method 1 proposed in this paper. It is not difficult to find that although this improvement has increased by 0.2 in mAP@0.5, the value of mAP@0.5:0.95 has decreased by 0.2. The calculation formula of GIOU [34] used by YOLOv5 is shown in (4).

$$L = 1 - IoU + \frac{|A - P \cup P^{gt}|}{|A|} \tag{4}$$

where P and P^{gt} express the area of the prediction box and ground truth box, respectively, and A is the minimum area of the area composed of P and P^{gt}. We hope the value of L to be infinitely close to 0. In this case, the lower the value of IOU, the lower the requirement for the position coincidence when verifying the accuracy of the algorithm.

Table 3. Performance comparison of SF-YOLOv5 and other algorithms on small object datasets.

Methods	Size	mAP@0.5	mAP@0.5:0.95	Parameters (M)	FLOPs (G)	Inference Time (ms)
YOLOv5s	640	69.7	35.5	7.01	15.8	13.1
YOLOv5-N5	640	69.9	35.3	1.88	11.2	10.4
YOLOv5-PB	640	70.8	36.0	2.03	12.7	11.3
SF-YOLOv5	640	71.3	36.3	2.23	13.8	12.2
improvement	-	+1.6	+0.8	−68.2%	−12.7%	−6.9%
YOLOv5n	640	63.9	30.8	1.76	4.20	8.60
SF-YOLOv5	640	71.3	36.3	2.23	13.8	12.2
improvement	-	+7.4	+5.5	+26.7%	+228.6%	+41.9%
YOLOv7-tiny	640	68.4	33.0	6.01	13.0	13.9
SF-YOLOv5	640	71.3	36.3	2.23	13.8	12.2
improvement	-	+2.9	+3.3	−62.9%	+6.2%	−12.2%
YOLOv3	640	74.5	39.5	61.5	154.7	44.7
SF-YOLOv5L	640	75.1	39.7	15.5	91.2	49.4
improvement	-	+0.6	+0.2	−74.8%	−41.0%	+10.5%
YOLOv7	640	76.1	39.5	36.5	103.2	18.0
SF-YOLOv5L	640	75.1	39.7	15.5	91.2	49.4
improvement	-	−1.0	+0.2	−57.5%	−11.6%	+174.4%
ResNeXt-CSP	640	73.7	37.6	31.8	58.9	32.6
SF-YOLOv5L	640	75.1	39.7	15.5	91.2	49.4
improvement	-	+1.4	+2.1	−51.3%	+54.8%	+51.5%

It can be seen from Table 3 and formula (4), the improved method 1 can achieve significant weight improvements after removing the high-level feature map. Although the detection performance has been slightly affected: the model can identify more objects; the position of the prediction box will deviate to a certain extent.

YOLOv5-PB is a new algorithm obtained by setting the feature fusion path as PB-FPN on the basis of the improved method 1, corresponding to the improved method 2 in this paper. Compared with YOLOv5-N5, this algorithm slightly increases the computational resources and number of parameters, but greatly enhances the detection accuracy, the value of mAP@0.5 and mAP@0.5:0.95 by 0.9 and 0.7, respectively. It proves the feasibility of introducing a more efficient feature fusion path to enhance the ability of algorithm.

The final SF-YOLOV5 was obtained by adding SPPF module at the junction between the neck and head of YOLOV5-PB, corresponding to the improved method 3. Compared with the improved method 2, YOLOV5-PB slightly increases the number of parameters and further enhances the detection accuracy of the algorithm.

Figure 11 shows the visualization of ablation experimental data of improved methods 1, 2, and 3. Figure 12 shows the detection results of SF-YOLOv5.
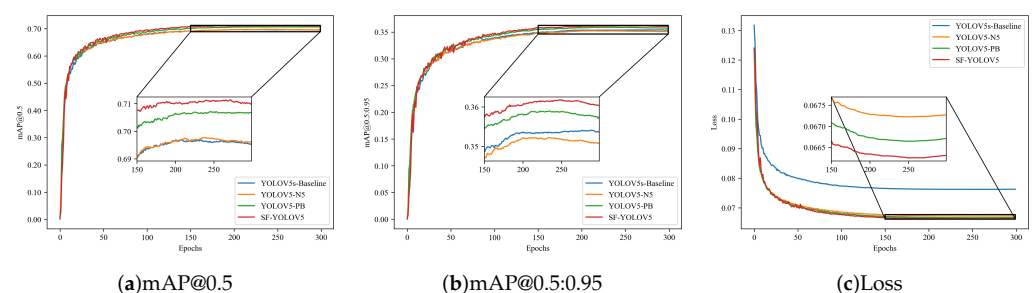

(**a**)mAP@0.5　　　　　　　(**b**)mAP@0.5:0.95　　　　　　　(**c**)Loss

Figure 11. The ablation experimental results of the improved methods 1, 2, and 3.

Figure 12. Detection effect of SF-YOLOv5 on WIDER FACE dataset.

4.6. Comparison with Other Classical Algorithms

We compare SF-YOLOv5 with several classical detection algorithms, and the results are expressed in Table 3. Among them, YOLOv5n is the latest lightweight algorithm of YOLOv5, and YOLOv3 is a relatively mature large-scale one-stage detection algorithm. YOLOv7 [28] is the latest algorithm of YOLO family at present, which has the strongest comprehensive performance in full-scale detection, and YOLOv7-tiny is a lightweight version of YOLOv7, which has similar parameter quantities and calculation quantities with SF-YOLOv5. ResNeXt-CSP is a new detector combined with classical algorithms ResNeXt [29] and CSPNet [32], which have excellent performance. It is not difficult to find that although YOLOv5n is lighter and faster, the detection accuracy is too low compared with SF-YOLOv5. Compared with YOLOv7-tiny, SF-YOLOv5 has advantages in detection accuracy, speed, and lightweight effect in the small object dataset, but it slightly increases the amount of calculation.

Because YOLOv3, YOLOv7, and ResNeXt-CSP are not lightweight algorithms, it is difficult to directly compare with SF-YLOv5, so we adjusted the scaling coefficient of SF-YOLOv5. This operation is adopted in YOLOv4, YOLOv5, and YOLOv7. The default value (0.33, 0.25) is adjusted to the same (1, 1) as YOLOv7. The adjusted SF-YOLOv5 was named SF-YOLOv5L. It has the same network structure and improvement idea as SF-YOLOv5, and the only difference is in the model size. The comparison of these algorithms is shown in Table 3. On the whole, SF-YOLOv5L is better than YOLOv3, ResNeXt-CSP, and its performance is close to that of the latest YOLOv7. This proved the feasibility and reliability of the improved method proposed in this paper.

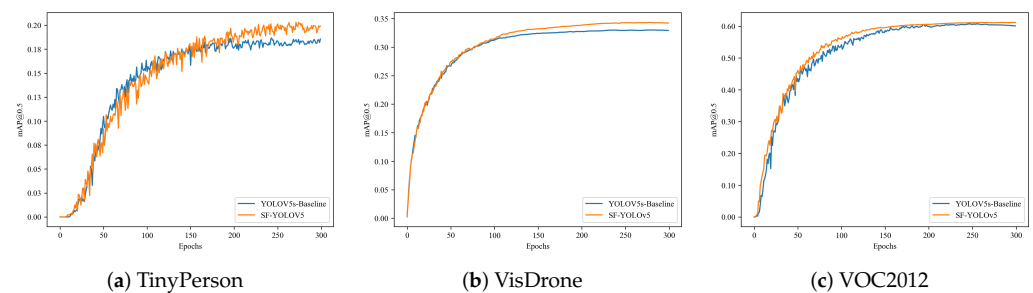*4.7. Performance of Novel SF-YOLOv5 on Other Datasets*

In this section, we further prove the ability of SF-YOLOv5 on other datasets, so as to judge whether the improvement made by SF-YOLOv5 in the direction of small object detection is universal.

TinyPerson [35], VisDrone [36], and VOC2012 [37,38] public datasets were selected for testing experiment in this section. Among them, TinyPerson and VisDrone contain a large number of small targets, which is suitable for the verification of SF-YOLOv5 proposed in this paper. Because there are not only small targets, but also a large number of large targets in VOC2012, it is mainly used to verify the performance of improved methods 2 and 3 proposed in this paper in the full-scale detection; therefore, in the experiment for VOC2012, we fine tuned the prediction head of SF-YOLOv5 and re added the P5 detection head. The algorithm is named SF-YOLOv5-P5. Table 4 shows the performance comparison of SF-YOLOv5 and YOLOv5s on these datasets.

Table 4. Performance comparison between YOLOv5s and SF-YOLOv5 on other datasets.

Dataset	Methods	Size	mAP@0.5	mAP@0.5:0.95	Parameters (M)	FLOPs (G)	Inference Time (ms)
TinyPerson	YOLOv5s	640	18.7	6.0	7.02	15.8	12.8
	SF-YOLOv5	640	20.0	6.5	2.23	13.8	11.0
	improvement	-	+1.3	+0.5	−68.2%	−12.7%	−14.1%
VisDrone	YOLOv5s	640	33.0	17.9	7.04	15.9	12.6
	SF-YOLOv5	640	34.3	18.2	2.24	13.8	11.5
	improvement	-	+1.3	+0.3	−68.2%	−13.2%	−8.7%
VOC2012	YOLOv5s	640	60.8	37.0	7.06	15.9	25.8
	SF-YOLOv5-P5	640	61.2	38.3	4.59	15.7	24.8
	improvement	-	+0.4	+1.3	−34.9%	−1.3%	−3.9%

To sum up, SF-YOLOv5 still achieved improvement in detection accuracy and comprehensive performance on TinyPerson, VisDrone and VOC2012 datasets. It is proved that the improved methods 1, 2, and 3 have obvious improvement effect in the direction for small object detection and have certain versatility in full-scale detection. Other experimental results in this section are shown in Figures 13 and 14.

(**a**) TinyPerson   (**b**) VisDrone   (**c**) VOC2012

Figure 13. Comparison of detection accuracy between SF-YOLOv5 and YOLOv5s on other datasets.

Figure 14. Detection effect of SF-YOLOv5 on other dataset.

5. Conclusions

In this paper, a lightweight small object detection algorithm based on an improved feature fusion mode is proposed, which is dedicated to improving the detection effect on small targets. When the algorithm is significantly lighter, the detection accuracy is also significantly improved. By reducing the convolution down sampling operation in the network structure and trimming the output feature map, the model parameter amount and computation amount are significantly reduced under the premise of ensuring the detection accuracy. Ablation experiments show that the contribution of high-level feature maps generated by multiple convolution downsampling can be replaced by more effective feature fusion methods when detecting small objects. Besides this, in order to improve the small object detection accuracy of the model, a new feature fusion method (PB-FPN) is proposed based on PANet and BiFPN. By setting a fast SPP operation at the junction of the neck and the head, the feature expression potential of the output feature map of the backbone is fully exploited, and the detection performance of the algorithm is further

improved. The revised model has good generalization ability and can also be applied to small object detection scenarios of other types of targets.

In the follow-up research, we will continue to improve the detection effect of the algorithm on ultra-small targets, and we will further explore the detection potential of the small object detection feature fusion method (PB-FPN) proposed in this paper on full-scale targets.

Author Contributions: Supervision, H.L.; Writing—original draft, F.S.; Writing—review and editing, J.G.; Validation, L.D. All authors have read and agreed to the published version of the manuscript.

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

References

1. Kowalski, M.; Grudzień, A.; Mierzejewski, K. Thermal–Visible Face Recognition Based on CNN Features and Triple Triplet Configuration for On-the-Move Identity Verification. *Sensors* **2022**, *22*, 5012. [CrossRef] [PubMed]
2. Zhang, Z.; Shen, W.; Qiao, S.; Wang, Y.; Wang, B.; Yuille, A. Robust face detection via learning small faces on hard images. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1361–1370.
3. Bai, T.; Gao, J.; Yang, J.; Yao, D. A study on railway surface defects detection based on machine vision. *Entropy* **2021**, *23*, 1437. [CrossRef] [PubMed]
4. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
5. Zhang, P.; Zhong, Y.; Li, X. SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.
6. Gu, Y.; Si, B. A novel lightweight real-time traffic sign detection integration framework based on YOLOv4. *Entropy* **2022**, *24*, 487. [CrossRef] [PubMed]
7. Liu, W.; Liao, S.; Ren, W.; Hu, W.; Yu, Y. High-level semantic feature detection: A new perspective for pedestrian detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 5187–5196.
8. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
9. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
10. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
11. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
12. 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] [PubMed]
13. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
17. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
18. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
20. Glenn, J. Yolov5-6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. 2022. Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.1 (accessed on 22 February 2022).

21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European conference on computer vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.

22. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3578–3587.

23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

24. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7036–7045.

25. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.

26. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.

27. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.

28. 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. *arXiv* **2022**, arXiv:2207.02696.

29. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.

30. Zhou, T.; Zhao, Y.; Wu, J. Resnext and res2net structures for speaker verification. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 301–307.

31. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 850–855.

32. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 390–391.

33. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.

34. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 658–666.

35. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1257–1265.

36. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef] [PubMed]

37. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. Available online: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (accessed on 30 July 2022).

38. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]