Multimedia Systems (2023) 29:49–58
https://doi.org/10.1007/s00530-022-00983-x

REGULAR PAPER


# Multiscale object detection based on channel and data enhancement at construction sites

Hengyou Wang[1,3] · Yanfei Song[1] · Lianzhi Huo[2] · Linlin Chen[1,3] · Qiang He[1,3]

Received: 4 December 2021 / Accepted: 10 July 2022 / Published online: 28 July 2022


**Abstract**
Object detection based on computer vision techniques plays an important role in the safety monitoring of large-scene construction sites. However, current object detection algorithms typically have poor performance on small targets. In this study, an enhanced multiscale object detection algorithm is developed to solve the problem of poor detection performance due to scale changes at construction sites. First, a scale-aware data automatic augmentation is defined to learn a data augmentation strategy. Then, to mitigate information loss caused by channel reduction when using feature pyramid network, we propose a method based on subpixel convolution to perform channel enhancement and upsampling, and add a bottom-up path to enhance the complete feature hierarchy with accurate localization signals in the lower layers. Experimental results show that the proposed algorithm achieves better accuracy on the construction site (MOCS) data set and the MS COCO data set. For example, compared with the Faster R-CNN detector with the ResNet-50 backbone network on the MOCS data set and MS COCO data set, the average accuracy increased by 8.0% and 1.5%, respectively. In particular, the average accuracy of small targets increased by 10.3% and 3.4%, respectively.

**Keywords** Multiscale object detection · Data enhancement · Feature pyramid · Subpixel convolution · Channel enhancement

## 1 Introduction

In recent years, object detection has become a hot research topic in the field of computer vision, and its main task is to classify and locate the target object in images or videos. With the development of deep learning, object detection algorithms based on convolutional neural networks (CNNs) have made marked progress. Object detection has been widely used in video surveillance [1], face detection [2], intelligent robots [3], safety monitoring of construction sites [4–6] and many other fields.

To ensure the safety monitoring of construction sites, many applications have been developed, such as hazard prevention [4], earth-moving equipment analysis [5] and helmet wear detection [6]. However, these detectors are trained and tested on their own data sets, which are small-scale data sets that contain only a few hundred images. They are also customized for specific objects from limited locations, viewpoints, or weather and lighting conditions. Therefore, detectors trained with limited data sources yield high variances and be difficult to generalize to typical construction environments. To solve the problem of safety monitoring at construction sites, a new image data set called the image data set of moving objects in construction sites (MOCS) [7] was presented by Tsinghua University. This data set consists of five different types of equipment and various viewpoints, lighting conditions and weather conditions and contains 41668 images collected from 174 construction sites. The data set also accurately annotates 13 moving target objects and contains 222,861 annotated examples, and the training and validation data set is available for use by any researchers.

Communicated by B-K Bao.

✉ Hengyou Wang
wanghengyou@bucea.edu.cn

[1] School of Science, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

[2] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China

[3] Institute of Big Data Modeling and Technology, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

Although the construction sites data set has been built, existing object detection methods still cannot achieve ideal performance with small targets, which makes it difficult to use these methods to monitor construction site safety. Small objects are represented by only a few pixels in an image and are thus described by very little information. Thus, small-scale object classification [8, 9] do not benefit from deeper network architecture, more filters, and larger filter sizes. Thus, the problem of multiscale target detection remains difficult and challenging issue in object detection.

To solve the problem of low detection accuracy caused by scale changes and small object instances, some previous studies primarily investigated network structures and data augmentation methods. In terms of network structure, feature pyramid networks(FPNs) [10] and adaptive receptive fields are commonly used, which make the network scale invariant. There are also some studies that used FPNs, such as PANet [11], Libra R-CNN [12] and NAS–FPN [13]. In terms of data augmentation, current data augmentation methods can be divided into color operations and geometric operations. Previous data augmentation studies [14, 15] also improved frame-level enhancement by enriching foreground data. However, these data augmentation methods typically rely on a large amount of expert experience. There are still some limitations in improving the detection accuracy only through data enhancement or network structure improvement. In particular, the detection of small objects still has some room for improvement. Although the FPN network has brought a certain improvement to multi-scale target detection, its network still has some limitations. For example, the number of channels from the $C_5$ layer to the $F_5$ layer is reduced from 2048 to 256. This large reduction in the number of channels leads to serious loss of channel information. In addition, the images of construction site are often affected by illumination, occlusion and other factors. Especially when the site is large enough, some targets are far away from the camera and will become small targets in the image, which brings challenges to target detection. Thus it is necessary to enhance the images first. In this work, we develop an enhanced multi-scale object detection algorithm. First, it enhances the input images for the problem of small target pixels. Inspired by [16], we define a scale-aware automatic enhancement method, which seeks the best augmentation strategy through evolutionary algorithms without requiring a large number of hyperparameters. Second, we design an improved feature pyramid network based on subpixel convolution and bidirectional fusion to reduce the loss of semantic information due to channel reduction. Specifically, in the stage from $C_5$ to $P_4$ of the FPN network, the sub-pixel upsampling method is used to replace the traditional bilinear interpolation upsampling method. Therefore, in the convolution process from $C_5$ to $F_5$, the number of channels only be reduced to 1024, which reduces the loss of channel information to a certain extent. Inspired by [11], we also enhance the entire feature hierarchy via bottom-up path augmentation. Finally, to verify the performance of the proposed methods, extensive experiments are performed on the MOCS data set and MS COCO data set. Furthermore, we also compare the proposed method with the yolo series of detectors and the Faster R-CNN detectors with conventional multi-scale training methods. Experimental results demonstrate that the proposed algorithm can effectively improve the average accuracy of the detector on the construction site (MOCS) data set and the MS COCO data set.

## 2 Related work

Recently, deep learning based object detection algorithms primarily include one-stage detection algorithms and two-stage detection algorithms. The former uses a CNN to obtain features of different scales and then finds target positioning and classification. These classical algorithms mainly include YOLOv3 [17], SSD [18], DSSD [19], etc. The latter first generates a series of candidate regions and then uses a CNN to extract features. Finally, category classification and position regression were performed on each candidate region. The R-CNN series algorithms are well-known algorithms among the two-stage detection algorithms, including R-CNN [20], Fast R-CNN [21], Faster R-CNN [22], and Mask R-CNN [23]. The two types of methods have certain advantages: the former is superior in algorithm speed, and the latter is superior in detection accuracy and positioning accuracy. However, the scale change of objects in real scenes brings some challenges to object detection. Previous researchers have mainly addressed this problems from two aspects: network improvement and data augmentation.

For network structure improvements, a feature pyramid network (FPN) is a typical method, which uses a top-down fusion path to build multiscale features by spreading semantic information from the high level to the low level, which could improve the overall performance of the detector. PANet [11] adds an additional top-down path to increase the deep low-level information. Libra R-CNN [12] is designed with a more balanced feature pyramid and uses equal attention to multiscale features through feature integration and refinement. NAS–FPN [13] uses neural architecture to learn better integration between all cross-scale network structure connections. AugFPN [24] proposed a series of enhancement methods for FPN. Although detection accuracy can be improved by improving the FPN, there is still some room for improvement in security monitoring at construction sites.

Regarding data enhancement methods, such as SNIPER [25], only the context area around the real target of the image pyramid was processed, and combined with the idea of SNIP [26]. SNIP [26] only selects real samples that are

of the appropriate scale to participate in model training to achieve better detection. Dwibedi [15] et al. improved detection performance using cutting and pasting strategies on the enhancement of the anchor frame level. InstaBoost [27] uses annotated instance masks with position probability maps to enhance training images. However, these data enhancement methods require hyperparameter adjustment and expert experience. Yukang Chen et al. [16] proposed a data enhancement method that can automatically perceive scales and avoid setting a large number of hyperparameters.

## 3 Enhanced multiScale faster R-CNN

Inspired by existing methods, we combine network structure improvement with data enhancement methods and propose an enhanced multiscale object detection method that has a better performance on the MOCS and MS COCO data sets, and its framework is shown in Fig. 1. This method improves the application of deep learning in the field of architecture.

In this section, we introduce the proposed enhanced multiscale Faster R-CNN object detector, which considers the influence of scale-aware data automatic augmentation and makes full use of channel information based on subpixel convolution and bidirectional fusion. This proposed method is referred to as EMS Faster R-CNN. Then, the data enhancement module and improved FPN module are introduced in subsections 3.1 and 3.2, respectively.

### 3.1 Scale-aware data automatic augmentation method

Automatic enhancement methods typically find the best strategy [28, 29], which can be described as a search problem, which contains a search space, search algorithm, and estimation metric. For search algorithms, reinforcement learning [30] and evolutionary algorithms [31] are typically used to explore the search space. When finding the best strategy, each submodel that is optimized using the searched strategy $p$ is evaluated according to the given metric to estimate its effectiveness.

To solve the problem of scale changes, the search space set used in this study includes both data enhancement at the image level and anchor box level. Common object detectors typically use image pyramids to deal with scale changes; thus, image-level enhancement in the search space includes the functions of zoom in and zoom out. We define $K$ ($0.5 \le K \le 1.5$) as the ratio of zoom-in and zoom-out; when $0.5 \le K \le 1$, the input image is zoomed-in and when $1 \le K \le 1.5$, the input image is zoomed-out. In addition, whether the input image is zoomed-in or zoomed-out is represented by the probability $P$, and the probability of not performing any operation is represented by $P_{ori}$. The probability of performing the zoom-in operation is represented by $P_l$, and the probability of performing the zoom-out operation is represented by $P_s$, where $0 \le P_l, P_s \le 0.5$, and satisfies

$$P_{ori} = 1 - P_l - P_s \tag{1}$$

Therefore, the zoom-in and zoom-out operations on the input image are determined by $K$ and $P$. In each training iteration,

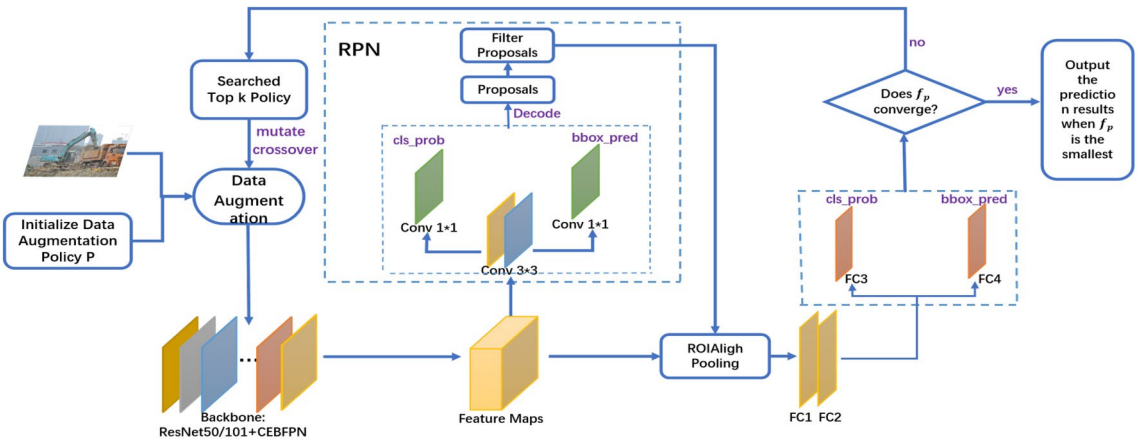



**Fig. 1** Proposed EMS Faster R-CNN framework. Input the original image data, including the training set and validation set, and initialize the data enhancement substrategy set P. Then the enhanced image data are input to the backbone network, RPN network, ROI pooling layer and two fully connected layers, and then use two fully connected layers for classification and regression, where *cls_prob* and *bbox_pred* represent the tensors of classification probability scores and prediction box regression parameters, respectively. [22]. Next, we evaluate this model by $f_p$ (4); select the data enhancement strategies with the top k scores as the parents of the next generation; generate the new child strategies through mutation and crossover between parent strategies; perform the previous operations until $f_p$ convergence is reached; and finally output the prediction result when $f_p$ is the smallest

the searched $K$ and $P$ are used to randomly sample from the zoom-in, zoom-out, and original images. In addition, to reduce the additional computational complexity of large-scale images, we retain the original shape by performing random cropping in the zoom-in function.

The enhancement of the anchor box level can be divided into two types: geometric operations and color operations. Each operation has two parameters, which are denoted as probability $P$ and amplitude $M$. Specifically, geometric operations can be divided into six types: horizontal flip, Rotate, ShearX, ShearY, TranslateX, and TranslateY. Color operations can be divided into eight types: Brightness, Color, Contrast, Cutout, Equalize, Sharpness, Solarize, and SolarizeAdd. The probability and amplitude are determined by $P_i(1 \le i \le 14, i \in Z)$ and $M_i(1 \le i \le 14, i \in Z)$, respectively. Because the traditional data enhancement at the box-level is often rectangular, there will be an obvious hard boundary between the enhanced and original areas. Thus, a Gaussian map was used to blend the original and transformed pixels, so that the boundary was smoother. Specifically, the enhanced area is determined by the following formula:

$$R = r(x, y) \cdot I + (1 - r(x, y)) \cdot C \tag{2}$$

where $R$ is the region to be enhanced; and $r(x, y)$, $I$ and $C$ are the Gaussian map, input image and transformation function, respectively.

We denote the height and width of the given image as $H$ and $W$. The anchor box can be defined by the four parameters, $(x_c, y_c, h, w)$, where $x_c$ and $y_c$ are the horizontal and vertical coordinates of the center of the box, respectively. $h$ and $w$ are the height and width of the box, respectively, and $r(x, y)$ can be calculated as follows:

$$r(x, y) = \exp(-(\frac{(x - x_c)^2}{2\sigma_x^2} + \frac{(y - y_c)^2}{2\sigma_y^2})) \tag{3}$$

To be consistent with the conventional method [29], anchor box level enhancement also has five substrategies, which are all composed of color operations and geometric operations. Each operation contains two parameters, probability and amplitude, where the probability is sampled from the set 0 to 1.0 with 0.2 intervals. Amplitude is set as the six discrete values, from 0 to 10 with 2 intervals.

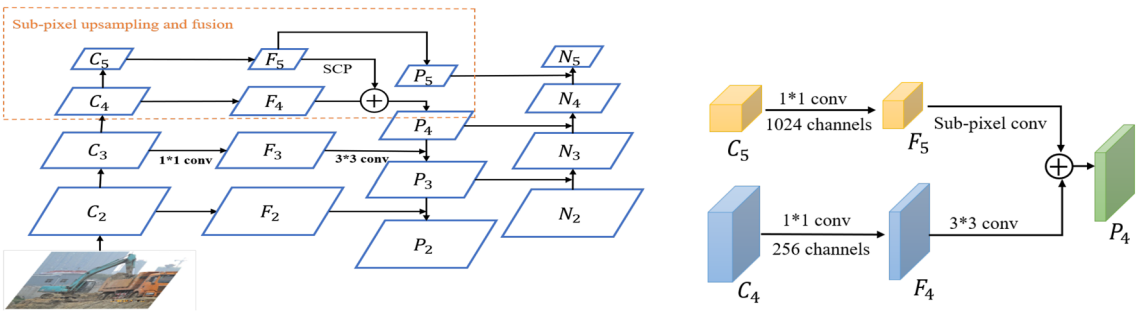
The verification accuracy of a small subset of training images is often used as the search metric of automatic enhancement methods. However,the scale change cannot be described in this manner. Inspired by [16], we use the scale perception index to evaluate the search policy. Specifically, a concept called Pareto scale balance is introduced to describe the goal of this study: optimization at a given scale cannot be improved without compromising the accuracy of any other scale. Thus, penalty factor $\Phi$ is used to punish the scale $\hat{S}$ whose accuracy drops after fine-tuning by the strategy $p$; thus, the objective function of the measurement is formulated as follows:

$$\min_p f(\{L^p_{i\in S}\}, \{AP^p_{i\in S}\}) = \min_p \sigma(\{L^p_{i\in S}\}) \cdot \phi(\{AP^p_{i\in \hat{S}}\}) \tag{4}$$

where $AP^p_{i\in S}$ is the average verification accuracy of each scale when the strategy is set as $p$; $\sigma(\{L^p_{i\in S}\})$ is the standard deviation of the loss for different scales; $\Phi\left(\left\{AP^p_{i\in \hat{S}}\right\}\right) = \prod_{i\in \hat{S}} \frac{AP_i}{AP^p_i}$); and $\frac{AP_i}{AP^p_i}$ is the ratio of the original and fine-tuning precision.

## 3.2 Feature pyramid network of channel enhancement bidirectional fusion based on subpixel convolution

In this section, we introduce the feature pyramid network of channel enhancement bidirectional fusion based on subpixel convolution (CEBFPN). As shown in Fig. 2a, we use $\{C_2, C_3, C_4, C_5\}$ to represent the output feature layers of the backbone network. The step size of downsampling is defined as $\{4, 8, 16, 32\}$ for the input image, the number of channels is set as $\{256, 512, 1024, 2048\}$, and then the number of channels can be reduced by 1*1 convolution to obtain the feature layers $\{F_2, F_3, F_4, F_5\}$. However, with the




(a) CEBFPN framework (b) subpixel convolutional upsampling fusion

**Fig. 2** CEBFPN framework and its subpixel convolutional upsampling fusion method. (Where SCP is subpixel convolution.)

traditional FPN, the number of reduced channels is set as {256, 256, 256, 256}, whose channel reduction resultes in serious information loss. Thus, to reduce the loss of semantic information caused by channel reduction, we set the number of reduced channels of the $C_5$ layer to 1024, and the other layers retain 256 channels. Then, through subpixel convolution, the $F_5$ feature layer is upsampled and merged with the $F_4$ layer to obtain the feature layer $P_4$. Thus, making full use of the channel of the layer $C_5$ information, the $P_3$ and $P_2$ feature layers can be obtained through the top-down path. In addition, we use bottom-up path enhancement with low-level accurate positioning signals to enhance the entire feature level. To make the proposed algorithm easier for readers to understand, the subpixel convolution and its feature fusion method will be introduced in detail.

**Subpixel convolution:** Subpixel convolution is a clever method of image or feature map enlargement that can convert a low-resolution image into a high-resolution image, as shown in Fig. 3. First, we input the original low-resolution image with $H \times W$ size and then use $1 \times 1$ convolution to expand the number of channels to $r^2$. Finally, the feature image of $H \times W \times r^2$ size is rearranged into a high-resolution image of $rH \times rW$ size via shuffling pixels on the $r^2$ channels. Similarly, the elements of $H \times W \times C \cdot r^2$ tensor can be rearranged into a $rH \times rW \times C$ tensor by subpixel convolution, which is defined by the mathematical formula as follows:

$$PS(I)_{x,y,c} = I_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \bullet r \bullet mod(y,r) + C \bullet mod(x,r) + c} \tag{5}$$

where $r$ is the expansion factor, $I$ is the input feature, $PS(I)_{x,y,c}$ is the output feature, and the pixel value corresponds to $(x, y, c)$.

**Subpixel upsampling and fusion:** Based on the observation of subpixel convolution, the $C_2$ and $C_3$ layers must increase channels first to be reduced to 256 channels by subpixel convolution, which will result in many extra calculations. While the channels of the $C_4$and $C_5$ layers are sufficient to directly perform the subpixel convolution operation, we only perform subpixel convolution on the $C_5$ layer to reduce the amount of calculation and then fuse it with the features of the $C_4$ layer to obtain the $P_4$ layer. As shown in Fig. 2b, the $1 \times 1$ convolution is used to convert the $C_5$ layer to the $F_5$ layer that has 1024 channels; then, the number of channels of the $C_4$ layer is reduced to 256 as in the traditional FPN. Finally, the $F_5$ feature layer performing subpixel convolutional upsampling and the $F_4$ feature layer are merged into the $P_4$ layer, which has 256 channels.

## 4 Experimental analysis

### 4.1 Data set and evaluation metrics

To verify the performance of the proposed algorithm, two data sets were used. First, The MOCS data set is a large-scale and public image data set that is designed for detecting objects in construction sites and was developed by Tsinghua University. The MOCS data set contains 13 categories with a total of 41,668 images collected from 174 different construction sites. We selected 19,404 images for training and 4,000 images for verification. Second, the MS COCO data set contains 80 categories, consisting of 115,000 images for training (train-2017) and 5000 images for verification (val-2017). The evaluation metrics used in this study are the mean average accuracy (mAP) of the different intersection ratios (IOU) and the different target sizes. According to the difference in the IoU threshold, mAP is divided into $AP$, $AP_{50}$, and $AP_{75}$, where $AP$ is the average precision whose IoU is between 0.5 and 0.95, $AP_{50}$, which means that the method's average precision of IoU threshold is 0.50, and similarly, $AP_{75}$ is 0.75. Based on the target size, the $AP$ is categorized into $AP_s$, $AP_m$, and $AP_l$ for the average precision of small targets, medium targets, and large targets, respectively. Where the small target is that pixel area is less than 32*32, the medium target is that pixel area is between 32*32 and 96*96, and the large target is larger than 96*96.

### 4.2 Implementation details

Because this paper primarily focuses on the multiscale target detection problem of construction sites, the primary data set

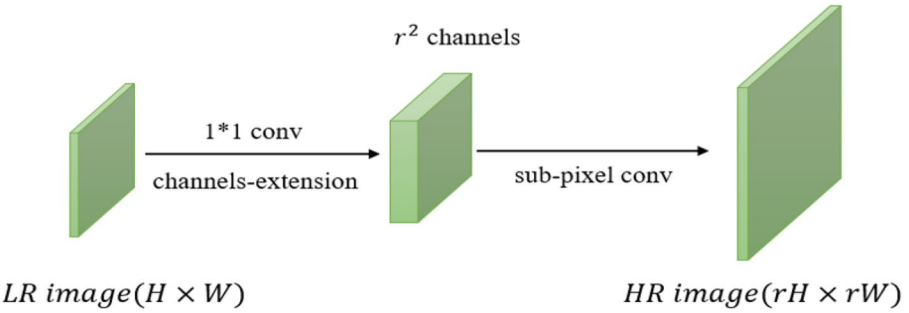



**Fig. 3** Subpixel convolution

is the MOCS data set, and the proposed method is also verified on the common MSCOCO data set. The experimental hardware environment used in this study includes an Intel Xeon Gold 5218R CPU, 32 GB of RAM, and an NVIDIA Quadro RTX 5000 GPU; and the software environment is Ubuntu 20.04, Python 3.6, CUDA 10.1, and the Maskrcnn-benchmark object detection toolbox based on Pytorch 1.2.0. In addition, to fairly compare the performance of different methods, all experiments are executed on one GPU. The batch size is set to 2 during the training stage and is set to 1 when tested. In addition, the learning rate of the prpoposed algorithm follows the Maskrcnn benchmark, where the initial learning rate of the scaling criterion is set to 0.0025. The maximum number of iterations is set to 720, 000. The weight attenuation parameter is set to 0.0001, which is performed at the 480000 and 640, 000 iterations.

### 4.3 Main results

To evaluate the effectiveness of the proposed algorithm, we perform experiments on the construction site data set MOCS and the MS COCO data set, and compare the proposed algorithm with other object detection methods, including one-stage object detection algorithm and two-stage multi-scale target detection algorithms, of which the one-stage target detection algorithms include YOLOv3, YOLOv5 [32] and YOLOX [33]. In addition, the experiments of YOLOv5 and YOLOX selected the most basic network structure of yolov5-s and yolox-s, respectively. The two-stage multi-scale target detection algorithms include Faster R-CNN with FPN, Faster R-CNN replace FPN with PAFPN, and Faster R-CNN with scale-aware data enhancement. Since the proposed method is mainly constructed based on the two-stage target detection model of Faster R-CNN, similar with other two-stage methods, we will also use two backbone networks for validation and evaluation, including ResNet-50 and ResNet-101. Experimental results are shown in Tables 1 and 2. Table 1 compares the average detection accuracy of different multiscale target detection algorithms on the construction site data set MOCS. Table 1 shows that while using the backbone network ResNet-50, the *AP* value of the original Faster R-CNN with FPN is 46.6% on the MOCS data set, where the average accuracy of small target detection is only 15.1%. After replacing FPN with PAFPN, the overall detection accuracy only increases by 0.1%, and the detection accuracy of large targets increases by 0.2%. The detection accuracy of small targets drops by 1.6%. Thus, the improvement of the FPN network is limited. In addition, the performance of Faster R-CNN with a scale-aware data enhancement strategy markedly improved. Finally, compared with the original Faster R-CNN with FPN, the overall value *AP* of the proposed method increases from 46.6 to 54.6%, improved by 8%. Concurrently, $AP_{50}$,$AP_{75}$,$AP_s$,$AP_m$ and $AP_l$ increases by 8.5%, 9.4%, 10.3%, 10.7%, and 6.6%, respectively. In addition, it can be seen from Table 1 that the detection accuracy of the proposed algorithm is also higher than that of the yolo series algorithms.

To intuitively show the improvement produced by the proposed algorithm on the MOCS data set, we also provide the visual detection results of five mentioned algorithms, including YOLOv5, YOLOX, Faster R-CNN+FPN, Faster R-CNN+SA and the proposed model EMS Faster R-CNN. Among them, the two-stage methods uniformly use the backbone network ResNet-50. It can be seen from Fig. 4 that our method has the best detection results, and the other four methods have problems, such as missed detection, false detection and so on. For example, when detecting the third image in Fig. 4, YOLOX and Faster R-CNN+FPN did not detect the small vehicle successfully. Moreover, only our proposed method and the YOLOv5 detect the two people, which contain few pixels in the upper left of the fourth image. However, the detection accuracy of YOLOv5 is not as high as ours. Especially, for the fifth image, only our proposed method can detect correctly, and the YOLOv5 model falsely detects a person in the lower right, while other methods have the problem of missing detection.

**Table 1** Comparison of detection average accuracy of different algorithms on MOCS data set

| Method | Backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|
| YOLOv3 [17] | DarkNet-53 | 40.6 | 67.7 | 42.2 | 11.4 | 28.4 | 52.8 |
| YOLOv5 [32] | Modified CSP v5 | 46.2 | 61.3 | 49.0 | 16.7 | 37.6 | 55.4 |
| YOLOX(PAFPA) [33] | CSPDarknet | 51.4 | 73.2 | 55.2 | 15.3 | 38.7 | 64.9 |
| Faster R-CNN+FPN [10] | ResNet-50 | 46.6 | 71.0 | 51.2 | 15.1 | 32.9 | 59.0 |
| Faster R-CNN+PAFPN [11] | ResNet-50 | 46.7 | 70.6 | 51.5 | 13.5 | 32.8 | 59.2 |
| Faster R-CNN+SA[16] | ResNet-50 | 54.1 | 79.0 | 59.9 | 24.7 | 43.0 | 65.0 |
| EMS Faster R-CNN (ours) | ResNet-50 | **54.6** | **79.5** | **60.6** | **25.4** | **43.6** | **65.6** |
| Faster R-CNN+FPN [10] | ResNet-101 | 48.3 | 71.5 | 52.9 | 14.9 | 35.3 | 61.2 |
| Faster R-CNN+PAFPN [11] | ResNet-101 | 48.6 | 71.5 | 53.0 | 14.8 | 34.1 | 61.5 |
| Faster R-CNN+SA [16] | ResNet-101 | 54.6 | **78.9** | 59.8 | **25.3** | 43.6 | 65.1 |
| EMS Faster R-CNN (ours) | ResNet-101 | **54.7** | 78.6 | **60.4** | 23.0 | **43.7** | **65.3** |

The best performance compared with other method based on the same backbone

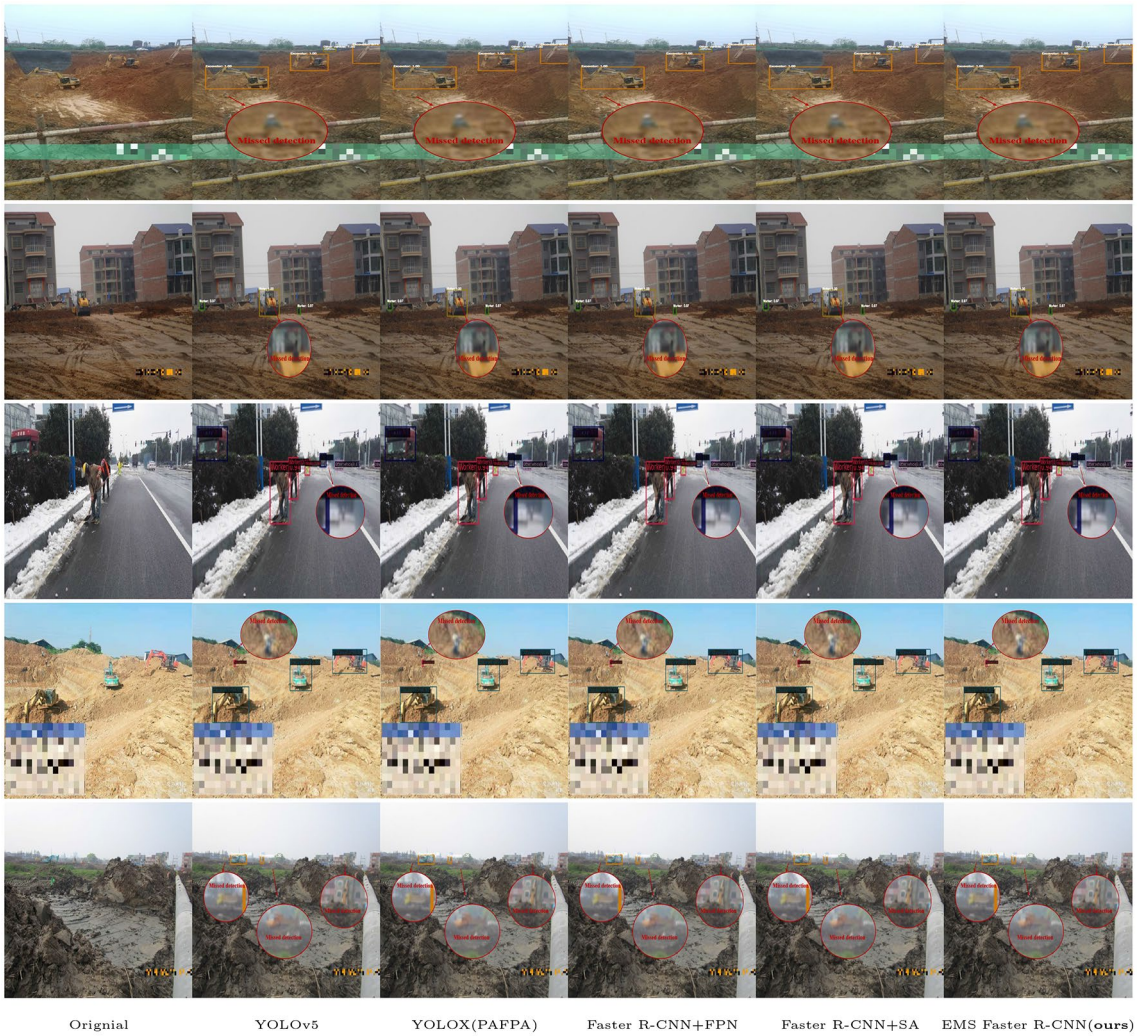



**Fig. 4** Comparison of the detection effects of different algorithms on the MOCS data set. From the left to the right columns are the original images, and the detection results of YOLOv5, YOLOX, Faster R-CNN+FPN, Faster R-CNN+SA, and EMS Faster R-CNN, respectively

**Table 2** Comparison of detection average accuracy of different algorithms on MS COCO data set

| Method | Backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|
| YOLOv3 [17] | DarkNet-53 | 33.7 | 56.6 | 35.3 | 19.4 | 36.8 | 44.3 |
| YOLOv5 [32] | Modified CSP v5 | 37.4 | 57.1 | 40.3 | 21.2 | 42.3 | 49.0 |
| YOLOX(PAFPA) [33] | CSPDarknet | 37.6 | 56.0 | 40.5 | 20.0 | 41.9 | 50.2 |
| Faster R-CNN+FPN [10] | ResNet-50 | 36.9 | 58.7 | 39.9 | 20.9 | 40.1 | 48.1 |
| Faster R-CNN+PAFPN [11] | ResNet-50 | 37.2 | 58.8 | 40.2 | 21.4 | 40.3 | 48.6 |
| Faster R-CNN+SA [16] | ResNet-50 | 38.1 | 59.9 | 41.5 | 23.7 | 40.9 | 49.2 |
| EMS Faster R-CNN (ours) | ResNet-50 | **38.4** | **60.1** | **41.8** | **24.3** | **41.4** | **49.6** |
| Faster R-CNN+FPN [10] | ResNet-101 | 39.3 | 61.2 | 42.6 | 22.8 | 42.8 | 51.4 |
| Faster R-CNN+PAFPN [11] | ResNet-101 | 39.3 | 61.1 | 43.0 | 22.7 | 42.9 | **51.8** |
| Faster R-CNN+SA [16] | ResNet-101 | 40.4 | 62.5 | 44.2 | 25.4 | **43.9** | **51.8** |
| EMS Faster R-CNN (ours) | ResNet-101 | **40.5** | **62.6** | **44.4** | **26.3** | 43.7 | 51.7 |

The best performance compared with other method based on the same backbone

**Table 3** Effect of each component on MOCS and MS COCO data sets based on ResNet-50

| CEBFPN | SA | Data set | mAP | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|---|
| | | MOCS | 46.6 | 71.0 | 51.2 | 15.1 | 32.9 | 59.0 |
| ✓ | | | 46.7 | 70.9 | 51.5 | 15.0 | 33.1 | 59.3 |
| | ✓ | | 54.1 | 79.0 | 59.9 | 24.7 | 43.0 | 65.0 |
| ✓ | ✓ | | **54.6** | **79.5** | **60.6** | **25.4** | **43.6** | **65.6** |
| | | MS COCO | 36.9 | 58.7 | 39.9 | 20.9 | 40.1 | 48.1 |
| ✓ | | | 37.2 | 58.7 | 40.0 | 21.0 | 40.3 | 49.2 |
| | ✓ | | 38.1 | 59.9 | 41.5 | 23.7 | 40.9 | 49.2 |
| ✓ | ✓ | | **38.4** | **60.1** | **41.8** | **24.3** | **41.4** | **49.6** |

*CEBFPN* FPN of Channel Enhancement Bidirectional Fusion by Sub-pixel Convolution, *SA* scale-aware data automatic augmentation module.

The best performance compared with other method based on the same backbone

We also perform comparative experiments on the common data set MS COCO to further verify the effectiveness of the proposed algorithm. Experimental results are shown in Table 2 and Fig. 5, which compare the average detection

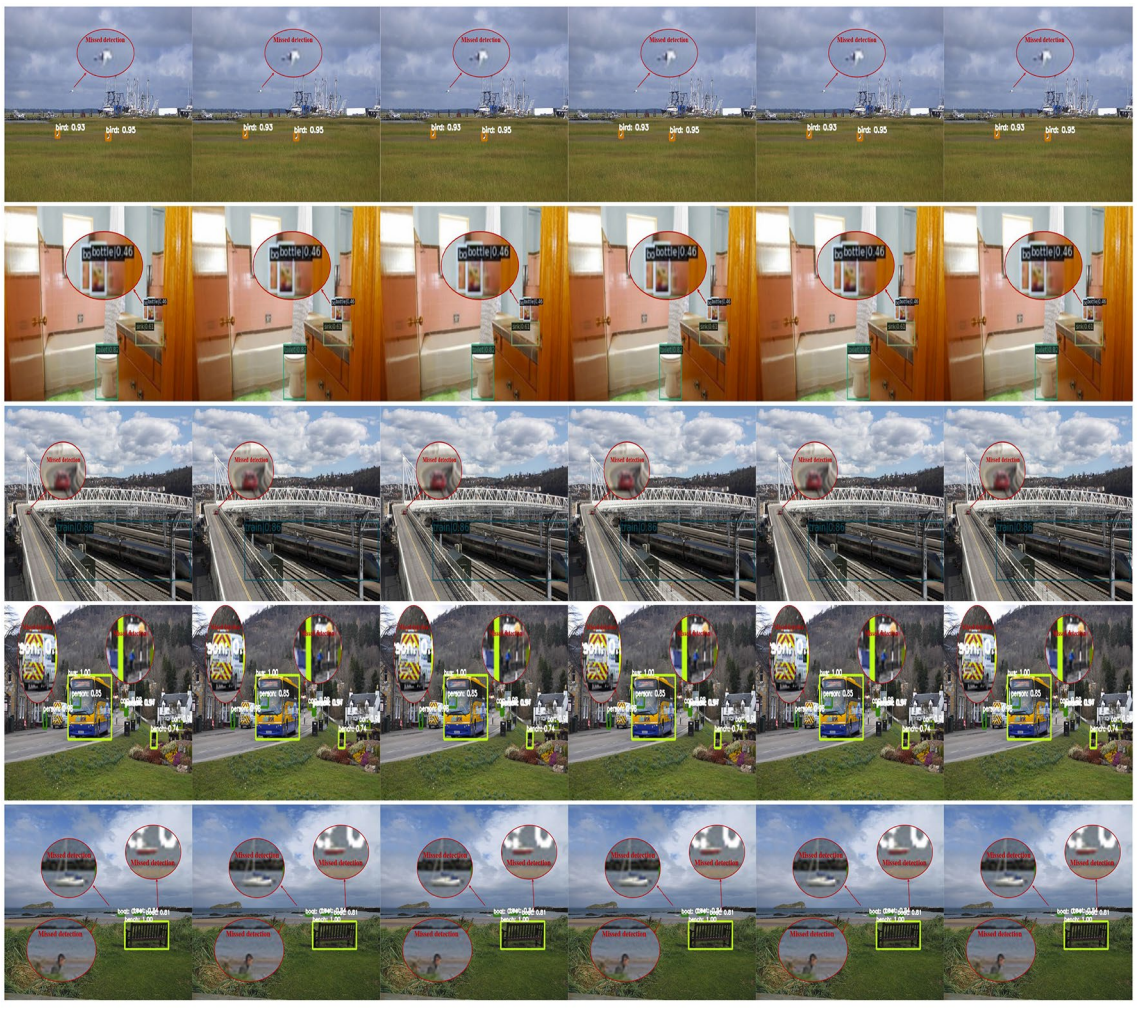



**Fig. 5** Comparison of the detection effects of different algorithms on the MS COCO data set. From the left to the right columns are the original images, and the detection results of YOLOv5, YOLOX, Faster R-CNN+FPN, Faster R-CNN+SA, and EMS Faster R-CNN, respectively

accuracy of seven methods and the visual detection results of the five algorithms, respectively. The proposed algorithm can also produce more consistent improvement on the MS COCO data set. Specifically, when using the ResNet-50 backbone network, the proposed algorithm achieves an accuracy of 38.4%, where the average accuracy of the small object achieves 24.3%(improvements of 1.5% and 3.4%, respectively) compared with that of Faster R-CNN with FPN. At the same time, the detection accuracy of the proposed algorithm is higher than that of the one-stage model. In addition, it can be seen from Fig. 5 that the detection effect of the proposed algorithm is better than that of the other four methods. For example, only our proposed method can successfully detect the kite in the first image, and the person and car with few pixels in the fourth image. In addition, we perform experiments on these two data sets using the ResNet-101 backbone, and the experimental results are shown in Tables 1 and 2. The proposed algorithm achieves the highest average precision compared with the other three two-stage methods on the MS COCO data set, which reaches an average precision of 40.5%. Although the overall accuracy has improved, the $AP_s$ of the proposed method is worse than Faster R-CNN with SA on the MOCS data set when using the ResNet-101 backbone, but the $AP_s$ on the COCO data set is better than Faster R-CNN with SA. One of the reasons is that the training of the model first refers to the MS COCO data set, and then trains on the MOCS data set, so the model works better on the MS COCO data set, especially for deeper networks. This phenomenon is a challenge to solve in our future work, we will continue to investigate the cause and resolve this issue.

### 4.4 Ablation experiments

We also analyzed the impact of each proposed component of EMS Faster R-CNN on the MOCS and MS COCO data sets. The overall ablation research report is shown in Table 3. We have gradually added the feature pyramid network of Channel Enhancement Bidirectional Fusion Based on the subpixel convolution module (CEBFPN) and the scale-aware data enhancement module(SA) on the Faster R-CNN network with FPN. Concurrently, the improvements brought about by the combination of CEBFPN and SA are also introduced to demonstrate the effectiveness of the proposed method, and the ablation experiments all use the backbone ResNet-50 and are all compared under the same settings for fair comparison. As shown in Table 3, after replacing FPN with CEBFPN, the average precision on the MOCS and MS COCO data sets has been improved by 0.1% and 0.3%, respectively. These results verify that the upsampling and fusion method of CEBFPN based on subpixel convolution can make better use of the channel information. Then the experiment is performed by introducing the scale-aware data enhancement module based on the Faster R-CNN with FPN, whose experimental results show that the average detection accuracy on the MOCS and MS COCO data sets are 54.1% and 38.1%, improved by 7.5% and 1.2%, respectively. These results also verify the effectiveness of the scale-aware data enhancement module. Finally, we perform experiments on Faster R-CNN that combines the CEBFPN module and the SA module. Its detection results on the MOCS and MS COCO data sets indicate the highest average accuracies, which reach 54.6% and 38.4%, respectively. $AP_s$ reaches 25.4% and 24.3% (improvements of 0.7% and 0.6% compared with the Faster R-CNN with the SA module). Therefore, it can be concluded that reducing the channel loss by replacing FPN with CEBFPN will produce further improvements. These experiments provide more evidence of the effectiveness of the proposed method.

## 5 Conclusions

For safety monitoring at construction sites and to address the problem of poor detection of small targets in existing object detection algorithms, we proposed a method that combines data enhancement and FPN network improvement called the EMS Faster R-CNN object detector. Extensive experiments were performed on the MOCS data set and the MS COCO data set. Compared with conventional multiscale Faster R-CNN object detectors, the proposed algorithm can markedly improve detection accuracy, which promotes the application of deep learning in safety monitoring of construction sites. The proposed algorithm based on FPN ResNet-50 can also produce effective and consistent accuracy improvements on both the MS COCO and MOCS data sets, demonstrating the proposed algorithm's strong generalizability.

**Acknowledgements** This study was supported in part by the National Natural Science Foundation of China (Nos. 62072024, 41971396, and 61971290), the Research Ability Enhancement Program for Young Teachers of Beijing University of Civil Engineering and Architecture (No. X21024), the Outstanding Youth Program of Beijing University of Civil Engineering and Architecture, the BUCEA Post Graduate Innovation Project, and R &D Program of Beijing Municipal Education Commission(Nos. KM202110016001, KM202210016002).

## References

1. Vasuhi, S., Vaidehi, V.: Target detection and tracking for video surveillance. WSEAS Trans. Signal Process. **10**, 179–188 (2014)
2. Zhang, D.D., Lei, L.I.: Face detection system based on pcanet-rf. Comput. Technol. Dev. **26**(2), 31–34 (2016)
3. Martinez-Martin, E., Del Pobil, A.P.: Object detection and recognition for assistive robots: experimentation and implementation. IEEE Robot. Automat. Magazine **24**(3), 123–138 (2017)
4. Kim, D., Liu, M., Lee, S., Kamat, V.R.: Remote proximity monitoring between mobile construction resources using camera-mounted uavs. Autom. Constr. **99**, 168–182 (2019)

5. Roberts, D., Golparvar-Fard, M.: End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level. Autom. Constr. **105**, 102811 (2019)
6. Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T.M., An, W.: Detecting non-hardhat-use by a deep learning method from far-field surveillance videos. Autom. Constr. **85**, 1–9 (2018)
7. Xuehui, A., Li, Z., Zuguang, L., Chengzhi, W., Pengfei, L., Zhiwei, L.: Dataset and benchmark for detecting moving objects in construction sites. Autom. Constr. **122**, 103482 (2021)
8. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1958–1970 (2008)
9. Zou, W.W., Yuen, P.C.: Very low resolution face recognition problem. IEEE Trans. Image Process. **21**(1), 327–340 (2012)
10. 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, pp. 2117–2125 (2017)
11. 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, pp. 8759–8768 (2018)
12. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 821–830 (2019)
13. 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, pp. 7036–7045 (2019)
14. Fang, H.-S., Sun, J., Wang, R., Gou, M., Li, Y.-L., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 682–691 (2019)
15. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1301–1310 (2017)
16. Chen, Y., Li, Y., Kong, T., Qi, L., Chu, R., Li, L., Jia, J.: Scale-aware automatic augmentation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9563–9572 (2021)
17. Farhadi, A., Redmon, J.: Yolov3: An incremental improvement. In: Computer Vision and Pattern Recognition, pp. 1804–2767. Springer Berlin/Heidelberg, Germany (2018)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016). Springer
19. Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
20. 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, pp. 580–587 (2014)
21. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. Adv. Neural Info. Process. Syst **28**, 91–99 (2015)
23. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
24. Guo, C., Fan, B., Zhang, Q., Xiang, S., Pan, C.: Augfpn: Improving multi-scale feature learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12595–12604 (2020)
25. Singh, B., Najibi, M., Davis, L.S.: Sniper: Efficient multi-scale training. arXiv preprint arXiv:1805.09300 (2018)
26. 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, pp. 3578–3587 (2018)
27. Fang, H.-S., Sun, J., Wang, R., Gou, M., Li, Y.-L., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 682–691 (2019)
28. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113–123 (2019)
29. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T.-Y., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. In: European Conference on Computer Vision, pp. 566–583 (2020). Springer
30. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
31. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proceedings of the Aaai Conference on Artificial Intelligence, vol. 33, pp. 4780–4789 (2019)
32. Glenn Jocher: Yolov5. https://github.com/ultralytics/yolov5, (2021)
33. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)