IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 March 2024, accepted 6 April 2024, date of publication 10 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386788

RESEARCH ARTICLE

MSF-CSPNet: A Specially Designed Backbone Network for Faster R-CNN

FEITAO LI, (Member, IEEE), TONG SUN, PING DONG, QIANG WANG, YANLING LI, AND CHANGXIA SUN

College of Information and Management Science, Henan Agricultural University, Zhengzhou 450046, China

Corresponding author: Feitao Li (lft045006@henau.edu.cn)

This work was supported in part by China Scholarship Council under Grant 201908420354, and in part by the Startup Project of Doctor Scientific Research of Henan Agricultural University under Grant 30501133.

ABSTRACT Although Faster R-CNN has undergone a lot of improvements, it still exists a significant gap in the performance between the detection of small and large objects, mainly because the low-level network lacks semantic information and small objects are only involved in a few images. To mitigate the above issues, we propose an object detection model based on Multi-Scale Feature fusion Cross Stage Partial Network (MSF-CSPNet) in this paper. The proposed MSF-CSPNet focuses on the fusion of concrete features and abstract features from multi-scale feature by learning shallow features at the shallow level and deep features at the deep level. Meanwhile, the data augmentation is performed by using random horizontal flip. On the basis, the improved Faster-RCNN model with Automatic Mixed Precision, Group Batch Sampler and MSF-CSPNet was formed. The proposed algorithm is valuated on the Microsoft Common Objects in Context (MS COCO) 2017 and obtained leading performance with 5.4% improvement in AP_{coco}, 5.9% improvement in AP_{50}, 6.9% improvement in AP_{75}, 5.8% improvement in AP_S, 6.1% improvement in AP_M, 5.8% improvement in AP_L compare to Faster R-CNN based on ResNet-50 with Feature Pyramid Network (FPN) backbone, and also outperformed previous reports on state-of-art Faster R-CNN series using other backbone networks, especially for small object detection. This research shows that the combination of a backbone with stronger learning ability and FPN is helpful to detect the expression of objects. Faster R-CNN based on MSF-CSPNet has high efficiency and better balance between accuracy and speed.

INDEX TERMS Convolutional neural network, cross stage partial network, faster R-CNN, object detection.

I. INTRODUCTION

Object detection [1], as a longstanding, fundamental and challenging problem in computer vision [2], has been an active field of research for several decades [3], [4]. The task of object detection is to identify object categories and predict the location of each object in an image by a bounding box, and there are many real world applications [5] based on this task, such as face detection and pedestrian detection [6]. Since deep learning [7] entered the object detection field, the milestone approaches primarily divided into two categories: one-stage detectors, like SSD [8], RetinaNet [9], You Only Look Once (YOLO) series, including YOLOv1 [10], YOLOv2 [11], YOLOv3 [12], YOLOv4 [13], YOLOv5 [14], YOLOv6 [15], YOLOv7 [16] and YOLOv8 [17], and two-stage detectors, such as Region-based Convolutional Neural Network (R-CNN) series, including R-CNN [18], Fast R-CNN [19], Faster R-CNN [20], R-FCN [21], FPN [22]. The main difference between the two categories is that two-stage detectors need to first generate proposals, and then perform fine-grain object detection, however, one-stage detectors directly extract features from network to predict object classification and position. Compared with the one-stage detectors, two-stage detectors usually have a relatively great performance, but are much slower and more unsuitable for real-time object detection applications.

Faster R-CNN [23], as a milestone of R-CNN serials detectors, was able to make predictions at a frame rate of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

5fps on a GPU by using a novel proposal generator (Region Proposal Network) and achieved state-of-the-art results on many public benchmark datasets, such as Pascal Visual Object Classes (Pascal VOC) 2007, 2012 and MS COCO. Subsequently, some more efforts have made to strengthen Faster R-CNN by bringing more computations into network. Region-base Fully Convolutional Networks (R-FCN) [21] generated a position sensitive score map which encoded relative position information of different classes, and used a Position Sensitive ROI Pooling layer (PSROI Pooling) to extract spatial-aware region features by encoding each relative position of the target regions so as to share the computation cost in the region classification step, which speed up inference when many proposals are used. while Multi Scale (MS) CNN [24] and FPN [22] construct feature pyramids by employing inherent multi-scale, pyramidal hierarchy to alleviate the scale mismatch between the RPN receptive fields and actual object size. The Mask R-CNN [25] was proposed by Kaiming He et al. to tackle pixel-wise object instance segmentation using a Region of Interest (RoI) align layer by extending Faster R-CNN. Then Zhang et al. [26] proposed Mask-Refined R-CNN to solve the problem that the difference in spatial information between receptive fields of different sizes was ignored in Mask R-CNN. A-Fast-R-CNN [27] proposed an adversarial network that generates examples with occlusions and deformations to improve image classification in semi-supervised setting. Cascade R-CNN [28] was proposed to address the problem of over-fitting due to exponentially vanishing positive samples during training, as well as the problem of inference-time mismatch between the IoUs for which the detectors is optimal and those of the input hypotheses. Light-Head R-CNN [29] was proposed to address the shortcoming of an intensive computation after or before RoI warping. The Genetic Algorithm Gabor Faster R-CNN (Faster GG R-CNN) [30] by embedding Gabor kernels into Faster R-CNN was proposed to address the texture interference problem of fabric detect detection.

Although Faster R-CNN has undergone a lot of improvements, but also has weakness of (relatively) slow speed and very low accuracy compare to recently emerged YOLO version. Among them, the network structures of YOLOv1 to YOLOv8 not only have relatively large changes, but also explore different backbone networks, for example, GoogleNet in YOLOv1, DarkNet19 in YOLOv2, DarkNet53 in YOLOv3, CSPDarkNet53 in YOLOv4, YOLOv5-S, YOLOv5-M, YOLOv5-L, YOLOv5-X in YOLOv5 and so on. Obviously, the YOLO series continuously breaks records in accuracy and speed cannot be separated from the network structures changing and a well-designed backbone. However, some researcher made attempt to explore different backbone networks in Faster R-CNN, such as VGG [31], ResNet [32], MobileNet [33], HyperNet [34], Performance Vs Accuracy Net (PVANet) [35], Inception Net [36] and DetNet [37], but have not achieved better results than YOLOv4.

Cross Stage Partial Network was proposed by Wang et al. [38], which reduced computations by 20% with equivalent or even superior accuracy on the ImageNet dataset and significantly outperformed the state-of-the-art methods in terms of AP_{50} on the MS COCO object detection dataset, but still underperformed with small objects. This paper first proposed to apply CSPNet into Faster R-CNN to get better performance. FPN mechanism was introduced into CSPNet, named MSF-CSPNet. Different from CSPNet in YOLOv4, we made small modification that added a branch of FPN into stage 2 of CSPNet, so the proposed MSF-CSPNet was able to focus on the fusion of concrete features and abstract features by learning shallow features at the shallow level and deep features at the deep level. Compared to the Faster R-CNN model based on ResNet50 backbone with FPN, the final Faster R-CNN model improves the AP_{coco} from 37.1% to 42.5%, the AP_{50} from 58.0% to 63.9%, the AP_{75} from 39.9% to 46.5%, the AP_S from 21.5% to 27.3%, the AP_M from 40.7% to 46.8%, the AP_L from 47.5% to 53.3%.

II. METHODOLOGY

A. MOTIVATION

No matter what it is one stage detectors or two stage detectors, they usually depend on a backbone network that is pretrained on the ImageNet classification dataset. Different from the task of ImageNet classification, the task of object detection is discovering ''where'' and ''what'' each object instance is when give an image by using bounding-box. Therefore, the design that the spatial resolution of the feature maps is gradually decreased for the standard image classification networks will harmful for localization task. In order to alleviate the above issue, many techniques like FPN and dilation technology are proposed and applied to these networks to maintain the spatial resolution. However, there also exists the following three challenges when trained with these backbone networks.

1) STRENGTHENING LEARNING ABILITY OF TRADITIONAL BACKBONE

The accuracy of existing CNN is greatly degraded after lightweightening, so how to strengthen CNN's learning ability to maintain sufficient accuracy while being lightweightening has become extremely critical.

2) STRENGTHENING FUSION ABILITY OF ConvNet's PYRAMIDAL FEATURE HIERARCHY

As we know, different layer of ConvNet's pyramidal feature hierarchy has different semantics. We need to design an architecture that can combine low-resolution, semantically strong features with high-resolution, semantically weak features.

3) INVISIBILITY OF SMALL OBJECTS

Despite object detection has made impressive progress, there is also a major gap in the performance between the detection

of small and large objects. Large stride of backbone network will result into decreasing the spatial resolution of the feature maps and integrating the large context, which are easily weaken the information from the small objects. FPN adopting bottom-up pathway can combine low semantic information from shallow layers that predict small object with context cues of high-representations from deeper layers that are sufficient to recognize the category of the object instances. However, these context cues will lose simultaneously when the small objects cannot be found in deeper layers.

MSF-CSPNet is proposed to address these problems, which has the several advantages compared with traditional backbone networks like ResNet for object detection. First, CSPNet with equivalent or even superior accuracy on the ImageNet dataset can reduce computations by 20%. Second, benefited by bottom-up pathway of FPN, MSF-CSPNet is more potent in locating and recognizing the small objects.

B. CROSS STAGE PARTIAL MODEL BASED ON FPN

Based on the backbone network Darknet-53 in YOLOv3, the CSPDarknet-53 was proposed by referring to CSPNet, which contains five Cross Stage Partial blocks. Next, We will introduce the detail architecture of Multi-Scale feature Fusion Cross Stage Partial Network (MSF-CSPNet).

1) STRUCTURE OF CONVOLUTION BATCH-NORMALIZATION MISH BLOCK

The convolution batch-normalization mish (CBM) block is a basic component unit in CSPDarknet-53, which contains convolution layer, batch-normalization layer and mish function. The structure of CBM is shown in figure 1. Batch normalization is the normalization of each batch data of convolution layer, which includes 4 steps. Firstly, the mean of each batch of training data can be calculated by using (1). x_i, m means the input of each batch of images $\{x_1, x_2, \ldots, x_n\}$ and the number of the data in (1), respectively.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i. \tag{1}$$

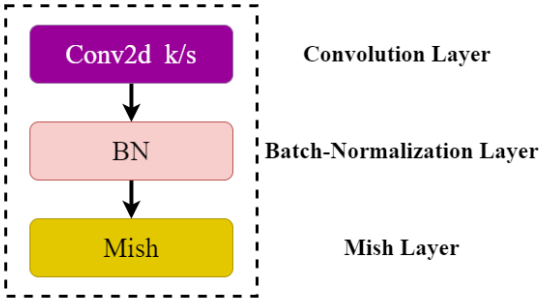

FIGURE 1. The architecture diagram of Convolution Batch-Normalization Mish (CBM) block. Standard CBM block usually contains convolution layer with batch-normalization layer and mish activate layer. k,s means kernel size and stride of Conv2d layer, respectively.

Then, the variance of each batch of training data can be found by employing (2).

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2. \tag{2}$$

Next, x_i is subtracted from the mean μ_B and then divided by the variance σ_B according to the obtained results, which is shown in (3). Therefore, the obtained data will exhibit a normal distribution, where ξ is a very small value in order to avoid zero variance.

$$\hat{x_i} = \frac{x_i - \mu_i}{\sqrt{\sigma_B^2 + \xi}}. \tag{3}$$

Finally, the y_i is obtained by multiplying $\hat{x_i}$ by γ and added β, where γ is the scaling factor, and β is offset. The values of both γ and β are iteratively updated when the network is trained.

$$y_i = \gamma\hat{x_i} + \beta. \tag{4}$$

Activation functions enable neural networks to effectively address complex problems by introducing non-linearity to the linear transformed input in a layer of a neural network. Compared to the performance of Swish, ReLu, and Leaky ReLU across different tasks in different in computer vision, Mish function as a novel self regularized non-monotonic activation function was proposed by Misra [39] and demonstrated better performance and stability, which is mathematically defined as:

$$f(x) = x * \tanh\left(\ln\left(1 + e^x\right)\right). \tag{5}$$

2) STRUCTURE OF CROSS STAGE PARTIAL BLOCK

A stage of Cross Stage Partial Networks (CSPNet) usually contains several CBM blocks and a Res unit, whose architecture is shown in figure 2. The feature maps of the first CBM block in a stage are divided into two parts though channel $x_0 = [x_0', x_0'']$. Between x_0' and x_0'', the former passes though a CBM block and link to the end of the stage, and the latter will go though CBM block, Res unit and CBM block, respectively. Finally, the two part are concatenated by a CBM block and undergo another CBM block, and then generate output. The $\times$ n means in each stage of CSPNet the number of repeated Res units.

3) STRUCTURE OF CROSS STAGE PARTIAL MODEL WITH FPN

By improving the basic structure of CSPNet, Multi-Scale Feature fusion Cross Stage Partial Network (MSF-CSPNet) is constructed. The top-down pathway and lateral connections were added into the feature activation output by the last layer of all CSP blocks except for CSP block1, and create the final set of feature maps *P2, P3, P4, P5* that are respectively of the same spatial sizes. On the basis of *P5* feature map, the P6 are generated by a down-sampling of stride = 2. The MSF-CSPNet structure is shown in Figure 1. The FPN starts from the second CSPNet stage to the last CSPNet stage

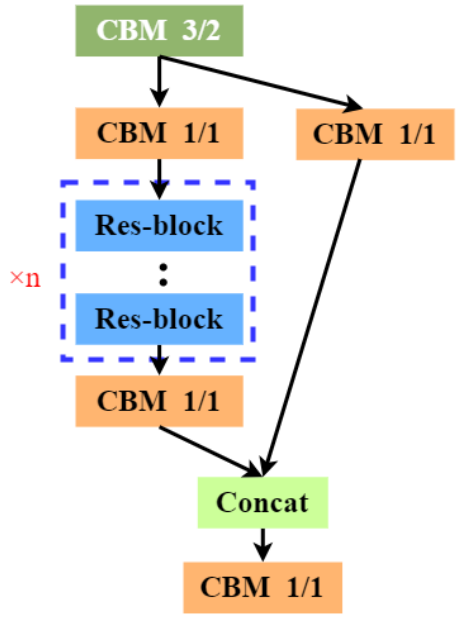

FIGURE 2. The architecture diagram of cross stage partial block. CBM means convolution layer with batch-normalization and mish, the two numbers in the CBM block represent the kernel and stride of Conv2d, respectively. ×*n* res unit means residual block are internally stacked n times in one CSP block, Concat means two branches of a CPS block are concatenated by using a CBM block with convolution kernel of 1.

can achieve multi-scale feature fusion from low-resolution, semantically strong features to high-resolution, semantically weak features. By creating P2 and P6 output, MSF-CSPNet is able to extract shallow layer characteristic information and deep layer characteristic information better than the BottleneckCSP structure. MSF-CSPNet has better feature extraction ability, which can be applied to a lot of object detection tasks as a basic module with minor modifies. The CSPDarknet-53 is adopted as the representation of the CSPNet in this paper. The assignment strategy of RoI of different scales in FPN follows the (6).

$$k = [k_0 + \log_2 \sqrt{wh}/224] \tag{6}$$

Here w, h is the width and height of RoI in P_k level feature pyramid, respectively. 224 is the canonical Image pre-training size, and k_0 means the target level on which an RoI with w×h = 224 should be mapped into. Different from the FPN-based Faster R-CNN system that set k_0 to 4, we set k_0 to 5. The mainly purpose of this design is that CSPNet has 5 stage CSP blocks.

There are two limitations to make an effective and efficient backbone for Faster RCNN detector. On one hand, heavy inference computations for deep neutral network greatly relies on costly computation resources. On the other hand, reducing the down-sampling factor of FPN need to remain the same as reducing the valid receptive filed of every stage of CSPDarkNet-53, which will be harmful for image classification in object detection.

MSF-CSPNet is carefully designed to overcome the two limitations. Specifically, the MSF-CSPNet follows the same setting as CSPDarkNet-53 in YOLOv4 from the first Cross Stage Partial block to the fifth Cross Stage Partial block, the differences lies in the second CSP block and the extra stage, e.g. *P6*, and an overview of MSF-CSPNet backbone for Faster RCNN can be shown in Figure 3.

C. FASTER-RCNN OBJECT DETECTION MODEL BASED ON MSF-CSPNET FEATURE EXTRACTOR

How to effectively detect objects of different sizes in images remains a major challenge in object detection. A few techniques, such as FPN and dilation, are considering as a standard solution for detecting objects of different scales. By adding MSF-CSPNet feature pyramid before Region Proposal Network (RPN) to extract image features at different levels, the Faster RCNN detector can better detect objects of different scales. Figure 4 displays the structure of Faster-RCNN network based on MSF-CSPNet.

The object detection process of Faster-RCNN is as following: First, the shorter edge of the image is resized to 800 pixels and the longer edge of the image is limited to 1333 pixels in order to avoid too much memory cost. The images within mini-batch were padded to the same size by filling zeros into the right-bottom of the image. Then each batch of images goes though the MSF-CSPNet performing feature extraction. The feature layers of *P2, P3, P4, P5, P6* will be created via MSF-CSPNet. We adopt Region Proposal Network (RPN) by replacing the single-scale feature map with FPN. A head of the same design (3×3 convolutional and two sibling 1×1 convolutional layers for box regression and box classification, respectively.) was attached to each level on the feature pyramid of FPN. The head slides densely over all the locations in all pyramid levels, therefore, it is not necessary to have multi-scale anchors on a specific level. Anchors with areas of 16^2, 32^2, 64^2, 128^2, 256^2 pixels are allocated to *P2, P3, P4, P5, P6* respectively. Anchors of multiple aspect ratios 1:2, 1:1, 2:1 at each level is used in our experiments, which is the same as in [22]. In total there are 15 anchors over the pyramid and 256 anchors per image. The proposals generated by RPN and the output features of FPN are fed into the Region of Interest (RoI) pooling layer followed by Two Multi-Layer Perceptron (MLP) Head whose function is to convert the number of input channels into the number of output channels (In general, the number of output channels is 1024). Fast RCNN predictor included two sibling 1×1 convolutional layers used to predict a class and class-specific box refinement for each proposal. The results from Fast RCNN predict are post-processed by Non-Maximum Suppression (NMS) in order to filter low scores object, then mapped predicted bounding box back to original image.

III. EXPERIMENTS, RESULTS, AND ANALYSIS

A. HARDWARE AND SOFTWARE

The experiments was performed on an integrated development environment called Pycharm installed in a local

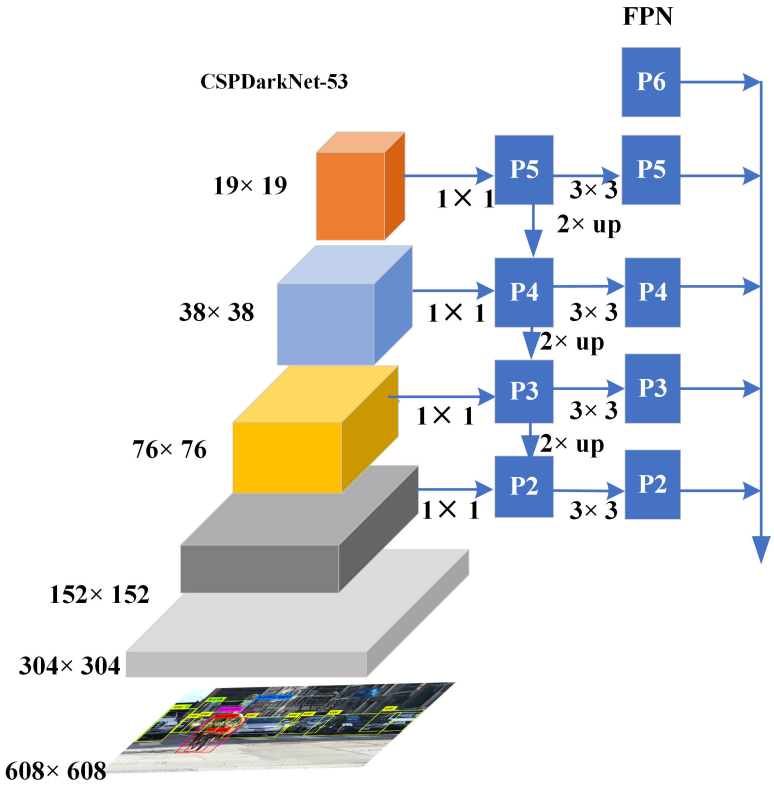

FIGURE 3. The architecture diagram of our proposed backbone network. The pyramidal feature hierarchy are built by last layer output of each stage of the CSPDarknet53 as shown on the left side of Figure 3. FPN is indicated by the dark blue on the right side of figure 3.

Dell Precision 7920 workstation having a NVIDIA GeForce RTX 3090 GPU with RAM capacity of 24 GB. The Pycharm is preconfigured with leading deep learning libraries on Ubuntu 20.04 64-bit, such as PyTorch, Numpy, as well as Matplotlib. The frameworks of all deep learning models in this paper are developed and run in Python, by employing Pytorch library, making use of automated differentiation on graphs of varying computations.

B. DATASET

The experiments are performed on the MS COCO Detection dataset. The MS COCO 2017 Detection dataset contains 164k images, including 118,287 images for training, 5000 images for validation, and 40670 test images, and 860k objects annotated with ground-truth bounding boxes from 80 categories. There are 41.43% small objects, 34.4% medium objects and 24.2% large objects among all the objects of the training dataset. Although the number of the small objects are large, their distribution in the training images is extremely nonuniform. In other words, only about half of the training images contain any small objects, while 70.07% and 82.28% of the training images include medium and large objects, respectively. We use the train dataset for training and report the performance on the validation dataset.

In general, mean Average Precision (mAP) is a crucial metric for evaluating object detection models, measuring their performance and accuracy. For VOC2007, VOC2012 and ImageNet, Intersection over Union (IoU) threshold of mAP is set to 0.5. Instead of using a fixed IoU threshold, MS COCO has six evaluation scores which demonstrates different capability of detection algorithms, including performance on different IoU thresholds and on different scale objects. For example, AP_{coco} (averaged precision over ten intersection-over-union thresholds from 0.5 to 0.95 with interval of 0.05), AP_{50}, AP_{75} (AP at different IoU thresholds), and AP_S (AP for area of objects smaller than 32^2), AP_M (AP for area of objects between 32^2 and 96^2), AP_L (AP for area of objects bigger than 96^2).

C. NETWORK PARAMETER SETTING

The detection effect of Faster-RCNN is greatly influenced by the detection parameters, the default hyper-parameters are set as follows: the training epochs is 43; The experiments use a stochastic gradient descent optimizer. The momentum value is 0.9, and the weight-decay coefficient is set as 1e-4. The step decay learning rate scheduling strategy is adopted with initial learning rate 0.005 and multiply with a factor 0.1 at the 12 epochs and 22 epochs, respectively; All architectures use a single GPU to execute multi-scale training with the batch size of 8 while the batch size is set as 1 in the validation data-loader due to the GPU memory limitation. Automatically Mixed Precision (AMP) is used in our training experiment in order to alleviate the shortage of GPU memory. For all the experiments in this paper, we only use a Nvidia

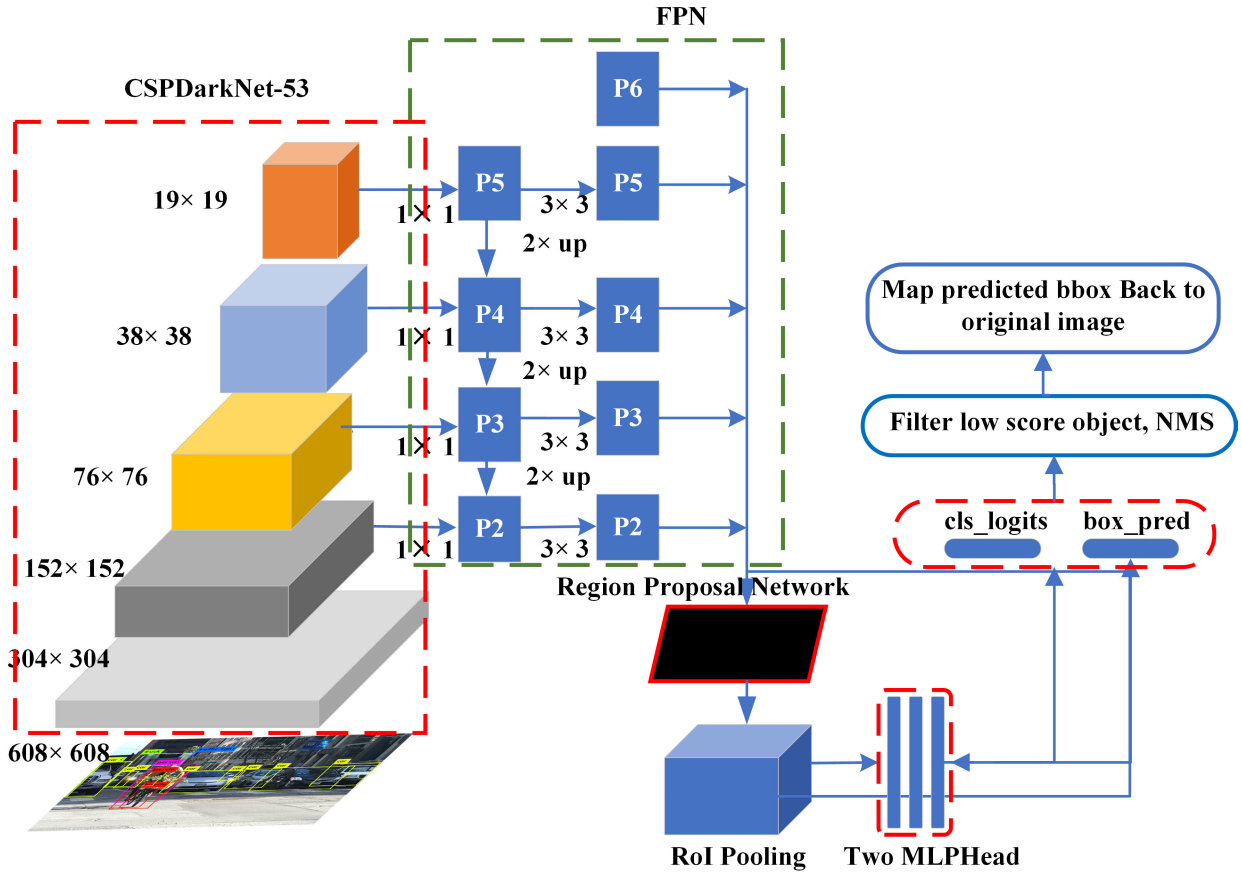

FIGURE 4. The architecture diagram of faster RCNN based on our proposed backbone network.

RTX 3090 GPU for training, so techniques such as sync-BN that optimizes multiply GPUs are not used.

D. EXPERIMENTAL RESULTS AND ANALYSIS

1) GRADIENT-WEIGHTED CLASS ACTIVATION MAPPING (GRAD-CAM) EXPERIMENT

Grad-CAM can produce 'visual explanations' for a large class of Convolution Neural Network (CNN) models in a coarse heat-map of the same size as the convolution feature maps. We used the Grad-CAM to explain the reason why we adopted the CSPDarkNet53 as our backbone network. The results of different layers of both network are shown in Figure 5. We can see from figure 5 that the low shallow network in ResNet50 pay little attention to the cat, while the low shallow network in CSPDarkNet53 has started to pay attention to edge detection information of the cat. Meanwhile, the deep network in ResNet50 makes attention to the neck of the cat, however, the deep network in CSPDarkNet53 make attention to the face and legs of the cat. The shallow network has small receptive field and high resolution, and the deep network has large receptive field and low resolution. If the convolution neural network is able to achieve better semantic information and edge detection information in the shallow network, and obtain rich semantic information in deep network, the CNN is more suitable as the backbone in object detection network framework. Through the comparison of Grad-CAM: ResNet50 and CSPDarkNet53, we can draw the conclusion that CSPDarkNet53 is more suitable as the backbone than ResNet50.

2) ABLATION EXPERIMENTS

The research results of ablation experiments are presented in Table 1. First, we run Faster R-CNN with single-scale map of CSPDarkNet-53 (stage5) on a single GPU, and obtain 37.1% AP_{coco}, 58.0% AP_{50}, 39.9% AP_{75}, 21.5% AP_S, 40.7% AP_M, 47.5% AP_L. The overall performance is significantly improved (from 37.1% to 39.8% in AP_{coco}, from 58.0% to 62.0% in AP_{50}, from 39.9% to 43.5% in AP_{75}, from 21.5% to 24.1% AP_S, from 40.7% to 43.4% AP_M, from 47.5% to 51.7 in % AP_L) by the three-scale feature fusion (stage3, stage4 and stage5). When we fuse four-scale feature layers (stage2, stage3, stage4 and stage5), compared to three-scale feature fusion, the AP_{coco}, AP_{50}, AP_{75}, AP_S, AP_M, AP_L increases by 2.7%, 1.9%, 3.0%, 3.2%, 3.4%, 1.6%, respectively. To ensure fair comparison, they are using the same hyper-parameters in addition to varying the number of fused feature layers. Meanwhile, We are still conducting the same experiments on 4 GPUs in order to study the effect of Cross-GPU Batch Normalization (Sync-BN) on the performance of Faster R-CNN. We observe from Table 1 that the influence of Sync-BN on the performance of Faster R-CNN gradually decreases with the increase of fusing

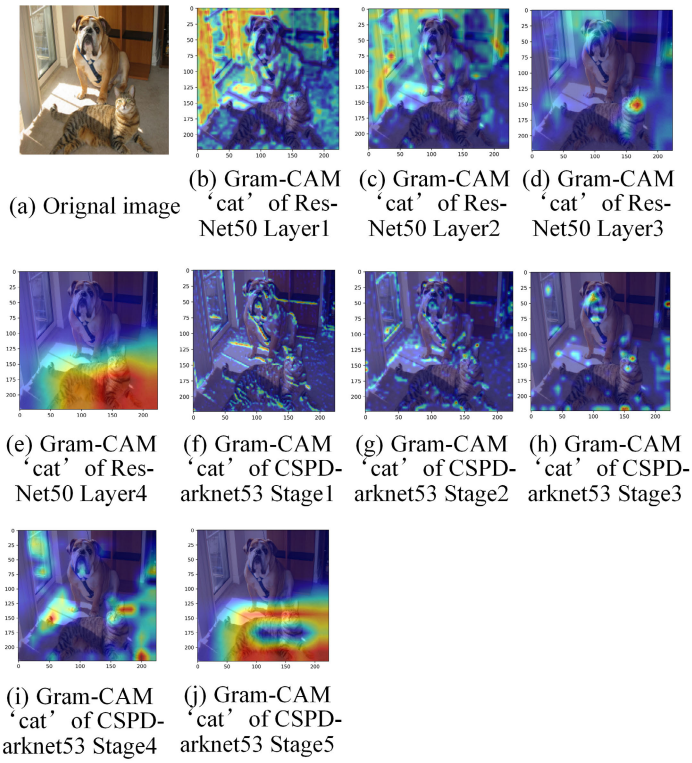

FIGURE 5. (a) Original image with a cat and dog. (b-e) Support for the cat category according to various layers of ResNet50. (f-j) Support for the cat category according to various stages of CSPDarkNet53. Note that in (b-j), red regions corresponds to high score for class.

feature layers. For example, the AP_{coco} of Faster R-CNN with three-scale feature fusion of CSPDarkNet-53 backbone has been improved by 2.9%, and the AP_{coco} of Faster R-CNN with single-scale map and four-scale feature fusion has been improved by 2.4% and 0.2%, respectively. The AP_{50} of Faster R-CNN with single-scale map has been improved by 3.1%, and the AP_{50} of Faster R-CNN with three-scale feature fusion and four-scale feature fusion has been improved by 1.5% and 0.1%, respectively. The experiment results demonstrate the performance of Faster-RCNN has been gradually improved with the number of fused feature layers increasing, which explains the reasons that we select four stages as the input of the FPN.

TABLE 1. Results of different maps based on FPN. Meanwhile, we also train the backbone with different scale maps on four GPUs combining with sync-BN technology.

Backbone	scale	Gpu	AP_{coco}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CSPDarkNet53	one	single	37.1	58.0	39.9	21.5	40.7	47.5
CSPDarkNet53	three	single	39.8	62.0	43.5	24.1	43.4	51.7
CSPDarkNet53	four	single	42.5	63.9	46.5	27.3	46.8	53.3
CSPDarkNet53	one	four	39.5	61.1	42.6	20.8	44.9	54.3
CSPDarkNet53	three	four	41.7	63.5	45.4	26.1	45.8	51.6
CSPDarkNet53	four	four	**42.7**	**64.0**	**46.7**	**27.3**	**46.8**	**53.6**

3) MAIN RESULTS

We first download the pre-training weights of both CSPDarkNet-53 and ResNet-50 on ImageNet classification from git-hub website. CSPDarkNet-53 has 76.5% the top-1 accuracy at the cost of 4.71G FLOPs, however, ResNet-50 has 75.9% the top-1 accuracy at the cost of 4.1G FLOPs. Then we start to train FPN with CSPDarkNet-53, and compare it with ResNet-50 based FPN. All the results obtained by both methods are shown in Figure 6. From Figure 6 we can see CSPDarkNet-53 has superior performance than ResNet-50.

We use FPN with ResNet-50 as our baseline because this method has achieved state-of-the-art results on the COCO detection benchmark, surpassing all the R-CNN series at that time. In order to validate the effectiveness of the CSPDarkNet-53 with FPN, an additional output which involves the second stage of CSPDarkNet-53 was added into FPN compared with CSPDarkNet-53 in YOLOv4. More design details are introduced in Section III. Then ResNet-50

TABLE 2. Results of different backbones based on FPN. The standard Top-1 error on the ImageNet classification is inversely proportional to accuracy on the ImageNet classification (the lower error is, the better accuracy is in classification). The computation complexity of algorithm is denoted as FLOPs. The COCO evaluation metrics is utilized to study effectiveness of these backbone for faster RCNN.

Backbone	Classification		FPN results					
	Top1 err	FLOPs	AP_{coco}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50	24.1	4.1G	37.9	60.0	41.	22.9	40.6	49.2
ResNet-101	23.0	7.6G	39.8	62.0	43.5	24.1	43.4	51.7
DetNet-59	23.5	4.8G	40.2	61.7	43.7	23.9	43.2	52.0
CourNet		29.8G	33.5	52.9	34.9	23.6	44.1	48.5
CSPDarkNet-53	23.5	4.7G	**42.7**	**64.0**	**46.7**	**27.3**	**46.8**	**53.6**

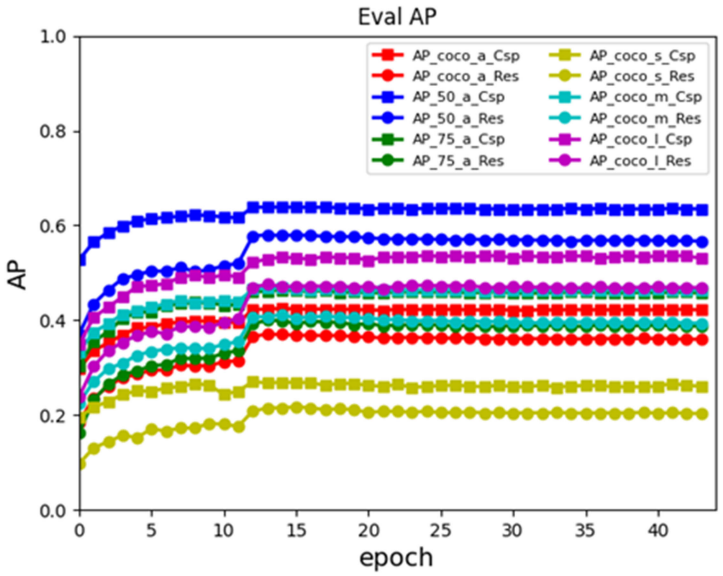

FIGURE 6. Comparisons of the accuracy faster RCNN with different backbone (CSPDarknet53 vs ResNet50) on the MS COCO dataset. AP_{coco}, AP_{50}, AP_{75} has the same meaning as AP_{coco}, AP_{50}, AP_{75}, respectively; s, m, l, a means small, mediate, large and all objects, respectively.

TABLE 3. Performance comparison of object detectors results between our method and state-of-the-art on MS COCO2017 dataset. Based on our simple and effective backbone CSPDarkNet53 with FPN, our model performance has achieved equivalent or even superior average precision compare with all previous state-of-the-art.

Method	Backbone	FLOPs	AP_{coco}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN	ResNet-101	7.6G	34.9	55.7	37.4	15.6	38.7	50.9
Faster-RCNN	ResNet-101 with FPN	7.6G	36.2	59.1	39.0	18.2	39.0	48.2
Faster-RCNN by G-RMI	Inception-ResNet-v2		34.7	55.5	36.7	13.5	38.1	52.0
Cascade-RCNN	ResNet-101	7.6G	42.8	62.1	46.3	23.7	45.5	55.2
Faster-RCNN	DetNet-59	4.8G	40.2	61.7	43.7	23.9	43.2	52.0
Faster-RCNN	MA-ResNet	1.54G	28.7	48.9				
FCOS	ResNet-50-FPN	4.8G	40.6	58.5	44.4	22.9	43.5	51.5
YOLOv4	CSPDarknet-53	4.7G	**43.5**	**65.7**	**47.3**	26.7	46.7	53.3
YOLOv5s	CSP+Focus	16.3G	30.4	51.3	31.1	19.0	41.6	56.5
YOLOv5m	CSP+Focus	23.4G	30.7	51.4	31.9	21.3	42.7	46.1
YOLOv5l	CSP+Focus	46.7G	32.1	52.6	33.4	21.8	43.9	48.3
YOLOv5x	CSP+Focus	85.3G	31.6	52.1	32.7	22.3	43.3	48.6
MSFYOLO	CourNet	41.2G	33.5	52.9	34.9	23.6	44.1	48.5
YOLOv7-tiny	CSP+Focus+E-ELAN	5.8G	35.2	52.8	37.3	15.7	38.0	53.4
Faster-RCNN (ours)	**CSPDarknet-53 with FPN**	**4.7G**	42.7	64	46.7	**27.3**	**46.8**	**53.6**

backbone is substituted by CSPDarkNet-53 while keep the same architecture as original FPN.

Since CPSDarkNet-53 has more parameters than ResNet-50, a natural hypothesis is that the improvement in performance owing to more parameters. In order to validate our hypothesis, we have access to a large number of documents and found that FPN with ResNet-101 which has 7.6G FLOPs complexity still has inferior performance than our method. Even if DetNet-59 has the same top-1 accuracy and FLOPs as the CSPDarkNet-53, its overall performance is still inferior than our method. The comparison of all results obtained by the two methods are displayed in Table 2. These results further demonstrate that CSPDarkNet-53 is more suitable for Faster R-CNN than both ResNet and DetNet.

E. COMPARISON TO STATE-OF-THE-ART

There is a tradition to show the State-of-the-Art comparing in order to validate the effectiveness of MSF-CSPNet, the prime results of comparison are displayed in Table 3. The performance of Faster R-CNN with MSF-CSPDarkNet53 backbone is compared with other state-of-the-art object detectors on MS COCO objection dataset. Faster R-CNN with the proposed backbone obtains 42.7 % AP_{coco}, 63.9% AP_{50} and 46.5% AP_{75}, outperforming Faster R-CNN with different backbones, including ResNet-101 [40], ResNet-101 with FPN [22], Inception-ResNet-v2 [41], DetNet-59 [37] and MA-ResNet [42], and its variants [28], also outperforming one-stage object detectors, such as FCOS [43], YOLOv5 [14], YOLOv7-tiny [16] and MSFYOLO [44]. The AP_S of the proposed method reaches 27.3%, which is higher

than the best results (26.7%) of YOLOv4 [13] (over 0.6 points gains in AP_S). Faster R-CNN with the proposed backbone achieves the state-of-the-art results in AP_M (46.8%) and AP_L (53.6%), which is also better than AP_M (46.7%) and AP_L (53.3%) of YOLOv4. However, our proposed method performs slightly more inferior than YOLOv4 in AP_{coco}, AP_{50} and AP_{75}. It is worth noting that CSPDarkNet-53 has only 4.7G FLOPs complexity while is lower than other backbones except for MA-ResNet. The reasons for the above phenomenon can be summarized as follows. On one hand, YOLOv4's success is attributed to the use of many training tricks, such as Cross mini-Batch Normalization (CmBN), self-adversarial-training (SAT), Mish activation, Mosaic data augmentation, DropBlock regularization, CIoU loss, we only use horizontal flipping for data augmentation in this paper. On the other hand, the main reason why our method can improve in AP_S is the fusion of shallow layer of CSPDarknet53 in order to find missing small objects, which yields 0.6 points gain (27.3 vs 26.7) in AP_S for small object.

IV. CONCLUSION

This paper improves CSPNet to raise the ability of object feature extraction and the accuracy of different scale object detection on MS COCO dataset. A novel backbone network, called MSF-CSPNet, is constructed by introducing FPN mechanism into CSPNet. Then, the new backbone is used in Faster R-CNN to improve the object detection ability of Faster R-CNN. By performing experiments and analysis of Grad-CAM, model complexity and identification ability, the following conclusions are finally obtained:

(1) CSPNet has better network performance and classification accuracy than ResNet, which is suitable for feature extractor for multiple scale object detection tasks.

(2) The MSF-CSPNet through the combination of deep network and shallow network and muti-scale prediction, not only achieves better accuracy than competitive with the Faster R-CNN counterpart on the COCO benchmark, but also much faster during both training and inference.

(3) Faster R-CNN with MSF-CSPNet feature extractor can obtain better accuracy than YOLOV4 and YOLOV7-tiny in small object detection.

This study achieved impressive results of object detection on the MS COCO benchmark. We hope this paper inspire developers and researchers to develop Faster R-CNN with better performance, and also push forward Faster R-CNN to apply in real-world scenarios.

REFERENCES

[1] Y. Ren, C. Zhu, and S. Xiao, "Object detection based on fast/faster RCNN employing fully convolutional architectures," *Math. Problems Eng.*, vol. 2018, pp. 1–7, Jan. 2018.

[2] X. Zhang, L. Li, H. Liu, P. Yang, and Y. Gao, "Disentangling classification and regression in Siamese-based network for visual tracking," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 27, pp. 1–12, Dec. 2022.

[3] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," in *Proc. 9th Int. Conf. Adv. Comput. Inf. Technol. (ACITY)*, Dec. 2019, pp. 119–133.

[4] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[5] X. Zhang, L. Cheng, B. Li, and H.-M. Hu, "Too far to see? Not really! Pedestrian detection with scale-aware localization policy," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3703–3715, Aug. 2018.

[6] X. Zhang, S. Cao, and C. Chen, "Scale-aware hierarchical detection network for pedestrian detection," *IEEE Access*, vol. 8, pp. 94429–94439, 2020.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9905, 2016, pp. 21–37.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[13] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[14] G. Jocher. (Oct. 2022). *Ultralytics/yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation (v7.0)*. [Online]. Available: https://ultralytics.com/

[15] N. Barazida, "YOLOv6: Next generation object detection—Review and comparison," Tech. Rep., 2022.

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[17] S. Rath, "YOLOv8 Ultralytics: State-of-the-Art YOLO Models," Tech. Rep., 2023, pp. 1–19. [Online]. Available: https://Learnopencv.Com/Ultralytics-Yolov8/

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Apr. 2015, pp. 1440–1448.

[20] 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.

[21] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. 30th Int. Conf. Neural Inf. Syst. (NIPS)*, Dec. 2016, pp. 379–387.

[22] T.-Y. 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)*, Jul. 2017, pp. 936–944.

[23] Z. Chen, S. Huang, and D. Tao, "Context refinement for object detection," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11212, 2018, pp. 74–89.

[24] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9908, Jul. 2016, pp. 354–370.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[26] Y. Zhang, J. Chu, L. Leng, and J. Miao, "Mask-refined R-CNN: A network for refining object details in instance segmentation," *Sensors*, vol. 20, no. 4, p. 1010, Feb. 2020.

[27] X. Wang, A. Shrivastava, and A. Gupta, "A-Fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3039–3048.

[28] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.
[29] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head R-CNN: In defense of two-stage object detector," 2017, *arXiv:1711.07264*.
[30] M. Chen, L. Yu, C. Zhi, R. Sun, S. Zhu, Z. Gao, Z. Ke, M. Zhu, and Y. Zhang, "Improved faster R-CNN for fabric defect detection based on Gabor filter with genetic algorithm optimization," *Comput. Ind.*, vol. 134, Jan. 2022, Art. no. 103551.
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 770–778.
[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[34] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 845–853.
[35] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, and M. Park, "PVANet: Lightweight deep neural networks for real-time object detection," 2016, *arXiv:1611.08588*.
[36] H. Li, Y. Liu, W. Ouyang, and X. Wang, "Zoom out-and-in network with map attention decision for region proposal and object detection," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 225–238, Mar. 2019.
[37] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*.
[38] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
[39] D. Misra, "Mish: A self regularized non-monotonic activation function," 2019, *arXiv:1908.08681*.
[40] B. Jiang, J. Xia, and S. Li, "Few training data for objection detection," in *Proc. 4th Int. Conf. Electron. Inf. Technol. Comput. Eng.*, Nov. 2020, pp. 579–584.
[41] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3296–3297.
[42] L. Yang, J. Zhong, Y. Zhang, S. Bai, G. Li, Y. Yang, and J. Zhang, "An improving faster-RCNN with multi-attention ResNet for small target detection in intelligent autonomous transport with 6G," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–9, 2022.
[43] W. Lin, J. Chu, L. Leng, J. Miao, and L. Wang, "Feature disentanglement in one-stage object detection," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109878.
[44] Z. Song, Y. Zhang, Y. Liu, K. Yang, and M. Sun, "MSFYOLO: Feature fusion-based detection for small objects," *IEEE Latin Amer. Trans.*, vol. 20, no. 5, pp. 823–830, May 2022.

FEITAO LI (Member, IEEE) received the Ph.D. degree from the University of Chinese Academy of Sciences, in 2016. He is currently a Lecturer with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. His main research interests include optical sampling and artificial intelligence.

TONG SUN received the Ph.D. degree from Chongqing University, Chongqing, China, in 2020. He is currently a Lecturer with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. His main research interests include digital agriculture, the agricultural Internet of Things, and agricultural AI.

PING DONG received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. She is currently a Lecturer with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. Her research interests include agricultural informatization and deep learning in agriculture.

QIANG WANG received the master's degree from Zhengzhou University, in 2009. He is currently an Associate Professor with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. His main research interests include smart agriculture and the agricultural Internet of Things.

YANLING LI received the Ph.D. degree from Xidian University, in 2013. She is currently an Associate Professor with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. Her main research interests include big data processing and artificial intelligence.

CHANGXIA SUN received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2013. She is currently an Advisor and an Associate Professor with the College of Information and Management Science, Henan Agricultural University, Zhengzhou, China. Her research interests include information security, artificial intelligence, and block chain technology.