Liu *et al.*
EURASIP Journal on Image and Video Processing (2022) 2022:9
https://doi.org/10.1186/s13640-022-00586-6

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Performance analysis of different DCNN models in remote sensing image object detection

Huaijin Liu[1*], Jixiang Du[2*], Yong Zhang[1] and Hongbo Zhang[3]

*Correspondence:
lhjhqdx@163.com;
jxdu@hqu.edu.cn

[1] College of Mechanical Engineering and Automation, Huaqiao University, Xiamen 361021, China
[2] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
[3] Fujian Key Laboratory of Big Data Intelligence and Security, Huaqiao University, Xiamen 361021, China

Abstract

In recent years, deep learning, especially deep convolutional neural networks (DCNN), has made great progress. Many researchers use different DCNN models to detect remote sensing targets. Different DCNN models have different advantages and disadvantages. In this paper, we use YoloV4 as the detector to "fine-tune" various mainstream deep convolutional neural networks on two large public remote sensing data sets—LEVIR data set and DOTA data set to compare the advantages of various networks. This paper analyzes the reasons why the effect of "fine-tuning" convolutional neural networks is sometimes not good, and points out the difficulties of object detection in optical remote sensing images. To improve the detection accuracy of optical remote sensing targets, in addition to "fine-tuning" convolutional neural network, we also provide a variety of adaptive multi-scale feature fusion methods to improve the detection accuracy. In addition, for the large number of parameters generated by deep convolutional neural network, we provide a method to save storage space.

Keywords: Object Detection, Feature Fusion, Deep Convolutional Neural Network, Remote Sensing Image

1 Introduction

High precision target detection in remote sensing image is very important in both military and civil fields. In military, the target detection of optical remote sensing images can be used to detect targets of non-metallic materials and provide enemy information, such as the number of military aircraft and oil tanks. For civil use, object detection based on optical remote sensing images can be used in urban planning, resource exploration and fighting against smuggling crimes. With the development of artificial intelligence, object detection technology based on deep convolutional neural network (DCNN) has been developed favourably. A common DCNN-based target detector usually consists of two parts, a backbone network for extracting image features and a head for bounding box prediction and classification prediction. For detectors running on the GPU platform backbone networks might be VGG [1], Inception [2, 3], ResNet [4], and ResNext [5]. For those detectors running on CPU platforms, their backbone networks might be lightweight networks Squeezenet [5], ShuffleNet [6], MobileNet [7, 8], and GhostNet

[9]. For the head part, it is usually divided into two categories, namely, two-stage target detector and single-stage target detector. The most representative two-stage target detectors include Faster R-CNN [10], R-FCN [11] and Mask R-CNN [12]. For single-stage target detector, the most representative models include SSD [13], RetinaNet [14] and YOLO [15, 16]. In recent years, anchor-free target detectors have been developed, such as keypoint-based single-stage target detectors CenterNet [17] and CornerNet [18], etc. The target detectors developed in recent years often insert some layers between the backbone network and the head. These layers are mainly used for feature fusion between different layers, which is called the neck in related papers. In general, the neck consists of multiple bottom-up paths and multiple top-down paths. Algorithms to achieve this network fusion mainly include the feature pyramid network (FPN) [19] and path aggregation network (PAN) [20].

With the increase of optical remote sensing image data sets published by aerospace and aviation companies in recent years, articles on optical remote sensing image target detection have gradually appeared. For example, Yang et al. [21] used convolutional neural network to detect aircraft in optical remote sensing images, Ding et al. [22] used deep convolutional neural network to detect cars in optical remote sensing images, and Dai et al. [23] used deep convolutional neural network to detect roads in optical remote sensing images. In the field of remote sensing, many scholars usually compare DCNN model with traditional machine learning. In fact, it is more practical to compare the detection performance of different DCNN models. In this paper, we compare different mainstream DCNN models mainly through two publicly available large-scale remote sensing data sets–three classes of LEVIR data sets and 15 classes of DOTA data sets. In addition, we use the single-phase target detector YoloV4 as the framework.

The main contributions of this paper are as follows: (1) we use YoloV4 as the detector to analyze the object detection performance comparison of different DCNNs on optical remote sensing images; (2) we propose an adaptive spatial feature fusion mechanism, which better integrates different scale features; (3) we fine-tune the backbone network CSPdarknet53 of YoloV4, adding dilated convolution and grouped convolution, reducing storage space and improving detection performance; and (4) we propose an efficient DCNN-based optical remote sensing object detection method that outperforms most of the state-of-the-art object detectors.

The rest of this paper is organized as follows: In Sect. 2, we will describe the development of DCNN and the network flow and basic principles of YoloV4. Section 3 describes how to fine-tune the DCNN model and design feature fusion, and provides a way to save storage space. In Sect. 4, the experimental results are analyzed and compared. Finally, Sect. 5 summarizes and discusses the article.

2 Related work

2.1 Development of deep convolutional neural network

As a feature extractor, the backbone network plays an important role in the performance of the detection model. With the development of network architecture in recent years, there are many excellent backbone networks. Therefore, it is of great significance to study the influence of different deep convolutional neural networks on target detection of optical remote sensing images. In the following, we will summarize the popular

backbone networks in recent years, including most of the mainstream deep convolutional neural networks.

1) Repeat network: developed by networks such as VGG, that is, stacking the same topological structure, and the whole network becomes a modular structure, which is adopted by almost all subsequent networks. The mainstream of such networks are VGG16 [1] and VGG19 [1].
2) Multi-path network: developed from the inception series, the input of the previous layer is divided into different branches for feature extraction, and finally the output results are spliced. Such networks include InceptionV3 [2] and InceptionV4 [3].
3) Skip-connection network: establish a transmission channel for shallow information and deep information, and change the original single linear structure. This kind of network mainly includes ResNet series and extremely improved series, such as ResNet50 [4], ResNet101 [4], ResNext50 [5], ResNext101 [5], SENet [24], SKNet [25], Res2Net [26].
4) Lightweight networks: this type of network mainly uses depth-wise seperable convolution to reduce network parameters and improve speed. Representative networks include SqueezeNet [5], ShufflenetV2 [6], MobileNetV2 [7], MobileNetV3 [8], and GhostNet [9].
5) Other networks: networks designed by drawing on the advantages of each branch, mainly EfficientNet [15], Darknet53 [27], CSPDarknet53 [28].

2.2 The principle of YoloV4

2.2.1 YoloV4

YoloV4 is an efficient target detection method. It mainly consists of four parts: Input, Backbone, Neck and Prediction. Backbone is used for feature extraction, Neck is used for multi-scale feature fusion, and Prediction is used for classification and bounding box prediction. YoloV4 adopts CSPDarknet53 as the feature extraction network, Neck adopts the structure of spatial pyramid pooling (SPP) [29], feature pyramid network (FPN) [19] and path aggregation network (PAN) [20] for feature fusion. Prediction generates anchor frame through clustering method, uses binary cross entropy loss for category prediction, and uses dimensional clustering machine to predict boundary frame. The network flow chart of YoloV4 is shown in Fig. 1.

2.2.2 FPN

FPN fuses the deep feature information with the shallow feature information through upsampling, thereby constructing the feature pyramid structure of different sizes. To better integrate features, YoloV4 has added a PAN after FPN. Combined in this way, FPN conveys strong semantic features from the top to the bottom, while PAN conveys strong localization features from the bottom to the top. Together, they work together to aggregate features from different backbone layers to different detection layers. To more intuitively understand the feature fusion work of FPN and PAN, we drew a flowchart of the Neck part, as shown in Fig. 2.

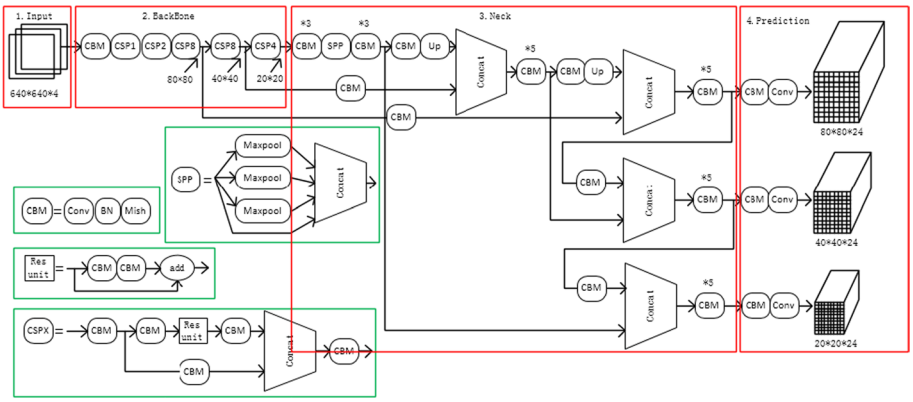

Fig. 1 Network flow chart of YoloV4

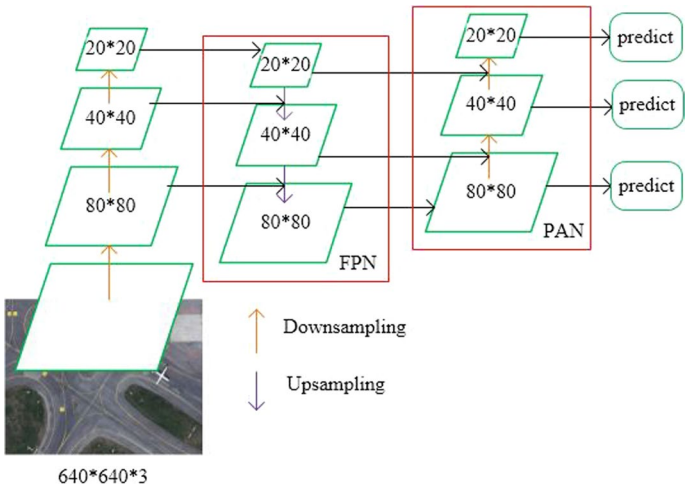

Fig. 2 Fusion structure of feature pyramid network and path aggregation network

2.2.3 Dimensional clusters

YoLo uses a dimensional cluster to predict the bounding box, as shown in Fig. 3. First, the YoLo model decomposes the image into S∗S grids, each of which is assigned three bounding boxes. Then, four coordinate values are predicted for each bounding box by dimensional clustering: t_x, t_y, t_w, t_h, where (t_x, t_y) is the predicted coordinate offset and (t_w, t_h) is the scale. The central coordinates (b_x, b_y) and length and width (b_w, b_h) of the prediction box can be calculated according to Equation (1-4). Where, p_w and p_h are the length and width of the bounding box, and (c_x, c_y) are the offset of the cell where the bounding box is located. Finally, the confidence can be obtained through the intersection and association ratio (IoU) between the prediction box and the real box, and the prediction box with low confidence can be eliminated by non-maximum suppression (nms).

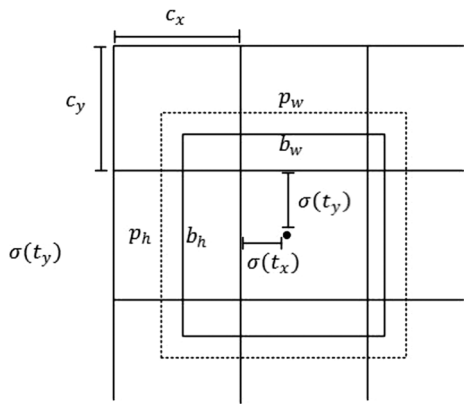

Fig. 3 Schematic diagram of dimensional clusterer

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

3 Methods

3.1 Fine-tuning the backbone structure in YoloV4

With the development of DCNN, some new DCNNs have emerged, and we try to use the new DCNNs in recent years for the feature extraction of YoloV4. At present, new and mainstream DCNNs architectures, such as Inception, SENet, MobileNet, EfficientNet, etc., cannot be directly applied to YoloV4. This is because their structural parameters are different, making their network outputs unsuitable for multi-scale feature fusion in the Neck stage, so we need to adjust these DCNNs frameworks. When different DCNNs are applied to YoloV4, the fine-tuning of the network structure is also different.

For VGG16, VGG19, ResNet50, and ResNet101, we removed the last feature pooling layer and the full connection layer, and then connected directly to the neck network. VGG networks mainly increases the network depth by stacking convolutional layers to improve detection accuracy. For example, the backbone network of VGG16 consists of 5 convolutional blocks and 5 max-pooling layers. The convolutional blocks respectively contain (2, 2, 3, 3, 3) convolutional layers with convolution kernel 3×3 and stride 1, as shown in Fig. 4a. The backbone network of VGG19 is also composed of 5 convolution blocks and 5 max-pooling layers, which, respectively, contain (2, 2, 4, 4, 4) convolutional layers with convolution kernel 3×3 and stride 1. ResNet networks use residual modules to fuse shallow information with deep information to solve the degradation problem of deep networks. For example, the backbone network of Resnet50 consists of a 7×7 convolutional layer with stride 2 and padding 3, a max-pooling layer and 4 residual blocks, where the 4 residual blocks contain (3, 4, 6, 3) resBottleneck modules respectively, as

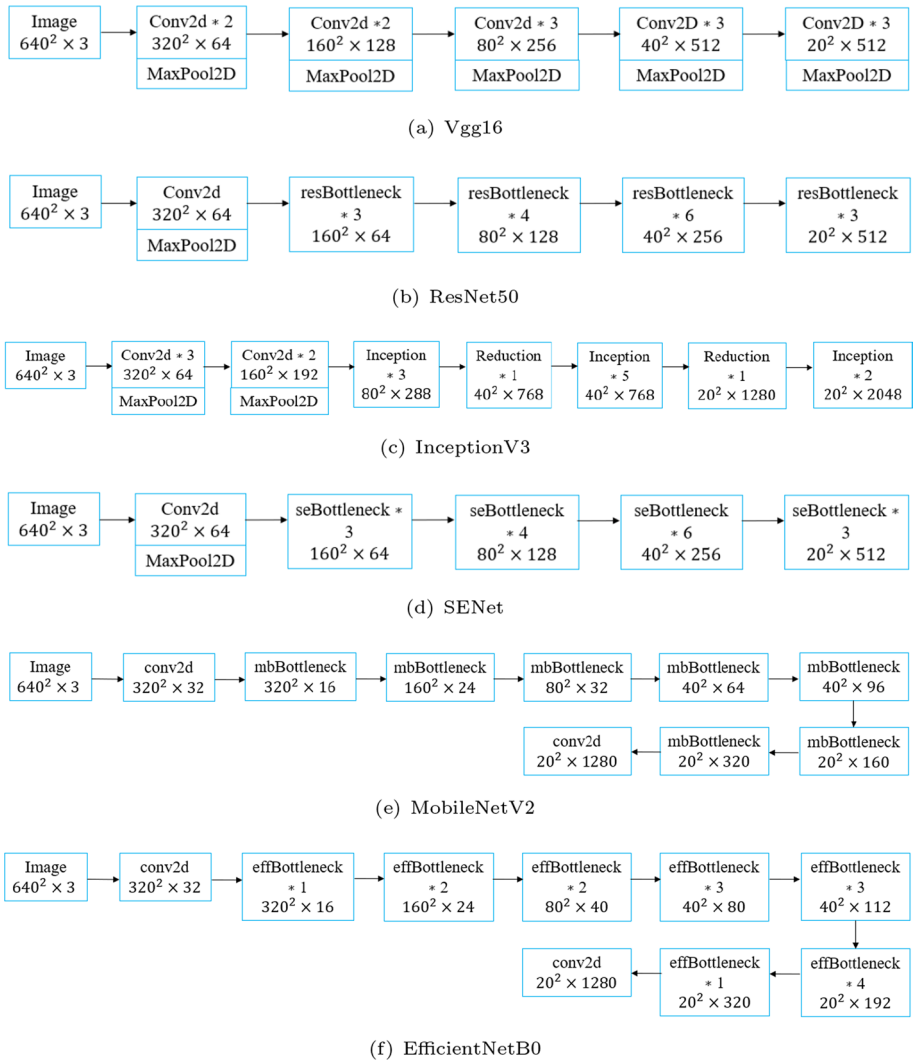

Fig. 4 Flowchart of different types of DCNNs

shown in Fig. 4b. The resBottleneck module uses convolution with stride 2 for downsampling, and performs residual learning between three convolution layers. The convolution kernels of the three convolution kernels are 1×1, 3×3 and 1×1 respectively.

For InceptionV3 and InceptionV4, to connect the neck network for feature fusion, we changed the effective convolution of the Inception module to the same convolution. Effective convolution is actually a convolution without padding, and the same convolution is a convolution with zero padding. The convolutional layers of the Inception backbone network do not have zero padding, resulting in the output feature maps not suitable for multi-scale feature fusion networks. Therefore, we use zero padding for the 3×3 convolutional layers in the Inception backbone. The backbone network of InceptionV3 consists of 5 convolutional layers, 2 max-pooling layers, 10 Inception blocks and 2 Reduction blocks, as shown in Fig. 4c. The backbone network of InceptionV4 consists of 3 convolutional layers, 17 Inception blocks and 2 Recuction blocks. The Inception block uses convolution kernels of different sizes to extract features from the upper layer

separately, and then concatenates them to obtain better results. The Recuction block uses a 3×3 convolutional layer with stride 2 and a max-pooling layer to downsample the feature maps of the upper layer.

For the Resnet evolution series, the ResNet modules in the four residual blocks of the Resnet backbone network are mainly replaced by the ResNeXt module, SENet module, SKNet module and Res2Net module. Figure 4d shows the backbone network of SENet, which consists of a 7×7 convolutional layer with stride 2 and padding 3, a max-pooling layer and 4 residual blocks. The four residual blocks contain (3, 4, 6, 3) seBottleneck modules, and each seBottleneck module mainly embeds an SE block in the resBottleneck module of ResNet to model the interdependence between channels. The SE block contains a global average pooling layer and two fully connected layers.

For SqueezeNet, ShuffleNetV2, MobileNetV2-V3 and GhostNet lightweight network models, we remove the last GlobalPool, Conv2d and FC layers. Figure 4e shows the backbone network of MobileNetV2, which mainly consists of a 3×3 convolution layer with stride 2, 7 mbBottleneck blocks and a 1×1 convolutional layer. The mbBottleneck block reduces network parameters and improves network speed by splitting the 3×3 standard convolution into a depthwise convolution and a point-wise convolution. The depthwise convolution is actually a grouped convolution, and the pointwise convolution is a 1×1 convolution.

There are also some other methods that use networks designed by borrowing the advantages of each branch. For example, EfficientNet improves detection accuracy by increasing the size of network depth, network width, and input image resolution. As shown in Fig. 4f, EfficientNetB0 consists of a 3×3 convolution layer with stride 2, 16 effBottleneck modules and a 1×1 convolutional layer. The effBottleneck module mainly consists of an SE block and a depth-wise separable convolutional block to scale the depth and width of the network model.

3.2 Adaptive multi-scale feature fusion method

YoloV3 uses the top-down FPN structure for feature fusion, as shown in Fig.5a. YoloV4 adds a bootom-up PAN structure on the basis of FPN for feature fusion, as shown in Fig.5b. To better fuse features of different scales, we design a new adaptive spatial feature fusion module (ASFF for short) inspired by spatial feature fusion [30]. The adaptive spatial feature fusion module allows the network to learn how to spatially filter the useless information of other layers and retain only the useful information for fusion. We first use the proposed ASFF module on the basis of FPN+PAN to further fuse features, as shown in Fig.5c. At the same time, we also use the ASFF module behind the FPN to fuse the features, as shown in Fig.5d. Experiments show that using the adaptive spatial feature fusion module behind the FPN can better improve the detection accuracy than using the adaptive spatial feature fusion module behind the PAN. This shows that the combination of FPN and ASFF modules can better fuse features.

The proposed adaptive spatial feature fusion module can be represented by formula (5) and formula (6). Equation (5) means that for each level, the features of all other levels will be adjusted to the same shape, and feature fusion will be performed according to the learnable weight parameters. Specifically: 1) for the level-l feature map (c, h, w), we first need to perform upsampling or downsampling operations on the feature maps of the

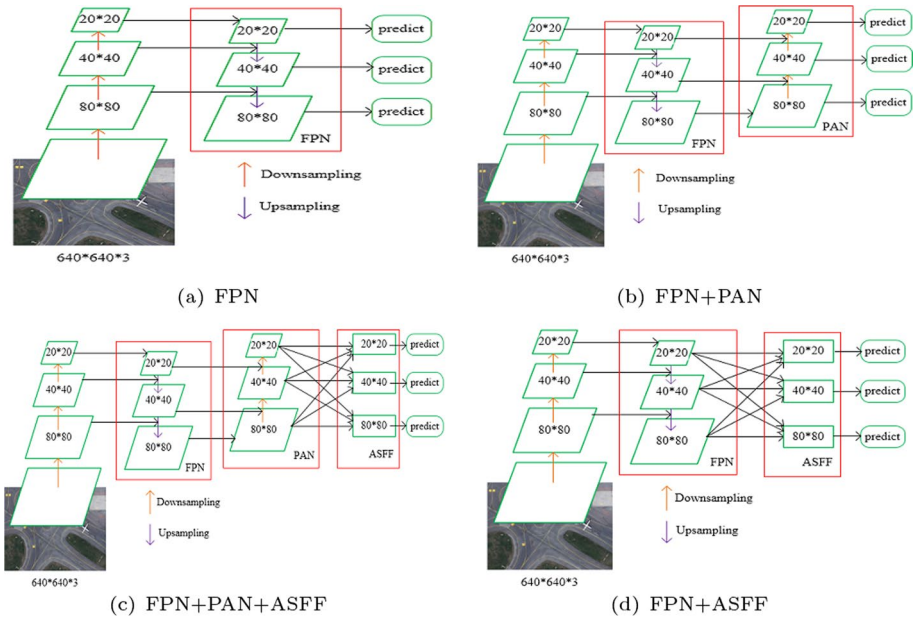

Fig. 5 Multi-scale feature fusion method. Figure 5a is the pyramid structure, Fig.5b is the combination structure of pyramid and path aggregation, Fig.5c is the combination structure of pyramid, path aggregation and adaptive spatial feature fusion, Fig.5d is the combination structure of pyramid and adaptive spatial feature fusion

remaining layers to resize them to the level-l output size and 2) then, the three adjusted feature maps are connected and a 1×1 convolutional layer is used for dimensionality reduction to obtain a $3 \times h \times w$ feature map, and then normalized by the softmax activation function to obtain the weight vectors of parameters α, β and γ; 3) Finally, the weight vectors α, β and γ are multiplied and summed with the three feature maps respectively to obtain the fused feature map.

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot x_{ij}^{1\to l} + \beta_{ij}^{l} \cdot x_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot x_{ij}^{3\to l} \tag{5}$$

$$\alpha_{ij}^{l} = \frac{e^{\lambda_{a_{ij}}^{l}}}{e^{\lambda_{\alpha_{ij}}^{l}} + e^{\lambda_{\beta_{ij}}^{l}} + e^{\lambda_{\gamma_{ij}}^{l}}} \tag{6}$$

3.3 Receptive field improvement method based on dilated convolution

There are many small target sizes in optical remote sensing images. With the deepening of neural network depth, the features of small targets are easily lost. To improve the accuracy of small target detection, we replace the standard convolution of the fifth stage of CSPDarknet53 with dilated convolution. Dilated convolution can improve the resolution without increasing the number of parameters. For the input image of $640 * 640$, the resolution of the output convolution feature layer in the fifth stage of CSPDarknet53 was reduced to $1/32 * 1/32$ of the original image. We replaced the standard convolution in the fifth stage of CSPDarknet53 with dilated convolution, and the resolution was only reduced to $1/16 * 1/16$ of the original image, and the process of feature learning was

deepened at the same time. The modified structure of the network is shown in Fig. 6. To simplify understanding, only the modification to the basic network part is drawn, and the structure of the entire detector is no longer drawn.

3.4 Model parameter reduction method based on grouped convolution

The size of the storage space required by the fine-tuned deep convolutional neural network model affects the applicability of deep convolutional neural network. To reduce the storage space, we refer to the innovation point of MobileNet module and replace the traditional convolution computation method with Depthwise convolution [8], where the grouping number is equal to the maximum common divisor of the number of input convolution channels and the number of output convolution channels. Compared with standard convolution, Depthwise convolution can reduce the amount of computation exponentially without affecting the accuracy, so as to reduce the number of model parameters and improve the operation speed. Finally, Pointwise convolution is used to solve the problem of "non-flow of information" in Depthwise convolution. This operation is equivalent to a regularization of the features extracted by grouped convolution, which is more conducive to the flow of information. The modified structure of the backbone network is shown in Fig. 7. To simplify understanding, the structure of the entire detector is no longer drawn.

4 Experimental results and discussion

All experiments in this section use Linux 18.04 system, RTX 3090 graphics card, Intel (R) Core (TM) i7-10700K (3.8GHz) CPU, 64G memory, PyTorch [31] framework commonly used for deep learning, and software, such as Python 3.8, CUDA 11.0, and Torch 1.7. This section compares the detection performance of different DCNNs on optical remote sensing images. The selected DCNN models include VGG series, Inception series, ResNet and improved series, lightweight series, EfficientNet, Darknet53, CSP-Darknet53. The network parameters were set as the input image size is 640, the cycle iterations is 100 times, the size of each batch is 16, the optimizer selects SGD, the network learning rate is set to 0.01, and the learning momentum is set to 0.937. The data set

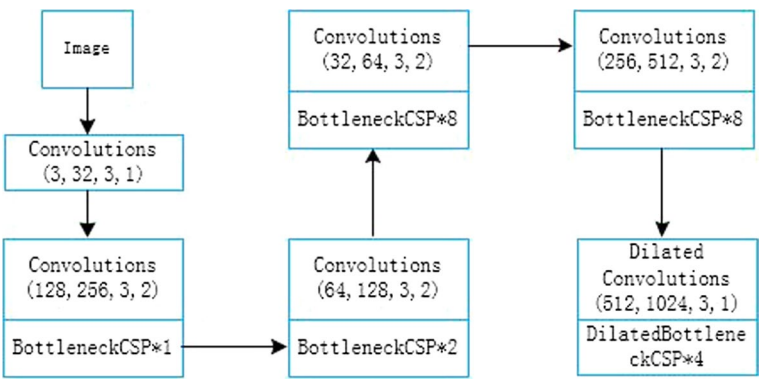

Fig. 6 Modified network structure diagram, in which BottleneckCSP is the basic structure of CSPDarknet53, and the standard convolution in DilatedBottleneckCSP is replaced with dilated convolution

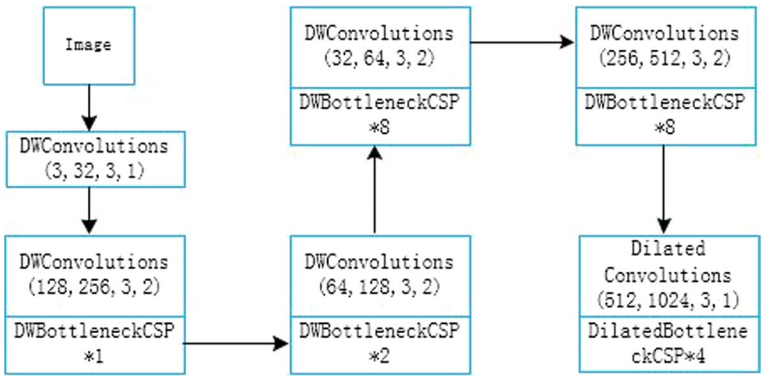

Fig. 7 Modified network structure diagram, where DWConvolutions represents grouped convolution, and the standard convolution in DWBottleneckCSP is replaced with grouped convolution

is LEVIR [32] of 3 classes and DOTA [33] of 15 classes of optical remote sensing data set published in 2018.

4.1 Qualitative analysis

First of all, we qualitatively analyze the performance of various deep convolutional neural networks on the optical remote sensing LEVIR data set, which is composed of 800∗600 pixels and more than 22,000 pictures, covering most types of ground features of human living environment. There are three target types in the data set: airplanes, ships and oil tanks, including 4724 airplanes, 3025 ships and 3279 oil tanks. The average number of objects per image is 0.5. Since there are many networks for experimental comparison, some test results of DCNN model are randomly selected here for effect display and qualitative analysis of experimental results. For the detection model parameters, we set the confidence threshold at 0.001 and the IoU threshold at 0.5. The detection results of LEVIR data set of different DCNN models are shown in Fig. 8.

By observing the test results of the randomly selected DCNN model on LEVIR data set, it can be concluded that DCNN can better obtain the detection results of LEVIR data set, and the positioning effect is more accurate. When the target size changes to some extent, YOLO detectors with different DCNN can still obtain better detection results. However, some DCNNs have some missed detections and false detections during the detection process, which indicates that the corresponding DCNNs need to further improve their classification ability.

Second, we will qualitatively analyze the representation of each DCNN on the optical remote sensing DOTA data set. The optical remote sensing data set contains a total of 21,046 images of 15 target types, with approximately 188,000 targets, and the image size is 800∗800 pixels. The detection model parameters were set as 0.001 confidence threshold and 0.5 IoU threshold. The detection results of different DCNN models on optical remote sensing DOTA data sets are shown in Fig. 9.

By observing the test results of the network model of the randomly selected DCNN models on the DOTA data set, it is known that it is sometimes difficult to accurately detect directly using the existing deep convolutional neural network, and there are many missed and false detection. The main reasons for the poor detection results are as

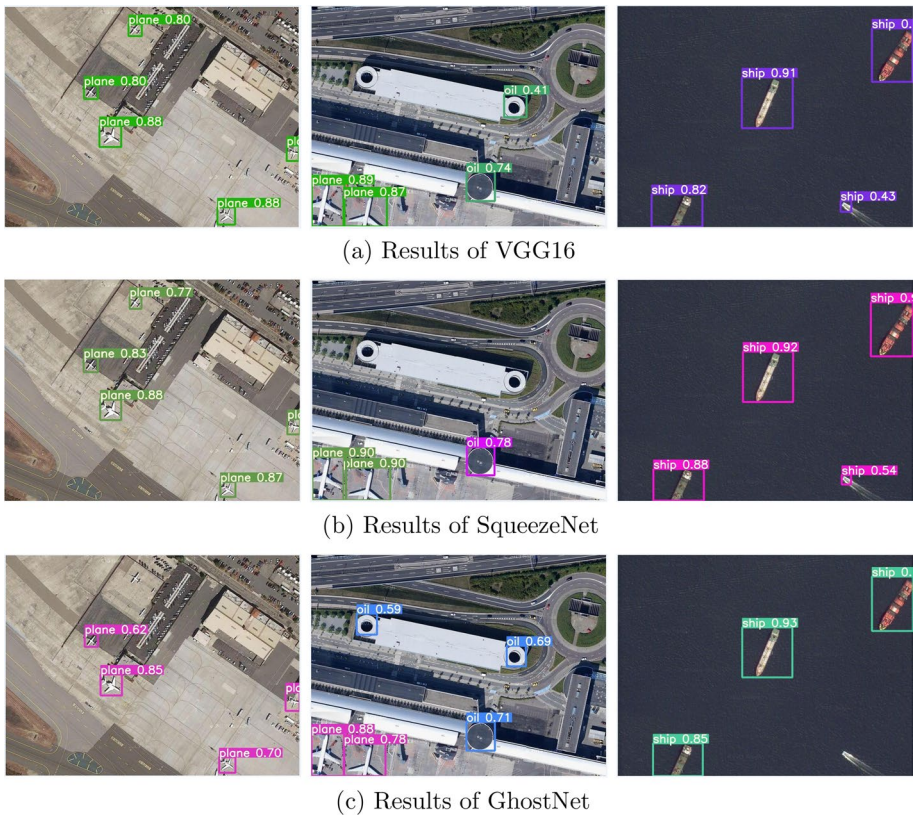

Fig. 8 Visualization results of some randomly selected deep convolutional neural network models on the LEVIR data set

follows: 1. After the small size target passes through the multi-layer convolutional neural network, the effective positioning information is lost seriously, so it is difficult to obtain accurate results directly by deepening and broadening the network. 2. The contrast between some targets and the surrounding environment is relatively low, and the classification ability of some deep convolutional neural networks is insufficient, so it is difficult to carry out a good classification operation. 3. DOTA data set is stored into many relatively dense small-size targets, and the dense distribution poses a certain challenge to the accuracy of target detection.

4.2 Quantitative analysis

Qualitative detection results can only give people a certain intuitive feeling, but lack persuasion. Therefore, quantitative analysis must be conducted to judge the advantages and disadvantages of each deep convolutional neural network. Mean average precision (mAP) is a common criterion for target detection in optical remote sensing images. AP is the area under the curve with accuracy on the vertical axis and recall rate on the horizontal axis. mAP is the average of AP's for all categories. The test time and storage space are very important for the real-time performance and practical application of the target detector, and we also use them as performance indicators. The quantitative structure of each DCNN model in the LEVIR and DOTA data sets is shown in Tables 1 and 2. Where

(a) Results of VGG16

(b) Results of SqueezeNet

(c) Results of GhostNet

Fig. 9 Visualization results of some randomly selected deep convolutional neural network models on the DOTA data set

mAP@0.5 means that when the detector IoU threshold is set to be greater than 0.5, the average precision AP of each category is calculated, and then, the average AP of all categories is calculated to get the mAP. mAP@[.5:.95] represents the average mAP at different IoU thresholds (from 0.5 to 0.95 with a step size of 0.05).

As can be seen from Table 1, the mAP of most DCNNs in LEVIR data set exceeds 80%. As for LEVIR data set, the target type is relatively single, the contrast with the background is relatively high, and the interference from the surrounding environment is relatively low, so it is not difficult to distinguish the background from the target. At the same time, LEVIR's target number is small and the target distribution is sparse, which is conducive to target detection. In addition, the study found that increasing the depth or width of DCNN does not necessarily improve the accuracy of target detection, such as VGG19, InceptionV4 and Resnet101. Meanwhile, for some improved residual convolutional networks, such as ResNeXt50 and SK-ResNet50, the target detection accuracy is not improved much. In addition, for some of the latest DCNN models, such as GhostNet

Table 1 Performance comparison of different DCNN models on LEVIR data set

Base-network	mAP@.5	mAP@[.5:.95]	Test-time(inference/nms/total)	Memory
VGG16	0.815	0.592	14.9 ms/1.2 ms/16.1 ms	321.2 M
VGG19	0.798	0.576	17.7 ms/1.1 ms/18.8 ms	363.7 M
InceptionV3	0.874	0.639	9.8 ms/1.1 ms/11.1 ms	394.0 M
InceptionV4	0.728	0.502	15.9 ms/1.1 ms/17.0 ms	543.0 M
ResNet50	0.83	0.60	9.50 ms/1.1 ms/10.6 ms	398.2 M
ResNet101	0.795	0.566	12.9 ms/1.1 ms/14.0 ms	558.2 M
ResNeXt50	0.782	0.557	13.8 ms/1.2 ms/15.0 ms	401.7 M
ResNeXt101	0.797	0.568	34.7 ms/1.1 ms/35.8 ms	559.5 M
SqueezeNet	**0.905**	**0.673**	6.0 ms/1.0 ms/6.9 ms	217.9 M
ShuffleNetV2	0.856	0.618	**3.8 ms/1.1 ms/4.9 ms**	**217.2 M**
DarkNet53	0.868	0.539	11.7 ms/1.2 ms/12.9 ms	532.7 M
MobileNetV2	0.873	0.634	4.0 ms/1.1 ms/5.1 ms	217.9 M
MobileNetV3	0.869	0.633	4.9 ms/1.1 ms/6.0 ms	217.9 M
SE-ResNet50	0.852	0.619	10.4 ms/1.4 ms/11.8 ms	426.0 M
SK-ResNet50	0.823	0.592	9.2 ms/1.1 ms/10.3 ms	260.1 M
CSPDarknet53	0.882	0.639	11.1 ms/1.2 ms/12.3 ms	420.8 M
EfficientB0	0.757	0.537	6.4 ms/1.2 ms/7.6 ms	241.1 M
EfficientB1	0.835	0.60	7.7 ms/1.0 ms/8.8 ms	261.3 M
GhostNet	0.809	0.579	4.6 ms/1.1 ms/5.7 ms	229.4 M
Res2Net50	0.761	0.536	11.2 ms/1.3 ms/12.5 ms	407.1M

The best results are in bold, the second best results are underlined

Table 2 Performance comparison of different DCNN models on DOTA data sets

Base-network	mAP@.5	mAP@[.5:.95]	Test-time(inference/nms/total)	Memory
VGG16	0.657	0.405	13.5 ms/1.4 ms/14.9 ms	321.8 M
VGG19	0.66	0.41	15.2 ms/1.4 ms/16.6 ms	364.2 M
InceptionV3	0.669	0.416	9.8 ms/1.3 ms/11.0 ms	394.5 M
InceptionV4	0.644	0.389	18.2 ms/1.2 ms/19.5 ms	543.5 M
ResNet50	0.677	0.424	11.8 ms/1.3 ms/13.1 ms	406.4 M
ResNet101	0.631	0.384	16.3 ms/1.4 ms/17.7 ms	558.7 M
ResNeXt50	0.677	0.424	20.7 ms/1.4 ms/22.2 ms	402.2 M
ResNeXt101	0.653	0.401	68.8 ms/1.3 ms/70.1 ms	558.7 M
SqueezeNet	0.651	0.396	7.3 ms/1.3 ms/8.6 ms	218.5 M
ShuffleNetV2	0.629	0.374	**4.4 ms/1.3 ms/5.7 ms**	**217.8 M**
Darknet53	0.68	0.432	15.3 ms/1.2 ms/16.5 ms	533.2 M
MobilenetV2	0.648	0.396	4.7 ms/1.3 ms/6.1 ms	218.5 M
MobileNetV3	0.658	0.399	5.9 ms/1.4 ms/7.3 ms	218.4 M
SE-ResNet50	0.672	0.42	12.5 ms/1.3 ms/13.8 ms	426.5 M
SK-ResNet50	0.663	0.411	13.0 ms/1.3 ms/14.3 ms	260.7 M
CSPDarknet53	**0.705**	**0.45**	14.1 ms/1.3 ms/15.4 ms	421.4 M
EfficientB0	0.646	0.394	7.9 ms/1.3 ms/9.2 ms	241.7 M
EfficientB1	0.646	0.392	10.0 ms/1.4 ms/11.4 ms	261.9 M
GhostNet	0.594	0.35	5.5 ms/1.4 ms/6.8 ms	229.9 M
Res2Net50	0.661	0.41	13.4 ms/1.4 ms/14.7 ms	407.6M

The best results are in bold, the second best results are underlined

and Res2Net50, the improvement of target detection accuracy is not necessarily effective. For lightweight DCNN models, such as ShuffleNet and MobileNet, it is effective to improve the detection speed, but the detection accuracy is not as good as CSPDarknet.

It can be seen from Table 2 that, for the detection of DOTA data set, many DCNN models do not achieve ideal results, and some networks achieve poor results. On mAP@.5, except CSPDarknet53, none of the other DCNN models exceeded 70%, and some of the DCNN models have mAP@.5 less than 65%. The main reasons for this result are as follows: 1. Compared with LEVIR data, the situation of DOTA data is more complex, with a larger number of targets and smaller target size. 2. The contrast between DOTA targets and the surrounding environment is relatively low and more dense, making it more difficult to distinguish targets. Similarly, it is found that increasing the depth or width of the convolutional neural network does not necessarily improve the accuracy of target detection, such as Inceptionv4 and Resnet101. Meanwhile, for some improved residual convolutional networks, such as ResNeXt50 and SK-ResNet50, the target detection accuracy is not improved much. For lightweight convolutional neural networks, such as Shufflenet and MobileNet, it is effective to improve the detection speed, but the detection accuracy is not as good as CSPDarknet. In addition, for some of the latest DCNN models, such as GhostNet and Res2Net50, the improvement of target detection accuracy is not necessarily effective. This further validates the conclusion analysis in Table 1.

It can be seen from Tables 1 and 2 that to achieve high precision performance on optical remote sensing data sets, it is not only necessary to increase the depth or width of the network, or simply change the structure of the convolutional neural network. Therefore, convolutional neural networks wants to achieve higher detection accuracy on optical remote sensing data sets, not only related to the network depth, width and network structure, but also related to the network feature fusion mode.

4.3 The test results of the proposed method in LEVIR data set are analyzed

In this section, each scheme we designed will be compared in detail. In the next section, we will compare with other commonly used detection methods based on deep convolutional neural network, such as Faster R-CNN, CenterNet, etc. First, we conducted an experimental comparison of the three proposed multi-scale adaptive spatial feature fusions on the LEVIR data set, and the experimental results are shown in Table 3. The basic network of all the following comparative experiments was CSPDarknet53 and the detector was YoloV4. Network parameters were set to loop iteration 100 times, the size of each batch was 16, the optimizer selected SGD, and the network learning rate was

Table 3 Effect of different multi-scale feature fusion methods on the detection result of LEVIR data set

Method	mAP@.5	mAP@[.5:.95]	Time	Memory
FPN	0.863	0.628	**11.7 ms**	**419 M**
FPN+PAN	0.882	0.639	12.3 ms	420.8 M
FPN+PAN+ASFF	0.89	0.638	14.2 ms	497.5 M
FPN+ASFF	**0.907**	**0.662**	19 ms	501 M

The best results are shown in bold

set to 0.01. The division ratio of training set, validation set and test set is 6:2:2. For the test parameters, we set the confidence threshold to 0.001 and the IoU threshold to 0.5. It can be seen from Table 3 that using PAN on the basis of FPN improves mAP@0.5 by 1.9% and mAP@[.5:.95] by 1.1%, and using ASFF on the structure of FPN+PAN can further improve mAP@0.5 by 0.8%. This shows that our proposed adaptive spatial feature fusion module is effective for feature fusion. At the same time, it can be seen from the table that using ASFF on the basis of FPN has the highest detection accuracy, which improves mAP@0.5 by 1.7% and mAP@[.5:.95] by 2.4% compared with the stucture of FPN+PAN+ASFF. This shows that the structure of FPN+ASFF can better perform feature fusion.

Next, we performed ablation experiments on the fine-tuned backbone network CSPdarknet53. The experimental results are shown in Table 4. The multi-scale feature fusion method uses the best-performing feature pyramid network and adaptive spatial feature fusion structure. For comparison, we use CSPdarknet53 as the baseline, and get 0.882% of mAP@0.5 and the pre-trained model size is 420.8 M. As can be seen from the second row of Table 4, we replaced the standard convolution of the fifth stage of CSPdarknet53 with dilated convolution, and the detection accuracy was improved by 1.2%. This is because dilated convolution can increase the resolution without increasing the amount of parameters, thereby improving the accuracy of small target detection. Then, as can be seen from the third row of Table 4, we replace the standard convolution of CSPdarknet53 with group convolution to greatly reduce the storage space. This is because compared to standard convolution, grouped convolution can reduce the amount of calculation exponentially without affecting accuracy. Finally, after the various schemes are integrated, the accuracy is improved by 2.6% and the storage space is reduced by 111.7 M compared with the original method.

4.4 Performance comparison with other DCNN-based detection methods

In this section, we compare our proposed method with several popular DCNN-based object detection methods. Our proposed method for optical remote sensing image object detection uses YoloV4 as the detector, fine-tunes the CSPDarknet53 backbone network and adopts the FASN structure for multi-scale spatial feature fusion. The detectors selected for comparison include two-stage detectors and single-stage detectors. The two-stage detectors is Faster R-CNN [10], and the single-stage detectors include RetinaNet [14], YoloV3 [15], YoloV4 [16], and the anchorless CenterNet [17].

Table 4 Effect of dilated convolution and grouped convolution on the detection results of LEVIR data set

CSP	DC	GC	FASN	mAP@.5	mAP@[.5:.95]	Time	Memory
√	×	×	×	0.882	0.639	**12.3 ms**	420.8 M
√	√	×	×	0.894	0.641	14.5 ms	371.1 M
√	√	√	×	0.892	0.642	13.7 ms	**212.8 M**
√	√	√	√	**0.908**	**0.667**	15.4 ms	291.1 M

CSP here stands for multi-branch convolutional network, DC stands for dilated convolution, GC stands for grouped convolution, and FASN stands for feature pyramid network and adaptive spatial feature fusion structure. The best results are shown in bold

Table 5 Comparison of detection accuracy and detection time of different target detection methods

Method	Backbone	mAP@.5	mAP@[.5:.95]	Time
Faster R-CNN	ResNet50	0.750	0.497	96.5 ms
RetinaNet	ResNet50	0.816	0.591	22.6 ms
CenterNet	Harglass52	0.771	0.557	14.2 ms
YoloV3	DarkNet53	0.768	0.539	13.4 ms
YoloV4	CSPDarknet53	0.882	0.639	**12.3 ms**
Our	CSPDarknet53*	**0.908**	**0.667**	15.4 ms

*Denotes our optimized CSPDarknet53 backbone network. The best results are shown in bold

The comparison detector we selected will compare with the deep convolutional neural network commonly used by the detector. The corresponding detection structure of each detection method is shown in Table 5. From the detection results, the detection results of Faster R-CNN, CenterNet and YoloV3 are relatively poor, while our method has good results in both accuracy and efficiency, and has lightweight characteristics.

5 Conclusions

In the target detection of optical remote sensing image based on DCNN, two parts should be considered generally, one is the selection of deep convolutional neural network, the other is the selection of detector. In this paper, the single-stage detector YoloV4 is chosen as the detector considering the real-time performance of the project. Although the single-stage detector is not good at small scale target detection in optical remote sensing images, it can meet the requirements of accuracy as well as efficiency with the fine-tuning of network structure. With different researches on detectors, DCNN model is also developing continuously. It is significant to study the influence of DCNN model on optical remote sensing image target detection. Through the study, it is found that the deep convolutional neural network with multi-scale feature fusion is suitable for the single-stage detector YoloV4. We fine-tuned the network structure based on YoloV4 and tested two optical remote sensing data sets. The experimental results show that the proposed method can achieve good detection results in both simple and complex cases.

Abbreviations

DCNNs	Deep convolutional neural networks
FPN	Feature pyramid network
PAN	Path aggregation network
CSP	Multi-branch convolutional network
DC	Dilated convolution
GC	Grouped convolution
FASN	Feature pyramid network and adaptive spatial feature fusion structure
IoU	Intersection and association ratio
NMS	Non-maximum suppression
mAP	Mean average precision

Acknowledgements

We would like to thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Author contributions

All authors participated in the design of the analytics, performance measures, experiments, and writing of the manuscript. All authors read and approved the final manuscript.

Funding
This work is supported by the Natural Science Foundation of China (Nos. 61673186 and 61871196), the Natural Science Foundation of Fujian Province of China (No. 2019J01082) and the Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University (ZQN-YX601).

Availability of data and materials
The data set, benchmark and related open-source codes are available at https://github.com/liuhuaijjin/PADCNN.

Declaration

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Received: 18 October 2021 Accepted: 25 April 2022
Published online: 07 June 2022

References

1. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
2. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
3. C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence (2017)
4. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
5. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
6. X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6848–6856 (2018)
7. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
8. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
9. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1580–1589 (2020)
10. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
11. J. Dai, Y. Li, K. He et al., R-fcn: Object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems **29**, 379–387 (2016)
12. K. He, G. Gkioxari, P. Dollar et al. Mask rcnn[C]. Proceedings of the IEEE international conference on computer vision. pp. 2961-2969 (2017)
13. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016). Springer
14. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
15. J. Redmon, A. Farhadi, Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
16. A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
17. K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6569–6578 (2019)
18. H. Law, J. Deng, Cornernet: detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 734–750 (2018)
19. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
20. S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
21. Y. Li, K. Fu, H. Sun, X. Sun, An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images. Rem. Sens. **10**(2), 243 (2018)
22. P. Ding, Y. Zhang, P. Jia, X.I. Chang, A comparison: different dcnn models for intelligent object detection in remote sensing images. Neural Process. Lett. **49**(3), 1369–1379 (2019)
23. J. Dai, R. Ma, H. Ai, Semi-automatic Extraction of Rural Roads From High-Resolution Remote Sensing Images Based on a Multifeature Combination. IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2020)

24. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
25. X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 510–519 (2019)
26. S. Gao, M.-M. Cheng, K. Zhao et al., Res2net: A new multi-scale backbone architecture. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 652–662 (2019)
27. M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114 (2019). PMLR
28. C.-Y. Wang, H.-Y.M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, Cspnet: a new backbone that can enhance learning capability of cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 390–391 (2020)
29. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
30. S. Liu, D. Huang, Y. Wang, Learning spatial fusion for single-shot object detection. arXiv preprint arXiv:1911.09516 (2019)
31. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. **32**, 8026–8037 (2019)
32. Z. Zou, Z. Shi, Random access memories: a new paradigm for target detection in high resolution aerial remote sensing images. IEEE Trans. Image Process. **27**(3), 1100–1111 (2017)
33. G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, Dota: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3974–3983 (2018)

Publisher's Note