Automatic Detection of Ship Based on Rotation Invariant RetinaNet

Luyang Zan, Kaixuan Lu, and Zhengchao Chen

Abstract Ship detection in the aerial image is an active yet challenging in remote sensing image processing. Due to the difficulties of locating the arbitrarily rotated ships and the complexity of the background around the ship, the existing deep learning-based object detection algorithm is difficult to accurately identify the ship target. For multi-angle ships, the horizontal bounding box contains not only ship objects but also a lot of irrelevant backgrounds, they are learned by the deep convolutional network as a learning object, and this leads to the common misalignment between the final classification confidence and localization accuracy. In this paper, we propose a new module called rotation RetinaNet (RRNet) to handle this problem. Firstly, we start from the popular one-stage RetinaNet approach, with ResNet50 as a basic network. Then, we apply a rotation head to RetinaNet to guarantee the rotation invariance of the model. Finally, we add an angular loss to the original loss so that the model can learn the angular offset of the bounding box. As a consequence, the proposed RRNet achieves high performance on the open-source DOTA datasets.

Keywords Ship detection · Convolutional neural network · Rotation RetinaNet

1 Introduction

As a comprehensive technology of earth observation, remote sensing has a large range of high-resolution imaging capabilities. With the rapid growth of the number of high-resolution satellites in orbit, the acquisition speed of remote sensing image data has been accelerated, and the data volume has increased significantly. As a result, the comprehensive observation capability of human beings to the earth has reached an unprecedented level. Ship detection is the focus of research in remote sensing object

L. Zan · K. Lu · Z. Chen (✉)
Airborne Remote Sensing Center, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China
e-mail: chenzc@radi.ac.cn

L. Zan
College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

© Springer Nature Singapore Pte Ltd. 2020

L. Wang et al. (eds.), *Proceedings of the 6th China High Resolution Earth Observation Conference (CHREOC 2019)*, Lecture Notes in Electrical Engineering 657,
https://doi.org/10.1007/978-981-15-3947-3_39

detection. It is very important in the fields of military defense and port monitoring. Over the past years, experts and scholars have proposed many methods to detect ships [1–3], these works have achieved remarkable results. These methods only focus on the position information of ships but ignore the direction information, which is also of great significance. The ships on the sea are easy to detect because of the simple background; however, the inshore ships are more difficult to be detected than other objection detections [4]. The inshore ships are often of large aspect ratio, tightly assigned, and arbitrary rotated, other than this, the harbors and ships are similar in gray level and texture. Those factors will bring great challenges to rotated ship detection.

In recent years, with the development of deep learning technology, the feature learning algorithm based on deep neural network has made a breakthrough in many research fields such as computer vision and natural language processing. In essence, deep learning is a deep neural network containing multiple hidden layer nodes. Through feature extraction step by step, the optimal representation of data is obtained. In the context of remote sensing big data, the intelligent deep learning algorithm has significant advantages in performance and precision in the information extraction of complex high-resolution remote sensing images. As an efficient target feature extraction network, convolutional neural network has been widely used in target detection model. In recent years, Pascal VOC [5], COCO [6], DOTA [7], and DIOR [8], the dataset of object detection in the field of large-scale computer vision or remote sensing, have made a breakthrough in the object detection algorithm based on deep learning. According to whether candidate regions are generated or not, the existing deep learning-based object detection model can be divided into two categories: candidate region-based method and location-based regression method. Since R-CNN was proposed, target detection models based on candidate regions have achieved great success in the field of natural images, including R-CNN [9], fast R-CNN [10], faster CNN [11], FPN [12], HRNet [13], etc. This method divides object detection into two phases: the first phase focuses on generating a set of candidate regions that may contain objects; the second phase aims to further classify the candidate regions obtained in the first phase and adjust the coordinates of the boundary boxes. The method based on position regression is to use a single-stage position detector for object instance prediction to simplify the detection to a regression problem. Compared with the method based on candidate regions, this method is simpler and faster, including YOLO [14], SSD [15], RetinaNet [16], etc. Among them, RetinaNet is the state-of-the-art detector, which can guarantee speed and accuracy at the same time. These methods can be applied to ship detection of remote sensing images and can significantly improve the efficiency and automation level of ship recognition. However, these methods cannot inference the angle information of the ship.

Based on the framework of target detection network RetinaNet, we proposed an arbitrary rotated network named as rotated RetinaNet (RRNet), which is more suitable for ship detection.

The rest of the paper is organized as follows. Section 2 introduces the details of the dataset and the proposed method. Section 3 describes an experiment conducted on

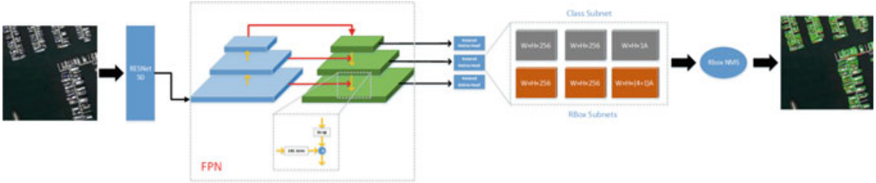


Fig. 1 Pipeline of RR, including FPN and Rotated Retina Head

DOTA datasets to evaluate the performance of the method. Finally, Sect. 4 concludes the results of the method.

2 Methodology

2.1 *Pipeline*

Our rotation RetinaNet's overall structure is illustrated in Fig. 1. Our network is a one-stage method based on RetinaNet. By adding a Rotated Retina Head structure to make the network learn target's direction information, the final detection result changed from the horizontal bounding box (HBB) to the oriented bounding box (OBB). We use ResNet50 as our backbone network to generate a multi-scale convolution feature pyramid, then we use an FPN module to fusion different scale feature maps, on each FPN level we add two additional small FCN subnet, one for classes prediction the other for RBox regression. After the five-parameter regression and the rotation non-maximum suppression (R-NMS) operation for each RBox in the RBox regression subnet, we get the final detection results.

2.2 *Network Design*

2.2.1 Data Augmentation and Multi-scale Training

To ensure the robustness of the model and avoid overfitting of the network, the training set images are randomly flipped and randomly mirrored in the pre-processing stage. Because the proportion of ships in the image is too small and the distribution is too dense, to ensure the accuracy of the ship detection, we will expand the image from 640 × 640 to 1024 × 1024 during the training process. Small ships are zoomed into make them easier to detect. The experimental results show that the model can reduce missed detection after the image is enlarged.

2.2.2 Feature Pyramid Networks

On the one hand, because the ship's target is usually small, the shallow network responsible for detecting small targets usually contains less semantic information, so the detection network cannot detect small ship targets well. On the other hand, because the image contains a large number of ships of different scales, it is difficult for the network to cope with the problem of multiple different scales of the same object. Therefore, in this paper, the FPN module is added to the detection model, and the FPN is used to make the detection network overcome the differences mentioned above in scale and enhance the semantic information of the shallow network.

The basic idea of FPN: Using different scales of feature information to predict targets of different scales, the specific structure in FPN is mainly divided into two processes: bottom-up and top-down. Bottom-up process: The process by which a common convolutional network extracts features through feedforward calculations. Top-down process: Contains two steps of upsampling and horizontal join. Firstly, the features extracted from the bottom-up process are sampled so that the scale of the high-level features can satisfy the scale of the horizontal connection with the features of the lower layer, and then, the obtained sampling features and the underlying features are added to the pixels, that is, the horizontal connection gets a multi-level feature map.

2.2.3 RBox Parameterized Description and RBox Regression

For the object detection method using the HBB to locate the target position, the HBB is usually parameterized into four variables consisting of the coordinates of the center point and the width and height, for example, box $= (x_c, y_c, w, h)$. The objects with arbitrary direction on the remote sensing image, such as the ship target in the image with a tilt angle of 45°, may account for less than 40% of the minimum external bounding box. The learning target of the network leads to a decrease in the detection accuracy of the network. Therefore, this paper uses the arbitrary quadrilateral to mark the target object. The arbitrary quadrilateral is a more accurate definition method of the target position. The labeling of the arbitrary quadrilateral is usually represented by a total of eight parameters of its four endpoints coordinates $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$, and then, in the bounding box regression, we simplify the above-mentioned arbitrary quadrilateral to a OBB with the following five parameters (x_c, y_c, w, h, θ). As show in Fig. 2, where the box Contains the coordinates of the center point (x_c, y_c), the width w and the height h and the direction angle θ.

For RBox regression [13], the following five coordinates are used for parameterization:

$$t_x = (x - x_a)/w_a,\ \ t_y = (y - y_a)/h_a \tag{1}$$

$$t_w = \log(w/w_a),\ \ t_h = \log(h/h_a) \tag{2}$$

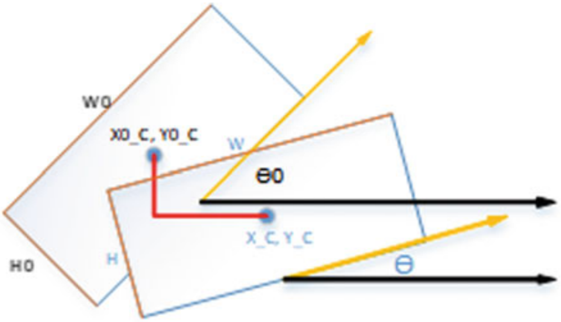


Fig. 2 RBox parameterization

$$t_{\theta} = \sin(\theta - \theta_a) \tag{3}$$

$$t_x^* = \left(x^* - x_a\right)/w_a,\ \ t_y^* = \left(y^* - y_a\right)/h_a \tag{4}$$

$$t_w^* = \log\left(w^*/w_a\right),\ \ t_h^* = \log\left(h^*/h_a\right) \tag{5}$$

$$t_{\theta}^* = \sin\left(\theta^* - \theta_a\right) \tag{6}$$

where the variables x, x_a, x^* are for the prediction box, default box, and ground truth box (as for x, y, w, h, θ).

2.2.4 Rotated Retina Head

Rotated Retina Head consists of two similarly structured sub-networks, one for classification and the other for RBox regression. We add this Rotated Retina Head module on each FPN level output layer to generate a series of candidate rboxes and classes.

Classification Subnet

The class prediction sub-network discriminates the objects in each spatial position in the feature map. The subnet attaches a small FCN to each FPN layer. The parameters of this subnet are shared at all pyramid levels. The input feature map comes from each FPN layer and has 256 channels. Then, four 3 × 3 conv layers are applied. Each layer has 256 filters. After each filter, ReLU is activated, and then, a 3 × 3 conv layer has two filters. Finally, the softmax activation function is used to calculate a set of category confidence for each region to complete the classification task.

RBox Regression Subnet

Similar to the class prediction sub-network, the position regression sub-network adds a small FCN structure to each FPN layer to return the offset between the anchor box and the nearest ground truth object. Each position has a 5A output. Relative to each a anchor of each spatial position, there are 5 offsets calculate by the prediction anchor and the ground truth box to achieve accurate positioning. Classification and regression have the same structure and different parameters.

2.2.5 Multi-task Loss Function

In this paper, the multi-task loss function with angle penalty is used for the end-to-end training of the network, and the loss function is shown as follows:

$$L = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, l_i) + \frac{1}{N_{\text{reg}}} \sum_i p_j \big(L_{\text{reg}}(v_j^*, v_j)\big) \tag{7}$$

Here, l_i represents the object category, p_i represents the probability distribution of different categories calculated by softmax function, v_j represents the coordinate vector deviation of model prediction output, and v_j^* represents the real vector deviation. Where classification loss L_{cls} is the cross entropy loss, regression loss L_{reg} is smooth L1 loss.

3 Experiment

3.1 Dataset and Setting

Object detection dataset of aerial remote sensing image. It contains 2806 expert interpreting aerial images from different sensors and platforms. Each image ranges in size from 800 × 800 to 4000 × 4000 pixels and contains a wide range of scales, directions, and shapes. These DOTA images are then annotated by aviation image interpretation specialists with an arbitrary quadrilateral. We selected the images containing ship targets in the DOTA dataset, obtained a total of 1572 + 4902 samples, divided the training set and test set, and obtained 4902 training set samples and 1572 test set samples. We divide the images into 640 × 640 sub-images with an overlap of 100.

All the experimental contents in this paper were completed under the PyTorch deep learning framework. The hardware environment is a server with four NVIDIA GeForce Titan XP GPU (12 GB memory) and Intel Xeon E5 CPUs. We use ResNet50 imagenet pre-training model to initialize the network. For the DOTA dataset, we trained 60 epochs in total.

3.2 Performance Evaluation

We evaluate the performance of RRNet by the ship's validate set of DOTA. The performance evaluation of neural network includes mean average accuracy, precision, and recall. The classification results of each assessment can be divided into four categories:

- True positives (TP): the number of ships correctly classified as ships.
- False positives (FP): the number of ships incorrectly classified as ships.
- True negatives (TN): the number of ships correctly classified as non-ships.
- False negatives (FN): the number of ships incorrectly classified as non-ships.

Precision is often used to measure the ability of a model to distinguish binary classes (that is, ship or non ship). In this study, combined with TP and FP calculation:

$$\text{Precison} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

Recall is a commonly used method to quantify the performance of algorithms in machine learning. It measures the proportion of correctly detected ships in all positive samples, i.e., a missing metric.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

mAP is a specific value that can more intuitively describe the performance of the detector, and it is the area of the precision–recall curve and the horizontal axis. The definition is described as follow:

$$\text{mAP} = \int_0^1 P(R)\mathrm{d}R \tag{10}$$

3.3 Evaluation of RRNet

We use 4375 training sets and 1357 validation sets to train RRNet, where resize is set to 640 * 640. We use the pre-training model ResNet50 to initialize the network and train the model by using the optimizer SGD with 0.9 momentum, 0.0001 weight decay, and batch size 8, and maximum number of epochs is set to be 50. The initial learning rate at the beginning of training is set to be 0.01, we use the learning policy "warm up" to adjust the learning rate, the warm-up step is set to be [10, 30, 50], and the warm-up ratio is 1/3.

On one hand, for the small dense ships because of its small volume quantity, in the process of feature extraction network constantly down sampling of the figure map will be lost in a large number of target information, thus detection model needs to be done the ship detection task using the characteristics of the previous figure map, but due to the characteristics of the previous figure map layer contained in the semantic information is too shallow, it is easy to make the detection model can't locate the ship object precisely. On the other hand, for large ships with characteristics of strong semantic, due to the complexity of its semantic information, so the model need contain deep characteristic figure of high-level semantic information to complete judgment, and high-level semantic feature maps cannot very accurately locate the position information of large ships. The RRNet is a multi-scale detection network, and it chooses five different feature maps to combine, the lowest layer is mainly used for small ship detection, and the highest layer is mainly used for large ship detection. When training RRNet on the ship datasets, the trend curve of mAP is shown as Fig. 3, and RRNet achieves 66.8% mAP after 50 epochs. In order to prove our proposed method is more outstanding, we compare the RRNet with other methods, and the results are shown in Table 1. In Table 1, the SSD, YOLOv2, and R-FCN [15] are horizontal region detectors, and the R-DFPN [16] is rotation region detector. The result shows that the mAP of our RRNet is better.

In addition to using mAP to evaluate the detection accuracy of the model, we also calculate the accuracy and recall rate of the model to evaluate the performance of the model and record the accuracy and recall rate under different thresholds in the ship validation set. Figure 4 shows the precision–recall curve, it can be seen that the best precision rate and recall rate of our proposed method are 93.9% and 72.1%, respectively.

Figure 5 shows some detection results in different scenes, including tightly arranged ships and arbitrary oriented ships, ships in the harbor and on the sea. These

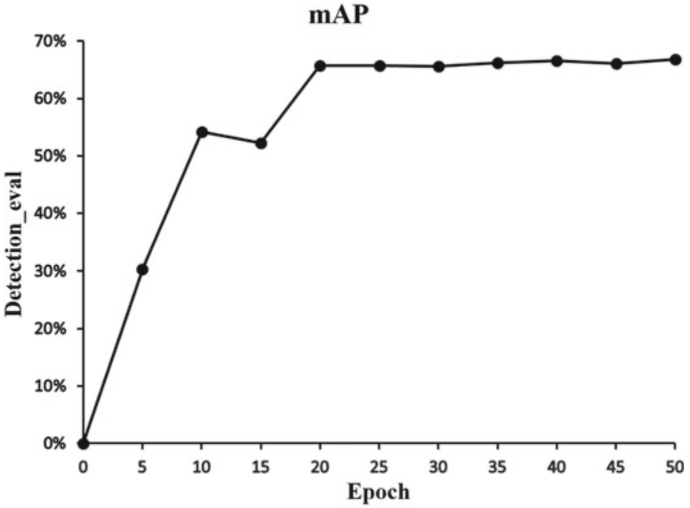


Fig. 3 mAP of RRNet

Table 1 Detection performance comparisons of different models

Methods	AP of ship
SSD	13.21
YOLOv2	7.37
R-FCN	7.45
R-DFPN	54.78
Ours	66.80

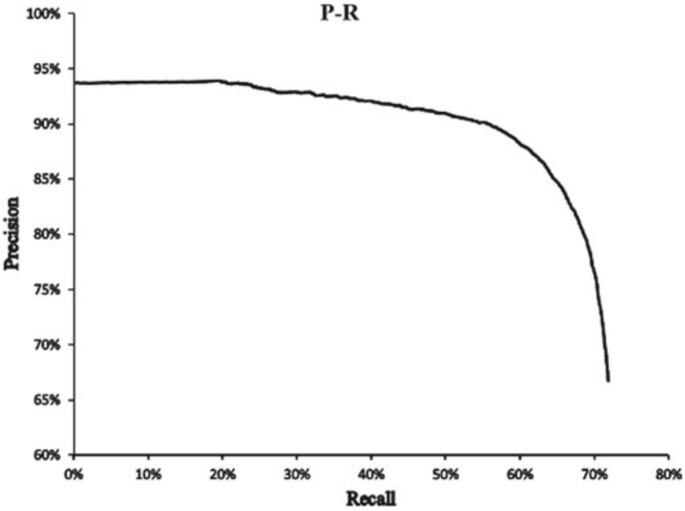
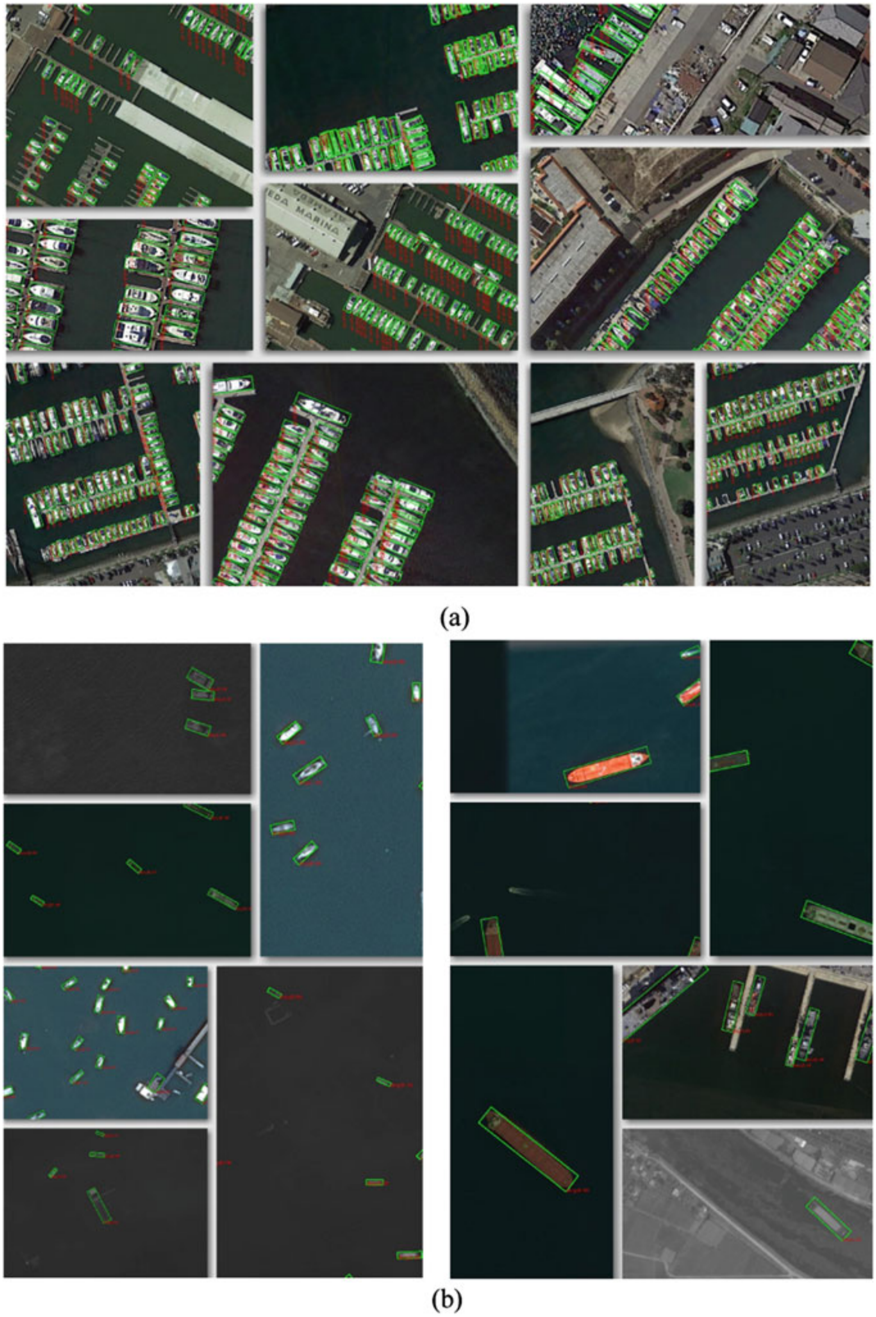


Fig. 4 Precision–recall curve of RRNet

complex scenes and various ships have a great influence on the performance of RRNet.

4 Conclusion

In this paper, we propose a rotated neural network named RRNet to detect arbitrary-oriented ships. We apply a rotation head to guarantee the rotation invariance of the model and add an angular loss to the original loss so that the model can learn the angular offset of the bounding box. The results of the experiment based on the DOTA dataset show that our proposed RRNet has a great performance on rotated ship detection. However, due to the difficulty of features in ship detection, our method has a relatively low recall rate, there still exists some missed inspections, and we need to explore how to effectively increase recall rate in the future.


(a)

(b)

Fig. 5 Detection results of ships in different scenes. **a** Detection results of arbitrary oriented and tightly arranged ships; **b** detection results of ships in the harbor and on the sea

Acknowledgements This work was jointly supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA19080302 and No. 23100304).

References

1. Zhang R, Yao J, Zhang K, Feng C, Zhang J (2016) S-CNN ship detection from high-resolution remote sensing images. ISPRS Int Arch Photogramm Remote Sens Spatial Inf Sci XLI-B7:423–430
2. Kang M, Ji K, Leng X, Lin Z (2017) Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection. Remote Sens 9(8):860
3. Tang J, Deng C, Huang GB, Zhang B (2014) Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. IEEE Trans Geosci Remote Sens 53(3):1174–1185
4. Chen Z, Lu K, Gao L, Li B, Gao J, Yang X, Yao M, Zhang B (2019) Automatic detection of track and fields in China from high-resolution satellite images using multi-scale-fused single shot multibox detector. Remote Sens 11:1377
5. Everingham M, Van Gool L, Williams CKI et al (2010) The pascal visual object classes (voc) challenge. Int J Comput Vision 88(2):303–338
6. Lin TY, Maire M, Belongie S et al (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, Cham, pp 740–755
7. Xia G-S, Bai X, Ding J, Zhu Z, Belongie S, Luo J, Datcu M, Pelillo M, Zhang L (2018) DOTA: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, 18–22 June 2018
8. Li K, Wan G, Cheng G, Meng L, Han J (2019) Object detection in optical remote sensing images: a survey and a new benchmark
9. Girshick R, Donahue J, Darrell T et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
10. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
11. Ren S, He K, Girshick R et al (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
12. Lin TY, Dollár P, Girshick R et al (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
13. Huang J, Zhu Z, Huang G (2019) Multi-stage HRNet: multiple stage high-resolution network for human pose estimation. arXiv preprint arXiv:1910.05901
14. Redmon J, Divvala S, Girshick R et al (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
15. Liu W, Anguelov D, Erhan D et al (2016) SSD: single shot multibox detector. In: European conference on computer vision. Springer, Cham, pp 21–37
16. Lin T-Y, Goyal P, Girshick RB, He K, Dollár P (2017) Focal loss for dense object detection. In: ICCV