# Parallel Cascade R-CNN for object detection in remote sensing imagery

# Parallel Cascade R-CNN for object detection in remote sensing imagery

**Jingyou Hou[12], Hongbing Ma[12*] and Shengjin Wang[12]**

[1] Department of Electronic Engineering, Tsinghua University, Beijing, 100086, China

[2] the Beijing National Research Center for Information Science and Technology (BNRist)

* hbma@tsinghua.edu.cn

**Abstract.** Object detection in remote sensing image is a challenging task in computer vision. Remote sensing images have different characteristics compared with conventional images. Especially object detection in remote sensing image needs to focus on small targets with different ratio and orientation. In this paper, we propose a novel detection architecture for remote sensing imagery to solve the problem of scale diversity. On the basis of Cascade R-CNN, we have developed Parallel Cascade R-CNN. In the second stage, parallel detection heads are used for separate detection, and their RoIAlign modules have different output sizes. In addition, different preprocessing methods are applied according to the shape and quantity characteristics of different classes. We evaluated our algorithm on DOTA dataset. Experiments have shown that our algorithm can achieve performance improvement on a high baseline. And the detection performance of those categories with smaller objects has been improved. In the detection task of horizontal bounding box, we obtained the mAP of 78.96%, which reached the state of the art. Our algorithm is simple and has good performance, easy to be migrated to various network structures.

## 1. Introduction

With the advancement of remote sensing technology, remote sensing images have become more accessible, and the resolution of remote sensing images has been increasing, which provides conditions for the analysis of remote sensing images. With the continuous development of computer hardware and network technology, as well as the rapid increase of data available for training, deep learning technology develops rapidly, and new algorithms improve the accuracy. The location and size of targets such as aircraft, bridges and oil tanks in remote sensing images play an important role in mapping, information analysis and urban planning. Therefore, the study of remote sensing image object detection algorithms is of great significance.

Compared with traditional object detection algorithms, the object detection algorithms based on deep neural network has better effect. With the development of detection algorithms, they continuously improve the detection effect on some common datasets, such as Pascal VOC[1], COCO[2], etc. Object detection algorithms usually use classification networks such as VGGNet[3], GoogLeNet[4]-[7], ResNet[8] and MobileNet[9] to extract features from input images. These classification networks have good effects on ImageNet[10] and other classification datasets.

Object detection algorithms based on deep learning can be roughly divided into single-stage and two-stage algorithms. The single-stage algorithms extract features from images and obtains classification

and bounding boxes from features at a time, which has higher efficiency but relatively lower performance, such as You Only Look Once (YOLO)[11]-[13] whose later versions have improved performance. In order to improve the detection performance of multi-scale targets, SSD[14] uses multiple boundary boxes to process feature maps of different scales. Seung-Wook Kim *et al*.[15] propose a parallel feature pyramid network, combining feature maps of different scales based on SPP-Net [16]. RetinaNet[17] proposed Focal loss and assigned different weights to samples with different difficulty. Anchor is a common idea to select the target area from the feature map. By setting a variety of anchors with different proportions, targets of different shapes can be detected. However, there are a large number of anchors, which overlap with each other. So we need to use method such as NMS(non maximum suppression) to remove duplicates, thus reducing efficiency. Therefore, the algorithm based on the anchor free thought abandoned the anchor. For example, CornerNet[18] regarded the points in the upper left corner and the lower right corner of the boundary box as key points. CenterNet[19] takes the center point as the key point for detection and achieves a good trade-off between performance and efficiency.

In general, the two-stage algorithm first obtains RoI by RPN network in the first stage, and then fine-tuning RoI in the second stage to obtain the final detection results, such as Faster R-CNN[20],[21]. Two-stage algorithm has better performance, but its speed is relatively slower. FPN[22] integrates feature maps of different scales and has good effects on targets of various scales, so it can be applied to various network structures. Mask R-CNN[23] uses RoIAlign to optimize RoI pooling and carry out segmentation tasks at the same time. Deformable ConvNets[24] can learn features better by adding bias and weights into convolution and RoI Pooling. TridentNet[25] constructs parallel branches of shared parameters with different receptive field to solve the problem of scale diversity. Cascade R-CNN[26] uses cascade detection heads to gradually improve the accuracy of detection results by continuously increasing the threshold of IoU.

There are some differences between remote sensing images and conventional images. 1) Remote sensing images may come from different geographic regions, so the background may be very complex, while the background of general image is relatively simple. 2) The camera for remote sensing images have a large distance from the targets, and most of the general images are shot at a relatively close distance, so the remote sensing image is relatively low in resolution. 3) In general images, the number of targets in each image is relatively balanced. For remote sensing images, there may be a large number of targets in some images, while there are few targets in some images. 4) The acquisition of remote sensing images needs relatively high conditions, while general images are easy to be acquired. This leads to a relatively smaller number of images in the dataset of remote sensing images for deep learning algorithms based on large amounts of data. 5) There may be quantitative or scale differences among different categories of remote sensing images. Therefore, the object detection of remote sensing image is a great challenge. DOTA dataset[39] is one of popular remote sensing datasets. And we can use it to evaluate our algorithm.

In this paper, we focus on the problem of multi-scale objects in remote sensing images and propose a novel detector architecture for detection of horizontal bounding boxes. Our contributions are as follows:

1) we proposed the Parallel Cascade R-CNN, which uses multiple cascade detector head branches in parallel by using different RoIAlign. Parallel Cascade R-CNN can improve the detection performance of small objects. This structure is simple but efficient and easy to be imported to various network structures.

2) we implemented our horizontal bounding boxes detection algorithm on the DOTA dataset, and obtained a mAP of 78.96 on the test set, reaching the state of the art.

## 2. Related work

### *2.1. Generic object detection*

*2.1.1 RoIAlign.* For two-stage detection networks, the output of the first stage is usually RoI, with different sizes and shapes. If they are directly input into the second-stage network for detection, it is not conducive to the parallel processing of GPU. Therefore, Fast R-CNN[40] uses RoI pooling to convert feature maps of different sizes into feature maps of fixed sizes.

Mask R-CNN is an instance segmentation algorithm whose detection branch uses RoIAlign to improve RoI pooling. RoI pooling carries out quantization operation when transforming feature maps, which makes coordinate calculation not accurate enough. RoIAlign does not round decimals when calculating coordinates. After dividing the feature maps into small bins, it samples in each small area by bilinear interpolation, and the final result is obtained by pooling. RoIAlign is widely used in later object detection algorithms.

*2.1.2. Cascade R-CNN.* In the object detection algorithm, positive and negative samples can be distinguished according to the IoU threshold. Previous detection algorithms used a threshold of 0.5. Some common datasets such as COCO are evaluated under multiple IoU thresholds. When the evaluation threshold matches the detector threshold, the detection effect is better. If the IoU threshold of 0.5 is used directly in the detector, for evaluating with higher IoU thresholds, they may not be ideal. And the direct use of high threshold for training will lead to too few positive samples, resulting in overfitting. Therefore, Cascade R-CNN cascades several detectors with different IoU thresholds to gradually improve the accuracy of detection results and realize the improvement of overall detection performance. In terms of implementation, FPN and RoIAlign are used on the basis of Faster R-CNN.

### *2.2. Object Detection in Remote Sensing Imagery*

The object detection algorithm based on deep neural network for remote sensing image is widely studied[27]-[31]. Some studies focus on horizontal bounding boxes(HBB). For example, IoU-Adaptive Deformable R-CNN[32] combines Cascade R-CNN and Deformable convolution, and determines the amount of dilated convolution and the intersection over union (IoU) threshold of the detector according to the IoU of anchors. Fu *et al*.[33] proposed a new top-down module to combine multi-scale features and the Dense Inception module to extract features. Qiu *et al*.[34] proposed a novel aspect ratio attention network based on RoI. Dong *et al*.[35] proposed Sig-NMS to improve the precision of detecting small targets. Yang *et al*.[36] propose a lightweight network called SlimNet for feature extraction. Some studies also detect oriented bounding boxes. Li *et al*.[37] propose a feature-attentioned framework to improve its performance. Zhang *et al*.[38] propose a context-aware detection network, learning global and local contexts of objects.

## 3. Proposed method

### *3.1. Parallel Cascade R-CNN*

As shown in figure 1, we designed a Parallel Cascade R-CNN architecture. This architecture is an extension of Cascade R-CNN. After the detection network in the first stage, RoI of different sizes can be obtained. We then input these RoIs into each parallel branch. These parallel branches use different RoIAlign modules and they have different output sizes. For example, one possible setting is that the RoIAlign module of the first branch outputs 11×11 feature maps and the RoIAlign module of the second branch outputs 7×7 feature maps. Each parallel branch is a cascade of detectors. The branches detect independently and are finally merged together. When testing, the same ensemble strategy as Cascade R-CNN was used. We copy the first convolutional layers of each branch to the subsequent convolutional layers and combine the output of each detection head as the final result. For parallel branches, the loss function is defined as:

$$L = \frac{1}{N_{branch}} \sum_{I} L_i \tag{1}$$

Here, $N_{branch}$ is the number of parallel branches. $L_i$ is the loss function of the i-th branch, which is

the same as that of Cascade R-CNN.

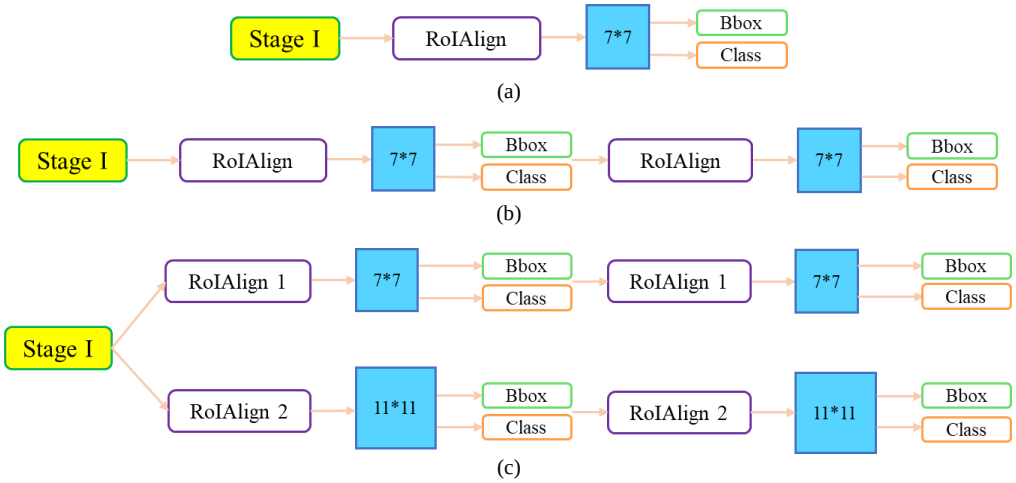



Figure 1. (a) Faster R-CNN with RoIAlign (b) Cascade R-CNN with two cascade heads. (c) Parallel Cascade R-CNN. The parallel branches use different RoIAlign modules and they have different output sizes.

To improve results of the detection, it is common practice to train multiple models and fuse the results they produce. Although this can improve the effect to a certain extent, it needs several times of training and testing time, and needs to design multiple networks and adjust multiple sets of hyper-parameters. Our model has a better balance between efficiency and accuracy. It is similar to the bagging algorithm in thought and plays a similar role of model ensemble. At the same time, it will hardly introduce new hyper-parameters, and the model complexity and training time are within the acceptable range. Our approach is efficient and simple and easy to be migrated to other network structures.

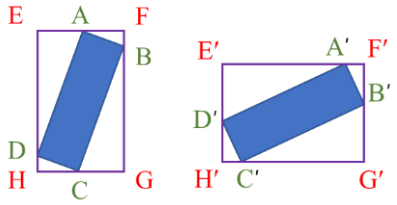
In the second stage, different parallel branches uses the same size of convolution kernels, so the receptive field size is the same. After generating feature maps of different sizes with RoIAlign, the ratio of the receptive field size to the feature map size is different, so each branch can process feature of different scales, which can improve the detection performance of small objects in remote sensing images.

For some classic network structures, such as Faster R-CNN and Cascade R-CNN, one anchor can only detect one category at most. For some targets in remote sensing images, such as football field and playground, their bounding boxes may be closer. Thus, as shown in figure 2, there may be more than one category corresponding to one anchor. Therefore, some objects may be missed. By using parallel detection heads, it is possible to detect different categories respectively, which can reduce the occurrence of missing detection to some extent.

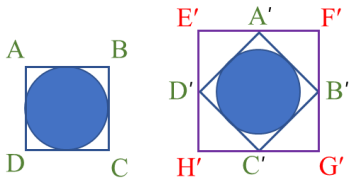

Figure 2. The soccer-ball-field and ground-track-field are corresponding to the same anchor.

### *3.2. Class-Specific Augmentation*

The label of the dataset is quadrilateral, and our aim is to localize the axis-aligned bounding boxes over original oriented bounding boxes. Directly converting a quadrilateral into a horizontal rectangle box will lose information. Therefore, we also built another part of the training set, and added the rotated images on the basis of the original training set. The original image as well as oriented bounding boxes was rotated 45 degrees and cut into sub images, then they are added to the original training set. At the same time, we take into account that the targets such as the oil tanks and the roundabouts are circular. As shown in figure 3, if the outer rectangular box is directly taken after rotation, the rectangular box will be larger than the ground truth and the area will be twice of the ground truth. Therefore, we calculate the rotated rectangle by calculating the center and radius of the circle.




(a)




(b)

Figure 3. (a) After rotating the oriented bounding box of the rectangle ABCD and taking outer rectangle box, we can get E′F′G′H′. The result is correct. (b)After rotating the bounding box of the circle for 45 degrees, it is the square A′B′C′D′. If it is taken outer rectangle box as E′F′G′H′ directly, the result will be wrong.

In remote sensing images, the number of targets of some categories may be relatively small, while the number of targets of some categories may be large, which leads to the problem of unbalanced samples. To alleviate this problem, some categories are oversampled before input to the network. If the total number of samples in the image is less than the threshold, it is copied so that the probability of it being sampled increases.

## 4. Experiments

### *4.1. Dataset*

We conducted experiments on DOTA dataset. It contains 15 types of targets. The training set contains 1411 remote sensing images. The validation set contains 458 remote sensing images. And the test set contains 937 remote sensing images. The training and validation sets provide annotations. Each kind of annotation is in the form of quadrilateral box, which contains the horizontal and vertical coordinates of the four vertices of the quadrilateral.

The image size of DOTA dataset is between 800×800-4000×4000, which is larger than that of other remote sensing image datasets. At the same time, there are more target instances in DOTA data set, and the background is more complex. DOTA dataset also has some characteristics such as denser targets, unbalanced number of targets of different categories, and big difference in scale of objects. Therefore, object detection of DOTA datasets is a more challenging task. Using DOTA datasets is a good way to measure network performance.

Table 1. Ablation study on the Number of detection heads.

| Method | 1×1 [a] head | 2×1 heads | 1×2 heads | 2×2 heads |
|---|---|---|---|---|
| plane | **90.19** | 90.06 | 90.00 | 90.03 |
| baseball-diamond | 85.48 | **85.99** | 84.87 | 84.71 |
| bridge | 61.51 | 62.33 | 62.36 | **62.58** |
| ground-track-field | 72.73 | 70.87 | 74.22 | **75.01** |
| small-vehicle | 76.72 | 77.02 | 77.88 | **78.29** |
| large-vehicle | 83.41 | 83.94 | 84.45 | **84.95** |
| ship | 86.21 | 86.23 | **87.35** | 86.31 |
| tennis-court | 90.39 | **90.57** | 89.84 | 90.33 |
| basketball-court | 85.96 | **86.17** | 85.45 | 85.49 |
| storage-tank | 82.01 | 81.73 | 82.86 | **83.48** |
| soccer-ball-field | 62.02 | 62.65 | 62.55 | **64.85** |
| roundabout | 72.56 | **73.09** | 68.14 | 71.37 |
| harbor | 78.34 | 79.30 | **81.06** | 80.99 |
| swimming-pool | 80.84 | 79.18 | **80.85** | 80.80 |
| helicopter | 61.29 | 65.48 | **67.33** | 65.14 |
| mAP | 77.98 | 78.31 | 78.61 | **78.96** |

[a] m×n heads mean there are m parallel branches and each branch has n cascade detection heads.

Table 2. Computational efficiency of different models.

| Method | 1×1 head | 2×1 heads | 1×2 heads | 2×2 heads |
|---|---|---|---|---|
| Memory(MB) | 6971 | 7945 | 7564 | 10060 |
| Training Speed(iter/s) | 1.37 | 1.20 | 1.30 | 1.04 |

### *4.2. Evaluation Metrics*

Our research task is to detect objects in remote sensing images and mark them with a rectangular box whose edge is consistent with the direction of the coordinate axis. At the same time, the relative

probability of each detected object needs to be calculated. After that, we evaluated each category separately. The evaluation method is to sort all the detection results according to the order of probability from large to small. Then we evaluate them one by one. If the IoU between the detected rectangular box and the ground truth is larger than the threshold 0.5, it is considered true. Otherwise, it is false. If an object is detected for multiple times, only the first detection result is considered true, and the subsequent detection result is considered false. The definition of IoU is

$$\frac{area_{\mathrm{bbox}\cap\mathrm{gt}}}{area_{\mathrm{bbox}\cup\mathrm{gt}}} \tag{2}$$

where bbox ∩ gt is the intersection of the detected rectangular box and the ground truth， bbox ∪ gt is the union of the detected rectangular box and the ground truth.

We can calculate the precision rate and recall rate of each category, and draw the precision-recall curve where

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

The area of the curve and the first quadrant of the coordinate axis is AP. mAP can be obtained by averaging the AP of each category. We use the mAP on the test set as the final evaluation indicator.

The author of DOTA dataset provides us with an online evaluation server. We can submit the test results of the test set online and obtain the mAP of the test set after calculation by the server.

Table 3. Ablation study on the shape of feature map.

| Method | 1×1 head | 2×1 heads | 4×1 heads[a] |
|---|---|---|---|
| plane | **90.19** | 90.06 | 90.17 |
| baseball-diamond | 85.48 | 85.99 | **86.12** |
| bridge | 61.51 | 62.33 | **63.71** |
| ground-track-field | **72.73** | 70.87 | 69.10 |
| small-vehicle | 76.72 | **77.02** | 76.19 |
| large-vehicle | 83.41 | 83.94 | **84.40** |
| ship | 86.21 | **86.23** | 86.14 |
| tennis-court | 90.39 | 90.57 | **90.68** |
| basketball-court | 85.96 | **86.17** | 85.28 |
| storage-tank | 82.01 | 81.73 | **83.12** |
| soccer-ball-field | 62.02 | 62.65 | **63.68** |
| roundabout | 72.56 | **73.09** | 68.40 |
| harbor | 78.34 | 79.30 | **79.99** |
| swimming-pool | 80.84 | 79.18 | **80.97** |
| helicopter | 61.29 | 65.48 | **66.78** |
| mAP | 77.98 | 78.31 | **78.32** |

[a]4×1 heads have 4 parallel branches. And the shapes of the outputs of RoIAlign are 7×7, 11×11, 5×10 and 10×5.

### *4.3. Implementation Details*

As the image is too large to be processed directly in the GPU, it needs to be cut into small images for detection, and then the detection results are merged. During cutting, the object may be cut into two parts, which will affect the detection. Therefore, during the cutting, we make half of the adjacent images overlap, so as to ensure that an object exists in at least one sub-image completely. This also played a role in the data augmentation. For an object that is cut into multiple parts, we only consider it a positive

sample when the remaining part exceeds 70%.The cut side length of the image is 1024.

We did data augmentation during training. The image is flipped horizontally and vertically. The augmentation methods for testing include horizontal and vertical flip and multi-scale test. The test ensemble method was also applied. We copy the first convolutional layers of each parallel branch to the subsequent convolutional layers and merge the results of each detection head.

The test results are deduplicated using NMS. When the objects are dense, the bounding box will overlap. The number and intensity of different categories varies, so different NMS thresholds are used for different categories, which can get by experiments on the validation set. After NMS, all the results including those with low confidence score are retained.

Our algorithm is implemented with the Detectron framework based on Caffe2. The backbone, ResNeXt-101, was pretrained on ImageNet. Our experiment uses 3 GTX 1080Ti with 11G video memory, and batch size of each GPU is 1.With the SGD optimizer with momentum, the learning rate is set to 0.00375 and momentum is set to 0.9. The learning rate decays to one tenth after 50,000 iterations. The total number of iterations is about 80,000. The final training uses the training set and the validation set.

Table 4. Comparison with the State-of-the-Art.

| Method | ICN[41] | SCRDet[42] | $A^2$RMNet | Li *et al*. | Cascade R-CNN | Ours | Ours |
|---|---|---|---|---|---|---|---|
| backbone | ResNet-101 | ResNet-101 | ResNet-101 | ResNet-101 | ResNeXt-101 | ResNet-101 | ResNeXt-101 |
| plane | 89.97 | **90.18** | 89.84 | 90.15 | 90.00 | 89.71 | 90.03 |
| baseball-diamond | 77.71 | 81.88 | 83.39 | 78.60 | 84.87 | **85.93** | 84.71 |
| bridge | 53.38 | 55.30 | 60.06 | 51.92 | 62.36 | 61.39 | **62.58** |
| ground-track-field | 73.26 | 73.29 | 73.46 | 75.23 | 74.22 | **76.51** | 75.01 |
| small-vehicle | 73.46 | 72.09 | **79.25** | 73.60 | 77.88 | 77.99 | 78.29 |
| large-vehicle | 65.02 | 77.65 | 83.07 | 71.27 | 84.45 | 84.22 | **84.95** |
| ship | 78.22 | 78.06 | **87.88** | 81.41 | 87.35 | 87.53 | 86.31 |
| tennis-court | 90.79 | **90.91** | 90.90 | 90.85 | 89.84 | 90.30 | 90.33 |
| basketball-court | 79.05 | 82.44 | **87.02** | 83.94 | 85.45 | 85.83 | 85.49 |
| storage-tank | 84.81 | 86.39 | **87.35** | 84.77 | 82.86 | 83.05 | 83.48 |
| soccer-ball-field | 57.20 | 64.53 | 60.74 | 58.91 | 62.55 | 64.21 | **64.85** |
| roundabout | 62.11 | 63.45 | 69.05 | 65.65 | 68.14 | 71.01 | **71.37** |
| harbor | 73.45 | 75.77 | 79.88 | 76.92 | **81.06** | 80.39 | 80.99 |
| swimming-pool | 70.22 | 78.21 | 79.74 | 79.36 | **80.85** | 79.56 | 80.80 |
| helicopter | 58.08 | 60.11 | 65.17 | **68.17** | 67.33 | 61.21 | 65.14 |
| mAP | 72.45 | 75.35 | 78.45 | 75.38 | 78.61 | 78.59 | **78.96** |

### *4.4. Ablation Study*

In order to explore the role of the model and the impact of various factors on its performance, we did some ablation study. The results were evaluated on the test set.

*4.4.1. the Number of Detection Heads.* As can be seen from table 1, both cascade detection heads and parallel branches can bring about performance improvements. When the detection head is 1×1, the network is optimized Faster R-CNN, and the mAP increased by 0.33 after adding parallel branches. When the detection head is 1×2, the network is Cascade R-CNN, and when parallel branches are added, the mAP increases by 0.35. It can be seen that our parallel architecture can still bring improvement on

a high baseline. table 2 shows their computational efficiency. The increase of memory and the decrease of training speed are acceptable.

*4.4.2. the Shape of Detection Heads.* In table 3, 4×1 heads uses feature maps of different shapes at the same time without causing significant performance changes. The shape output by RoIAlign does not directly represent the shape information of the objects as the anchors do.

*4.5. Comparison with the State-of-the-Art*

We compared the HBB results of test set with other work and baseline. The results are listed in table 4. For each category, the best result is highlighted in bold. The three categories with the smallest size in DOTA dataset are storage-tank, small-vehicle and bridge. It can be seen from table 4 that after adding parallel branches on the basis of Cascade R-CNN, the detection performance of these categories has been improved, indicating that the algorithm proposed can effectively solve the problem of small objects .As can be seen, our method reaches the state of the art. We had the best results in some categories such as bridge, soccer-ball-field and roundabout. The visualization of some detection results is shown in figure 4.

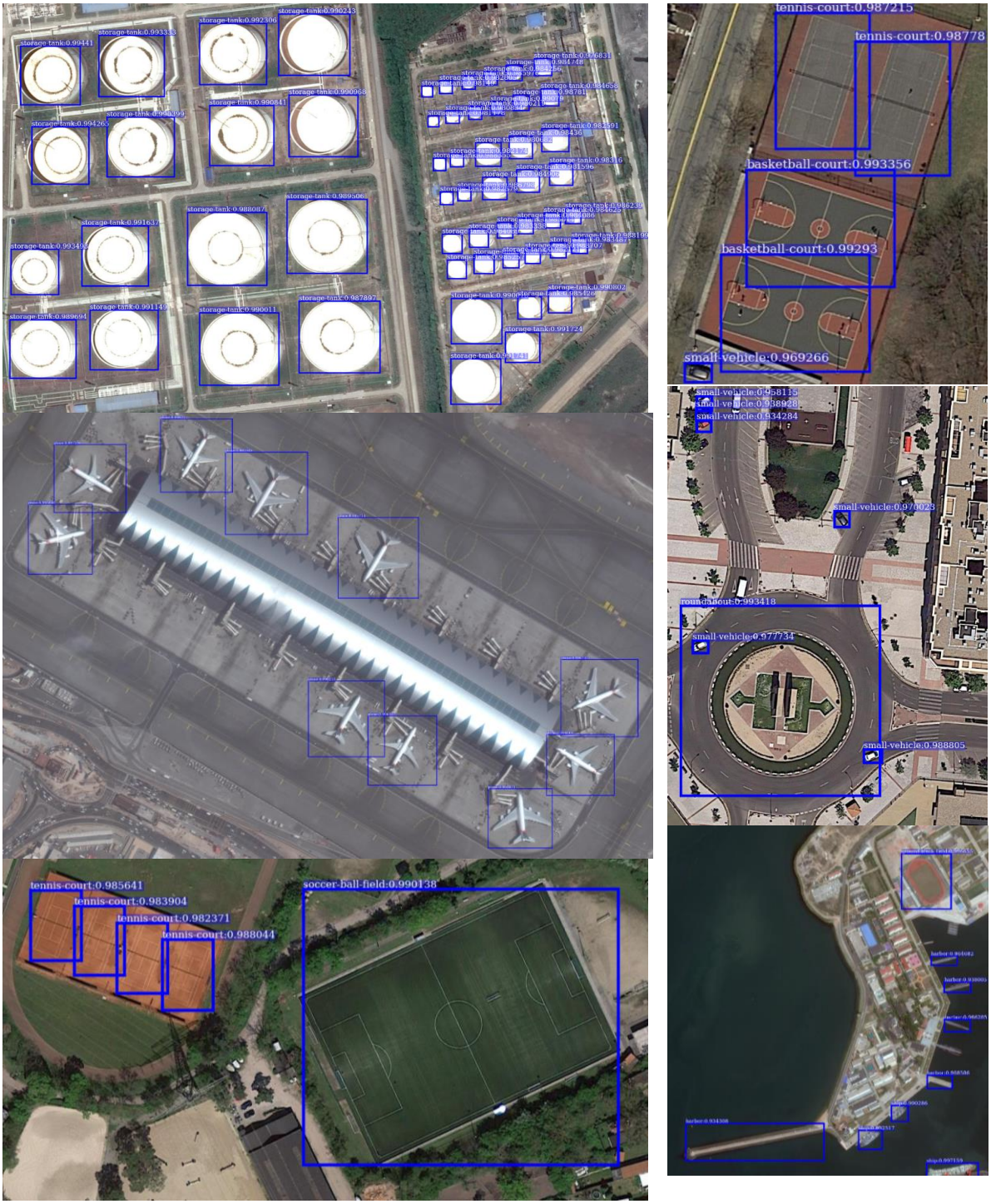



Figure 4. The visualization of detection results.

## 5. Conclusion

In this paper, we use parallel branches to improve two-stage detector such as Cascade R-CNN in remote sensing images. This structure is easy to migrate to other network structures. Experiments show that the parallel structure can effectively improve the detection performance of the network, especially for small objects. We conducted experiments on DOTA dataset and our model reached the state of the art. In the

following study, we will optimize the network structure in the first stage, such as FPN, so as to better solve problem of various scale.

**Acknowledgments**
This work was supported by the National Natural Science Foundation of China under Grant Nos. 61701277、61771288 and the state key development program in 13th Five-Year under Grant No. 2017YFC0821601.

**References**

[1] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision. 88: 303-338.
[2] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In: European conference on computer vision. Zurich. pp. 740–755.
[3] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In: European conference on computer vision. Zurich. pp. 818–833.
[4] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In: the IEEE conference on computer vision and pattern recognition. Boston. pp. 1–9.
[5] Ioffe, S., & Szegedy, C. (2015, June). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: International Conference on Machine Learning. Lille. pp. 448–456.
[6] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In: the IEEE conference on computer vision and pattern recognition. Las Vegas. pp. 2818–2826.
[7] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence. San Francisco, pp. 12.
[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: the IEEE conference on computer vision and pattern recognition. Las Vegas. pp. 770–778.
[9] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. https://arxiv.org/abs/1704.04861
[10] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In: the IEEE conference on computer vision and pattern recognition. pp. Miami. 248–255.
[11] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: the IEEE conference on computer vision and pattern recognition. Las Vegas. pp. 779–788.
[12] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In: the IEEE conference on computer vision and pattern recognition. Hawaii. pp. 6517–6525.
[13] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. https://arxiv.org/abs/1804.02767
[14] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In: European conference on computer vision. Amsterdam. pp. 21–37.
[15] Kim, S. W., Kook, H. K., Sun, J. Y., Kang, M. C., & Ko, S. J. (2018). Parallel feature pyramid network for object detection. In: the European Conference on Computer Vision. Munich. pp. 234–250.

[16] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence. 37: 1904–1916.
[17] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2018). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 42: 318–327
[18] Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In: the European Conference on Computer Vision. Munich. pp. 734–750.
[19] Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. https://arxiv.org/abs/1904.07850
[20] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In: the IEEE conference on computer vision and pattern recognition. Columbus. pp. 580–587.
[21] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence. 39: 1137–1149.
[22] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In: the IEEE conference on computer vision and pattern recognition. Hawaii. pp. 936–944.
[23] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In: the IEEE international conference on computer vision. pp. 2980–2988.
[24] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In: the IEEE international conference on computer vision. Venice. pp. 764–773.
[25] Li, Y., Chen, Y., Wang, N., & Zhang, Z. (2019). Scale-aware trident networks for object detection. In: the IEEE International Conference on Computer Vision. Seoul. pp. 6054-6063.
[26] Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In: the IEEE conference on computer vision and pattern recognition. Salt Lake City. pp. 6154–6162.
[27] Cui, Z., Li, Q., Cao, Z., & Liu, N. (2019). Dense Attention Pyramid Networks for Multi-Scale Ship Detection in SAR Images. IEEE Transactions on Geoscience and Remote Sensing. 57: 8983–8997.
[28] An, Q., Pan, Z., Liu, L., & You, H. (2019). DRBox-v2: An Improved Detector With Rotatable Boxes for Target Detection in SAR Images. IEEE Transactions on Geoscience and Remote Sensing. 57: 8333–8349.
[29] Pang, J., Li, C., Shi, J., Xu, Z., & Feng, H. (2019). $\mathcal{R}^2$-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing. 57: 5512–5524.
[30] Deng, Z., Sun, H., Zhou, S., & Zhao, J. (2019). Learning deep ship detector in SAR images from scratch. IEEE Transactions on Geoscience and Remote Sensing. 57: 4021–4039.
[31] Li, Q., Mou, L., Xu, Q., Zhang, Y., & Zhu, X. X. (2019). R 3-Net: A Deep Network for Multioriented Vehicle Detection in Aerial Images and Videos. IEEE Transactions on Geoscience and Remote Sensing. 57: 5028–5042.
[32] Yan, J., Wang, H., Yan, M., Diao, W., Sun, X., & Li, H. (2019). IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. Remote Sensing. 11: 286.
[33] Fu, K., Chen, Z., Zhang, Y., & Sun, X. (2019). Enhanced Feature Representation in Detection for Optical Remote Sensing Images. Remote Sensing. 11: 2095.
[34] Qiu, H., Li, H., Wu, Q., Meng, F., Ngan, K. N., & Shi, H. (2019). A2RMNet: Adaptively aspect ratio multi-scale network for object detection in remote sensing images. Remote Sensing. 11: 1594.
[35] Dong, R., Xu, D., Zhao, J., Jiao, L., & An, J. (2019). Sig-NMS-based faster R-CNN combining transfer learning for small target detection in VHR optical remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing. 57: 8534–8545.

[36] Yang, Z., Liu, Y., Liu, L., Tang, X., Xie, J., & Gao, X. (2019). Detecting Small Objects in Urban Settings Using SlimNet Model. IEEE Transactions on Geoscience and Remote Sensing. 57: 8534–8545.
[37] Li, C., Xu, C., Cui, Z., Wang, D., Zhang, T., & Yang, J. (2019). Feature-Attentioned Object Detection in Remote Sensing Imagery. In: IEEE International Conference on Image Processing. Taipei. pp. 3886–3890.
[38] Zhang, G., Lu, S., & Zhang, W. (2019). CAD-Net: A context-aware detection network for objects in remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing. 57: 10015-10024.
[39] Xia, G. S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., ... & Zhang, L. (2018). DOTA: A large-scale dataset for object detection in aerial images. In: the IEEE Conference on Computer Vision and Pattern Recognition. Munich. pp. 3974–3983.
[40] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision. Santiago. pp. 1440–1448.
[41] Azimi, S. M., Vig, E., Bahmanyar, R., Körner, M., & Reinartz, P. (2018). Towards multi-class object detection in unconstrained remote sensing imagery. In: Asian Conference on Computer Vision. Perth. pp. 150-165
[42] Yang, X., Yang, J., Yan, J., Zhang, Y., Zhang, T., Guo, Z., ... & Fu, K. (2019). Scrdet: Towards more robust detection for small, cluttered and rotated objects. In: the IEEE International Conference on Computer Vision. Seoul. pp. 8232–8241.