ROTATED REGION BASED CNN FOR SHIP DETECTION

Zikun Liu[⋆†] *Jingao Hu*[⋆†] *Lubin Weng*[⋆] *Yiping Yang*[⋆]

[⋆] Institute of Automation, Chinese Academy of Sciences
[†]University of Chinese Academy of Sciences

ABSTRACT

The state-of-the-art object detection networks for natural images have recently demonstrated impressive performances. However the complexity of ship detection in high resolution satellite images exposes the limited capacity of these networks for strip-like rotated assembled object detection which are common in remote sensing images. In this paper, we embrace this observation and introduce the rotated region based CNN (RR-CNN), which can learn and accurately extract features of rotated regions and locate rotated objects precisely. RR-CNN has three important new components including a rotated region of interest (RRoI) pooling layer, a rotated bounding box regression model and a multi-task method for non-maximal suppression (NMS) between different classes. Experimental results on the public ship dataset HRSC2016 confirm that RR-CNN outperforms baselines by a large margin.

***Index Terms*—** Rotated region, convolutional neural network, ship detection.

1. INTRODUCTION

Recently, advances in object detection are driven by progresses of backbone deep convolutional neural networks [1, 2, 3, 4] and improvements of object detection frameworks including R-CNN [5], Fast R-CNN [6], Faster R-CNN [7] and SSD [8], etc. Object detection is a more complicated task than image classification, due to the demand of precise localization. To solve this problem, these detection frameworks combined with various backbones are based on different feature extraction strategies of regions.

Feature extraction strategy of regions is an important difference between these detection frameworks. R-CNN [5] is based on special object proposal method which typically relies on simple feature and fast evaluation. See, for example, Selective Search (SS) method [9]. Then, convolution operation is carried out on each proposal region generated directly from images without sharing computation, which leads to inefficiency. Fast R-CNN [6] is also based on special object proposal method, but performs convolution operation on the entire image and extract convolution features for each proposal in each feature map by a region of interest (RoI) pooling layer. Compared with R-CNN, Fast R-CNN saves much computation time, exposing proposal generation time as a bottleneck. Faster R-CNN [7] introduces a region proposal network (RPN) to instead of special region proposal method. The RPN shares convolution computation with the detection network and improves the quality of proposals. SSD [8] also performs convolution operation on the entire input image, but produces a fixed-size collection of bounding boxes based on a small set of default boxes at each location in several feature maps. The role of default boxes without computation in SSD is the same as RPN in Faster R-CNN. Object detection based on natural images in which objects are always stand-alone has been greatly advancing by these detection networks. However, they are not strong enough to finish the task of strip-like rotated assembled object detection, which is common in remote sensing images, especially ship detection as shown in figure 1. The existing works [10, 11, 12, 13] in ship detection cannot accurately locate ships in a cluster shown in figure 1 or are hand-crafted work without robustness.

(a) (b) (c)

Fig. 1: The challenge of rotated object detection: (a) ground truths, (b) detection results of Fast R-CNN based on SRBBS method [14], (c) Our results.

Table 1: Properties of networks: feature extraction accurately for rotated regions, sharing convolution computation and end-to-end training.

Networks	Rotated region	Sharing computation	End-to-end
R-CNN			
Fast R-CNN		√	
Faster R-CNN		√	√
SSD		√	√
Desired models	√	√	√

The desired model for ship detection should share convolution computation, be trained end-to-end, and most of all, should extract features accurately for rotated regions. In table 1, we list the properties of current state-of-the-art CNN detection frameworks introduced previously. There are two challenges, which are also our motivations, to detect ships in complex backgrounds using these networks: the demand of rotated region feature extraction without noise information and the rotated bounding box (RBB) regression to accurately locate ships. In figure 1(b), we can see that it is difficult for Fast R-CNN combined with SRBBS proposal method [14] to distinguish the two ships moored together.

In this paper, we introduce a rotated region based CNN for ship detection, which can accurately extract features for rotated bounding box and share convolution computation between regions. We list our contributions as follows:

1. We introduce RR-CNN which is significantly more accurate than state-of-the-art detectors and baselines.

2. We propose the RRoI pooling layer, an auxiliary structure, to extract features of rotated regions.

3. The auxiliary structures proposed in this paper can be plugged into all the detection frameworks in table 1.

2. THE ROTATED REGION BASED CNN

2.1. Model

RR-CNN approach includes auxiliary structures and a backbone network which could be any classical model, such as Alexnet [1] and VGG-16 [2], etc. The auxiliary structures includes three key components: the RRoI pooling layer (section 2.1.1), the RBB regression model (section 2.1.2) and the multi-task for NMS (section 2.1.3).

2.1.1. The RRoI pooling layer

It is important to accurately extract features of ship regions. However, both Fast R-CNN and Faster R-CNN extract region features by RoI pooling layers based on bounding boxes which always cover too much noise information in ship detection task. R-CNN extracts region data from images before convolution layers, which equates to putting a special RoI pooling layer between the data layer and the first convolution layer. SSD uses default boxes to instead of RoI pooling. When there are ships moored together in a bounding box region, it is difficult for classifiers to distinguish the ships. The pixel-level free-form RoI pooling used in [15] can accurately extract features, but it still pools features from left to right, top to bottom, leading to the lack of robustness. Liu et al. [14] proved the advantages of the bounding way of rotated bounding boxes (RBBs) for ships. In this paper, we propose a rotated region of interest (RRoI) pooling layer based on RBB.

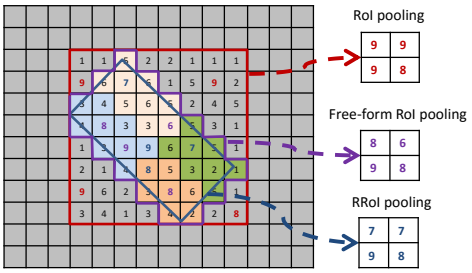


Fig. 2: The differences between RoI pooling, free-form RoI pooling and our RRoI pooling. To facilitate the discussion, we assume that the fixed spatial extent is 2×2.

The RRoI pooling layer also gets a small feature map with a fixed spatial extent of $H \times W$ from the features inside the valid region of interest by max pooling, as in RoI pooling layer [6]. However, our layer is based on RBB. We define each RRoI as a five-tuple (x, y, w, h, a) specifies the RBB's the center position (x,y), the width (the longer side), height and the rotation angle (between $-90°$ and $90°$). When the rotation angle is zero, the special case is equivalent to the RoI pooling layer. The bins of the fixed spatial extent are arranged from left to right, top to bottom with one of the longer sides as the horizontal line, which can be seen in figure 2.

RRoI pooling is applied independently to each rotated region inside each feature map channel, as in RoI pooling [6]. The process of forwarding data through RRoI pooling layer is shown in figure 2 and modeled as

$$y_{mrj} = max_{i \in B(m,r,j)}(x_i) \tag{1}$$

where m is the m-th feature map, r is the r-th RBB region, y_{mrj} is the j-th output corresponding to the j-th bin belonged to the fixed spatial extent of the region, $B(m, r, j)$ is a set of pixels belonged to the j-th bin of the r-th region in the m-th feature map.

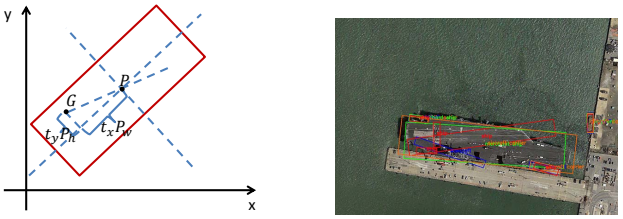
The RRoI pooling layer supports back-propagation which is defined as

$$\frac{\partial L}{\partial x_{mi}} = \sum_r \sum_j [i = argmax_{i' \in B(m,r,j)}(x_{i'})] \frac{\partial L}{\partial y_{mrj}} \tag{2}$$

where x_{mi} is the ith input in the mth feature map, L is the loss function, the partial derivatives of the loss function with respect to each output $\partial L / \partial y_{mrj}$ are computed by the next layer. To accelerate the computation, we can store the indexes of max-pooling during the forward pass.

2.1.2. The RBB regression model

The state-of-the-art detection networks in table 1 predict bounding boxes (BBs) to improve localization performances. The BB regression model was introduced in [5] and improved



(a) (b)

Fig. 3: (a) The demo for the scale-invariant translation (SIT) of the ground truth RBB relative to a proposal; (b) The results of no suppression between classess.

in [6]. They used a scale-invariant translation (SIT) and log-space height/width shift relative to an object proposal as regression targets and a L_1 function as localization task loss.

However, as introduction in [14], RBB can locate ships more accurately than BB. Furthermore, NMS between BBs leads to sparse windows to locate all the ships as shown in the figure 1(b). In this letter, RR-CNN outputs RBBs. We define the RBB regression model as

$$L_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h,a\}} smooth_{L_1}(t_i^u - v_i), \quad (3)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases}, \quad (4)$$

$$t_x = (\cos\alpha(G_x - P_x) + \sin\alpha(G_y - P_y))/P_w, \quad (5)$$

$$t_y = (-\sin\alpha(G_x - P_x) + \cos\alpha(G_y - P_y))/P_h, \quad (6)$$

$$t_w = \log(G_w/P_w), \quad (7)$$

$$t_h = \log(G_h/P_h), \quad (8)$$

$$t_a = (G_a - P_a)/(\lambda 180), \quad (9)$$

where u is the labeling class of a training RRoI, $v = (v_x, v_y, v_w, v_h, v_a)$ is the ground-truth RBB regression target, $t^u = (t_x^u, t_y^u, t_w^u, t_h^u, t_a^u)$ is a predicted tuple for v, $P = (P_x, P_y, P_w, P_h, P_a)$ is the RBB proposal, α equals to P_a, $G = (G_x, G_y, G_w, G_h, G_a)$ is the ground-truth RBB, the meaning of elements inside P and G is the same as the ones of the five-tuple in section 2.1.1 and λ is a constant number ($\lambda = 0.5$). The L_1 function in equation (4) is the same as the one in [6]. In figure 3(a), we show the intuitive understanding of the SIT in equation (5) and (6). In the baseline method BL2 of the publication dataset HRSC2016 (introduced in section 3) also proposed a RBB regression model whose SIT is different from ours. In method BL2 , the SIT for the center of the proposal P is the horizontal/vertical distance between G and P divided by the projection of P's width and height in the horizontal/vertical direction. However, the projection is unstable because of rotation leading to inconsistent normalization. In our method, we normalize the distance's projection in width/height direction directly by P's width/height, which is more stable.

In the test stage, we can transform an input proposal P into a predicted ground-truth RBB $\hat{G} = (\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h, \hat{G}_a)$ by the transformation

$$\hat{G}_x = t_x P_w \cos\alpha - t_y P_h \sin\alpha + P_x \quad (10)$$

$$\hat{G}_y = t_x P_w \sin\alpha + t_y P_h \cos\alpha + P_y \quad (11)$$

$$\hat{G}_w = P_w exp(t_w) \quad (12)$$

$$\hat{G}_h = P_h exp(t_h) \quad (13)$$

$$\hat{G}_a = \lambda 180 t_a + P_a \quad (14)$$

2.1.3. *Multi-task for NMS*

The state-of-the-art methods in table 1 and the method BL2 all include non-maximum suppression (NMS) stage which only keeps suppression within classes. No NMS between classes on ship detection task leads to unacceptable results shown in figure 3(b). In this paper, we add multi-task loss to learn a NMS score for each proposal. We define the loss function as

$$L(p, u, t^u, v, C) = L_{cls}(p, u, C) + \lambda_1[u \geq 1]L_{loc}(t^u, v) + \lambda_2[C > 2]L_{nms}(p, u) \quad (15)$$

where p is a discrete probability distribution computed by a softmax, C is the class number including background class, λ_1 and λ_2 is hyper-parameters, v and t^u are the same as the ones in section 2.1.2, L_{nms}, L_{loc} and L_{nms} are the loss functions for classification task, location task and NMS task respectively. In this paper, the L_{nms} is the same as the one in [6], the L_{loc} is defined in section 2.1.2, and L_{nms} is defined as L_{cls} with $C = 2$.

2.2. Training

In this paper, we plugged our auxiliary structures into Fast R-CNN framework by using our components instead of the corresponding ones. However, our components can be also applied to Faster R-CNN or SSD, but with additional adjustments. We select the SRBBS method [14] as our proposal method, which is the same as the BL2 method. The training samples for L_{nms} are the same as the ones for L_{cls}, but with all the positive samples labeled as "1" and negative ones labeled as "0". The training process has no difference with Fast R-CNN.

3. EXPERIMENTS

We evaluate our methods on the public dataset HRSC2016[1] [14]. The dataset contains 1061 images collected from Google Earth. The training, validation and test set include

[1] http://www.escience.cn/people/liuzikun/DataSet.html

Table 2: The AP values of all the classes on the L3 task.

class	CP	BL2	RC1	RC2	class	CP	BL2	RC1	RC2	class	CP	BL2	RC1	RC2
ship	35.5	**41.9**	34.3	32.9	air.	18.2	41.2	**53.9**	40.3	mer.	20.5	12.8	**27.4**	18.2
war.	0.8	3.9	3.9	**4.8**	Arl.	55.8	**71.0**	65.3	61.0	Aus.	32.6	43.2	**48.7**	42.7
Car.A	35.9	53.1	**61.0**	56.7	Car.B	61.0	**67.8**	**67.8**	57.6	Com.A	43.1	**57.1**	50.5	38.4
Con.	30.3	36.8	**56.2**	40.2	Con.A	50.8	49.6	**62.7**	56.3	Ent.	12.8	3.0	**23.0**	18.2
Med.	30.1	78.8	**98.5**	62.5	Nim.	32.3	44.6	59.3	**65.0**	Per.	37.3	43.9	**47.6**	42.3
San.	36.5	**51.4**	45.5	31.5	Tar.	31.4	50.9	62.4	**67.0**	Tic.	41.1	**57.4**	48.6	49.8
Whi.	30.8	50.9	**51.8**	45.8										

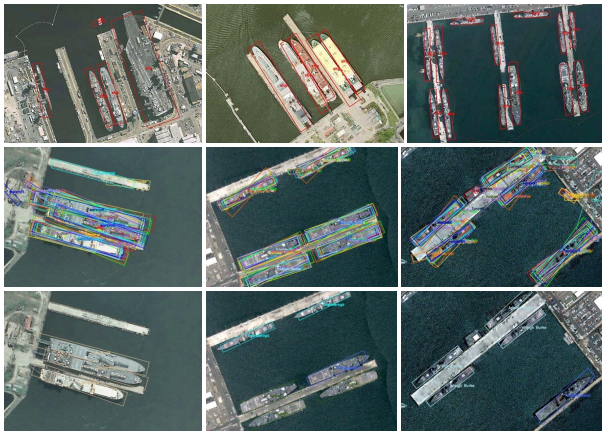


Fig. 4: The samples of detection results. The samples in the first row were generated by RC2 on L1 task; The ones in the second row were got by RC1 on L3 task; The images in the third row were the results of RC2 on L3 task, which correspond to the ones in the second row.

436 images with 1207 samples, 181 images with 541 samples and 444 images with 1228 samples respectively. The ship detection tasks in HRSC2016 include three levels which are L1, L2 and L3 task. We exclude submarine, hovercraft classes and samples with "difficult" label as they did in [14]. And the three tasks contain 1 class, 4 classes and 19 classes respectively.

In this paper, we select BL2 method in HRSC2016 as our baseline which is also evaluated by RBB. BL2 is based on Fast R-CNN combined with SRBBS proposal generation method and a RBB regression model. Furthermore, to compare with Fast R-CNN, we also take Fast R-CNN based on SRBBS as our comparsion method (CP). There are two variants of our method. Compared with BL2, the first variant (RC1) has additional components of the RRoI pooling layer and our new RBB regression model. The second one (RC2) has all the three components introduced in section 2.1. We use criteria average precision (AP) to measure the performance for one class and mean average precision (mAP) [5] for all classes. The parameters used in our experiments are the same as the ones used in BL2.

Table 3: The mAP values on all the three tasks.

task	CP	BL2	RC1	RC2
L1 (mAP)	55.7	69.6	**75.7**	**75.7**
L2 (mAP)	44.3	58.8	**63.6**	61.0
L3 (mAP)	33.5	45.2	**51.0**	43.7

Table 4: The AP values of all the classes on the L2 task.

method	ship	air.	war.	mer.	mean
CP	28.6	36.7	61.8	50.1	44.3
BL2	**45.1**	51.0	**75.2**	63.9	58.8
RC1	36.4	**74.5**	74.4	**69.3**	**63.6**
RC2	31.5	72.6	74.6	65.1	61.0

The experimental results are shown in table 3. The details of AP values on L2 and L3 task are shown in table 4 and table 2 respectively. We can see that variant RC1 outperforms CP and BL2 method on all the three tasks by +18.5 to +20.0 and 4.8 to 6.1 mAP points respectively. Compared with RC1, our method RC2 has lower mAP values on L2 and L3 task due to NMS between classes. However, RC2 still yields higher mAP values on L1 and L2 task than the ones reported for BL2. It is worthwhile to note that the detection samples got by RC2 are more suitable to practical application than the ones with overlapping between each other generated by CP, BL2 and RC1, which are shown in figure 3(b) and figure 4.

4. CONCLUSION

We have presented Rotated Region based CNN (RR-CNN) for efficient and accurate rotated object detection, especially for ship detection in remote sensing images. RR-CNN has three key features, namely RRoI pooling layer, a new RBB regression model and NMS between different classes. Compared with the state-of-the-art object detection framworks, RR-CNN has the advantage of feature extraction accurately for rotated regions. In the future work, we will try to combine RR-CNN with Faster RCNN or SSD framworks, design the speciall backbone network and learn rotation invariant CNN features.

5. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[4] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, "Densely connected convolutional networks," *arXiv preprint arXiv:1608.06993*, 2016.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] Ross Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed, "SSD: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2015.

[9] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[10] Ge Liu, Yasen Zhang, Xinwei Zheng, Xian Sun, Kun Fu, and Hongqi Wang, "A new method on inshore ship detection in high-resolution satellite images using shape and context information," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 3, pp. 617–621, 2014.

[11] Zhengxia Zou and Zhenwei Shi, "Ship detection in spaceborne optical image with SVD networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 5832–5845, 2016.

[12] Ruiqian Zhang, Jian Yao, Kao Zhang, Chen Feng, and Jiadong Zhang, "S-CNN ship detection from high-resolution remote sensing images," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 423–430, 2016.

[13] Shigang Wang, Min Wang, Shuyuan Yang, and Licheng Jiao, "New hierarchical saliency filtering for fast ship detection in high-resolution sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 351–362, 2016.

[14] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 8, pp. 1074–1078, Aug 2016.

[15] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Region-based semantic segmentation with end-to-end training," in *European Conference on Computer Vision*. Springer, 2016, pp. 381–397.