Two-Stage Object Detection Based on Deep Pruning for Remote Sensing Image

Shengsheng Wang[1], Meng Wang[1(✉)], Xin Zhao[1], and Dong Liu[2]

[1] College of Computer Science and Technology,
Jilin University, Changchun 130012, China
1030997649@qq.com
[2] Xiangnan University, Chenzhou, China

Abstract. In this paper, we concentrate on tackling the problems of object detection in very-high-resolution (VHR) remote sensing images. The main challenges of object detection in VHR remote sensing images are: (1) VHR images are usually too large and it will consume too much time when locating objects; (2) high false alarm because background dominate and is complex in VHR images. To address the above challenges, a new method is proposed to build two-stage object detection model. Our proposed method can be divided into two processes: (1) we use twice pruning to get region proposal convolutional neural network which is used to predict region proposals; (2) and we use once pruning to get classification convolutional neural network which is used to analyze the result of the first stage and output the class labels of proposals. The experimental results show that the proposed method has high precision and is significantly faster than the state-of-the-art methods on NWPU VHR-10 remote sensing dataset.

Keywords: Very-high-resolution remote sensing image · Computer vision
Object detection · Convolutional neural network · Deep learning

1 Introduction

The spatial resolution of optical remote sensing sensor has been greatly improved in the past 10 years, and a large number high-resolution images have been applied to resource survey, natural hazard, urban traffic control and other fields [1–7]. The background of remote sensing image is more complex than ground-based image and remote sensing images often contain a lot of noise. Remote sensing images have its own unique characteristics like, providing a vast visual field, covering large area, more visualized contents. The traditional object detection methods based on SIFT [8] and HOG [9] are not ideal for processing such complex images.

In recent years, deep learning began to replace the traditional image processing technology in many computer vision tasks such as object detection, classification. And the two-stage object detection method based on deep learning has been applied to detect objects from high resolution remote sensing image [10–14] recently. "Two-stage" means the region proposal stage [15] and the classification stage. The region proposal stage should extract region proposals rapidly because remote sensing images

W. Liu et al. (Eds.): KSEM 2018, LNAI 11061, pp. 137–147, 2018.
https://doi.org/10.1007/978-3-319-99365-2_12

are too large, and the classification stage should classify region proposals accurately because the background is complex.

Long et al. [10] first used Selective Search (SS) algorithm to generate region proposals, and then multiple targets were detected by CNN (Convolutional Neural Network) and SVM classifier for optical remote sensing images. Jiang et al. [11] proposed a method that first got proposals by a graph-based super pixel segmentation, and then classified proposals with a CNN. Ševo et al. [12] cut high-resolution satellite images into same size image patches directly and classified the image patches by a trained CNN. Wu et al. [13] used Edge Boxes to extract region proposals and input region proposals to CNN for classification. However, methods above have drawbacks: using SS and super-pixel segmentation to extract region proposals is slow; Cutting image into same size patches will loss image information; it is hard for Edge Boxes to handle images with complex background. In conclusion, the existing two-stage object detection methods based on CNN have the problems of high false alarm rate and slow speed when facing high-resolution remote sensing images.

To solve the above problems, we propose a two-stage Object Detection based on Deep Pruning (ODDP) for VRH remote sensing images. Firstly, a deep neural network pruning method Deep Pruning (DP) is proposed, and then the Learning Region Proposal Network algorithm (LRPN) is proposed based on DP. We use LRPN to train a highly sparse CNN to extract region proposals quickly. Finally, the Optimizing Classification Network algorithm (OCN) based on Deep Pruning is proposed, which is used to learn a more accurate classification network than normal training network. Combine the two networks to obtain the object detection model, and Fig. 1 is the test phase of the object detection model.

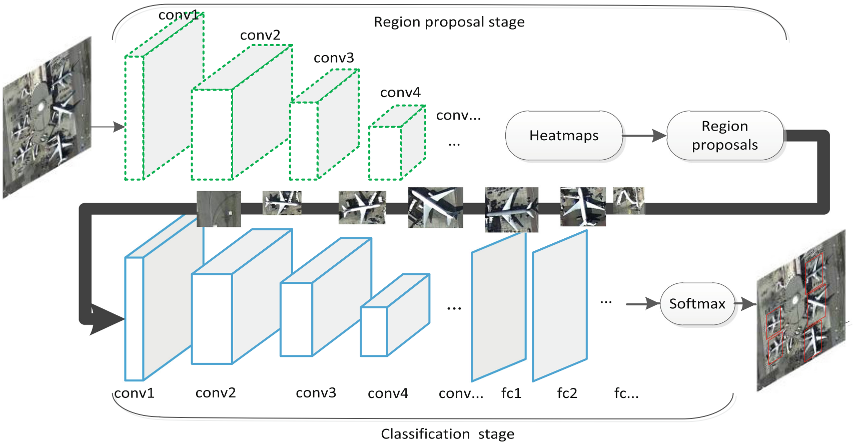

Fig. 1. Test phase of the two-stage model obtained by ODDP.

2 Two-Stage Object Detection Based on Deep Pruning Method

The proposed ODDP includes LRPN and OCN. As can be seen from Fig. 2, LRPN obtains region proposal network through four steps: pretraining, first pruning, reconstruction and secondary pruning. OCN uses the pretraining network to obtain the classification network through pruning and recovering. We construct object detection model by combining region proposal network and classification network.

2.1 Deep Pruning

Deep neural network is typically over-parameterized, and there is a significant redundancy [16], which may lead to overfitting and calculating slowly. Inspired by dropout method of neural network, Deep Pruning (DP) is proposed to reduce the network capacity. Specifically, the proposed method cuts off the connections below the adjustable threshold, which changes the structure of the network, and makes the network sparser. So, DP makes the network less likely to overfitting and accelerates network computing.

There are four steps of DP: (1) Make the network learn which connections are more important. (2) Set threshold, cut off unimportant connections, and then get a sparse network. (3) Train the sparse network to restore the precision. (4) Analyze the average precision and threshold, then choose a way by the result of C. We will describe the workflow of DP below.

Sparsity is the percentage of connections that will be cut off. The processes of choosing threshold are as follows: if there are M parameters in layer W, we sort the M parameters and then choose a parameter as the threshold T:

$$\begin{aligned} &T = sort(M)_{(1-S)\cdot size(M)}, where \\ &S = S + sig(\Delta P)\cdot (S_{\text{fixed}} - \theta \cdot S_{\text{conv}} - (1-\theta)\cdot S_{\text{fully}})^2 \end{aligned} \tag{1}$$

We define C here which will be used in the algorithm below:

$$\mathrm{C} = \begin{cases} 1, \Delta P < 0 \\ 2, \Delta P \geq 0 \&\& \, \theta \cdot S_{\text{conv}} + (1-\theta)\cdot S_{\text{fully}} < S_{\text{fixed}} \,\&\& T > 0 \\ 3, \Delta P \geq 0 \&\& (\theta \cdot S_{\text{conv}} + (1-\theta)\cdot S_{\text{fully}} \geq S_{\text{fixed}} \,||\, T = 0) \end{cases} \tag{2}$$

Where S is the sparsity which will change during two pruning processes, we set the initial value of S to 0.1, sort() is the sort function which is used to sort M parameters, size() is used to compute the number of parameters, sig() is a signal function which will return −1 or +1, S_{fixed} is the preset sparsity and the value of S_{fixed} should be different in different network structure (in our experiment we set $S_{\text{fixed}} = 0.8$), S_{conv} is the convolution layer sparsity of the network, S_{fully} is the fully connected layer sparsity of the network and we can compute S_{conv} and S_{fully} after each pruning, T is the threshold which is used to prune connections, ΔP is the average precision difference of the network before and after pruning, θ balances the contribution of S_{fixed} and S_{conv} to the sparse degree of the entire neural network and we select θ according to network structure (in our experiment we set $\theta = 0.2$).

```
Algorithm 1: workflow of Deep Pruning
Input: a network; training dataset
Output: a network
Step 1: Train the network by Stochastic Gradient Descent(SGD): SGD(W)
Step 2: Prune
    2.1: Sort the weights and find the threshold
    2.2: Cut off the connections with the weight less than threshold
Step 3: Train the sparse network: SGD(W)
Step 4: Calculate C in formula (2),
    if C == 1 : recover the cut off connections and move to step 1
    if C == 2 : move to step 2
    if C == 3 : end
```

2.2 Learning Region Proposal Network

LRPN makes the initial network to autonomously learn a convolutional neural network with new structure from the training dataset. The new structure network has the following advantages: (1) the network structure fits to the distribution of dataset; (2) avoid the blindness of artificially designed network structure; (3) combine the advantage of the classic initial network and the new sparse network. Due to the above advantages, our method speeds up network computing, while the accuracy does not decrease. The LRPN algorithm and its workflow will be described below.

```
Algorithm 2: workflow of LRPN
Input: initial network, pretraining dataset, training dataset
Output: pretrained network, RPN
Step 1: Pretraining
    1.1 Initialize the weight of initial network: $W \sim N(0, 0.01^2)$
    1.2 Use Stochastic Gradient Descent(SGD) and the pretraining dataset to train the initial network: SGD(W)
    1.3 Output the pretrained network
Step 2: First pruning
    Deep Pruning takes as input the training dataset and the pretrained network to get the sparse network.
Step 3: Reconstruction
    Combine conv_fc_class and conv_fc_bbr with convolution layer of the spare network.
Step 4: Secondary pruning
    4.1 Deep Pruning takes as input the training dataset and the reconstruction network to get the sparser network.
    4.2 Take the final network's convolutional network as RPN and then output RPN.
```

Pretraining. Deep neural networks always have millions of parameters and training so many parameters is a problematic with hundreds of remote sensing images. Fortunately, we can transfer pretrained networks to our task because the low level convolutional kernels extract the similar features. So, this step provides primary feature for the next step to make the network converge faster even if the training set is really small. Pretraining uses the remote sensing dataset AID (Aerial Image Dataset) [17] rather than ImageNet dataset [10, 18] to train the initial network for getting better feature.

First Pruning. We use deep pruning to change the initial network structure. The training dataset is cropped from NWPU VHR-10 remote sensing dataset [18]. We input the training dataset with its class label, and the pretrained network to Deep Pruning to get a spare network.

In the first pruning, we remove the connections that is not important to the binary classification (object/background) and leave space for the localization task by setting a smaller sparsity.

Reconstruction. Reconstruction allows the network to have the ability to handle both binary classification and localization simultaneously. In this step, we first select the convolution layer of the network that previous step output. And then add two branches after the final convolution layer: one is used to distinguish background and object called cls_fc_class; another provides coordinate offset named cls_fc_bbr. The output of cls_fc_class is entered to softmax classifier during training, and the softmax gives the probability of the image as object or background. cls_fc_bbr uses L1 loss function in training to perform bounding box regression.

Secondary Pruning. The secondary pruning changes the network structure again. We input the cropped training dataset with its class label and ground truth bounding box, and the reconstruction network to Deep Pruning to get a sparser network. During deep pruning, first of all, train the network so that we can know which connections are effective not only for the binary classification, but also to the localization. Since there are two tasks, the corresponding objective function should be multi-task loss function. Therefore, the multi-task loss function that optimizes the binary classification and localization tasks at the same time is:

$$L_{\mathrm{ODDP}}(loc, p_{\mathrm{bic}}) = L_{\mathrm{bic}}(p_{\mathrm{bic}}) + \alpha L_{\mathrm{bbox}}(loc) \tag{3}$$

Where loc is the predicted tuple for bounding box regression, p_{bic} is the class confidence calculated by the softmax classifier, L_{bic} is the softmax loss for binary classification of object and background, L_{bbox} is smooth L1 loss:

$$\begin{gathered} L_{\mathrm{bbox}}(loc) = f_{\mathrm{L1}}(loc - loc_{\mathrm{t}}),\ where \\ f_{\mathrm{L1}}(x) = \begin{cases} 0.5x^2, if\ |x| < 1 \\ |x| - 0.5,\ otherwise \end{cases} \end{gathered} \tag{4}$$

Where loc_{t} is the ground truth tuple for bounding box regression.

The background is meaningless in the back propagation, but it will cause the model to converge prematurely. Therefore, set the background sample $\alpha = 0$ and the object sample $\alpha = 0.5$.

Then the secondary pruning is performed to remove the redundant connections to the localization and the binary classification tasks. The secondary pruning method and the first pruning both use Deep Pruning method, while the sparsity of the two pruning process are different.

With the above training completed, Region Proposal Network (RPN) is constructed by the final network's convolution layers. So, the RPN is actually a fully convolutional neural network. At test, RPN takes image Gaussian pyramid to obtain heat maps, and we can get central coordinates of proposals from heat map's local maximal positions.

2.3 Optimizing Classification Network

The benefits of complex networks are very expressive and can capture the highly nonlinear relationship between features and output. The drawback of large network is that it tends to capture the noise in the training dataset. This noise cannot be generalized to new datasets, resulting in overfitting, high variance and weak generalization ability. Simply reducing the capacity of the model leads to another extreme that the network will miss the correlation between the features and output, resulting in underfitting and high bias.

In the existing work [10–13], when training classification network, they selected a network based on experience. So the dataset and network capacity do not match well, which may lead to overfitting or underfitting. We propose a training algorithm Optimizing Classification Network algorithm (OCN) to regulate the network, so that the network can better learn the distribution of dataset. The proposed OCN and the workflow are described in detail below.

In order to achieve consistency with the region proposal network, we use the same pretrained network with LRPN. Firstly, fine-tune the pretrained network, and then set a bigger sparsity than first pruning of LRPN to prune the network by Deep Pruning. Compared with region proposal network, classification network needs to learn the unique features of each class, and to distinguish the subtle differences between multiple classes. So, we need a bigger capacity model which has more connections and that is why after pruning and retraining the network, the cut off connections are recovered.

```
Algorithm 3: workflow of OCN
Input: training dataset, the pretrained network
Output: classification network
Step 1: Deep Pruning takes as input the training dataset and the pretrained network to
get the sparse network.
Step 2: Recover the cut off connections and initialize the weights
Step 3: Train the new network SGD(W)
```

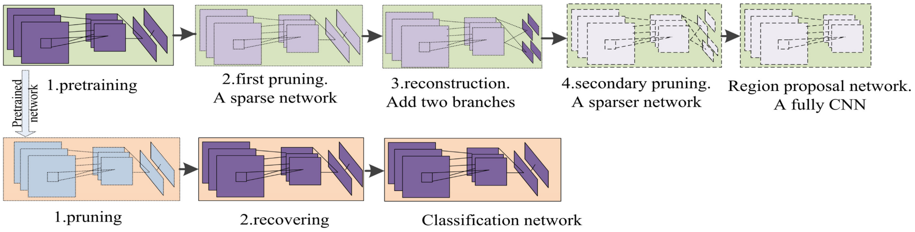

Fig. 2. Framework of ODDP. ODDP includes LRPN and OCN algorithms.

3 Experiment and Analysis

We evaluate the proposed ODDP on a public VHR remote sensing image dataset named NWPU VHR-10 [18], which has 10 classes. We use number 1 to 10 to represent 10 classes during training phase. In this experiment, we only use positive samples of the dataset, which include 650 VHR images. Some of the pictures are from Google maps, and their spatial resolution is between 0.5 m and 2 m. The other part is from Vaihinge dataset, and their spatial resolution is 0.08 m. In our experiment, we select 50% of the whole dataset for training set, 20% for validation set and 30% for test set. To evaluate ODDP, we use average running time, Average Precision (AP) and mean Average Precision (mAP).

We select AlexNet as the initial network and AID as the pretraining dataset. We define image patches as positive or negative samples based on IoU $\geq$ 0.7 or IoU $<$ 0.3. IoU is an evaluation of object detection, which is the overlap rate of two boxes, that is

$$I = \frac{G \cap D}{G \cup D} \tag{5}$$

Where, I is the overlap rate, D (Detection result) is the box predicted by the object detection model, G(Ground truth) is the ground truth box of the detected image.

The hardware environment is 2.8 GHz, 6 core CPU, 32 GB memory, GTX Titan X.

Figure 3 is the detecting result of some images of Vaihinge dataset with 0.08 m spatial resolution. It shows that our proposed method can well detect objects in VHR images. The rectangles in the images are the predicted bounding boxes. The first value in the upper left corner above rectangle is class label, and the second value is the probability that the area within the rectangle belongs to the class. The yellow solid ellipses mark false negative and the green dotted ellipses mark the false positive. It can be seen that although the objects are very different in size, shape and texture, ODDP still detected most objects. We compare ODDP with four current optimal methods, that are COPD [19], transferred CNN [20], RICNN with or without fine-tuning [18]. To be convincing, ODDP and four comparative methods all adopted the same training and test dataset. In addition to COPD, the three comparison methods adopted AlexNet network structure.

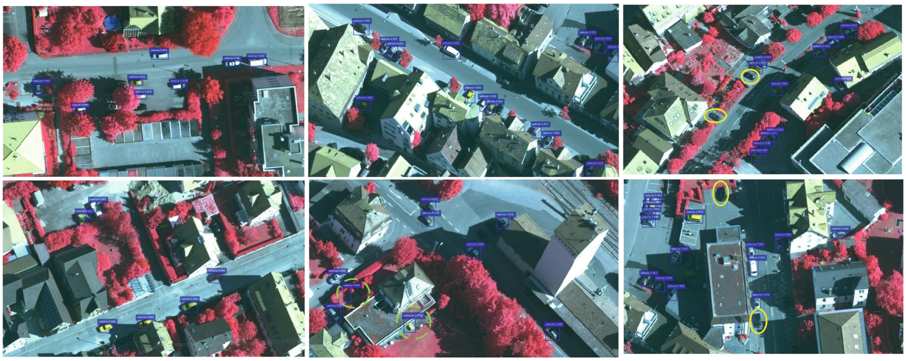

Fig. 3. Object detection results with the proposed method, and the images are from Vaihinge dataset (0.08 m spatial resolution). (Color figure online)

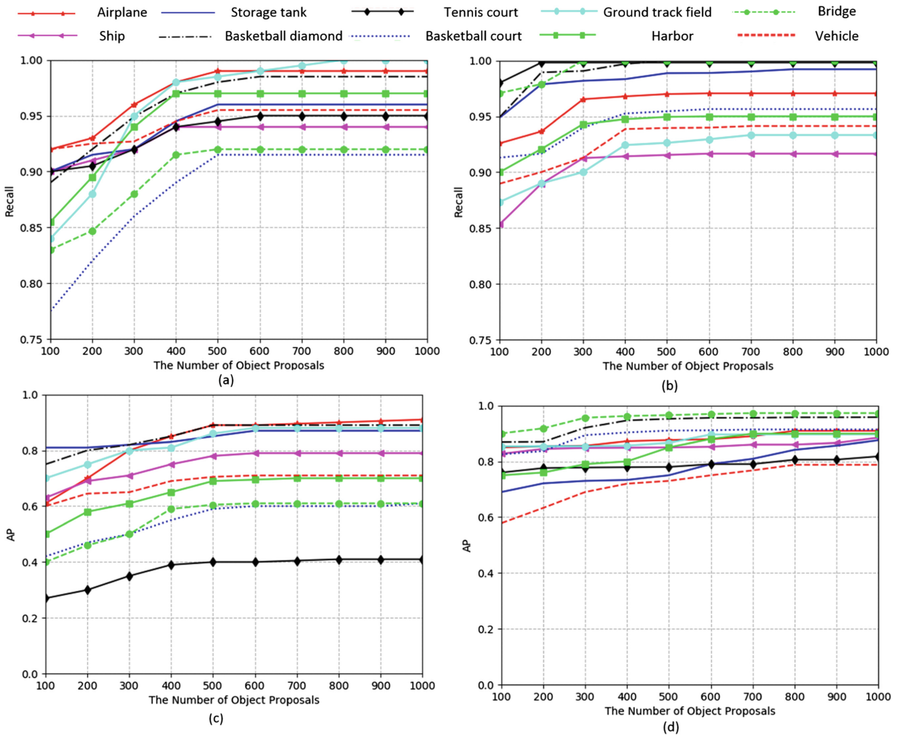

Fig. 4. (a), (b), (c), (d) show the relationship between the hyperparameter (the number of object proposals) and the detecting results (recall and AP). (b), (d) are the detecting results of our proposed method ODDP, and (a), (c) are the results of the RICNN with fine-tuning in [18].

Tables 1, 2 and Fig. 4 are comparison results. From Table 2 we can see that ODDP has better AP than the other four methods. Compared with RICNN with fine-tuning, the average precision of ten classes predicted by the proposed method improve 5.52%,

Table 1. Computation comparisons of five different methods

Algorithm	COPD	Transferred CNN	RICNN without fine-tuning	RICNN with fine-tuning	ODDP
Average running time/s	1.07	5.24	8.77	8.77	0.39

Table 2. The average precision of five object detection methods for 10 classes. The highest score is bold in each row (%)

	COPD	Transferred CNN	RICNN without fine-tuning	RICNN with fine-tuning	ODDP
Airplane	60.44	64.48	84.33	85.39	**90.91**
Ship	67.85	55.71	74.53	75.38	**88.46**
Storage tank	60.31	81.38	66.87	84.95	**87.53**
Baseball diamond	81.20	79.78	85.28	85.69	**95.79**
Tennis court	52.23	56.46	63.82	65.70	**81.79**
Basketball court	36.71	46.73	57.18	57.96	**91.39**
Ground track field	82.71	78.26	82.77	86.32	**89.61**
Harbor	82.72	78.26	82.79	84.12	**89.90**
Bridge	17.13	45.97	58.25	61.28	**97.27**
Vehicle	44.09	43.01	66.56	69.88	**78.76**
Mean AP	58.54	63.00	72.24	75.45	**89.14**

13.08%, 2.58%, 10.1%, 16.09%, 33.43%, 3.29%, 5.78%, 19.53%, 35.99%, 8.88%, respectively, and mAP improves 13.69%, indicating that ODDP has better detection ability. Figure 4 shows the tradeoff between "the number of object proposals" and recall, AP respectively. "The number of object proposals" is the hyperparameter of two-stage object detection method. Figure 4 also compare our method with RICNN with fine-tuning [18]. We can see from (a), (b) that as the horizontal axis increases, the recall curve of our method quickly reaches a plateau, while the contrast method requires a larger hyperparameter to reach a plateau, which indicates that our method locate objects more accurate. It can be seen from (c), (d) that our method has higher AP curve in every class and can reach a high AP earlier than the contrast method. The average running time is used to evaluate the speed of each object detection method. As can be seen from Table 1, ODDP runs faster.

4 Conclusion

In this work, a two-stage object detection method for VHR remote sensing image is proposed. Inspired by the dropout method, we first propose Deep Pruning, which can reduce the network capacity and the probability of overfitting and accelerate network computing. Then, we propose the object detection training algorithm based on Deep Pruning, including LRPN algorithm and OCN algorithm. The region proposal network and classification network can be obtained by inputting the initial network into LRPN and OCN respectively. And the two networks are combined into the two-stage object detection model. The experimental results show that ODDP outperform the existing object detection methods in remote sensing dataset.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (61472161), Science & Technology Development Project of Jilin Province (20180101334JC), Natural Science Foundation of Hunan Province (No. 2018JJ3479).

References

1. Yang, Y., Zhuang, Y., Bi, F., Shi, H., Xie, Y.: M-FCN: effective fully convolutional network-based airplane detection framework. IEEE Geosci. Remote Sens. Lett. **14**(8), 1293–1297 (2017)
2. Zhong, Y., Fei, F., Zhang, L.: Large patch convolutional neural networks for the scene classification of high spatial resolution imagery. J. Appl. Remote Sens. **10**(2), 025006 (2016)
3. Luo, Q., Shi, Z.: Airplane detection in remote sensing images based on object proposal. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1388–1391. IEEE Press (2016)
4. Liu, K., Mattyus, G.: Fast multiclass vehicle detection on aerial images. IEEE Geosc. Remote Sens. Lett. **12**(9), 1938–1942 (2015)
5. Cao, Y., Niu, X., Dou, Y.: Region-based convolutional neural networks for object detection in very high resolution remote sensing images. In: 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 548–554. IEEE Press (2016)
6. Zhang, R., Yao, J., Zhang, K., Feng, C., Zhang, J.: S-CNN ship detection from high-resolution remote sensing images. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLI-B7, pp. 423–430 (2016)
7. Chen, Z., et al.: Vehicle detection in high-resolution aerial images via sparse representation and superpixels. IEEE Trans. Geosci. Remote Sens. **54**(1), 103–116 (2016)
8. Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J.: Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Trans. Geosci. Remote Sens. **53**(6), 3325–3337 (2015)
9. Shao, W., Yang, W., Liu, G., Liu, J.: Car detection from high-resolution aerial imagery using multiple features. In: 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4379–4382. IEEE Press (2012)
10. Long, Y., Gong, Y., Xiao, Z., Liu, Q.: Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Trans. Geosci. Remote Sens. **55**(5), 2486–2498 (2017)
11. Jiang, Q., Cao, L., Cheng, M., Wang, C., Li, J.: Deep neural networks-based vehicle detection in satellite images. In: 2015 International Symposium on Bioelectronics and Bioinformatics (ISBB), pp. 184–187. IEEE Press (2015)
12. Ševo, I., Avramović, A.: Convolutional neural network based automatic object detection on aerial images. IEEE Geosci. Remote Sens. Lett. **13**(5), 740–744 (2016)

13. Wu, H., Zhang, H., Zhang, J., Xu, F.: Typical target detection in satellite images based on convolutional neural networks. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2956–2961. IEEE Press (2015)
14. Diao, W., Sun, X., Zheng, X., Dou, F., Wang, H., Fu, K.: Efficient saliency-based object detection in remote sensing images using deep belief networks. IEEE Geosci. Remote Sens. Lett. **13**(2), 137–141 (2016)
15. Girshick, R., Donahue, J., Darrell, T., Malik, J.: 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 (2014)
16. Denil, M., Shakibi, B., Dinh, L., De Freitas, N.: Predicting parameters in deep learning. In: Advances in Neural Information Processing Systems, pp. 2148–2156 (2013)
17. Xia, G.S., et al.: AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans. Geosci. Remote Sens. **55**(7), 3965–3981 (2017)
18. Cheng, G., Zhou, P., Han, J.: Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens. **54**(12), 7405–7415 (2016)
19. Cheng, G., Han, J., Zhou, P., Guo, L.: Multi-class geospatial object detection and geographic image classification based on collection of part detectors. ISPRS J. Photogramm. Remote Sens. **98**, 119–132 (2014)
20. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)