Target Detection Method for SAR Images Based on Feature Fusion Convolutional Neural Network

Yufeng Li, Kaixuan Liu, Weiping Zhao, Yufeng Huang

College of Electronics Information Engineering, Shenyang Aerospace University, China
liyufeng@sau.edu.cn, liukaixuan0524@163.com, sjzwpzm@sohu.com, yufengh_sau@sina.com

Abstract

For the image target of Synthetic Aperture Radar (SAR), it is more difficult to detect targets in complex background, large scene and more clutter. This paper designs a less layer convolutional neural network (CNN), the complete data validates its feature extraction effect, as a basis for feature extraction networks. In the training dataset, it supplements the target training samples in complex large scene, meanwhile a multi-level convolution feature fusion network is designed to enhance the detection ability of small targets in large scene. After the joint training of the region proposal network (RPN) and the target detection network, a complete model for SAR image target detection in different complex large scenes is obtained. The experimental results show that the proposed method has a good result on SAR image target detection and has an average precision (AP) value of 0.86 in the validation dataset.

Keywords: Synthetic aperture radar, Convolutional neural network, Target detection, Feature fusion, Complex large scene

1 Introduction

Synthetic Aperture Radar (SAR) [1] has all-day, all-weather and certain penetration capabilities. With the rapid development of SAR technology, it is more widely used in military and civilian fields such as investigation and remote sensing. Higher requirements are also put forward for SAR image processing technology. How to quickly and accurately detect SAR image targets has become an important issue to be studied in the field of SAR image target detection [2].

In the traditional SAR image target detection method, the Constant False Alarm Rate (CFAR) detection algorithm [3-5] is widely used. Based on local sliding windows, most CFAR algorithms perform pixel-by-pixel detection on SAR images to determine whether a certain threshold is exceeded. The target and background need to have a certain contrast, and the background clutter has certain statistical characteristics. However, in complex large scene, there are many background clutters, and its statistical characteristics are difficult to obtain, and the detection effect is also poor.

CNN is an artificial neural network in deep learning. It has developed rapidly in recent years and has attracted widespread attention and has powerful feature extraction capabilities. Krizhevsky [6] proposed an image classification algorithm for deep convolutional neural networks, which achieved good results in the ILSVRC2010 competition, greatly improved the accuracy of image classification and target detection. Girshick et al. [7] proposed the R-CNN method, which adopted Selective Search (SS) to select candidate regions, extracted feature from candidate regions by CNN, recognized target categories by Support Vector Machine (SVM) classifier and performed bounding-box regression on the candidate regions to improve detection efficiency. Kaiming He et al. [8] proposed SPP-net that can generate fixed-length representation, solve the problem that CNN input requires a fixed size, and improve CNN-based image detection methods. Girshick et al. [9] improved the existing methods and proposed the Fast-RCNN method. After passing the whole image through CNN, the candidate region feature map was generated, which greatly improved the detection efficiency. Ren et al. [10] once again proposed the Faster-RCNN method to improve the generation of candidate regions, so that the detection efficiency was further improved.

CNN has excellent feature extraction performance for many different types of images [11-16], but most of them are based on deep convolutional network models [17-19]. The rest of this paper is organized as follows, according to the characteristics of SAR image dataset, this paper designs a CNN with less layers and easy training. Based on this, a multi-level convolution feature fusion network is proposed. Combined with the idea of RPN and Fast RCNN, an end-to-end approximate joint training method is used to train a complete SAR image target detection system under the acceleration of GPU. However, using only a relatively complete SAR image slice dataset doesn't well achieve SAR target detection in complex large scenes. Therefore, in the SAR image dataset, we supplement

[*] Corresponding Author: Yufeng Li; E-mail: liyufeng@sau.edu.cn

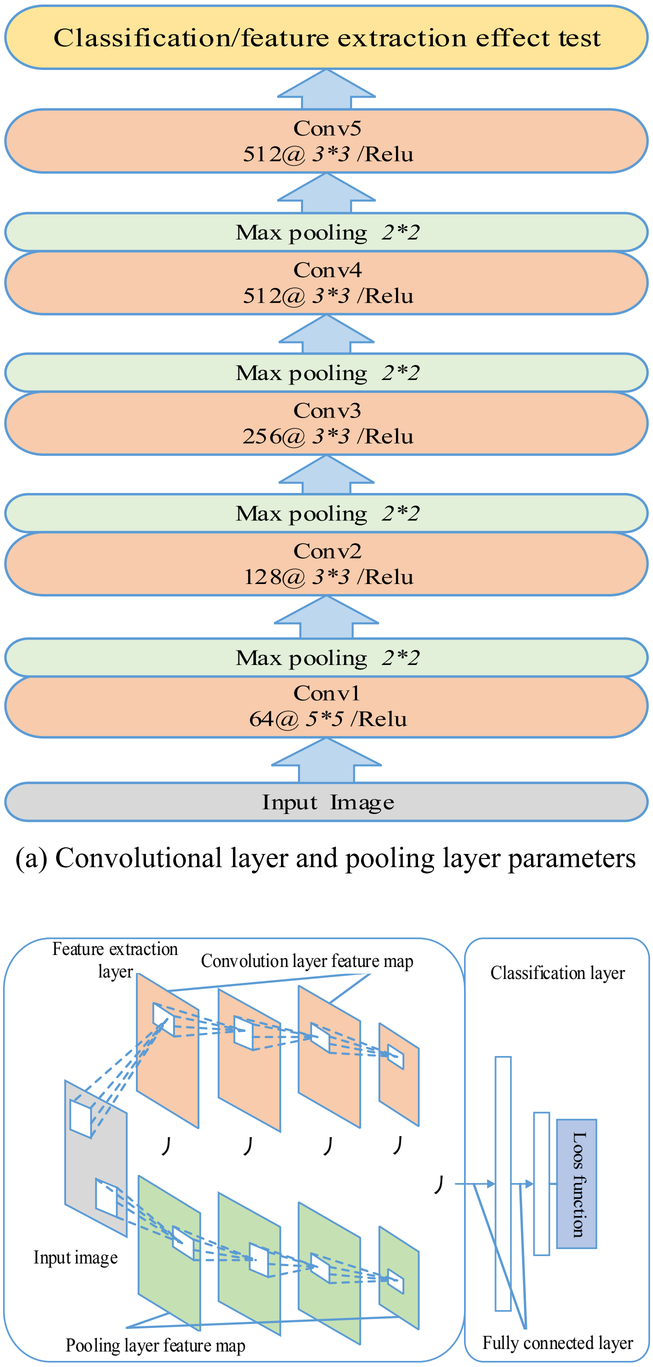
DOI: 10.3966/160792642020052103023

the target training samples in complex large scenes including wheat fields, trees, and buildings, thus enhancing the target detection capability of the network in complex large scenes. According to the experimental results, the proposed method has better robustness and generalization ability, and can detect SAR image targets in a variety of complex large scenes and has better detection results.

2 Feature Extraction Network

2.1 Design of Basic Convolution Feature Extraction Network

Different from large sample datasets such as ImageNet and PASCAL VOC, SAR image dataset is small [20]. According to the actual requirements and data characteristics, we design a convolutional neural network model using features with fewer layers. The specific structure is shown in Figure 1. The model includes an input layer, 5 convolutional layers, 4 pooling layers, and an output layer. After the input data layer normalizes input data, the first convolution layer discards large convolution kernel used by the majority of the network in the first convolutional layer, and adopts *5*5* size convolution kernel, with smaller receptive field. The remaining convolutional layers use *3*3* convolution kernel, and the number of convolution kernels is 64, 128, 256, 512, 512, respectively. All convolution kernels have a sliding step size of 1. The first four convolutional layers are each followed by a maximum pooling layer (max pooling), the local sensing area is *2*2*, the step size is 2, and the two fully connected layers are connected after the last convolutional layer. Each convolutional layer and the first fully connected layer use the Rectified Linear Unit (ReLU) [21] function as the activation function, which can effectively solve the saturation problem in the experiment. The classification layer is used to verify the feature extraction effect of the designed CNN model. When the classification result reaches a certain precision, the extracted feature map can be used for the subsequent network.



(a) Convolutional layer and pooling layer parameters

(b) Feature extraction network structure

Figure 1. CNN feature extraction network

2.2 Feature Fusion Network Design and Data Improvement

The traditional Faster-RCNN only selects the feature map of the last convolutional layer, ignores other low-level convolutional feature maps, and loses part of the target information. The deep convolution characteristics have more abstract expressions on the semantic features of the image, which is beneficial to the classification of the target, which is not conducive to the target positioning. The low-level convolution features consider the target local detail texture and so on, which is conducive to position. In this paper, the detection of SAR targets is small, and it is necessary to detect targets in complex scene. A multi-level convolution feature fusion network of Figure 2 is designed, which has better feature representation considering different levels of features.

After selecting the feature maps of the first convolutional layer and the third convolutional layer, there are inconsistencies in the size of the different layer feature maps. Therefore, we perform a max pooling operation to obtain the same size as the fifth convolutional layer feature map. After multi-level feature fusion, the feature resolutions of different levels of feature maps are different, and local response normalization (LRN) operations are required.

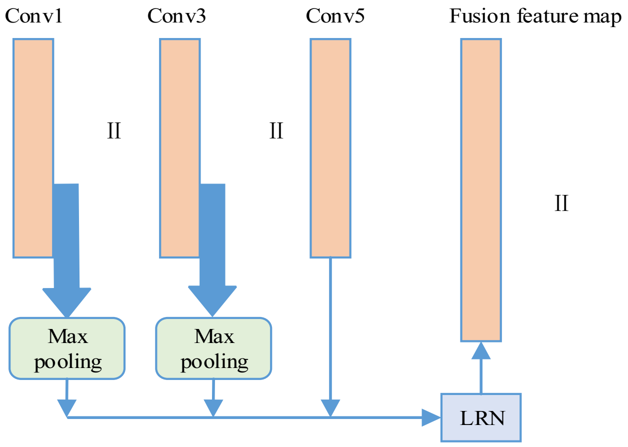


Figure 2. Multi-layer feature fusion network

3 Target Detection Network

The target detection part adopts the idea of RPN [10]. The candidate region of the target location is obtained from the input multi-level fusion feature map. Then use the network of the target detection part of Fast-RCNN to classify and perform bounding-box regression on the candidate regions, detect the target, and obtain a more realistic target region, where the two networks share the CNN convolutional layer feature.

3.1 Generation of Target Candidate Region

The RPN is used to directly obtain candidate regions of the included targets. The specific structural model is shown in Figure 3. The input is a feature map of the image after the multi-level feature fusion of 2.2 sections. After sliding through *n*n* sliding window, it slides on the input feature map and maps it to a lower latitude dimension feature vector. This feature vector is then input into the convolutional layers of two peers. The convolution kernel size is *1*1*, which is used for classification and bounding-box regression to achieve the original target region refinement. In order to make the receptive field and candidate regions of the sliding window as close as possible, to achieve more accurate classification and bounding-box regression, this paper selects *3*3* sliding window.

Compared with the traditional selective search [7], the use of RPN to generate regional proposals greatly shortens the time and improves efficiency. In the center of each small sliding window, we generate 9 original regions by setting 3 scales and 3 aspect ratios, called anchor. Since the SAR target to be detected in this paper is small, we have abandoned the original scale of 128/256/512, set the scale to 64/128/256, and the aspect ratio is still 1:1/1:2/2:1. Then calculate the Intersection over Union (IoU) of the reference region and the target real region, IoU greater than 0.7 is called positive label, less than 0.3 is called negative label, 1:1 is selected positive and negative label, and 256 anchors are trained as a mini-batch.

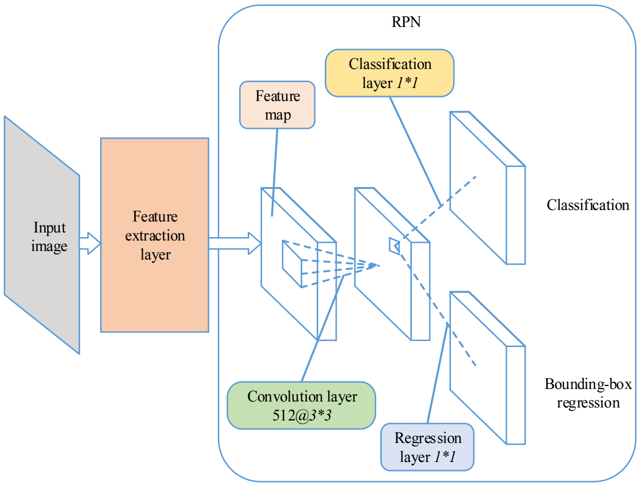


Figure 3. Candidate region generation network

We choose the multi-task loss function shown in Equation (1) to perform hyper-parameter optimization of the RPN network [10].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} L_{cls}(p_i,p_i^*) + \lambda \frac{1}{N_{reg}} \sum p_i^* L_{reg}(t_i,t_i^*), \quad \textbf{(1)}$$

Where i is the index of the target reference region in a mini-batch, N_{cls} indicates the number of samples contained in it, that is, the size of the mini-batch. N_{reg} denotes the number of anchors of all reference regions of the input network, and λ is a balance coefficient for controlling the weights of the two loss functions. In the Equation (1), the classification loss function is used to determine whether the region has a target, and the optimized cross entropy loss function is used in the experiment.

$$L_{cls}(p_i,p_i^*) = \sum_i -p_i^* \ln p_i, \qquad \textbf{(2)}$$

It applies to situations where each category is independent and exclusive. p_i indicates the probability that the i-th anchor is predicted to be the target region after softmax, and p_i^* is the real label of the reference target region. In equation (1), L_{reg} is the bounding-box loss function [9].

$$L_{reg}(t_i,t_i^*) = \sum_{\theta\in(x,y,w,h)} \text{smooth}_{L_1}(t_i - t_i^*), \qquad \textbf{(3)}$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & x<1 \\ |x|-0.5 & \text{others} \end{cases}, \qquad \textbf{(4)}$$

t_i is a vector representing the four parameterized coordinates of the prediction region. The t_i^* vector represents the four parameterized coordinates of the real region, $t_i = (t_x, t_y, t_w, t_h)$ corresponds to the predicted parameterized coordinates, $t_i^* = (t_x^*, t_y^*, t_w^*, t_h^*)$

corresponds to the real translation scaling parameterized coordinates, t_x, t_y are the translation parameters, and t_w, t_h are the logarithmic space scaling parameters, the specific forms are [7]:

$$\begin{cases} t_x = (x - x_a)/w_a, & t_y = (y - y_a)/h_a, \\ t_w = \log(w/w_a), & t_h = \log(h/h_a), \end{cases} \tag{5}$$

$$\begin{cases} t_x^* = (x^* - x_a)/w_a, & t_y^* = (y^* - y_a)/h_a, \\ t_w^* = \log(w^*/w_a), & t_h^* = \log(h^*/h_a), \end{cases} \tag{6}$$

x, y represent the abscissa and ordinate of the center point of the prediction region, w, h represent the width and height of the prediction region, x_a represents the abscissa of the initial reference anchor, and x^* represents the true abscissa. The ordinate, width, and height are the same as this definition.

There are a lot of overlaps in the proposed regions generated by the RPN network. Non-Maximum Suppression (NMS) is adopted to remove some regions, and the first N pieces with the highest probability of being the target region determined by the classification layer are selected as the final proposal region output.

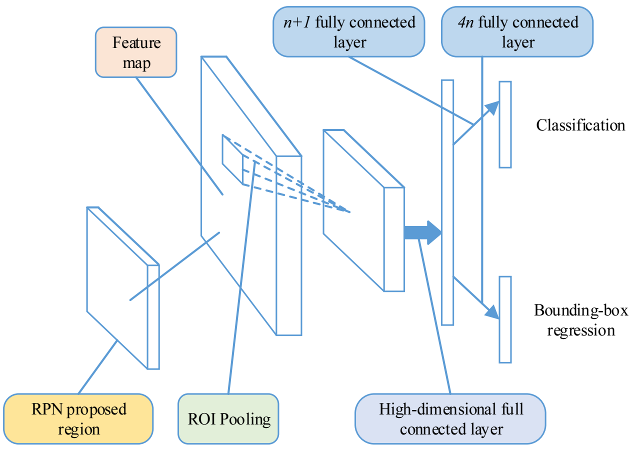
3.2 Generation of Target Candidate Region Description of the Target Detection Network

As shown in Figure 4, the candidate region generated by the RPN network is mapped back to the convolution feature map generated by the network in Section 2.1 according to the pre-set feat_stride parameters. Due to the multi-scale characteristic of the anchor, the size of the proposed feature maps is not fixed, and the Region of Interest (ROI) Pooling layer is required to implement fixed-size input. Then, after the fully connected layer, it is mapped to a feature vector, and a high-dimensional feature vector is used in this paper. The resulting feature vector is used for the next two parallel hierarchical classification layers (*n+1* class targets) and the bounding-box regression layer. The former is used for target classification, and the discrete probability P of the proposed regional target classification result is obtained, and the classification results are mutually exclusive. The latter obtains four parameterized coordinates of the target region, and the same Equation (5) is used for target localization.

4 Experiment and Result Analysis

4.1 Data Preparation and Explanation

In order to verify the effectiveness of the proposed method, the experiment selected *128*128* size stationary target slice images from the Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset, including 7 sub-categories, 3 major categories of different azimuth angles of the vehicle target image, as shown in Figure 5. MSTAR also contains images of complex large scenes with a size of *1748*1478*, including wheat fields, grasslands, trees, moors, buildings, etc. But these images do not contain targets. Therefore, we embed many target images of size *128*128* into the entire scene image [12]. This operation is reasonable because the target image and the scene image are both acquired by the same SAR sensor, and the resolution is unified to 0.3M. In this way, the m_MSTAR dataset required for the experiments in this paper is constructed and combined with the MSTAR dataset to form the complete dataset used in this paper. After the data enhancement technology, the m_MSTAR and MSTAR datasets have a ratio of 1:1.5 for a total of 800 dataset images. Figure 6 shows examples of two dataset samples.



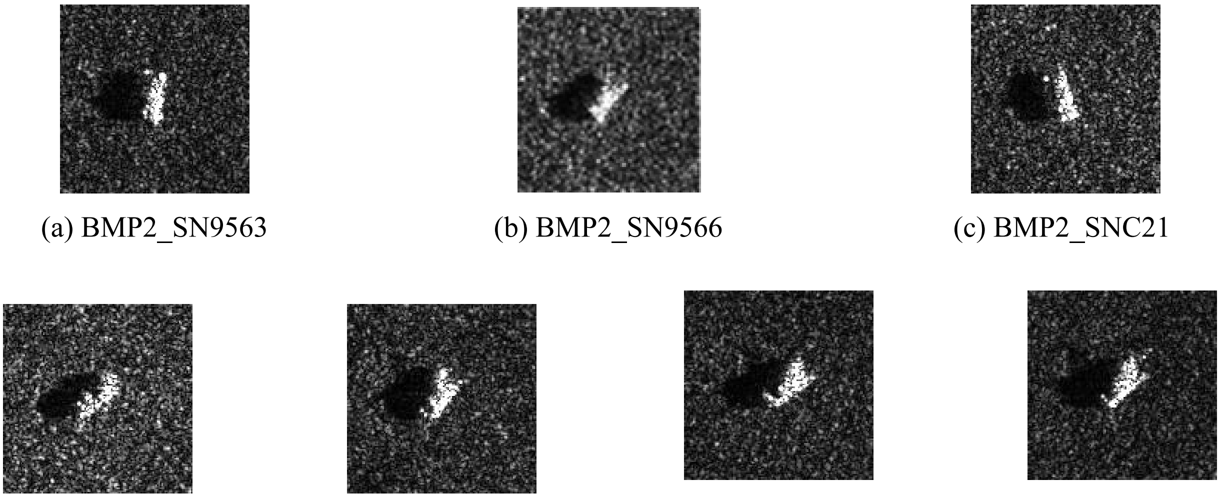
Feature map
n+1 fully connected layer
4n fully connected layer
Classification
Bounding-box regression
RPN proposed region
ROI Pooling
High-dimensional full connected layer

Figure 4. Target detection network

4.2 Multi-task Network Training

In the experiment, the approximate joint training multi-task network is used to share the fused convolution feature map, and the RPN network and the Fast-RCNN target detection network are merged into one training network. Because the derivative of the ROI pooling layer on the regional coordinates is neglected during the back-propagation process, the weight of this part cannot be updated, so it is called approximate joint training. This method can save time and improve efficiency.

In the training of the approximate joint training network, the initial learning rate is set to 0.01, and when the change in the loss value is small, the learning rate is updated, and the learning rate is divided by 10. For network loss function optimization, this paper chooses to use the Stochastic Gradient Descent (SGD) algorithm with momentum. When training the network, initializing according to the Gaussian distribution does not guarantee that the initial weight of the network is in a proper state. It may fall into a local minimum during training process and cannot achieve global optimality. For high-dimensional non-convex functions, there may



(a) BMP2_SN9563 (b) BMP2_SN9566 (c) BMP2_SNC21

(d) BTR70_SNC71 (e) T72_SN132 (f) T72_SN812 (g) T72_S7

Figure 5. The vehicle target images

(a) m_MSTAR (b) MSTAR

Figure 6. Sample data instances

be many saddle points. After the momentum is introduced, the SGD can escape from the local saddle point or not fall into the local minimum, and at the same time play a certain acceleration. The SGD weight after adding momentum is updated to

$$v = \text{mu} * v - \text{learning\_rate}*dw, \tag{7}$$

$$w = w + v, \tag{8}$$

v is initially 0, mu is a hyper-variable, set to 0.9 in this paper, and dropout is used during model training to prevent overfitting. During the training process, the activation value of a certain neuron stops working with a certain probability p, and the weight is not updated during the training process, nor does it participate in the calculation of the neural network, but its weight is retained. Dropout is only used in the training stage of the model. In this paper, it is only used for the classification layer of the target detection network stage and the fully connected layer before the bounding-box regression layer. The experiment uses the Keep_proc parameters to set the probability that a neuron is selected.

4.3 Experimental Result Analysis

4.3.1 Experimental Parameters Settings

The main parameters settings in the experiment are shown in Table 1. According to the experimental experience, when the influence of the number of iterations on the precision is stable, the number of iterations can be no longer increased. Therefore, the number of iterations selected in this paper is 30000.

Table 1. Experimental parameters values

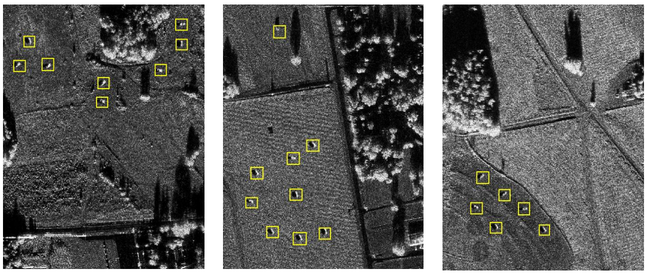
Operating system	Program environment	GPU	Learning rate
Ubuntu16.04	Tensorflow	GeForce GTX 980	0.01

Momentum	Target score threshold	RPN positive overlap	Batch size	Number of iterations
0.9	0.8	0.7	128	30000

4.3.2 SAR Target Detection in Complex Large Scenes

Use the new data to verify the effectiveness of the proposed network model for the SAR target detection.

Three complex scenes in the validation dataset are selected, including a plurality of different posture targets to be detected, and the detection effect is shown in Figure 7. The model achieves automatic detection of different posture SAR targets for scene 1 and scene 3, and the target position is more accurate. Scene 2 has many trees, fields and other interference backgrounds. Although there are false inspections, trees or other disturbances are detected as targets, but there is no missed inspection. Subsequently, the target of the false inspections can be manually processed, or the network can be adjusted, and the hyper-parameter can be optimized to reduce the false detection rate.



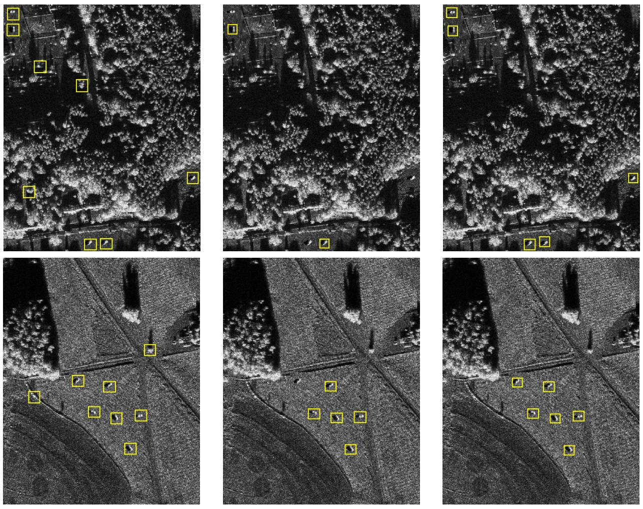
(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 7. Target detection result of SAR images in different scenes

Compared with other feature extraction networks: the VGG net [22] was proposed by the Visual Geometry Group at Oxford University, and the VGG net contains 16-19 layers. Different from the deep network structure of the VGG net, the ZF net [23] has 5 convolutional layers and 3 pooling layers, which is similar to the basic CNN structure layers in this paper, and the training difficulty is roughly equal. Figure 8 shows the target detection results of different feature extraction networks. Compared with the method in this paper, when the other two structures are used as the basic feature extraction network, there are too many missed detections and false detections. Table 2 gives the detection results in the validation dataset. In the local experimental environment, for the same *1476*1748* SAR image, the proposed method takes 0.324s, and the VGG net structure takes 0.465s, and the detection efficiency is significantly reduced.

In the table, True Positives (TP) indicates the number of correctly detected targets, False Positives (FP) indicates the number of false inspections, AP indicates average precision value of the target detection, and the number of targets that the validation dataset requires to detect is *N=120*. The recall rate (R) and precision rate (P) are used to evaluate the SAR target detection results. The specific expressions are as follows:

$$R = \frac{TP}{N}, P = \frac{TP}{TP + FP}, \tag{9}$$



(a) The VGG net (b) The ZF net (c) The proposed

Figure 8. Different feature extraction structure detection results

Table 2. CNN detection results of different feature extraction layers

Method	TP	FP	Precision rate (%)	Recall rate (%)	AP
The VGG net	62	16	79.5	51.7	0.52
The ZF net	100	3	97.1	83.3	0.74
The proposed	115	6	95	95.8	0.86

From the detection results of Figure 8 and Table 2, for the dataset used in this paper, the more complicated the VGG net model structure is, the more difficult to optimize the hyper-parameter, the easier to mistake the noise for effective features, and the more false inspections appear (16). The generalization ability is poor when new data is used. Although the precision rate of the ZF net detection is high (97.1%), there are fewer cases of false detection (3), but the recall rate is insufficient (83.3%). The proposed method has a moderate number of false detections (6), and the precision rate is lower than that of the ZF net detection by 2.1%, but the recall rate is higher than the ZF net detection by 12.5%, and has a higher AP value (0.86).

Robustness is one of the important factors to consider in target detection. In the validation dataset of this paper, a Salt and Pepper noise with a noise density of 0.1 and a Gaussian noise with a mean of 0 and a variance of 65 are added to some images. In the experiment, the VGG net's detection ability is obviously weak in the face of noise, and there are cases where the target is not detected completely in the whole scene, and the interference wave detection such as trees is also targeted, and the recall rate is relatively poor. The ZF net is also insufficient in the expression of target features, and the recall rate is not ideal. In this method, the number of layers of the basic CNN is moderate, and the convolution feature extraction effect is better. At the same time, multi-level convolution feature fusion is added, which has richer feature expression. After the "two-step" network, which first obtains the candidate target region and then detects the target, it finally achieves a better detection result on the smaller SAR target in the large scene.

5 Conclusion

In this paper, we use a more complete SAR dataset, supplement the training sample information in complex large scenes, and use CNN's excellent feature extraction ability, discard the CNN structure with too many layers, and use a basic CNN with fewer layers. By multi-level convolution feature fusion network, more rich target feature information is obtained. After the target detection network, and using the GPU to accelerate the optimization training, a complete SAR image target detection model is obtained. The experimental data and results show that the proposed method has better robustness and generalization ability and can detect small SAR image targets in complex large scenes.

Acknowledgments

This work was supported by National Science and Technology Major Project of High Resolution Earth Observation (70-Y40-G09-9001-18/20), Liaoning Provincial Natural Science Foundation of China (20180550334), Key Project of Ministry of Education of China (2017A02002), Liaoning education department science and technology research project (L201701, L201705 and L201735). The authors deeply appreciate the supports.

References

[1] B. Zheng, M. D. Xing, T. Wang, *Radar Imaging Technology*, Publishing House of Electronics Industry, 2005.

[2] X. R. Xue, *Research on SAR Image Processing Technology*, Scientific and Technical Documentation Press, 2017.

[3] G. Gao, A Parzen-window-kernel-based CFAR Algorithm for Ship Detection in SAR Images, *IEEE Geoscience & Remote Sensing Letters*, Vol. 8, No. 3, pp. 557-561, May, 2011.

[4] Y.-G. Ji, J. Zhang, J.-M. Meng, X. Zhang, A New CFAR Ship Target Detection Method in SAR Imagery, *Acta Oceanologica Sinica*, Vol. 29, No. 1, pp. 12-16, January, 2010.

[5] M. Zhao, J. He, Q. Fu, Survey on Fast CFAR Detection Algorithms for SAR Images Targets, *Acta Automatica Sinica*, Vol. 38, No. 12, pp. 1885-1895, April, 2012.

[6] A. Krizhevsky, I. Sutskever, G.-E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *Communications of the ACM*, Vol. 60, No. 6, pp. 84-90, June, 2017.

[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, United States, 2014, pp. 580-587.

[8] K.-M. He, X.-Y. Zhang, S.-Q. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 37, No. 9, pp. 1904-1916, September, 2015.

[9] R. Girshick, Fast R-CNN, *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1440-1448.

[10] S.-Q. Ren, K.-M. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, June, 2017.

[11] S. Chen, H. Wang, F. Xu, Q. Y. Jin, Target Classification Using the Deep Convolutional Networks for SAR Images, *IEEE Transactions on Geoscience & Remote Sensing*, Vol. 54, No. 8, pp. 4806-4817, April, 2016.

[12] J. Ding, B. Chen, H.-W. Liu, M.-Y. Huang, Convolutional Neural Network with Data Augmentation for SAR Target Recognition, *IEEE Geoscience & Remote Sensing Letters*, Vol. 13, No. 3, pp. 364-368, January, 2016.

[13] J. Geng, J.-C. Fan, H.-Y. Wang, X.-R. Ma, High-resolution SAR Image Classification via Deep Convolutional Cutoencoders, *IEEE Geoscience & Remote Sensing Letters*, Vol. 12, No. 11, pp. 2351-2355, January, 2015.

[14] J. Wu, H. Liu, G. He, A Scale A Daptive Convolutional Neural Network for Target Detection of High Resolution Remote Sensing Images, *Microelectronics & Computer*, Vol. 11, No. 8, pp. 78-86, January, 2018.

[15] H.-Z. Jiang, E. Learned-Miller, Face Detection with the Faster R-CNN, *IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, United states, 2017, pp. 650-657.

[16] E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez, Convolutional Neural Networks for Large-scale Remote-sensing Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 55, No. 2, pp. 645-657, June, 2017.

[17] Y.-B. Dong, H. Zhang, C. Wang, Y.-Y. Wang, Fine-grained Ship Classification Based on Deep Residual Learning for High-resolution SAR Images, *Remote Sensing Letters*, 2019, pp. 1095-1104.

[18] C. Szegedy, W. Liu, Y.-Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, United states, 2015, pp. 1-9.

[19] C. Lee, H.-J. Kim, W. Kyeong, Comparison of Faster R-CNN Models for Object Detection, *Proceedings of International Conference on Control, Automation and Systems*, Gyeongju, Korea, 2016, pp. 107-110.

[20] Z.-L Lin, C.-L. Wang, Y.-J. Hu, Y. Zhang, Convolution

Neural Network Model for SAR Image Target Recognition, *Journal of Image and Graphics*, Vol. 23, No. 11, pp. 1733-1741, April, 2018.

[21] P. Wang, R-Q. Ge, X. Xiao, Y.-P. Cai, G.-Q. Wang, F.-F. Zhou, Rectified-linear-unit-based Deep Learning for Biomedical Multi-label Data, *Interdisciplinary Sciences Computational Life Sciences*, Vol. 9, No. 3, pp. 419-422, November, 2017.

[22] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-scale Image Recognition, *International Conference on Learning Representations(ICLR)*, 2015.

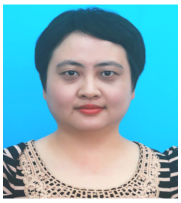
[23] M.-D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Network, *Proceedings of Computer Vision on ECCV 2014-13th European Conference*, Zurich, Switzerland, 2014, pp. 818-833.

Biographies

Yufeng Li is a professor in the School of Electronics and Information Engineering at Shenyang Aerospace University. He received the Ph.D. in Optical Engineering Department from Changchun Institute of Optics, Fine Mechanics and Physics in 2006. He received M.S. in communication systems from Jilin University in 2002. He received B.S. in communication engineering from Beijing Jiaotong University in 1991. His research interests include image processing and transmission technologies, wireless and mobile communication technologies, and avionics information technology.

Kaixuan Liu is a graduate student in the School of Electronics and Information Engineering at Shenyang Aerospace University. She received B.S. from Shenyang Aerospace University in 2017. Her research direction is image processing.

Weiping Zhao is a associate professor in the School of Electronics and Information Engineering at Shenyang Aerospace University. He received the Ph.D. in control theory and control engineering from Harbin Engineering University in 2004. In 1990 and 1997, he received B.S. and M.S. in speciality of power engineering from Northeast Electric Power University. His research interests include control theory, image processing, modern optimization algorithms, etc.



Yufeng Huang is a assistant professor in the School of Electronics and Information Engineering at Shenyang Aerospace University. She received the Ph.D. and M.S. from University of Science and Technology of China in 2007. She received B.S. from Hefei University of Technology. Her research direction is image processing and analysis.