DOI: 10.1051/itmconf/20171205001

Feature Fusion Based on Convolutional Neural Network for SAR ATR

Shi-Qi CHEN, Rong-Hui ZHAN, Jie-Min HU, Jun ZHANG

Automatic Target Recognition Laboratory,National University of Defense Technology,Changsha, China
*chenshiqi12@nudt.edu.cn. zhanrh@nudt.edu.cn,*hujiemindawang@163*.com,zhj64068*@sina.com

Abstract—Recent breakthroughs in algorithms related to deep convolutional neural networks (DCNN) have stimulated the development of various of signal processing approaches, where the specific application of Automatic Target Recognition (ATR) using Synthetic Aperture Radar (SAR) data has spurred widely attention. Inspired by the more efficient distributed training such as inception architecture and residual network, a new feature fusion structure which jointly exploits all the merits of each version is proposed to reduce the data dimensions and the complexity of computation. The detailed procedure presented in this paper consists of the fused features, which make the representation of SAR images more distinguishable after the extraction of a set of features from DCNN, followed by a trainable classifier. In particular, the obtained results on the 10-class benchmark data set demonstrate that the presented architecture can achieve remarkable classification performance to the current state-of-the-art methods.

1 Introduction

As a kind of active microwave imaging radar with all-day, all-weather and long-range detection capabilities, synthetic aperture radar (SAR) has occupied a leading role in areas such as early warning, surveillance and guidance, of which the most extensive application is in automatic target recognition (ATR).

With the emergence of well-performed classifiers such as support vector machine (SVM) [1], k-nearest neighbor (KNN) [2] and AdaBoost [3], machine learning technology has attracted much attention in SAR ATR studies. However, most of the work in machine learning approaches focus on designing a set of feature extractors. In SAR ATR field, the existing technique lacks the ability of extracting effective features and fusing them. Recently, a newly-developing learning method called convolutional neural networks (CNN) have been successfully applied to many large scale visual recognition tasks such as image classification [4]-[6], object detection [7][8]. Motivated by the model of mammal's visual system, CNNs consist of many hidden layers with the convolution operations which extract the features in the image and achieve state-of-the-arts results on the visual image data set such as ImageNet [4]. In this way, we consider employing CNN in SAR ATR field by means of designing reasonable network structure and optimizing training algorithms.

Generally, the architecture of CNNs can be interpreted as a two-stage procedure in image classification tasks. Unlike the previous methods which heavily rely on handcrafted features with human intervention, the training process in CNN automatically find appropriate features in search space, which are sent into trainable classifiers in the second stage, and thus to avoid the complexity of pre-processing and feature selection.

A variety of works have been done to achieve better performance in the past decades, however, it still remains a challenging task since modern advanced techniques require tens of thousands of examples to train adequately. Robert Wang [9] has augmented dataset to test the the adaptability on subsets under different displacement and rotation settings. Furthermore, the training based on very deep networks still faces problems for the reason that the stacking of non-linear transformations in typical feed-forward network generally result in poor propagation of activations as well as vanishing gradients. Hence it remains necessary to modify the architecture of deep feed-forward networks.

Owing to the background speckles all over the SAR images and the irregular distributions of strong scattering centers, SAR ATR is of great complexity especially when the networks get deeper. To enable the feasibility of training deeper network with limited data, we adopt CNN architectures which fuse the features extracted from different branches and these structures tends to perform well in maximally exploiting the input data and improving classification accuracy.

The remainder of this paper is organized as follows. Section II discusses the basic components of the CNN network as well as the training method. Section III gives an introduction into the two feature fusion categories and how they are constructed in the proposed networks. Experimental results conducted from SAR imagery are analyzed in Section IV to identify the desirable performance of the novel network architectures. Finally, we summarize this paper.

2 Structure And Training of Cnns

Among various of deep learning methods, CNN is an effective model which considers the dependencies between pixels in an image. The CNN consists of consecutive layers of neurons with learnable weights and bias, where the concept of perceptive field is applied. In this scenario, we will give a detailed description of the basic operation modules in CNN.

2.1Convolution Layer

Each CNN layer can be visualized as a 1-dimensional (1D) stack of 2-dimensional (2D) filters. Meanwhile, a subset of a layer's neuron works as receptive fields for the layers next to it.

The fundamental operation underpinning a CNN is the 2D convolution. In the convolution layer, we define the input feature maps of the previous layer as $O_m^{(l-1)}(m=1,...M)$ which are connected to the output feature maps $O_n^{(l)}(n=1,...N)$. $O_m^{(l-1)}(x,y)$ and $O_n^{(l)}(x,y)$ represent the unit of the *m*th input feature map and the *n*th output feature map at position (x,y) respectively. Each unit in the convolution layer is computed as:

$$O_n^{(l)}(x,y)=\sigma(\sum_{m=1}^{M}\sum_{p,q=0}^{F-1}k_{nm}^{(l)}(p,q)O_m^{(l-1)}(x-p,y-q)+b_n^{(l)}) \quad (1)$$

where the convolution kernel $k_{nm}^{(l)}(p,q)$ denotes the trainable filters, while other terms $b_n^{(l)}$ and σ represent the bias of the *n*th output feature map and the nonlinear activation function.

2.2Activation Function

To form a highly nonlinear mapping relationship between the input image and the output label, a proper activation function is added at each convolution layer [10]. Since a hyperbolic tangent function or a sigmoid function [11] may get trapped into saturating nonlinearities during the gradient propagation process, a piecewise linear function which does not suffer from vanishing gradients is introduced to make the computation more efficient. The rectified linear unit (ReLU) activation function [12] is defined as:

$$f(x)=\max(0,x) \quad (2)$$

2.3Pooling Layer

Pooling layers are introduced in CNN to reduce the resolution of images between successive layers. By either using averaging or choosing maximal value neuron in the group, pooling layers largely reduce the amount of computation through lessening the parameters. These layers also make CNN better at translation, shift and distortion invariance. We assume that each feature map in a convolution layer corresponds to a single map in the associated pooling layer. For example, the max pooling operation [13] is defined as

$$O_m^{(l+1)}(x,y)=\max_{p,q=0,...K-1}O_m^{(l)}(x\cdot s+p,y\cdot s+q) \quad (3)$$

where K is the pooling size, and s is the stride which indicates the internals between adjacent pooling windows.

2.4Error Function

As a training method for neural networks, the back propagation (BP) [14] algorithm in CNN uses the categorical cross entropy as the object function, which aims to measure the consistency between the prediction of network outputs and the truth labels.

Softmax layer [15] is a commonly used multi-classification layer in which the output probabilities over each class is computed at the final stage of the CNN architecture.

Given a training set $\{(x^{(i)},t^{(i)});i\in 1,...,N,t^{(i)}\in 1,...,T\}$, where $x^{(i)}$ is the *i*th input image, N is the number of examples, and $t^{(i)}$ refers to the target label among T classes. The prediction $y_j^{(i)}$ of *j*th class for *i*th input is passed through softmax function to yield a normalized probability distribution over classes. The softmax layer operation takes the form as [15]:

$$p_j^{(i)}=\frac{e^{y_j^{(i)}}}{\sum_{j=1}^{T}e^{y_j^{(i)}}} \quad (4)$$

We use the softmax output layer in conjunction with the categorical cross-entropy error function. With that as the background, the error function E is defined as:

$$\mathrm{E}=-\frac{1}{N}[\sum_{i=1}^{N}\sum_{j=1}^{T}1\{t^{(i)}=j\}\ln p_j^{(i)}] \quad (5)$$

2 Feature Fusion in Cnn

The combination of a variety of convolution layers and pooling layers plays an influential role in learning robust feature representations. Unlike natural image datasets, SAR images are more complex due to its complicated imaging mechanism. Even when the images are obtained from different azimuth angle, the consequent shapes of targets from the same type vary a lot. Furthermore, the speckling noise in SAR images makes it tougher to interpret each target.

Although CNN works as a good feature extractor, how to make use of multi-dimensional features and fuse them together still deserves exploration. To increase the diversity of extracted feature and understand the interactions between them, we formulate two patterns of feature fusion. Since the targets in SAR images may not be square shaped, use of convolution kernels sharing the same size are restricted to learn more adequate features, hence the inserted convolution kernels are modified as that with rectangular shape.

3.1Feature Fusion under Concatenation Pattern

Usually, the original convolution layer contains several square kernels, in this pattern we replace them with two convolution branches arranged by concatenation. This mode of feature fusion on convolution level is referred to as "conv-fusion" module. The concrete structure of its elements is demonstrated in Figure 1, where a represents the equivalent kernel size in standard CNN structure while n denotes the adjustable kernel size. Here, the convolution marked with S are same-padded while V signifies that are valid-padded.

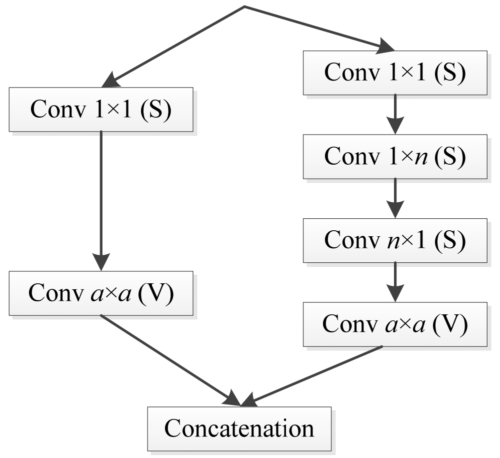

Figure 1. Conv-fusion module

Different from general convolution, S indicates that the size of the output grid is identical with that of input feature map. For the purpose of getting a same output size from two branches, the convolution operation under same mode is utilized to ensure the input and output dimension match. That is to say, when the feature map with size $m \times m$ is transferred into the module, the size of output feature map is $(m-a+1) \times (m-a+1)$.

In this module, the 1×1 convolutions placed before convolution operation of other sizes function as dimension reduction module [5], which allow for expanding the depth and width of CNN without increasing the computational complexity.

3.2Fusion under Summation Pattern

Deep networks stacked by several layers are usually tracked in degradation problem. Enlightened by the deep residual learning framework [16], we design the second pattern of feature fusion. Formally, the expected underlying mapping also known as unreferenced mapping is represented as F(x). H(x) denotes the residual mapping consisting of two consecutive convolution layers. When each convolution only computes corrective terms $H(x) = F(x) - x$ rather than the entire approximation F(x), the thought of shortcut connection which means skipping several layers is realized. By adopting the summation mechanism between layers, the information from previous layer flow more smoothly to the next layer without attenuation and the layers can learn the difference from the input feature map. Additionally, considering the ability of asymmetric kernels in extracting features with various scales and structures, we propose a new conception called asymmetric shortcut connection block (ASCB) as shown in Figure 2.

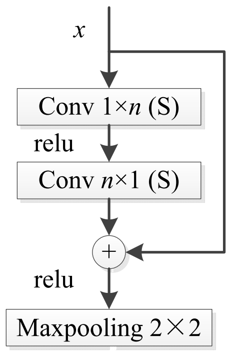

Figure 2. Asymmetric shortcut connection block

This means the untransformed input of previous layer can be propagated to the output and the feature maps between adjacent layer can be joined, hence making the deep network easier to be optimized and more efficient to be trained. To ensure that transformed output feature maps match the input size, we use two consecutive convolution operations under "same" mode.

Since the convolution kernel size differs in each layer with the deepening of networks, multi-scale features can be potentially extracted under the multi-level fusions, which intensify the links between features from adjacent layers and make them more adaptable to train deep networks.

3 Experiment Results

To design the appropriate number of feature maps and filter sizes used in convolutional layers as well as down sampling rates and numbers of stacking layers, several CNN architectures were tested. As the operations of forward and backward propagation are highly parallelizable, we utilize GPU processing based on deep CNN implementation to accelerate the testing process.

4.1Data Description

The CNN structure in this paper is applied to address the problem of SAR ATR for MSTAR dataset. The MSTAR benchmark data act as a standard data set to test and evaluate the performance of recognition algorithms. In our experiment, images for training are captured at 17 degree depression angle and that for testing are captured at 15 degree. The proposed algorithm is evaluated on the ten-target classification problem, and the corresponding number of images for training and testing are listed in Table I.

4.2Training Details

As has been presented in previous work, the implementation details play a decisive role in recognition performance. We aim to derive a set of best CNN structures both concerning low computation cost and high accuracy constraints towards the application of limited labelled SAR data. Except for some hyper parameters determined in CNNs, other details such as weight initialization, learning rate also count.

TABLE I. NUMBER OF TRAINING AND TESTING IMAGES FOR THE EXPERIMENTS

Target Types	No. Testing	No. Training
BMP2(9563)	195	233
BMP2(9566)	196	
BMP2(C21)	196	
T72(132)	196	232
T72(812)	195	
T72(S7)	191	
BTR70	196	233
BTR60	195	256
BRDM2	274	298
ZSU	274	299
T62	273	299
ZIL	274	299
2S1	274	299
D7	274	299
Total	3203	2747

To improve the training speed as well as prevent the network from becoming trapped in local minima, a set of fine-tuning methods have been applied to various of recognition tasks. Based on the gradient descent method, Adam [17] algorithm can dynamically adjust the updating amount of each parameter according to the moment estimation and thus achieve the efficiency and stability of CNN. In view of this, we chose the Adam technique as a substitute for normal mini-batch SGD using the given default value set in $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ [17].

Additionally, several subsets of training data are chosen stochastically from the whole data set as mini-batch and all the models are trained under a 50-element batch size.

As for weight initialization, the weights of all the layers are sent into an initializer [18] with uniform distribution $\left(-\sqrt{6/(fan\_in + fan\_out)}, \sqrt{6/(fan\_in + fan\_out)}\right)$, where fan_in and fan_out represents the number of input units and output units respectively.

4.3Experiments on Different Network Configurations

Here, we present the general layout of our baseline convolutional neural network (BCNN) and then describe the details of components used in BCNN. The schematic view of BCNN is depicted in Figure 3. To reduce the feature dimension, each convolution layer is followed by a pooling layer with the 2×2 pooling size and a stride of 2 pixels.

To avoid a fair amount of redundancy and spilling over of local information brought by relatively large receptive fields in each convolution layer, we choose convolution kernel size smaller than 7. In the structure, the fully connected layer is replaced by the global average pooling (GAP) layer since the correspondence between feature maps and categories is strengthened and no parameters need optimizing [19].

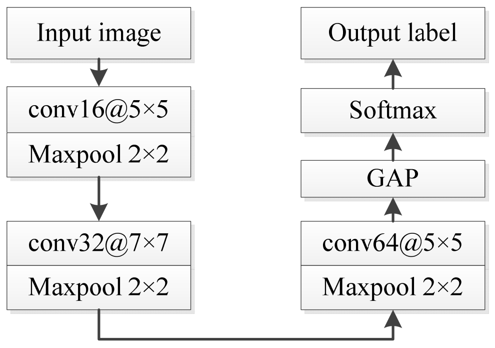

Figure 3. Overall architecture of BCNN (conv. (kernel depth) @ (kernel size))

For MSTAR dataset, the recognition accuracy on BCNN is obtained by the above hyper parameter setting. Next, the feature-fusion patterns are introduced into the BCNN and all these CNN-based methods share the same parameter setting except for different framework. The relevant composition of transformed conv-fusion CNN is listed in Table II.

TABLE II. COMPOSITION OF CONV-FUSION CNN

Layer name/Type	Output size	Filter size	Left Branch Filter size	Right Branch Filter size
Input image	64×64×1	/	/	/
Conv-fuse1	60×60×16	/	5×5	1×3 / 3×1
Maxpool1	30×30×16	2×2	/	/

Conv-fuse2	24×24×32		7×7	1×5 / 5×1
Maxpool2	12×12×32	2×2	/	/
Conv-fuse3	8×8×64		5×5	1×7 / 7×1
Maxpool3	4×4×64	2×2	/	/
GAP	64	2×2		

We have to mention that in the conv-fusion module, the number of feature channels from two branches are defaulted as the same. Hence, the feature map number of the merged convolution layer becomes twice.

As for the second pattern of feature fusion, an asymmetric shortcut connection block is inserted into the network between each convolution layer and pooling layer, with different asymmetric convolution set in each block. The whole network is called Module-residual CNN. Here we chose 1×3 / 3×1, 1×5 / 5×1, 1×7 / 7×1 kernel size in an ASCB and obtain an improvement in classification accuracy. Figure 4 is a flow chart of the proposed network.

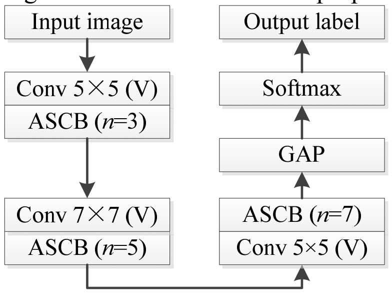

Figure 4. An Illustration of The Proposed Network

TABLE III. THE CONFUSION MATRIX UNDER CONCATENATION PATTERN

Target Types	Recognition Result										Accuracy (%)
	BMP2	T72	BTR70	BTR60	BRDM2	ZSU	T62	ZIL	2S1	D7	
BMP2(9563)	195	0	0	0	0	0	0	0	0	0	100.00
BMP2(9566)	195	0	0	0	1	0	0	0	0	0	99.49
BMP2(C21)	189	4	0	0	3	0	0	0	0	0	96.43
T72(132)	1	195	0	0	0	0	0	0	0	0	99.49
T72(812)	5	183	0	1	0	0	6	0	0	0	93.85
T72(S7)	8	177	0	1	0	0	4	0	1	0	92.67
BTR70	0	0	196	0	0	0	0	0	0	0	100.00
BTR60	0	0	7	188	0	0	0	0	0	0	96.41
BRDM2	1	0	0	0	272	0	0	1	0	0	99.27
ZSU	0	0	0	0	0	274	0	0	0	0	100.00
T62	0	1	0	0	0	1	271	0	0	0	99.27
ZIL	0	0	0	0	0	0	1	273	0	0	99.64
2S1	1	0	3	0	1	0	0	0	269	0	98.18
D7	0	0	0	0	0	0	0	0	0	274	100.00
Global Accuracy (%)					98.38						

TABLE IV. THE CONFUSION MATRIX UNDER SUMMATION PATTERN

Target Types	Recognition Result										Accuracy (%)
	BMP2	T72	BTR70	BTR60	BRDM2	ZSU	T62	ZIL	2S1	D7	
BMP2(9563)	195	0	0	0	0	0	0	0	0	0	100.00
BMP2(9566)	191	3	0	0	2	0	0	0	0	0	97.45
BMP2(C21)	189	2	2	1	1	0	0	0	1	0	96.43
T72(132)	0	196	0	0	0	0	0	0	0	0	100.00
T72(812)	2	190	0	0	0	0	3	0	0	0	97.44
T72(S7)	6	183	0	0	0	0	2	0	0	0	95.81
BTR70	0	0	194	1	1	0	0	0	0	0	98.98
BTR60	0	0	4	191	0	0	0	0	0	0	97.95
BRDM2	0	0	0	0	274	0	0	0	0	0	100.00
ZSU	0	0	0	0	0	274	0	0	0	0	100.00
T62	0	3	0	0	0	0	270	0	0	0	98.90
ZIL	0	0	0	0	0	1	0	273	0	0	99.64
2S1	0	2	0	0	1	0	3	0	268	0	97.81
D7	0	0	0	0	0	0	0	0	0	274	100.00
Global Accuracy (%)					98.72						

TABLE V. THE CONFUSION MATRIX OF UNDER DECISION-LEVEL FUSION

Target Types	Recognition Result										Accuracy (%)
	BMP2	T72	BTR70	BTR60	BRDM2	ZSU	T62	ZIL	2S1	D7	
BMP2(9563)	195	0	0	0	0	0	0	0	0	0	100.00
BMP2(9566)	195	0	0	0	0	1	0	0	0	0	99.49
BMP2(C21)	195	1	0	0	0	0	0	0	1	0	99.49
T72(132)	0	196	0	0	0	0	0	0	0	0	100.00
T72(812)	1	190	0	1	0	0	4	0	0	0	96.94
T72(S7)	4	187	0	0	0	0	0	0	0	0	97.90
BTR70	0	0	196	0	0	0	0	0	0	0	100.00
BTR60	0	0	3	192	0	0	0	0	0	0	98.46
BRDM2	0	0	0	0	274	0	0	1	0	0	100.00
ZSU	0	0	0	0	0	274	0	0	0	0	100.00
T62	0	0	0	0	0	0	273	0	0	0	100.00
ZIL	0	0	0	0	0	0	0	274	0	0	100.00
2S1	0	0	1	0	0	0	0	0	273	0	99.64
D7	0	0	0	0	0	0	0	0	0	274	100.00
Global Accuracy (%)	99.42										

To visually understand the intermediate state in the ASCB block and verify the reasonability of feature fusion, we display a set of hierarchical features learned from CNN. For instance, a BMP2 image is randomly chosen from the training dataset then the forward propagation process in each stage is visualized. In Figure 5, several output grids of size 8×8 in the second ASCB are displayed.

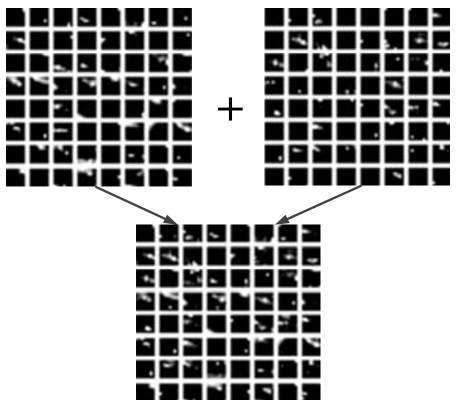

Figure 5. Visualization of 8×8 feature map in ASCB (n=7)

We can see from the figure 5 that the loss of visual information is suppressed after the last output map is fed through the block. By exploiting the complementarity of each map aggregated to the next stage, our proposed CNN structure can learn more specific and abstract features.

The recognition results for different CNN structures are shown in Table III, IV, V, where each row denotes the real target type and each column denotes the predictive label.

As we can conclude from the table, the overall accuracy of the two proposed architectures can reach 98.38% and 98.72% respectively. Whilst variations exist in testing set, the lowest accuracy is 95.81% for T72 in the second pattern feature fusion. For the targets without variants, the totally precise accuracy obtained in BMP2, T72, BRDM2, ZSU and D7 illustrates the effectiveness of our algorithms.

Since each CNN architecture extracts global and local features transmitted into classifier, we integrate three network architectures and make decision-level fusion of which we choose the largest posterior probability, a decision-level recognition rate of 99.42% can be obtained.

4.4Comparison with Previous Methods

To evaluate the feature extraction and fusion capability of the proposed CNN, we compare it with some widely cited methods including support vector machine (SVM) [20], Adaptive Boosting technique [21], polynomial kernel PCA (KPCA) [22], sparse representation based on monogenic signal (MSRC) [23] and Iterative graph thickening model of image representations (IGT) [24]. The recognition accuracies of these algorithms are shown in Table VI. Experiments conducted by Jun Ding [25] also suggest a typical CNN architecture but comparatively lower classification rate with data augmentation for this method.

TABLE VI. CLASSIFICATION RESULTS OF OTHER METHODS

Method	Accuracy (%)
SVM [20]	86.73
Adaptive Boosting [21]	92.70
KPCA [22]	92.67
MSRC [23]	93.66
IGT [24]	95.00
CNN with data augmentation [25]	93.16
BCNN	97.39
Conv-fusion CNN	98.38
Module-residual CNN	98.72

Decision fusion CNN	99.42

It is desirable that the feature fusion CNNs show the ability of classifying ten-class targets regardless of the existence of variants. Meanwhile, neither increasing the width of network nor the complexity of architectures, the excellent combination of high-level feature representations and learning of potential invariant feature can achieve particularly good performance even when the labelled training data is limited.

4 Conclusion

Understanding the multi-scale and hierarchical features learned by CNN and then dig thoroughly into the discriminative features help us finish target recognition task. In this paper, we first constructed novel CNN architectures which contain two independent modules called "conv-fusion" module and asymmetric shortcut connection block, then fined tune the hyper parameters in the deep CNN in the second stage and finally applied them to address the problem of recognition using SAR images. Experimental results indicate that the proposed network can gain superior performance compared with other state-of-the-art methods by extracting forceful feature representations. Furthermore, the CNN approach presented in this paper can be revised or extended to other practical applications and thus provide insightful points in the recognition tasks aimed at small dataset.

Acknowledgment

The research work is supported by the National Natural Science Foundation of China under grant No.61471370.

References

[1] Q. Zhao and J. C. Principe 2001. "Support vector machines for sar automatic target recognition," IEEE Transactions on Aerospace & Electronic Systems, vol. 37, no. 2, pp. 643–654.

[2] Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. 2003. Knn model-based approach in classification. Lecture Notes in Computer Science, 2888, 986-996.

[3] Y. Sun, Z. Liu, S. Todorovic, and J. Li 2007. "Adaptive boosting for sar automatic target recognition," IEEE Transactions on Aerospace & Electronic Systems, vol. 43, no. 1, pp. 112–125.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton 2012. "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., pp. 1106–1114.

[5] C. Szegedy et al. 2015. "Going deeper with convolutions," in Proc. IEEE Computer Vision Pattern Recognition, Boston, MA, USA, Jun. 8–10, pp. 1–9.

[6] K. Simonyan and A. Zisserman 2015. "Very deep convolutional networks for large-scale image recognition," presented at the Int. Conf. Learning Representations. [Online]. Available: http://arxiv.org/abs/1409.1556

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Computer Vision Pattern Recognition. , pp. 580–587.

[8] X. Chen, S. Xiang, C. Liu, and C. Pan 2014. "Vehicle detection in satellite images by hybrid deep convolutional neural networks," IEEE Geoscience Remote Sense Letter, vol. 11, no. 10, pp. 1797–1801.

[9] Du, K., Deng, Y., Wang, R., Zhao, T., & Li, N 2016. Sar atr based on displacement- and rotation-insensitive cnn. Remote Sensing Letters, 7(9), 895-904.

[10] Chen, S., Wang, H., Xu, F., & Jin, Y. Q. 2016. Target classification using the deep convolutional networks for sar images. IEEE Transactions on Geoscience & Remote Sensing, 54(8), 1-12.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner 1998. "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324.

[12] V. Nair and G. E. Hinton 2010. "Rectified linear units improve restricted boltzmann machines," in ICML.

[13] Y. LeCun, K. Kavukcuoglu, and C. Farabet 2010. "Convolutional networks and applications in vision," in Proc. IEEE International Symposium Circuits System, pp. 253–256.

[14] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson 1990. "Handwritten digit recognition with a backpropagation network," in Advances in Neural Information Processing Systems, pp. 396–404.

[15] C. M. Bishop 2006 Pattern Recognition and Machine Learning. New York, NY, USA: Springer-Verlag.

[16] He, K., Zhang, X., Ren, S., & Sun, J. 2015. Deep residual learning for image recognition. 770-778. .

[17] D. P. Kingma and J. Ba 2014. "Adam: A method for stochastic optimization," Computer Science.

[18] K. He, X. Zhang, S. Ren, and J. Sun 2015. "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in Proc. Int. Conf. Comput. Vis., pp. 1026–1034.

[19] Lin, M., Chen, Q., & Yan, S. 2013. Network in network. Computer Science.

[20] Bengio Y, Nicolas Boulanger-Lewandowski, Razvan Pascanu 2013. "Advances in optimizing recurrent networks," IEEE International Conference on Acoustics, Speech and Signal Processing, 8624-8628.

[21] Sun, Y., Liu, Z., Todorovic, S., and Li, J. 2007. "Adaptive boosting for sar automatic target recognition," Aerospace and Electronic Systems, IEEE Transactions on 43, 112–125.

[22] Mishra, A. K., & Motaung, T. 2015. Application of linear and nonlinear PCA to SAR ATR. Radioelektronika.IEEE.pp.349-354..

[23] D. Ganggang, W. Na, and K. Gangyao 2014. "Sparse representation of monogenic signal: With application to target recognition in sar images," vol. 21, no. 8, pp. 952–956.

[24] S. Umamahesh, M. Vishal, and R. Raghu, G. 2014. “Sar automatic target recognition using discriminative graphical models,” vol. 50, pp. 591–606.

[25] D. Jun, C. Bo, L. Hongwei, and H. Mengyuan 2016. “Convolutional neural network with data augmentation for sar target recognition,” in IEEE Geoscience and remote sensing letters, vol. 13, no. 3, pp. 364–368.