SAR Automatic Target Recognition Based on Deep Convolutional Neural Networks

Article *in* DEStech Transactions on Computer Science and Engineering · April 2017

DOI: 10.12783/dtcse/aita2016/7564

4 authors, including:

Zhan Ronghui
National University of Defense Technology
86 PUBLICATIONS **1,249** CITATIONS
SEE PROFILE

Jun Yang
Xidian University
323 PUBLICATIONS **4,254** CITATIONS
SEE PROFILE

2016 International Conference on Artificial Intelligence: Techniques and Applications (AITA 2016)
ISBN: 978-1-60595-389-2

SAR Automatic Target Recognition Based on Deep Convolutional Neural Networks

Rong-hui ZHAN*, Zhuang-zhuang TIAN, Jie-min HU and Jun ZHANG

Science and Technology on Automatic Target Recognition Laboratory, National University of Defense Technology, Changsha 410073, China

*Corresponding author

Keywords: SAR image recognition, Convolutional neural networks, Deep learning, Error back-propagation.

Abstract. Deep convolutional neural networks (CNN) have recently proven extremely competitive in challenging visible light image and speech recognition tasks. The goal of the present study is to explore the application of automatically learned convolutional network features to radar target recognition. Specifically, a two-stage convolutional-pooling network architecture is designed and error back-propagation algorithm with momentum acceleration strategy is used to learn the network weights in a supervised fashion. The effectiveness of the proposed method is assessed by SAR image classification tasks on the standard benchmark of MSTAR (Moving and Stationary Target Acquisition and recognition) database. Our experiments show the presented method has achieved encouraging results with a correct recognition rate of 95.64% for three classes of targets and 92.86% for ten classes of targets.

Introduction

Synthetic aperture radar (SAR) is a kind of coherently imaging radar with both high range and azimuth resolutions. Unlike optical sensor which cannot perform well in poor weather conditions, SAR is a kind radar system with all-time and all-weather operating ability, and it has played a great role in military and civil applications, such as reconnaissance, navigation, guidance, remote sensing, and resource exploration etc. With the development of SAR, the technology for images interpretation has become an urgent requirement. Since searching and discriminating targets of interest manually from massive SAR images is a very time-consuming and difficult task, the automatic target recognition (ATR) methods have to be taken into consideration.

However, for the special mechanism of backscatter imaging, the detailed structure of the target in the obtained SAR image will be covered with the speckle noise (resulting from the background), which will inevitably lead to a reduction of image quality. Moreover, SAR image is very sensitive to the variation of target pose, and will vary quickly and abruptly with small change in aspect angles. Therefore, SAR ATR is a challenging task and has become an important research topic for many applications.

Up to now, a variety of SAR ATR methods have been proposed in the literature and these methods are generally implemented via two processing steps, i.e., feature extraction and classification. There are many algorithms for SAR target feature extraction, such as principal component analysis (PCA) [1-3], independent component analysis (ICA) [4-6], Hu invariant moments [7], non-negative matrix factorization (NMF) [8-10] and sparse representation (SR) [11-13], etc. As for the target classification, theoretically any classification algorithms applied to pattern recognition can also be extended to the application of SAR image. The commonly used SAR image classification algorithms include mean square error (MSE) [14], hidden Markov models (HMM) [15], neural network (NN) [16, 17], support vector machine (SVM) [18, 19], Adaboost [20] and so on.

Over the past decades, great efforts for the purpose of ATR performance improvement have been paid to two aspects: 1) extracting more discriminative features; and 2) designing more effective classifier to distinguish the features obtained from different target classes. Contrast to the traditionally separate treatment of the ATR with feature extraction and classification, the recently developed deep learning architecture [21-23] can automatically learn representations or features from raw data, while jointly performing discrimination to improve the accuracy on different pattern recognition problems.

The deep learning models have been receiving increased attention in recent years, and the specific architecture with convolution and pooling (also referred to as deep convolutional neural network, CNN) is found to be highly effective and has been commonly used in computer vision [24-26] and speech recognition [27-28].

In view of the good adaptation of deep CNN to the general problem of pattern recognition, in this paper we propose to introduce the CNN into application of SAR ATR. Specifically, a two-stage convolutional network architecture is designed and used to automatically learn features from SAR images and classify them into different classes. The presented deep learning algorithm is tested on the benchmark MSTAR (moving and stationary target acquisition and recognition) database, and the results show that the new method is effective for SAR image ATR and involves no handcrafted feature extraction step.

Framework of the Deep CNN

The design of deep neural network architecture in this paper is to achieve high recognition accuracy for the target classification tasks. The overall framework design is illustrated in Fig. 1, where *C* represents convolutional layer (also referred to as feature extraction layer). The input of each neuron in *C* layer is connected with the receptive field of the previous layer and to extract the local feature. There are multiple feature maps in the *C* layer and each individual feature map represents unique target feature. For each feature map, the network weights are shared by using the same convolutional kernel, and different feature maps have different kernels. Thus, the extracted local features with rotation and shift invariant property are stored in *C* layer. *S* represents subsampling layer, where the features extracted from *C* layer are down-sampled while preserving the scale invariant of the features. In *S* layer, only the operation of rescale mapping is used, and the number of weights that need to be trained is small, so the computation is quite simple. The last layer of CNN is the fully connected layer, which combines all the pixel units of S_2 into a 1D output vector. At the output, each unit of the output vector corresponds to a specific class label.

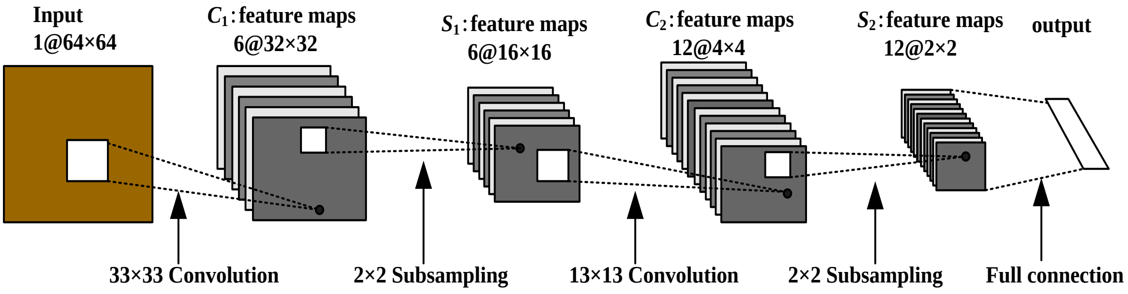

Figure 1. The proposed deep convolutional network architecture.

Convolutional Layer

Convolutional neuron layer is the key component of CNN. In image classification task, one or more 2D matrices (or channels) are treated as the input to the convolutional layer and multiple 2D matrices are generated as the output. The number of input and output matrices may be different. The process to compute a single output matrix is defined as:

$$O_j^{(l)}=f\left(\sum_{i=1}^{N} I_i^{(l\text{-}1)} * w_{i,j}^{(l)} + B_j^{(l)}\right) \tag{1}$$

where I_i is the input matrix, l is the index of neuron layers, $w_{i,j}$ is the trainable kernel matrix, $*$ represents convolution operator, B_j is the bias value added to each element of the resulting matrix, f is the nonlinear activation function applied to each element of the previous matrix to produce one output matrix O_j.

The most widely used activation functions are sigmoid function $f(x)=1/(1+e^{-x})$ and hyperbolic tangent function $f(x)=\tanh(x)$. Both sigmoid and hyperbolic tangent are saturating nonlinear functions that the output gradient drops close to zero as the input increases. In this paper, only the sigmoid function is considered.

Feature Pooling Layer

In the pooling layer, each feature map is treated separately and the operation of pooling plays an important role in CNN for feature dimension reduction. In order to reduce the number of output neurons in the convolutional layer, pooling algorithms include max-pooling and average-pooling. In this paper, the average-pooling algorithm with 2×2 kernel size computes the average values over a neighborhood in each feature map. This results in a reduced-resolution output feature map which is robust to small variations of the features in the previous layer.

Training Using Random Patches

The training method considered here is stochastic gradient descent algorithm, which computes the gradients by the back-propagation method. The details of the procedure are given as follows.

Back-Propagation Details

The main idea of back-propagation algorithm is to iteratively adjust the weights to minimize an error function

$$E=\frac{1}{2}\|d-y\|^2 \tag{2}$$

where d and y represent the desired output and the actual output of the network, respectively.

Recall that in (1), each convolutional layer has a number of units (or neurons), each of which takes all outputs of the lower layer as input, multiplies them by network weight, sums the result and passes it through a nonlinear activation function (sigmoid function). For simplicity of notation, this computation can be represented in a more compact vector form as:

$$O_j^{(l)}=f(O_j^{(l-1)}\cdot \mathrm{w}_j^{(l)}) \tag{3}$$

where the bias term is absorbed in the column weight vector by expanding the vector with an extra dimension of 1. Furthermore, all neuron activations in each layer can be represented in the following matrix form:

$$O^{(l)}=f(O^{(l-1)}\cdot \mathrm{W}^{(l)})\ (l=1,2,\cdots,L-1) \tag{4}$$

where $\mathrm{W}^{(l)}$ denotes the weight matrix of the l-th layer, with jth column $\mathrm{w}_j^{(l)}$ for any j.

The first (bottom) layer of the deep CNN is the input layer and the topmost (last) layer is the output layer. For output layer, because no nonlinear function is used, the equation (3) is reduced to

$$y_j=O_j^{(L)}=O_j^{(L-1)}\cdot \mathrm{w}_j^{(L)}. \tag{5}$$

The derivative of E with respect to each weight matrix $\mathrm{W}^{(l)}$ can be efficiently computed based on the well known error back-propagation method. If we use the stochastic gradient descent algorithm to minimize the objective function, the weight matrix update for each training sample can be computed as

$$\Delta \mathrm{W}^{(l)}=\alpha\cdot(O^{(l-1)})^{T}\cdot e^{(l)}\ \ (l=1,2,\cdots,L) \tag{6}$$

where α is the learning rate, and the error signal vector in the l–th layer $e^{(l)}$ is computed backwards from the sigmoid hidden unit as follows:

$$e^{(L)}=d-y \tag{7}$$

$$e^{(l)}=(e^{(l+1)}(\mathrm{W}^{(l+1)})^{\mathrm{T}})\cdot O^{(l)}\cdot(1-O^{(l)})\ \ (l=L-1,\cdots,2,1) \tag{8}$$

The overall update equation can now be written as:

$$\mathrm{W}_{t+1}^{(l)}=\mathrm{W}_{t}^{(l)}+\Delta \mathrm{W}^{(l)} \tag{9}$$

where t denotes the time index of iteration.

Fast Learning Strategy

Strategy for speeding up and stabilizing neural network training are considered in this paper, including batch learning, momentum and weight decay. When batch learning strategy is applied to improve learning speed and accuracy, multiple input samples (5~10 for instance) are formed as a batch to perform one single back propagation update, instead of updating the connection weights using only a single sample for each epoch.

To further speed up the learning, momentum weight updating is applied and the weight increment is now updated by:

$$\Delta \mathrm{W}_{t+1}^{(l)}=-\ \alpha\frac{\partial E}{\partial \mathrm{W}_{t}^{(l)}}+\eta\Delta \mathrm{W}_{t}^{(l)} \tag{10}$$

where $\partial E/\partial \mathrm{W}$ denotes the error gradient with respect to the weight vector, η is called the momentum parameter.

The momentum weight update term helps to speed up learning. In this following experiment, we choose the learning rate $\alpha=0.99$, the momentum rate $\eta=0.01$ for all layers. These learning parameters are selected through experimental tests.

Experiments and Analysis

Dataset Description

The experiment data set was collected by the Sandia National Laboratory (SNL) SAR sensor platform operating at X-band. The collection was jointly sponsored by Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory as part of the Moving and Stationary Target Acquisition and recognition (MSTAR) program [29-30]. The public released datasets include 10 different types of ground military targets (BMP2, BTR70, T72, BTR60, 2S1, BRDM2, D7, T62, ZIL131, ZSU23/4). For each target, images are captured at two different depression angles 15deg and 17deg over all the 360deg aspect angles with 1ft×1ft resolution.

In the experiments, all the images are cropped by extracting 64×64 patches from the center of the samples. The data collected at 17deg depression angles are used for training, and the data collected at 15deg depression angles are used for testing.

Experiment Results

The proposed algorithm is tested on two classification experiments. The first one is classifying three classes (i.e., BMP2, BTR70 and T72) of targets from the MSTAR database. The details of training and testing data set are shown in Table 1.

Table 1. Training set and testing samples for three classes of targets.

Training set	Sample number	Testing set	Sample number
		BMP2_c21	196
BMP2_c21	233	BMP2_9563	195
		BMP2_9566	196
BTR70_c71	233	BTR70_c71	196
		T72_132	196
T72_132	232	T72_812	195
		T72_s7	191

Since BMP2 and T72 contain three variants, respectively, i.e., BMP2 (sn_9563, sn_9566 and sn_c21), T72 (sn_132, sn_812 and sn_s7), we only select sn_9563, sn_c71 and sn_132 from the target classes of BMP2, BTR70 and T72 as the training set. In the testing set, however, all the three classes of targets including their variants are considered, which means that we have 698 images in the training set and 1365 images in the testing set.

In this paper, the recognition performance is comprehensively measured by probability of correct classification (PCC), average PCC (A-PCC) and over-all PCC (OA-PCC), which are defined as

$$PCC(k)=\frac{N_c(k)}{N_t(k)} \quad k=1,2,\cdots,K \tag{11}$$

$$A\text{-}PCC=\frac{1}{K}\sum\nolimits_k PCC(k) \tag{12}$$

$$OA\text{-}PCC=\frac{\sum\nolimits_k N_c(k)}{\sum\nolimits_k N_t(k)} \tag{13}$$

where K is the total class number of the target in the testing set, $N_t(k)$ and $N_c(k)$ represent the total and correctly classified number of sample images of the k-th target class, respectively.

The experiment results are shown in Table 2 and Table 3. For the purpose of performance comparison, the results from two other state-of-the-art methods, i.e., support vector machine classifier (SVM-C) and sparse representation classifier (SR-C), are also given in the tables. It is worth noting that for the classifiers SVM-C and SR-C, the same parameter setup as that in [12] is used in our experiments. For the detailed algorithm implementation of the two classifiers, the reader is referred to reference [12].

As can be seen from the above tables, all the three methods have high recognition performance, and especially for the target class BTR70, the PCCs of all the methods approach to or reach 100%. The reason that BTR70 achieves the best recognition accuracy is that there are no variants in the testing samples for this class of target. Clearly, the CNN outperforms SR-C and SVM-C both in mean and in standard error of the PCC. These results indicate that the CNN method is more effective in classifying the three classes of targets, and an important reason behind this fact is that more discriminative 2D features are obtained in CNN method, rather than the 1D features extracted in SR-C and SVM-C methods. Furthermore, the target features are automatically learned in CNN, thus the processing step of handcrafted feature extraction is avoided.

To further test the performance of the recognition methods, ten classes of targets (including the variants) in the MSTAR database are used in the second experiment. The details of training and testing data set are shown in Table 4, and we totally have 2747 training samples and 3203 testing samples.

Table 2. Confusion matrix for the testing set with three classes of targets.

Method	Testing image	Prediction result		
		BMP2	BTR70	T72
	BMP2	**556**	11	20
CNN	BTR70	0	**196**	0
	T72	44	2	**536**
	BMP2	**548**	17	22
SR-C	BTR70	0	**196**	0
	T72	80	11	**491**
	BMP2	**529**	15	43
SVM-C	BTR70	0	**195**	1
	T72	15	9	**558**

Table 3. Recognition accuracy for three classes of targets (%).

		method		
		SR-C	SVM-C	CNN
	BMP2	93.35	90.11	94.72
Testing classes	BTR70	100.0	99.48	100.0
	T72	84.36	95.87	92.10
Over-all PCC		90.47	93.92	**94.36**
Average PCC		92.57	95.15	**95.64**
Std		7.84	4.27	**4.02**

Table 4. Training set and testing samples for ten classes of targets.

Class ID	Training set	Sample number	Testing set	Sample number

1	BMP2(sn-c9563)	233	BMP2(sn-c9563)
			BMP2(sn-c9565)
			BMP2(sn-c21)
				195
				196
				196

2	BTR70	233	BTR70	196

3	T72(sn-132)	232	T72(sn-132)
			T72(sn-812)
			T72(sn-s7)
				196
				195
				191

4	BTR60	256	BTR60	195
5	2S1	299	2S1	274
6	BRDM2	298	BRDM2	274
7	D7	299	D7	274
8	T62	299	T62	273
9	ZIL131	299	ZIL131	274
10	ZSU23/4	299	ZSU23/4	274

The experiment results are shown in Table 5. It can be seen from tables that for most target classes, the CNN method can obtain good recognition performance with an average PCC higher than 90%. However, for the classes of BMP2 and T72, obvious performance deterioration is observed in PCC compared with the first experiment where only three target classes are considered. Meanwhile, the classes 2S1 and T62 also have relatively low PCCs. The results are consistent with the fact that the classes BMP2, T72, 2S1 and T62 have high similarity (all of them have long guns), which makes it more difficult to correctly distinguish the targets from each other.

Table 5. Confusion matrix and recognition accuracy for CNN method with ten classes of testing targets.

		Output Class									
		1	2	3	4	5	6	7	8	9	10
Target Class	1	**514**	15	30	13	3	1	8	2	1	0
	2	0	**193**	0	1	2	0	0	0	0	0
	3	4	7	**498**	5	14	0	5	48	0	1
	4	0	2	1	**188**	1	0	2	0	1	0
	5	7	4	11	3	**226**	4	7	12	0	0
	6	2	0	0	1	0	**257**	14	0	0	0
	7	0	0	0	0	0	0	**271**	0	0	3
	8	0	0	5	2	1	0	6	**251**	2	6
	9	0	0	0	0	0	1	5	0	**258**	10
	10	0	0	0	0	0	0	2	0	0	**272**
	PCC	87.56	98.47	85.57	96.41	82.48	93.80	98.91	91.94	94.16	99.27
Average PCC						92.86					
Over-all PCC						91.41					
Std						5.91					

Conclusion

In this paper, we have proposed to extend the deep CNN to the application of radar target recognition. Specifically, an effective network architecture with two-stage convolutional kernel and average-pooling strategy is designed to automatically learn the useful features from SAR images. The proposed CNN is tested on the MSTAR dataset, and an average PCC of 95.64% and 92.86% are achieved on the classification task for three and ten classes of targets with variants, respectively. The experiment results indicate the great potential of the CNN in the target recognition of SAR image. For the future work, the proposed method will be further extended to multi-view SAR ATR application.

Acknowledgement

The research work was financially supported by the National Natural Science Foundation of China under grant No.61471370 and No.61401479.

References

[1] V.V. Chamundeeswari, D. Singh, K. Singh, An analysis of texture measures in PCA-based unsupervised classification of SAR images, IEEE Geoscience and Remote Sensing Letters. 6 (2009) 214–218.

[2] L.P. Hu, X.Y. Xing, SAR target feature extraction and recognition based multilinear principal component analysis, Proceedings of SPIE. 9301 (2014) 93010L-1–93010L-6.

[3] X.G. Lu, P. Han, R.B. Wu, Two-dimensional PCA for SAR automatic target recognition, Proceedings of the 1st Asian and Pacific Conference on Synthetic Aperture Radar (2007) 513-516.

[4] H. Wang, Y. Pi, G. Liu, et al, Application of ICA for the enhancement and classification of polarimetric SAR images, Journal of Remote Sensing. 29 (2008) 1649-1663.

[5] J. Ji, Robust approach to independent component analysis for SAR image analysis, IET Image Processing. 6 (2012) 284-291.

[6] M. Tao, F. Zhou, Y. Liu, Z. Zhang, Tensorial independent component analysis-based feature extraction for polarimetric SAR data classification, IEEE Transactions on Geoscience and Remote Sensing. 53 (2015) 2481-2495.

[7] Y. Fu, M. Wang, C.Q. Zhang, SAR image target recognition based on Hu invariant moments and SVM, Proceedings of 2009 Fifth International Conference on Information Assurance and Security (2009) 585-588.

[8] R.H. Huan, Y. Pan, K.J. Mao, SAR image target recognition based on NMF feature extraction and Bayesian decision fusion, Proceedings of 2010 Second IITA International Conference on Geoscience and Remote Sensing (2010) 496-499.

[9] R. Vahid, S. Umamahesh, M. Vishal, SAR automatic target recognition via non-negative matrix approximations, Proceedings of SPIE. 8391 (2012) 83910M-2–83910M-6.

[10] Z.Y. Cui, Z.J. Cao, J.Y. Yang, et al, SAR target recognition using nonnegative matrix factorization with L1/2 Constraint, Proceeding of 2014 IEEE Radar Conference (2014) 382-386.

[11] J. Thiagarajan, K. Ramamurthy, P. Knee, Sparse representation for automatic target classification in SAR images, Proceeding of International Symposium on Communication, Control and Signal Processing (2010) 1-4.

[12] H.C. Liu, S.T. Li, Decision fusion of sparse representation and support vector machine for SAR image target recognition, Neurocomputing. 113 (2013) 97-104.

[13] G.G. Dong, N. Wang, G.Y. Kuang, Sparse representation of monogenic signal: with application to target recognition in SAR images, IEEE Signal Processing Letters. 21 (2014) 952-956.

[14] L.M. Novak, G.J. Owirak, W.W. Irving, Performance of 10-and 20-target MSE classifiers, IEEE Transactions on Aerospace and Electronic System. 36 (2000) 1279-1289.

[15] N. Chanin, H.P. Quoc, M.M. Russell, et al, Hidden Markov modelling for SAR automatic target recognition, Proceedings of 1998 IEEE International Conference on Acoustics, Speech and Signal Processing. 2 (1998) 1061-1064.

[16] P. Han, S. Gao, Automatic target recognition of SAR images based on the fuzzy neural networks, Proceedings of 2010 2nd IEEE International Conference on Information Management and Engineering (2010) 102-104.

[17] Y.C. Tzeng, K.S. Chen, A fuzzy neural network to SAR image classification, IEEE Transactions on Geoscience and Remote Sensing. 36 (1998) 301-307.

[18] T. Wu, X. Chen, X.W. Ruang, et al, Study on SAR target recognition based on support vector machine, Proceedings of 2009 2nd Asian-Pacific Conference on Synthetic Aperture Radar (2009) 856-859.

[19] Q. Zhao, J.C. Principe, Support vector machine for SAR automatic target recognition, IEEE Transactions on Aerospace and Electronic Systems. 37 (2001) 643-654.

[20] K. Chen, Y.H. Li, X.J. Xu, et al, A modified AdaBoost algorithm with new discrimination features for high-resolution SAR targets recognition, IEICE Transactions on Information & Systems. E98-D (2015) 1871-1874.

[21] G. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science. 313 (2006) 504-507.

[22] L. Martin, K. Lars, L. Amy, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognition Letters. 42 (2014) 11-24.

[23] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Networks. 61 (2015) 85-117.

[24] Y. Lecun, K. Kavukcuoglu, C. Farabet, Convolutional networks and applications in vision, Proceedings of 2010 IEEE International Symposium on Circuits and Systems (2010) 253-256.

[25] Y. Dong, Y. Wu, Adaptive cascade deep convolutional neural networks for face alignment, Computer Standards & Interfaces. 42 (2015) 105-112.

[26] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems (2012) 1097-1105.

[27] G. Hinton, L. Deng, D. Yu, et al, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Processing Magazine. 29 (2012) 82-97.

[28] N.S. Tara, K. Brian, S. George, et al, Deep convolutional neural networks for large-scale speech tasks, Neural Networks. 64 (2015) 39-48.

[29] J.C. Mossing, T.D. Ross, An evaluation of SAR ATR algorithm performance sensitivity to MSTAR extended operating conditions, Proceeding of SPIE. 3370 (1998) 554-565.

[30] V. Velten, T. Ross, J. Mossing, et al, Standard SAR ATR evaluation experiments using the MSTAR public released data set, Proceeding of SPIE. 3370 (1998) 566-573.