Convolutional Neural Network With Data Augmentation for SAR Target Recognition

Jun Ding, Bo Chen, *Member, IEEE*, Hongwei Liu, *Member, IEEE*, and Mengyuan Huang

***Abstract*—Many methods have been proposed to improve the performance of synthetic aperture radar (SAR) target recognition but seldom consider the issues in real-world recognition systems, such as the invariance under target translation, the invariance under speckle variation in different observations, and the tolerance of pose missing in training data. In this letter, we investigate the capability of a deep convolutional neural network (CNN) combined with three types of data augmentation operations in SAR target recognition. Experimental results demonstrate the effectiveness and efficiency of the proposed method. The best performance is obtained by using the CNN trained by all types of augmentation operations, showing that it is a practical approach for target recognition in challenging conditions of target translation, random speckle noise, and missing pose.**

***Index Terms*—Convolutional neural network (CNN), data augmentation, feature extraction, synthetic aperture radar (SAR) image, target recognition.**

I. Introduction

TARGET recognition in synthetic aperture radar (SAR) images has many potential applications in military and homeland security, such as friend and foe identification, battlefield surveillance, and disaster relief. A number of works have been done in the past decade [1]–[7], but it still remains a highly challenging task. The key step of target recognition is the feature extraction, whose performance is affected by many factors. There are various feature extraction methods in the literature. Zhao *et al.* in [1] utilize the raw pixels of images as the input of the support vector machine (SVM) classifier. The features for recognition in [2] are the magnitudes of the 2-D DFT coefficients of the cropped images. Zhou *et al.* [3] adopt the physically relevant scattering center features for recognition, which are predicted from the global scatter center model. Park *et al.* [4] propose 12 discriminative features which are based on the projected length and the pixels of the target. In [5], Patnaik *et al.* correlate a test image with the minimum noise and correlation energy (MINACE) filters in frequency domain. Dong *et al.* [6] develop an approach based on sparse representation of the monogenic signal (MSRC). Carmine *et al.* [7] extract the pseudo-Zernike moments of multichannel SAR images for target recognition. However, all of the previous methods heavily rely on hand-designed features. If more situations are considered, such as invariance under target translation, invariance under speckle variation in different observations, and tolerance of pose missing in training data, it is necessary for the experts to translate these requirements into features or model representation, which is challenging in general.

Convolutional neural networks (CNNs) have been heavily used once in the 1990s, but then, they fell out of fashion. In 2012, Krizhevsky *et al.* [8] made CNNs regain focus by showing superior image classification accuracy on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) data set. Recently, deep CNNs have been successfully applied to many pattern recognition tasks, such as image classification [8]–[10], object detection [11], and vehicle detection [12]. For image classification, the CNNs are almost of the same two-stage architecture proposed by LeCun [13], while the differences are their configurations, the depth, the number of units, and the form of nonlinear functions. The first stage of CNNs acts as a feature extractor, which learns rich hierarchical features mainly using a convolutional operation, and the last stage is a multilayer perceptron classifier. Szegedy *et al.* [9] introduced a 22-layer CNN for classification; meanwhile, Simonyan *et al.* [10] design 16-layer and 19-layer deep CNNs for large image recognition. All of these successful works show the powerful capability of CNN in feature extraction or representation learning.

In this letter, we thus utilize CNN with domain-specific data augmentation operations for issues in SAR target recognition, such as translation of target, randomness of speckle noise in different observations, and lack of pose images in training data. To the best of our knowledge, no general methodology addresses on these three problems all together before.

II. CNN

Convolution neural networks are obtained by stacking one or more computational layers, which can be viewed as a transformation from the input map to the output map. Here, we give a brief definition of these building blocks and discuss the configurations of our CNN used in SAR target recognition.

A. Computational Layers of CNN

Let $x \in \mathbb{R}^{h\times w\times c}$ be a multichannel image, where each dimension represents the height, width, and number of channels.

Manuscript received June 18, 2015; revised October 7, 2015 and November 22, 2015; accepted December 23, 2015. Date of publication January 26, 2016; date of current version February 24, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61372132 and Grant 61201292, by the One Thousand Young Talents Program, by the New Century Excellent Talents in University under Grant NCET-13-0945, and by the National Natural Science Fund for Distinguished Young Scholars under Grant 61525105.

The authors are with the National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China, and also with the Collaborative Innovation Center of Information Sensing and Understanding, Xidian University, Xi'an 710071, China (e-mail: dingjun410@gmail.com; bchen@mail.xidian.edu.cn; hwliu@xidian.edu.cn; mengyuanhuang@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2015.2513754

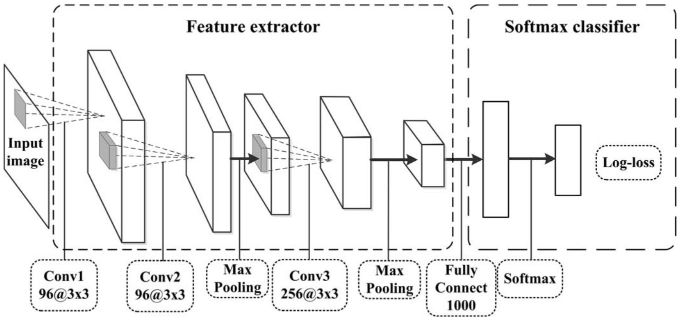

Fig. 1. Structure of the CNN used for SAR target recognition.

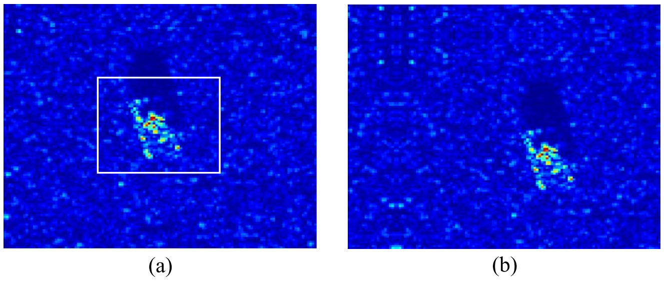

Fig. 2. Illustration of translation augmentation. (a) Image of the BMP2 target. (b) Translated image.

A block takes x and a set of parameters W as input and produces a new image $y \in \mathbb{R}^{h' \times w' \times c'}$ as output, i.e., $y = f(x, W)$.

1) Convolution: The convolution layer computes the convolution of the input x with a bank of filters $W \in \mathbb{R}^{\tilde{h} \times \tilde{w} \times \tilde{c} \times c'}$, also called kernels, and adds a bias $b \in \mathbb{R}^{c'}$. This computation can be formulated as

$$y_{i'j'k'} = f\left(b_{k'} + \sum_{i=1}^{\tilde{h}} \sum_{j=1}^{\tilde{w}} \sum_{d=1}^{c} W_{ijdk'} \times x_{i'+i,j'+j,d}\right) \tag{1}$$

where $f(\cdot)$ denotes a nonlinear function. Here, we use the rectified linear unit (ReLU) [14]

$$y_{ijk} = \max\{0, x_{ijk}\}. \tag{2}$$

2) Max-Pooling: The max-pooling layer computes the maximum response of each image channel in a $\tilde{h} \times \tilde{w}$ subwindow, which acts as a subsampling operation. It can be written as

$$y_{i'j'k} = \max_{1<i<\tilde{h}, 1<j<\tilde{w}} x_{i'+i,j'+j,k}. \tag{3}$$

3) Softmax: Softmax function is used to perform multiclass logistic regression

$$y_{ijk} = \frac{\exp(x_{ijk})}{\sum_{d=1}^{c} \exp(x_{ijd})}. \tag{4}$$

4) Log-Loss: Given the ground-truth class label t of input x, the log-loss is computed as

$$y = -\sum_{i=1}^{h} \sum_{j=1}^{w} \log x_{ijt}. \tag{5}$$

B. Configuration of Our CNN

Fig. 1 shows the structure of the CNN which we used in SAR target recognition. Here, CNN can be divided into two parts: feature extractor and softmax classifier. The former part consists of three convolution layers, each followed by a ReLU layer, and two max-pooling layers. The size of all convolutional filters is 3 × 3, and the number of units in the fully connected layer is 1000. We train the parameters of the CNN with stochastic gradient descent and back-propagation. It usually costs 1–2 days on our graphics processing unit (GPU) card for training and about several minutes for testing. The evaluation is conducted on a 2.7-GHz CPU with 16 GB of memory and a moderate GPU card.

III. Data Augmentation for SAR Target Recognition

Besides the lack of pose image in training data, we encounter more situations in SAR target recognition, such as translation of target and random speckle noising. To alleviate these problems, three types of data augmentation operations are developed.

A. Translation

A common step to cope with the misalignment is to perform registration between test data and training data based on some criteria, such as centroid centralization, which is also a difficult problem to solve, especially in complex scenarios. CNNs automatically provide some degree of small-scale translation invariance due to the convolution operation. When considering the shift of large-sized object like ground target, we synthesize the translation of training data to make the CNN "see" this type of variation explicitly. Given an image $I \in \mathbb{R}^{m \times n}$ of size $m \times n$, the translated image can be written as

$$I'_{ij} = I_{i+u,j+v} \tag{6}$$

where (u, v) is the shift coordinate and I' has the same size as I. Fig. 2 gives an illustration of BMP2 (infantry combat vehicle) images before and after translation. To guarantee that the target does not exceed the image boundary, a guard window is used. Here, we choose the size of the window to be 50 × 50 pixels [shown in white box in Fig. 2(a)].

B. Speckle Noising

SAR images suffer from the speckle noise due to the characteristic of the imaging system. Under simple assumption on single look images, the observed scene can be modeled with multiplicative noise model as follows:

$$I = s \times n \tag{7}$$

where I denotes the observed intensity and s is the underlying radar cross section of the scene and the speckle noise, which follows unit mean exponential distribution. To synthesize a new image, a median filter is applied on the observed image I to produce an estimate of s. Then, we pointwisely multiply the coarse-grained filtered image by random samples from

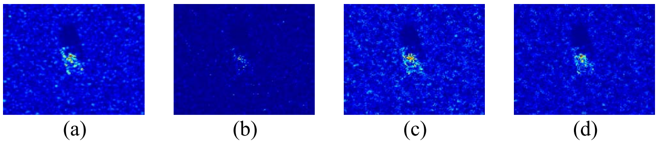

Fig. 3. Illustration of speckle noise augmentation. (a) Example image of the BMP2 target in the MSTAR database. (b) Noising the filtered image using exponential distribution. (c) Noising with truncated exponential distribution with $a = 0.5$. (d) Noising with truncated exponential distribution with $a = 1.5$.

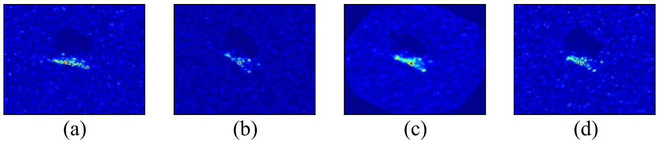

Fig. 4. Illustration of pose synthesis. (a) Example image of BMP2 at 284.492°. (b) Example image of BMP2 at 315.492°. (c) Synthesized image at 301.502° using (a) and (b). (d) Example image of BMP2 at 300.492°, which is the nearest to (c) in terms of azimuth angle.

exponential distribution. Fig. 3 shows some examples of the speckle noising augmentation. Due to the fact that the generation process results in the darkening of the image [see Fig. 3(b)], we replace the standard exponential distribution with the truncated exponential distribution, which imposes that the maximum intensity of noise samples does not exceed a given parameter a. Fig. 3(c) and (d) shows two examples generated with different a's.

C. *Pose Synthesis*

In real application, we always lack "pose images" which are defined as the images with unique target pose in training set. To alleviate this, a pose synthesis operation is introduced. Given a base image set $\mathbf{I} = \{I_{\theta_1}, I_{\theta_2}, \ldots, I_{\theta_N}\}$ of a target with known azimuth angle θ_i and a new azimuth angle θ^*, we generate the corresponding pose image as follows. First, select two images which have the closest angle to θ^*, e.g., I_{θ_a} and I_{θ_b}. Then, combine these two images linearly with each other to get

$$I_{\theta^*} = \mathrm{CR}_{\theta^*}\left(\frac{|\theta_a - \theta^*|R_{\theta_a}(I_{\theta_a}) + |\theta_b - \theta^*|R_{\theta_b}(I_{\theta_b})}{|\theta_a - \theta^*| + |\theta_b - \theta^*|}\right) \quad (8)$$

where $R_\theta(I)$ denotes the operation rotating the image I by θ degrees clockwise, $\mathrm{CR}_\theta(I)$ rotating image I counterclockwise, and the combination coefficients are the differences between the selected images and the desired angle. Fig. 4 illustrates the process with an example. In this example, the desired azimuth angle $\theta^* = 301.502$ and the closest two images are at 284.492° and 315.492°, shown in Fig. 4(a) and (b), respectively. The synthesized image is provided in Fig. 4(c), comparing with the real target image at 300.492° shown in Fig. 4(d).

IV. Experiment

A. *MSTAR Data Set*

To evaluate the performance of the proposed method, we use the Moving and Stationary Target Acquisition and Recognition (MSTAR) public data set. The data set consists of X-band SAR images with 0.3 m × 0.3 m resolution of multiple targets, which includes BMP2 (infantry combat vehicle), BTR70 (armored personnel carrier), T72 (main tank), etc. All images are of size 128 × 128. The number of images for training and testing in our experiments is summarized in Table I. The training images are obtained at an elevation angle of 15°, and the testing images are at 17°.

Besides the original test samples (denoted T_Orig), we produced another two types of data for test, translated test data, and speckle noised test data (denoted T_Trans and T_Noise, respectively). The former is generated by randomly translating each test image ten times, and the latter is synthesized by noising each test image nine times with parameter a set to 0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 1.5, 1.5, and 1.5, respectively.

B. *Single Type of Augmentation*

First, we evaluate the performances achieved by the CNNs with each augmentation technique separately. Fig. 5 shows the results of CNN with translation augmentation and linear SVM. In this experiment, we shift each training sample 6, 12, 21, 30, and 45 times, respectively, each time with a random translation in the guard window [see Fig. 2(a)]. The recognition accuracy achieved 94.29% on the original test data and 93.30% on the translated one, but the rate on the noising version is quite low. Note that the size of T_Trans is ten times that of the T_Orig, and the drop on recognition rate is unapparent. That is to say, the CNN trained with translation augmentation can handle target translation but not the random speckle noising. For comparison, we use the augmented data to train the SVM classifier with the liblinear solver [15]. The last two column recognition rates of SVM are unavailable due to the limitation of memory, since it operates on the whole big matrix with the size of 128 × 128 × 82410 and 128 × 128 × 123615, respectively. As shown in Fig. 5, the CNN method is superior to SVM on the same test set. This could be attributed to the fact that the CNN method handles the 2-D image directly and can exploit the context of each pixel through the convolution operation with the help of augmented data. Moreover, CNN also employs the max-pooling step to achieve the local translation invariance, which makes it robust to the small shift between targets from different images, while SVM only considers each input image as a vector.

Next, we perform the experiment on the CNN trained with speckle noising augmentation. The training set is expanded by 6, 12, 21, 30, and 45 times, respectively. One-third of the noising parameters a's for each sample are set to 0.5, one-third to 1.0, and the last third to 1.5. The result of this experiment is shown in Fig. 6. Again, we see that the CNN trained with speckle noising augmentation can perform well on the noising test data but the translated one and better than the SVM. Note that each output unit in the convolutional layer is calculated by summing the product of the corresponding filter and input unit in a local area, which serves as a smoothing operation to some extent. Meanwhile, CNN does not assume the distribution of noise in the input images, so it can be adaptable to more general and complex noise models, which is consistent with the empirical results in [16].

TABLE I
SUMMARY OF THE SETUP OF THE TEN-CLASS PROBLEM ON THE MSTAR DATABASE

Target Type	BMP2			BTR70	T72			BTR60	2S1	BRDM2	D7	T62	ZIL131	ZSU23/4
	snc21	sn9563	sn9566	c71	sn132	sn812	sns7							
Training	233	/	/	233	232	/	/	256	299	298	299	299	299	299
Testing	196	195	196	196	196	195	191	195	274	274	274	273	274	274

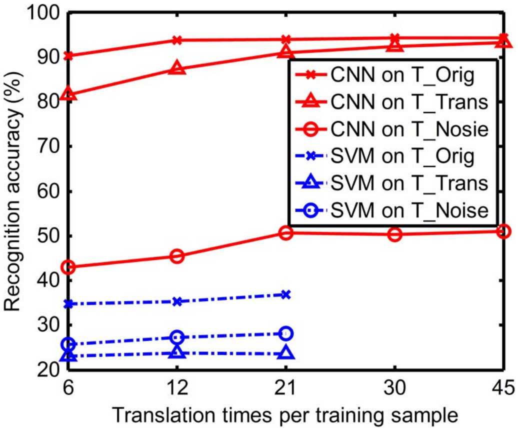

Fig. 5. Recognition accuracy of CNN with translation augmentation.

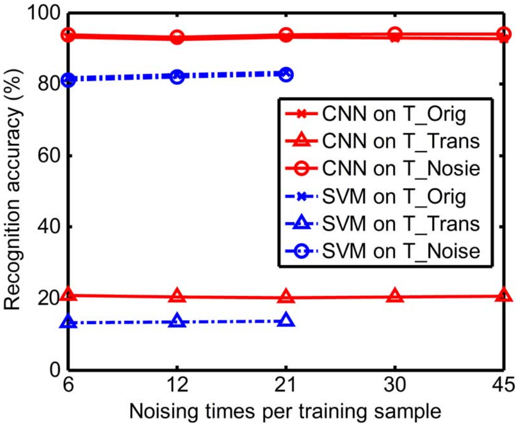

Fig. 6. Recognition accuracy of CNN with noising augmentation.

TABLE II
RECOGNITION ACCURACY OF CNN WITH POSE SYNTHESIS

Test data	K training samples per class				
	30	60	90	120	180
T_Orig	**75.12**	**81.70**	**85.58**	**89.26**	**89.73**
T_Trans	16.18	17.96	21.00	22.07	21.80
T_Noise	55.71	54.42	56.51	57.57	57.76

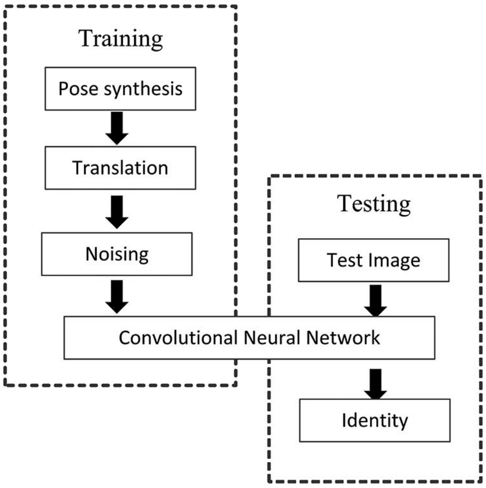

Fig. 7. Procedure of generating training data with all augmentation operations.

TABLE III
RECOGNITION ACCURACY OF THE ALL-IN-ONE CNN

Method	Test data		
	T_Orig	T_Trans	T_Noise
All-in-one CNN	**93.16**	**82.40**	**91.89**
CNN without augmentation	89.14	18.58	87.26
MINACE	80.80	79.86	51.09
SVM	75.68	17.05	70.58
MSRC [6]	93.66	-	-

To evaluate the effectiveness of the pose synthesis method, we randomly select K training samples per class as base images and generate 5000 synthesized images per class. The results with different K's are given in Table II. We can see from Table II that the accuracy archived to 89.26% when we know about only 120 pose images per class, about 55% of the training data. This shows the feasibility of pose synthesis in target recognition with CNN.

C. Combine All Types of Augmentation

In this section, we carry out more aggressive experiment by combining all three types of augmentation with CNN. Fig. 7 sketches the process of training data generating. In this experiment, we first set the number of base images per class to 120 and produce 1000 synthesized pose images per class. Then, we translated them five times randomly. Finally, speckle noising augmentation operations are performed on each translated images with parameter a's set to 0.5, 0.5, 1.0, 1.0, 1.5, and 1.5, six times per image in total. For convenience in describing, we refer to the CNN trained with all types of augmentation as the all-in-one CNN.

The recognition accuracies with different methods are summarized in Table III. Due to the computational limitation, the MINACE [5] and SVM algorithms are both trained on no

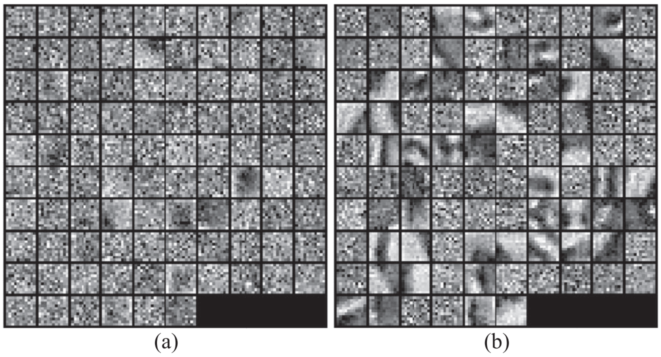

Fig. 8. Visualization of the filters in the lowest layer of CNN. (a) Filters in CNN without data augmentation. (b) Filters in the all-in-one CNN.

augmented training data without pose missing. Although the MINACE algorithm can achieve translation invariance, it cannot be distortion invariant to SAR images with the speckle noise model because of its Gaussian noise assumption. In contrast, the SVM algorithm manifests that it is insensitive to distortion but heavily sensitive to translation, while the performance of the all-in-one CNN is not affected significantly. As we can see, the performance of the all-in-one CNN is superior to the MINACE and SVM methods on all test data set and is comparable to the recently developed MSRC method, even when pose missing exists. To get a deeper view of the differences between the CNNs with and without data augmentation, we visually inspect the filters in their lowest layer in Fig. 8. The images in Fig. 8 are formed by concatenating the 96 kernels of size 11 × 11 and reshaping the obtained long image into a square one with 10 × 10 blocks. We can see that the learned filters are more informative and well structured in the all-in-one CNN than that of the CNN without augmentation. It is thus not surprising that the all-in-one model yields good performance on recognition. Note that the accuracies of the all-in-one CNN on different test data are almost comparable to the CNN trained with the corresponding single type of augmentation without pose missing (see Tables II and III).

V. Conclusion

In this letter, we have exploited the CNN for SAR target recognition with domain-specific data augmentation. For the real problems, such as translation of target, randomness of speckle noise, and pose missing in training data, three types of data augmentation operation are introduced. Experimental results demonstrate the effectiveness and efficiency of the proposed method. Moreover, the CNN trained with augmentation operations shows the feasibility to deal with the three problems in a unified way.

Acknowledgment

The authors would like to thank the Associate Editor who handled this letter and the anonymous reviewers for providing valuable comments and suggestions that greatly helped in improving the technical quality and presentation of this letter.

References

[1] Q. Zhao and J. C. Principe, “Support vector machine for SAR automatic target recognition,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 643–654, Feb. 2001.

[2] Y. Sun, Z. Liu, and J. Li, “Adaptive boosting for SAR automatic target recognition,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 112–125, Jan. 2007.

[3] J. X. Zhou, Z. G. Shi, X. Cheng, and Q. Fu, “Automatic target recognition of SAR images based on global scattering center model,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3713–3729, Oct. 2011.

[4] J. I. Park, S. H. Park, and K. T. Kim, “New discrimination features for SAR automatic target recognition,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 476–480, Oct. 2013.

[5] R. Patnaik and D. Casasent. “MINACE filter classification algorithms for ATR using MSTAR data,” in *Proc. SPIE Autom. Target Recog. XV*, Orlando, FL, USA, 2005, vol. 5807, pp. 100–111.

[6] G. Dong, N. Wang, and G. Kuang, “Sparse representation of monogenic signal: With application to target recognition in SAR images,” *IEEE Signal Process. Lett.*, vol. 21, no. 8, pp. 952–956, Aug. 2014.

[7] C. Clemente *et al.*, “Pseudo-Zernike based multi-pass automatic target recognition from multi-channel SAR,” *IET Radar Sonar Navig.*, vol. 9, no. 4, pp. 457–466, Sep. 2015.

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

[9] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, Boston, MA, USA, Jun. 8–10, 2015, pp. 1–9.

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

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, 2014, pp. 580–587.

[12] X. Chen, S. Xiang, C. Liu, and C. Pan, “Vehicle detection in satellite images by hybrid deep convolutional neural networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] V. Nair and G. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 807–814.

[15] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, “Liblinear: A library for large linear classification,” *J. Mach. Learn. Res.*, vol. 9, no. 8, pp. 1871–1874, Aug. 2008.

[16] V. Jain and H. S. Seung, “Natural image denoising with convolutional networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 769–776.