IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2018, accepted October 14, 2018, date of publication October 22, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877018

Moving Target Indication Using Deep Convolutional Neural Network

ZHE LIU[1,2], DOMINIC K. C. HO[2], (Fellow, IEEE), XIAOQING XU[1], AND JIANYU YANG[1], (Member, IEEE)

[1]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]EECS Department, University of Missouri, Columbia, MO 65211, USA

Corresponding author: Zhe Liu (liuzhe@uestc.edu.cn)

This work was supported in part by the Central University Fund under Contract ZYGX2016J033 and in part by the Natural Science Fund of China under Contract 61771113.

ABSTRACT Existing moving target indication (MTI) methods are limited either by the sufficient and homogeneous secondary data (SD) condition or by the high signal-to-clutter-ratio (SCR) requirement. In practice, such constraints are hard to be satisfied. In this paper, we propose a novel deep convolutional neural network (CNN)-based method for the MTI (CNN-MTI) to overcome the limitations of these methods. In the proposed method, the CNN architecture is deliberately designed, and the training data set is greatly augmented to ensure the CNN be deep and well-trained; in addition, the SD only act as the background interference during the augmentation of the training data set. These procedures ensure that the effective features of different classes of moving targets can be extracted, so that good MTI performance can be achieved without the requirement of the sufficient and homogeneous SD or the high SCR condition. Simulation results from the synthetic data and the experimental data demonstrate the validity and the robustness of the CNN-MTI with limited SD support in non-homogeneous and low SCR environment.

INDEX TERMS Moving target indication (MTI), convolution neural network (CNN), space-time adaptive processing (STAP), feature extraction.

I. INTRODUCTION

Moving target indication (MTI) is one of the most important tasks of airborne phased array radar. MTI concerns whether there is a moving target with a certain relative velocity in the interesting scenario, also known as cells under test (CUT). Due to the interference of severe clutter, noise and jamming, the moving target is generally buried in the interference and is hard to be directly detected. To address this problem, MTI based on space-time adaptive processing (STAP-MTI) scheme has been widely applied to improve the detection performance in the airborne radar.

In the pattern recognition point of view, STAP-MTI can be considered as a 2-class classification system. In this system, STAP, which performs adaptive filtering on the space-time observation (STO) at each candidate Doppler frequency shift caused by the target's possible relative velocity to suppress the interference [1], [2], can be regarded as the feature extraction procedure. While the subsequent constant-false-alarm-rate (CFAR) processing can be considered as a 2-class classifier, in which the amplitude of the STAP response is the extracted feature and the two classes represent the target-present and the target-absent, respectively. Owing to the STAP, the extracted feature has a much higher signal-to-interference-plus-noise ratio (SINR) comparing with the original STO; thus the difference between the features of the two classes becomes more distinct and can be much easier to be discriminated by the classifier. However, great challenges have been brought to the STAP by the requirements on both the abundance and the homogeneity of secondary data (SD), which are generally unavailable in practice due to the dense-target environment, the complex terrains, the limitation of the radar measurement or other factors [3]. Either heterogeneity or insufficiency can lead to mismatch between the actual clutter covariance matrix (CCM) and the one estimated from the sample covariance matrix (SMI) method [4]; thereby they can spoil the SINR and degrade the detection performance.

Emphasis of the STAP-MTI has been mainly laid on how to break the constraints on the SD during CCM estimation; and these efforts can be roughly divided into the following four groups. The first group of methods is used to decrease the required amount of the homogeneous SD. Methods in the first group generally take advantage of the properties of the CCM, such as the rank-deficiency [5]–[14],

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the Toeplitz space [15], the sparsity [16], [17], or the prior knowledge about the environment and the radar system [18], or their hybrid combinations [19]. The second group of methods addresses the heterogeneous SD with unexpected targets, i.e. outliers. Non-homogeneous detector (NHD), such as the generalized inner product (GIP) [20], the power selected training (PST) [21], the sample weighting and selecting methods [22], [23], sparsity recovery methods [24] and many others, have been proposed to detect and eliminate the cells of the SD with outliers. In the third group of methods, efforts are performed to deal with the heterogeneous SD with space-variant clutter. Typical methods include adaptive angle Doppler compensation (A2DC) method [25], space-time interpolation transformation (STINT) method [26], and registration based compensation (RBC) method [27]. The fourth group of methods, e.g. direct data domain (DDD) methods [28] and the maximum likelihood estimation detector (MLED) algorithm [29], perform operations only on the CUT without the estimation of the statistics of the interference; and thus could overcome the constraints of the abundance and heterogeneity simultaneously, but at the cost of performance degradation [30]. Due to the fact that methods in this field have their respective preconditions and characteristics, there is no perfect algorithm and the appropriate one needs be selected from a database of algorithms based on the specific application environment [31], [32].

Besides STAP-MTI, there are alternative MTI methods. In [33], the interference, the noise and the targets are considered as 3 classes. Feature extraction is performed through a 2-D fast Fourier transform (FFT) to convert the STO into the angle-Doppler domain, where the unsupervised classifier is used to discriminate among different classes. However, due to its detection performance being highly dependent on the manually-set threshold which is difficult to determine without prior knowledge to the environment, it is hard to be implemented in realistic scenario.

More recently, the linear-classifier based MTI (LI-MTI) [34] and polynomial-classifier based MTI (POLY-MTI) [35] were proposed. In [34], feature extraction is not performed and the STO passes directly to a linear classifier. In [35], the polynomial transformation and the principle component analysis (PCA) are used to extract the feature of the STO before the linear classifier. In both methods, the SD cells are distributed to different classes to construct training data. However, due to the shallow structure of the employed classifiers and the inadequate amount of the training data samples, these methods cannot obtain distinct features for different classes before classification; and this problem becomes especially severe in the low signal-to-clutter (SCR) case. Therefore, it is very hard to achieve satisfactory detection performance from them.

To overcome the constraints of SD and SCR and to improve the robustness of the MTI, we propose a deep convolution neural network (CNN) based MTI (CNN-MTI) method in this paper. The deep CNN has achieved extensive attention and tremendous success in the computer vision field [36]–[38]. In order to apply the CNN, the MTI is regarded as a multiple-class classification problem, in which each pairing of the candidate velocity and space information of the moving target is considered as one class. Deep CNN following the architecture of the well-known AlexNet [38] is designed to perform feature extraction and classification directly from the STO of the airborne radar echo. To tune the proposed deep CNN, large amount of training data are required but in practice there are limited amount of SD available. To solve this problem, we propose to augment the training dataset by taking the available SD as the background interference and combining them with the artificially generated moving target signal. Through the collaboration of the CNN and the training dataset, the MTI can be effectively realized without being constrained either by the heterogeneity and insufficiency effect of the SD or by the high SCR requirement. Thus, the robustness of the CNN-MTI can be highly improved comparing with the traditional STAP-MTI method and the POLY-method. Simulation results from the synthetic data and the experimental mountaintop data [39] demonstrate validity of the proposed method.

The organization of this paper is as follows. In Section II, the signal model of the STO from the airborne radar, as well as the fundamental of the CNN, is presented. In Section III, the proposed method of the MTI using deep CNN is discussed. In Section IV, the simulation results with synthetic data as well as experimental data are shown. Section V draws the conclusion.

II. FUNDAMENTALS

A. MATHEMATICAL MODEL OF THE SPACE-TIME OBSERVATION

The STO of the radar antenna array from a moving target in one range gate consists of two components, one of which is signal coming from the moving target and the other is the composite of clutter, jamming and noise. The STO can be organized into a 2-D matrix, whose column and row dimensions correspond to the pulse and the azimuth sampling, respectively. After matched filtering, the (m, n)-th element in the STO matrix $\mathbf{X}$ can be expressed as

$$X_{m,n} = a \exp(j2\pi m\varpi) \exp(j2\pi n\vartheta) \exp(j2\pi\varphi) + \chi_{m,n}, \tag{1}$$

where the first term represents the echo from the moving target signal and $\chi_{m,n}$ is the (m, n)-th element of the interference. $m = 1, \cdots, M$, $n = 1, \cdots, N$. M is the number of pulses transmitted within a coherent period interpulse (CPI) period and N is the number of antenna channels. m, n are the indices for pulse and antenna, respectively. a is the complex amplitude of the moving target. $2\pi\varphi$ is the random phase. ϑ and ϖ are the normalized space frequency (NSF) and the normalized Doppler frequency (NDF) given by

$$\vartheta = \frac{d}{\lambda} \sin\theta \cos\phi, \tag{2}$$

$$\varpi = \frac{2v}{\lambda f_r}. \tag{3}$$

In (2), d is the interval between the adjacent antennas. θ is the elevation angle and ϕ is the azimuth angle of the radar platform relative to the moving target. λ is the wavelength of transmitted signal corresponding to the center frequency. In (3), v is the velocity of the moving target relative to the platform. f_r is the pulse repetition frequency (PRF).

Due to the severe clutter, noise and jamming, the moving target signal are always buried in the interference. The goal of MTI is to detect the Doppler frequency and spatial frequency of the moving target from the STO.

B. CNN FUNDAMENTALS

CNN is an end-to-end classifier that performs feature extraction and classification directly from the observed data. Several types of CNN, e.g. LeNet [36], AlexNet [37], ResNet [40] and so on, have been proposed. Despite their variance in architecture, different types of CNN are commonly composed of the convolution layer, the pooling layer, the fully connected layer and the final classifier layer.

The convolution layers, pooling layers and fully-connected layers perform hierarchical nonlinear operations, so that features of different classes hidden in the observed data can be extracted. Such features are very helpful for the final classification layer to realize high performance classification.

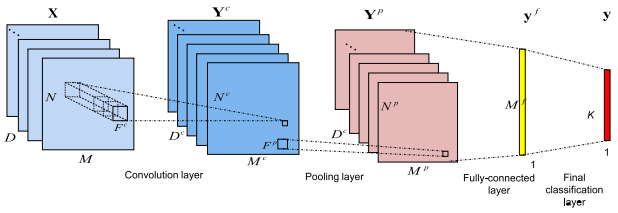


FIGURE 1. The illustration of CNN layers.

In Fig. 1, the CNN connection relationship is illustrated based on an example with 1 convolution layer, 1 pooling layer, 1 fully connected layer and 1 final classifying layer. The output of each layer is the input to the next layer. The input and the output of the convolution layer and the pooling layer are represented as data volumes, where the 3-elements of the subscript denote the 2-D position in the feature map and the index of feature map (channel), respectively. The input and output of the fully-connected layer and the final classification layer are vectors. The superscripts c, p and f denote the convolutional layer, the pooling layer and the fully connected layer, respectively. K is the total number of all possible classes.

1) CONVOLUTION LAYER

Convolution layer applies linear convolution and nonlinearization activation on each sub-region of the input volume. In the d-th channel of the output volume, the (m, n)-th node can be formulated as

$$Y^c_{m,n,d} = \gamma^c \left(\sum\nolimits_{\varsigma,\xi,\eta} X^{\dagger}_{(m-1)\times s^c+\xi,(n-1)\times s^c+\eta,\varsigma} W^c_{d,\varsigma,\xi,\eta} + b^c_d \right). \tag{4}$$

where γ^c is the elementwise and non-linear activation operator. $X^{\dagger}_{(m-1)\times s^c+\xi,(n-1)\times s^c+\eta,\varsigma}$ is a node of the input data volume after zero padding. $W^c_{d,\varsigma,\xi,\eta}$, b^c_d are the weighting parameter and the bias value, respectively. $\zeta = 1, \cdots, D$, $d = 1, \cdots, D^c$ are the channel index of the input volume and the output volume. ξ, η=1, ..., F^c and $F^c \times F^c$ is the kernel size of the input sub-region in one channel. S^c is the stride length of the convolution layer which measures the interval between two neighboring convolution sub-regions.

2) POOLING LAYER

Pooling layer performs down-sampling on a sub-region of its input volume. In the d-th channel of the output data volume, the (m, n)-th element is represented as

$$Y^p_{m,n,d} = \gamma^p \left(\mathbf{Y}^{c\dagger}, m, n; S^p, F^p \right). \tag{5}$$

where γ^p is the pooling operator of down-sampling. $\mathbf{Y}^c$ is the input volume of the pooling layer and $\mathbf{Y}^{c\dagger}$ is the volume after zero-padding of $\mathbf{Y}^c$. $F^p \times F^p$ is the kernel size of a sub-region. S^p is the stride length of the pooling layer which measures the interval between two neighboring sub-regions.

3) FULLY-CONNECTED LAYER

In the fully connected layer, every node of the input is connected to that of the output. The m-th element of its output vector is

$$y^f_m = \gamma^f \left(\left\langle \mathbf{y}^p, \mathbf{w}^f_m \right\rangle + b^f_m \right). \tag{6}$$

where γ^f is the nonlinear function operator. $\mathbf{y}^p$ is the vectorization of the input volume of the fully connected layer. $\mathbf{w}^f_m$ is the weighting vector and b^f_m the bias connecting the input vector with the m-th element of the output vector. $\langle \bullet \rangle$ denotes the inner product of two vectors.

4) FINAL CLASSIFIER LAYER

The softmax classifier is applied in this layer. The number of the elements of the output vector is K and it is also the total number of all possible classes and the k-th element is the estimated probability that the input belongs to the k-th class

$$y_k = p\left(k|\mathbf{X}\right) = \frac{\exp\left(\left\langle \mathbf{w}_k, \mathbf{y}^f \right\rangle + b_k\right)}{\sum_{k=0,1,\cdots,K-1} \exp\left(\left\langle \mathbf{w}_k, \mathbf{y}^f \right\rangle + b_k\right)}. \tag{7}$$

where $\mathbf{w}_k$ and b_k are the weighting vector and the bias value between the input vector and the k-th element of the output vector.

III. METHOD

In the proposed CNN-MTI, the pairing of one candidate NDF and one candidate NSF of the moving target are considered as one class, so that MTI can be regarded as a multiple-class classification problem; and the deep CNN is employed to perform feature extraction and classification. The proposed CNN-MTI method includes the following 4 steps: the construction of the deep CNN architecture, the generation of the

training data set, the training of the deep CNN and the test of the deep CNN.

A. THE CONSTRUCTION OF THE DEEP CNN ARCHITECTURE

The deep CNN architecture of the proposed CNN-MTI method is shown in Fig. 2. It follows the AlexNet model and is composed of four convolutional layers, two pooling layers, one fully connected layer and one final classification layer. Two convolution layers are stacked before one pooling layer.

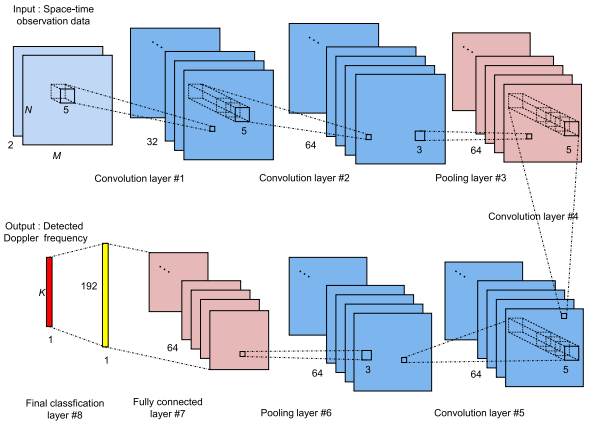


FIGURE 2. The CNN model in the proposed MTI-CNN.

In the CNN, the real part and the imaginary part of STO are fed into the two channels of the input volume. The number of channels of the input volume is 2, and the 2-D size of each channel is $M \times N$, i.e. the number of pulses transmitted in one CPI and the number of antenna channels. The output size of the final classification layer is K, which is the number of classes. In each convolution layer and each pooling layer, zero padding technique is used. The hyper-parameters, e.g. number of channels in each convolutional layer, are determined experimentally to achieve better performance. The hyper-parameters and the size of the output volume of each layer are listed in Table 1, where the output of one layer is the input of the next one and the operator $\lceil\rceil$ denotes the rounding-up operation.

TABLE 1. The description of the proposed deep CNN architecture.

Layer name	Kernel size/stride	Size of the output volume
Convolution layer #1	$5 \times 5/1$	$M \times N \times 32$
Convolution layer #2	$5 \times 5/1$	$M \times N \times 64$
Pooling layer #3	$3 \times 3/2$	$\lceil M/2 \rceil \times \lceil N/2 \rceil \times 64$
Convolution layer #4	$5 \times 5/1$	$\lceil M/2 \rceil \times \lceil N/2 \rceil \times 64$
Convolution layer #5	$5 \times 5/1$	$\lceil M/2 \rceil \times \lceil N/2 \rceil \times 64$
Pooling layer #6	$3 \times 3/2$	$\lceil M/4 \rceil \times \lceil N/4 \rceil \times 64$
Fully connected layer #7	None	192
Final classification layer #8	None	K

In the CNN, the rectified linear unit (Relu) function is used as the nonlinear activation function of all the convolution layers and the fully connected layer of the proposed CNN; while the max pooling is utilized in the pooling layers

$$\gamma^c(x) = \gamma^f(x) = \max(x, 0), \tag{8}$$

$$\gamma^P\left(\mathbf{Y}^{c\dagger}, m, n; S^p, F^p\right) = \max_{1\le\xi,\eta\le F^p}\left\{Y^{c\dagger}_{d,(m-1)\times S^p+\xi,(n-1)\times S^p+\eta}\right\}. \tag{9}$$

In the CNN-MTI, the features of different moving targets are extracted through the nonlinear operations in the convolution layers, the pooling layers and the fully-connected layer. The classification with the extracted features is realized in the final classification layer.

B. THE CONSTRUCTION OF THE AUGMENTED TRAINING DATA SET

In this subsection, we propose to artificially generate abundant training data using the limited available SD cells to augment the training dataset. For simplicity, we assume the NSF is already known.

The main idea is to regard the SD as the background interference where the moving target signal is buried. For the target-present classes, the training samples are constructed by repeatly adding the artificial moving target signal of different amplitudes, different NDF's and random phases to each available SD. For the target-absent case, small disturbance of clutter signal, which is the generated static clutter with much lower power, is added to the SD. The (m, n)-th element of the training sample generated from the q-th SD is

$$X^{(k,q,h)}_{m,n} = a^{(k,h)} \exp\left(j2\pi\varphi^{(k,q,h)}\right) \exp\left(j2\pi m\varpi^{(k)}\right) \times \exp\left(j2\pi n\vartheta\right) + \chi^{(q)}_{m,n}, \tag{10}$$

where the superscripts k, q, h are the indexes of the training samples indicating different NDF's, SD range cells and target amplitudes. $k = 0, \cdots, K-1$, $q = 1, \cdots, Q$, and $h = 1, \cdots, H$. K, Q, H are the total numbers of the candidate classes, the SD cells and the possible amplitudes of the moving target, respectively. k is also the class label of different NDF's. For the target-absent class, $k = 0$, $\varpi^{(0)} = \frac{2v_a}{df_r}\vartheta$, where v_a is the platform velocity; for the target-present classes, $k \geqslant 1$, and it corresponds to the k-th candidate NDF of the moving target which is

$$\varpi^{(k)} = -\frac{1}{2} + \frac{k-1}{K-1}, \quad k = 1, 2, \cdots, K-1. \tag{11}$$

In (10), the phase $2\pi\varphi^{(k,q,h)}$ randomly varies with different k, q and h within the range of $[0, 2\pi)$. $a^{(k,h)}$ is the h-th amplitude of the training samples for the k-th class. $a^{(k,h)} \in \left[a^{(k)}_-, a^{(k)}_+\right]$, where $a^{(k)}_- > 0, a^{(k)}_+ > 0$ are the lower bound and the upper bound of the training samples of the k-th class. Generally, the power boundary of moving target can be chosen to be 20-30 dB lower than the interference power which can be obtained from the power spectrum of the echo data. For $k \geqslant 1$, $a^{(k)}_- \gg a^{(0)}_-$, $a^{(k)}_+ \gg a^{(0)}_+$. Thus, the constructed training dataset is $\left\{\left(\mathbf{X}^{(k,q,h)}, k\right)\right\}$, $k = 0, \ldots, K-1$, $q = 1, \ldots, Q$ and $h = 1, \ldots, H$.

C. TRAINING OF THE CNN

In the training stage, we use the training dataset to obtain the values of the weighting parameters and the bias parameters in the CNN so that

$$\boldsymbol{\Theta} = \arg\min J\left(\boldsymbol{\Theta}\right), \tag{12}$$

where $\boldsymbol{\Theta}$ is the parameter vector which includes all the weighting parameters to be trained in the CNN. $J\left(\boldsymbol{\Theta}\right)$ is the cross-entropy loss function defined by

$$J\left(\boldsymbol{\Theta}\right) = -\sum_{k,q,h} k \ln p(k|\mathbf{X}^{(k,q,h)}; \boldsymbol{\Theta}). \tag{13}$$

In (13), $p\left(k|\mathbf{X}^{(k,q,h)}; \Theta\right)$ is the output of the last layer of the CNN network and it can be obtained from (7) for each training sample $\mathbf{X}^{(k,q,h)}$.

Gradient descent and the back projection method are used to iteratively solve the optimization problem in (12) by

$$\boldsymbol{\Theta}\left(i+1\right) = \boldsymbol{\Theta}\left(i\right) - \eta\left(i\right) \cdot \nabla_{\boldsymbol{\Theta}} J. \tag{14}$$

where $i = 1, 2, \cdots$ denotes the sequence number of iteration during gradient descent. The initial values of the parameter vector, $\boldsymbol{\Theta}(0)$, are randomly selected. $\eta(i)$ is the learning rate at the i-th iteration and $\nabla_{\boldsymbol{\Theta}}$ is the gradient operation operator with respect to $\boldsymbol{\Theta}$.

D. TEST OF THE MTI-CNN

In the test stage, the test sample $\mathbf{X}$, i.e. one STO from one range gate of the CUT, goes through each layer in Fig. 2. $\mathbf{X}$ is classified to the z-th class where

$$z = \underset{k=0,\cdots,K-1}{\arg\max}\ y_k = \underset{k=0,\ldots,K-1}{\arg\max}\ p\left(k|\mathbf{X}\right). \tag{15}$$

According to the classification results, the detected NDF as well as the velocity can be achieved by substituting the result to (11) and (3).

E. COMPUTATION COMPLEXITY AND MEMORY REQUIREMENT OF THE PROPOSED CNN-MTI

The computation burden of the proposed CNN-MTI mainly comes from the convolution operation during CNN training and test. The order of the computation complexity is the total numbers of the multiplication operations in all convolution layers [41]. Thus, according to Table 1 and Fig. 2, the computation complexity of the proposed CNN-MTI is in the order of $O(10^5 IMN)$, where I is the iteration numbers during training.

The memory requirement of the proposed CNN-MTI results from storing the weighting and bias parameters as well as the feature map of each layer. According to Table 1 and Fig. 2, the number of the weighting and bias parameters in the proposed CNN-MTI is about 2.56×10^5; while the total size of the saved feature map is $150MN$. The required memory of the proposed CNN-MTI is decided by the summation of the above two sizes.

F. REMARKS

Two important techniques are cooperated in the CNN-MTI to guarantee its accuracy and robustness against the limited SD support, the low SCR and the non-homogeneous clutter environment.

The first important technique is the deliberate design of the deep CNN in order to extract the useful features for the MTI. The two channels of the input volume enable that both the real part and the imaginary part of the STO data be simultaneously sent to the CNN for feature extraction. On the other hand, the decrease of the pooling operation, the zero padding and the small stride configuration in the CNN architecture can avoid the too rapid decrease of the feature dimensions and make full use of the feature on the edge. It is very important for the MTI application due to the fact that the dimensions of the STO data are generally in the quantity level of dozens. Such design of the CNN architecture can ensure the network to be deep enough so that more expressive and powerful features for the moving targets can be extracted.

The second important technique is the augmentation of the training dataset. Deep CNN requires large amount of training data to perform well, but the available training data of MTI are only the SD samples, which can be considered as the training data of the 0-th class. Through the proposed augmentation method, the amount of the constructed training data samples is KH times of that of the available SD, and the amount can be adjusted according to different requirements. Meanwhile, the class unequilibrium problem during training is avoided by generating the same amount of training data for each class. Also, in the augmented training dataset, the SD echo acts only as the interference background; so that the effect of the SD heterogeneity can be greatly reduced. Therefore, through the augmentation, the generated training dataset can provide the guarantee of not only the well training of the proposed deep CNN but also the robustness against homogeneous environment.

G. EXTENSION TO UNKNOWN NSF

In the above, the NSF of the moving target is assumed to be already known. For the unknown NSF case, the CNN-MTI can be extended.

First, the pairing of one candidate NDF and one candidate NSF, $(\varpi^{(k)}, \vartheta^{(l)})$, is considered as one class. $\vartheta^{(l)}$ can be obtained from (2) according to the value of the possible azimuth angles ϕ_l, $l = 1, \ldots, L$, where ϕ_l is the index of the candidate NSF and L is the total number of the candidate NSF. The label of each class is $\hat{k}$ which is decided by k and l

$$\hat{k}(k,l) = \begin{cases} 0, & k = 0 \\ (l-1)(K-1)+k, & k = 1, \ldots, K-1. \end{cases} \tag{16}$$

Then we can construct the training dataset by substituting each candidate NSF and NDF into (10)

$$X_{m,n}^{(k,l,q,h)} = a^{(k,l,h)} \exp\left(j2\pi\varphi^{(k,l,q,h)}\right) \exp\left(j2\pi m\varpi^{(k)}\right) \times \exp\left(j2\pi n\vartheta^{(l)}\right) + \chi_{m,n}^{(q)}. \tag{17}$$

Thereby, the training dataset is extended to be $\left\{\left(\mathbf{X}^{(k,l,q,h)}, \hat{k}\right)\right\}$, $k = 0, \ldots, K-1, q = 1, \ldots, Q, l = 1, \ldots, L, h = 1, \ldots, H$ and $\hat{k} = 0, \ldots, L(K-1)$.

As for the CNN classifier, the architecture shown in Fig. 2 keeps unchanged except that the number of the output nodes in the final classification layer is increased from K to $L(K-1)+1$.

In the training stage, the extended training dataset is used to obtain the CNN parameters; while in the test stage, the test sample $\mathbf{X}$ is classified to the z-th class where

$$z = \underset{\hat{k}=0,\cdots,L(K-1)}{\arg\max} \; y_{\hat{k}} = \underset{\hat{k}=0,\ldots,L(K-1)}{\arg\max} \; p\left(\hat{k}|\mathbf{X}\right). \quad (18)$$

From the classification result and (16), the index of the NDF and the NSF of the moving target can be found to be $l = \lfloor z/(K-1) \rfloor$ and $k = z-(l-1)(K-1)$ for $z \geq 1$, where the operator $\lfloor\rfloor$ denotes the ceiling operation. Accordingly, the velocity and azimuth position of the moving target can be achieved from (2), (3) and (11).

IV. RESULTS

In this section, the synthetic echo as well as the experimental Mountaintop data [39] are utilized to verify the proposed CNN-MTI method. The parameters of the synthetic echo and the experimental data are shown in Table 2. In both cases, the 33 range gates which contain the moving target are taken as the CUT. There are 5 more guard cells on each side of the CUT; and the SD range cells lie outside of the guard cells. Eight SD range cells are used in CNN-MTI. The CNN is realized via Tensorflow 1.3.0. This section includes three parts denoted as Part A, Part B and Part C to verify the CNN-MTI method with different clutter environments, different SCR scenarios and mountaintop data, respectively.

TABLE 2. System parameters.

Parameters	Values for the synthetic data	Values for the experimental data
Number of spatial channels	14	14
Number of pulses in one CPI	16	16
Jamming power (dB)	60	None
Jamming azimuth angle (degree)	25.45	None
Target azimuth angle (degree)	-15.45	-15.45
Noise power (dB)	0	NA
Pulse repetition frequency (Hz)	625	625
Platform velocity (m/s)	100	NA

To demonstrate the advantage of the proposed method, results obtained from the SMI-STAP-MTI [4], the SBL-STAP-MTI [16] as well as the POLY-MTI method are also provided for comparisons. In the SMI-STAP-MTI method and the POLY-MTI method, 96 SD range cells are used as SD; whereas, 8 range cells of SD are used in the SBL-STAP-MTI and the CNN-MTI. In the SMI-STAP-MTI and the SBL-STAP-MTI, the 2-D cell average CFAR (CA-CFAR) with the false alarm rate of 1e-4 [42] is applied on their respective STAP results. The NSF is known to be -0.1287. For the POLY-MTI and the CNN-MTI, the total number of classes is 17, where $k = 0$ represents the absent-target case and the labels $k \geqslant 1$ denote the NDF of $\varpi^{(k)} = -\frac{1}{2} + \frac{(k-1)}{16}$, $k = 1, 2, \cdots, 16$, respectively.

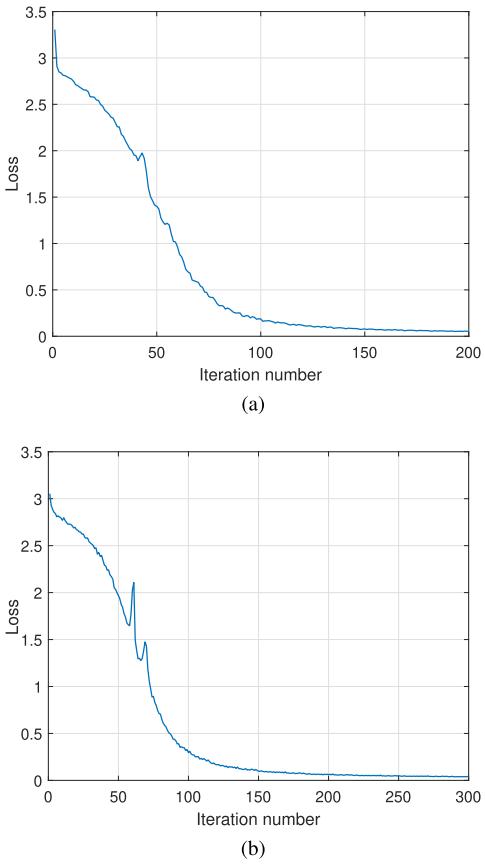


FIGURE 3. The value of the loss function in each generation of the training stage. (a) Curve of the loss function for the case 1 and case 2. (b) Curve of the loss function for the case 3.

A. DIFFERENT CLUTTER ENVIRONMENTS

To demonstrate the validity and the robustness of the proposed CNN-MTI, we simulate the echo of 3 different clutter environments. The interference environment is homogeneous in case 1; while case 2 has the space-variant clutter and case 3 contains target-like outliery. In case 2, the SD echo are the same with case 1, but the clutter in the CUT is generated with a different platform velocity of 120 m/s. As for case 3, the CUT echo is the same with case 1 but there is an unexpected target with the power of 30 dB in the 4-th range cell of the SD. For each case, the power of the clutter and the target is 50 dB and 28 dB; the target is located at the 17-th range gate with the NDF of 0.25 corresponding to the 13-th class.

Owing to the same SD, one training dataset needs to be constructed and the CNN is trained for case-1 and case-2 simultaneously; then the trained CNN is tested with the test datasets of case-1 and case-2 separately. Owing to the same CUT echo, the test dataset of case 1 and case 3 are the

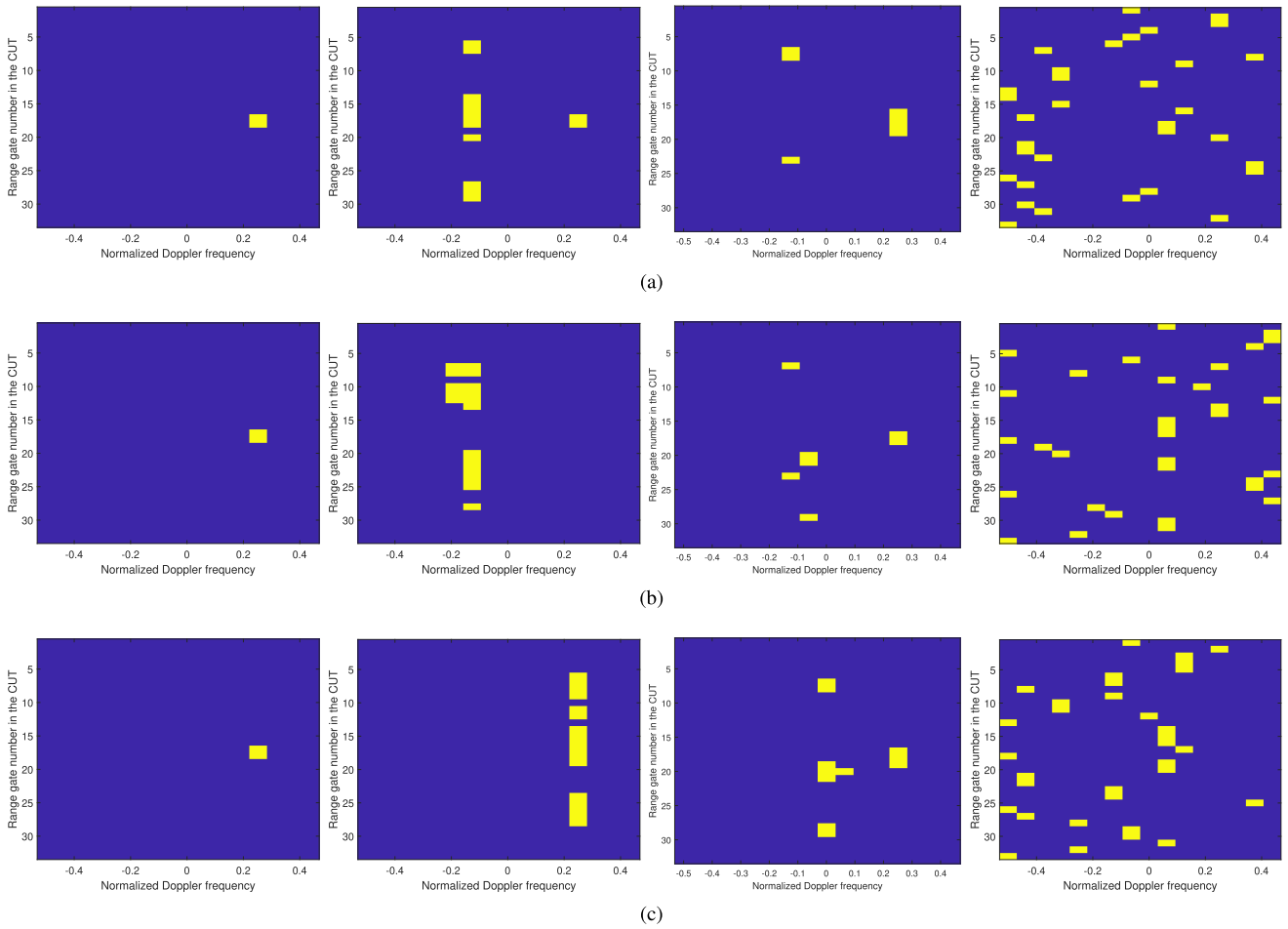


FIGURE 4. Processing results of different clutter environment with SCR=-20dB. In each row, from left to right are the result of the CNN-MTI method, the SMI-STAP-MTI method, the SBL-STAP-MTI and the POLY-MTI method. (a) Results of the different 3 methods in homogeneous environment (case-1). (b) Results of the different 3 methods in space-variant clutter environment (case-2). (c) Results of the different 3 methods in unexpected target-present clutter environment (case-3).

TABLE 3. Target power boundary of the training datasets of the CNN-MTI.

	Lower boundary (dB)		Upper boundary (dB)	
	For $k=0$	For $k=1,\cdots,16$	For $k=0$	For $k=1,\cdots,16$
Part A	0	30	20	50
Part B	0	8.5	5.5	50.5
Part C	0	50	15	70

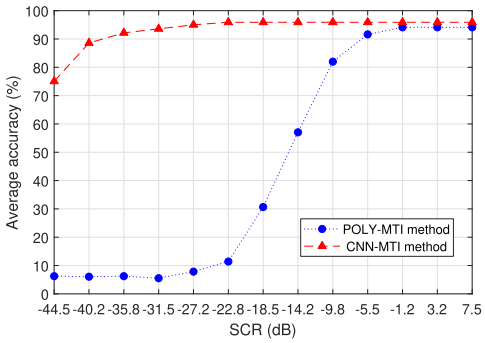


FIGURE 5. Performance comparison of the CNN-MTI method and the POLY-MTI method.

same, but the CNN needs to be trained separately for them. Therefore, there are 2 training datasets and 2 test datasets for the 3 different clutter environments. Each training dataset of the CNN-MTI is constructed according to the parameters listed in Table 2 and Table 3. The interval of the power for each class is equally spaced and the number of the different target power levels (amplitudes) for each class is 101. There are 808 training samples for each class and a total of 13736 samples in each training dataset. During each training, the batch gradient descent method is used to obtain the weights of the classifier and the size of the batch is 404. The learning rate is 1e-3. The values of the loss function of equation (13) for each of the training are evaluated and shown in Fig. 3.

After training for 200 iterations and 300 iterations for the two respective classifiers, the test datasets of the three cases are passed into the corresponding CNN classifier, and their classification results are shown in Fig. 4. In different clutter

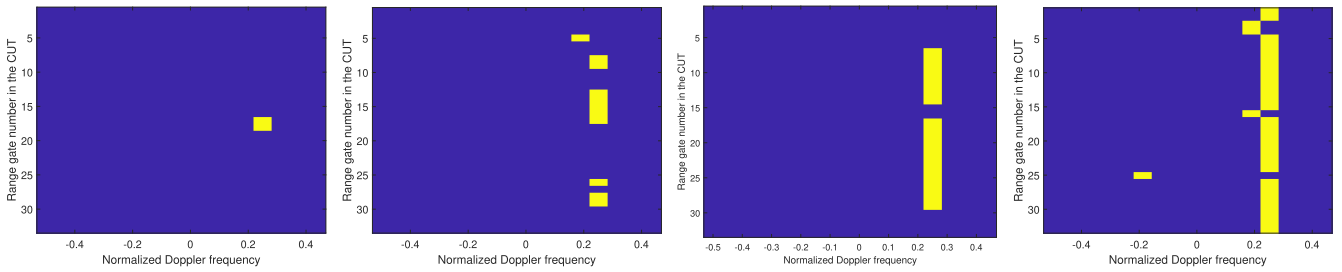


FIGURE 6. Processing results of the mountaintop data. (a) CNN-MTI method. (b) SMI-STAP-MTI method. (c) SBL-STAP-MTI method. (d) POLY-MTI method.

environments, the CNN classifier can detect the correct NDF for different range cells. Therefore, in the clutter scenario with the SCR of −22dB and the jamming-to-clutter-ratio (JCR) of −30 dB, the proposed CNN-MTI with 8 SD cells can correctly detect the moving target. The CNN-MTI method is accurate and robust against different clutter environments. On the contrary, the SMI-STAP-MTI cannot detect the target in case 2; and it can detect the target in case 1 and case 3 but with many false alarms in other range cells. The POLY-MTI fails to detect the target for all of the 3 cases. Although the SBL-STAP-MTI method can detect the moving target, false alarms occur in the 3 cases. Thus, the CNN-MTI is much more robust than the other three methods confronting with limited SD support or non-homogeneous clutter environment.

B. DIFFERENT SCR SCENARIOS

In this part, the performance of the CNN-MTI with regard to the different SCR is validated. There are artificially generated 13 test datasets with different SCRs, which is caused by the different target power under the same clutter power of 55 dB. The targets' powers in different test datasets vary from 10.5 dB to 62.5 dB with the equal interva. The jamming power is 60 dB. In each dataset, the test samples are generated by adding the target signals with the same power and candidate NDF's to the interference of each of the range gate in the CUT. Thus, there are 33 test samples for each of the 17 classes in each dataset.

The power boundary to construct the training dataset of the CNN-MTI method are shown in Table 3 and the number of training examples is the same with those of Part A. Due to the same clutter environment and the same SD, one training dataset is constructed and one CNN-classifier is trained for all of the 13 test datasets simultaneously; then the CNN-classifier is tested with the 13 test datasets separately. Therefore, there are 1 training datasets and 13 test datasets in this part. After training for 600 iterations, the samples of each test dataset go into the trained CNN and the detection performance is evaluated via average accuracy, which is the average percentage of the correctly classified test samples of each class in the test dataset [43]. Due to no constraints on the SCR, the SMI-STAP-MTI method and the SBL-STAP-MTI is not considered in this part.

The accuracy comparison of the CNN-MTI method and the POLY-MTI method is shown in Fig. 5. It can be observed that, the average accuracy of POLY-MTI is very low and it is less than 10% at SCR lower than −27.2 dB. On the contrary, for the CNN-MTI, the average accuracy approximately approaches 90% even with the SCR of −40.2 dB. With the increase of SCR, the performance of both methods improves. The average accuracy of the CNN-MTI remains to be 95.9% when the SCR is higher than −22.8 dB; and that of the POLY-MTI settles at 94.1% at SCR higher than −1.2 dB. The reason that the average accuracies of both methods remain unchanged as SCR becomes high is that only target-absent samples are mis-classified when the SCR is very high and the increase of the SCR has no impact on the accuracy enhancement. Thus the results demonstrate that, the average accuracy of the CNN-MTI exceed that of the POLY-MTI under different SCR conditions; and the advantage of the CNN-MTI is especially outstanding in the severely low SCR scenarios.

C. EXPERIMENTAL MOUNTAINTOP DATA

We also apply the CNN-MTI method to the Mountaintop dataset [39] and its parameters are listed in Table 2. Of the 17 classes of candidate Doppler frequency, the target Doppler frequency is the closest to the 13-th class.

In the stage of the generating the training dataset, the lowest and upper boundaries of the amplitude for classes 1-16 are 50 dB and 70 dB; whereas those for the class 0 is 0 dB and 15 dB. There are 808 training samples for each class and a total of 13736 training samples in the training dataset. The initial value of the learning rate is 1e-2 and it is rescaled by a factor of 1e-1 every 1000 iterations.

After training for 2000 iterations using the batch gradient method with the batch size of 404, the samples of the CUT range cells are sent to the trained CNN and the detection results are shown in Fig. 6. The proposed method can correctly detect the moving target located at the 17-th range gate and the NDF of 0.25; whereas in the results of the SMI-STAP-MTI, the SBL-STAP-MTI and the POLY-MTI, a lot of false alarm appear.

In the above results, the validity of the proposed CNN-MTI method has been illustrated in different clutter environments

and different SCR scenarios. Comparing with the SMI-STAP-MTI and the POLY-MTI, the proposed CNN-MTI method not only requires much less amount of the SD range cells but also can achieve much better performance confronting with the non-homogeneous and the low SCR scenario. The performance of the SBL-STAP-MTI method exceeds the SMI-STAP-MTI method and the POLY-MTI method, but have much more false alarms than the CNN-MTI method. Therefore, the CNN-MTI is more robust and can conquer against the inherent limitations of the traditional methods.

V. CONCLUSION

In this paper, we propose a deep CNN based MTI method, in which the MTI is regarded as a multiple-class classification problem. In the proposed CNN-MTI method, a deep CNN is utilized to realize the MTI by assigning each STO from the CUT to one class. The CNN architecture includes four convolutional layers, two pooling layers, one fully-connected layer and one final classification layer. To address the requirement of the large amount of training data during the tuning of the CNN, we propose to augment the training dataset by taking the SD as the background interference and combining them with the artificially generated moving target signal at different SCR levels, different Doppler frequencies and random phases. Simulation results from the synthetic data and the experimental data demonstrate that, with very small amount of SD, the proposed method can achieve good performance in the low SCR scenario and in the non-homogeneous clutter environment; and simulation results also demonstrate the great advantage of the CNN-MTI method over the SMI-STAP-MTI method, the SBL-STAP-MTI method and the POLY-MTI method.

ACKNOWLEDGMENT

The authors give the acknowledgement to the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] J. Ward, "Space-time adaptive processing for airborne radar," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 5, May 1995, pp. 2809–2812.

[2] R. Klemm, *Space-Time Adaptive Processing: Principles and Applications*. Piscataway, NJ, USA: IEEE Press, 1998.

[3] W. L. Melvin, "Space-time adaptive radar performance in heterogeneous clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 621–633, Apr. 2000.

[4] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-10, no. 6, pp. 853–863, Nov. 1974.

[5] H. Wang and L. Cai, "On adaptive spatial-temporal processing for airborne surveillance radar systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 3, pp. 660–670, Jul. 1994.

[6] J. R. Roman, M. Rangaswamy, D. W. Davis, Q. Zhang, B. Himed, and J. H. Michels, "Parametric adaptive matched filter for airborne radar applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 677–692, Apr. 2000.

[7] J. H. Michels, M. Rangaswamy, and B. Himed, "Performance of parametric and covariance based STAP tests in compound-Gaussian clutter," *Digit. Signal Process.*, vol. 12, nos. 2–3, pp. 307–328, 2002.

[8] J. S. Goldstein, I. S. Reed, and P. A. Zulch, "Multistage partially adaptive STAP CFAR detection algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 2, pp. 645–661, Apr. 1999.

[9] A. Haimovich, "The eigencanceler: Adaptive radar by eigenanalysis methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 2, pp. 532–542, Apr. 1996.

[10] J. R. Guerci, J. S. Goldstein, and I. S. Reed, "Optimal and adaptive reduced-rank STAP," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 647–663, Apr. 2000.

[11] D. W. Tufts, I. Kirsteins, and R. Kumaresan, "Data-adaptive detection of a weak signal," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 2, pp. 313–316, Mar. 1983.

[12] W. F. Gabriel, "Using spectral estimation techniques in adaptive processing antenna systems," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 291–300, Mar. 1986.

[13] I. P. Kirsteins and D. W. Tufts, "Adaptive detection using low rank approximation to a data matrix," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 1, pp. 55–67, Jan. 1994.

[14] X. Wang, E. Aboutanios, and M. G. Amin, "Slow radar target detection in heterogeneous clutter using thinned space-time adaptive processing," *IET Radar, Sonar Navigat.*, vol. 10, no. 4, pp. 726–734, 2016.

[15] D. M. Wikes and M. H. Hayes, "Iterated Toeplitz approximation of covariance matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, Apr. 1988, pp. 1663–1666.

[16] Q. Wu, Y. D. Zhang, M. G. Amin, and B. Himed, "Space–time adaptive processing and motion parameter estimation in multistatic passive radar using sparse Bayesian learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 944–957, Feb. 2016.

[17] X. Yang, Y. Sun, T. Zeng, T. Long, and T. K. Sarkar, "Fast STAP method based on PAST with sparse constraint for airborne phased array radar," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4550–4561, Sep. 2016.

[18] W. L. Melvin and J. R. Guerci, "Knowledge-aided signal processing: A new paradigm for radar and other advanced sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 3, pp. 983–996, Jul. 2006.

[19] B. Kang, V. Monga, and M. Rangaswamy, "Computationally efficient Toeplitz approximation of structured covariance under a rank constraint," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 775–785, Jan. 2015.

[20] W. L. Melvin, M. C. Wicks, and P. Chen, "Nonhomogeneity detection method and apparatus for improved adaptive signal processing," U.S. Patent 5 706 013 A, Aug. 9, 1996.

[21] D. J. Rabideau and A. O. Steinhardt, "Improved adaptive clutter cancellation through data-adaptive training," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 3, pp. 879–891, Jul. 1999.

[22] H. Xu, Z. Yang, S. He, M. Tian, G. Liao, and Y. Sun, "A generalized sample weighting method in heterogeneous environment for space-time adaptive processing," *Digit. Signal Process.*, vol. 72, pp. 147–159, Jan. 2018.

[23] B. Dai, T. Wang, J. Wu, and Z. Bao, "Adaptively iterative weighting covariance matrix estimation for airborne radar clutter suppression," *Signal Process.*, vol. 106, pp. 282–293, Jan. 2015.

[24] Z. Li, H. Liu, Y. Zhang, and Y. Guo, "Robust nonhomogeneous training samples detection method for space–time adaptive processing radar using sparse-recovery with knowledge-aided," *J. Appl. Remote Sens.*, vol. 11, no. 4, pp. 11–13, Oct. 2017.

[25] W. L. Melvin and M. E. Davis, "Adaptive cancellation method for geometry-induced nonstationary bistatic clutter environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 651–672, Apr. 2007.

[26] V. Varadarajan and J. L. Krolik, "Joint space-time interpolation for distorted linear and bistatic array geometries," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 848–860, Mar. 2006.

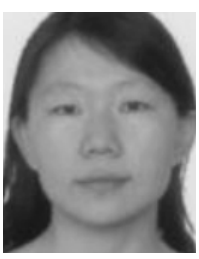
[27] F. D. Lapierre, P. Ries, and J. G. Verly, "Foundation for mitigating range dependence in radar space-time adaptive processing," *IET Radar, Sonar Navigat.*, vol. 3, no. 1, pp. 18–29, Feb. 2009.

[28] T. K. Sarkar *et al.*, "A deterministic least-squares approach to space-time adaptive processing (STAP)," *IEEE Trans. Antennas Propag.*, vol. 49, no. 1, pp. 91–103, Jan. 2001.

[29] J.-F. Degurse, L. Savy, R. Perenon, and S. Marcos, "An extended formulation of the Maximum Likelihood Estimation algorithm. Application to space-time adaptive processing," in *Proc. Int. Radar Symp.*, Sep. 2011, pp. 763–768.

[30] J.-F. Degurse, L. Savy, and S. Marcos, "Reduced-rank STAP for target detection in heterogeneous environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1153–1162, Apr. 2014.

[31] M. C. Wicks, M. Rangaswamy, R. Adve, and T. D. Hale, "Space-time adaptive processing: A knowledge-based perspective for airborne radar," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 51–65, Jan. 2006.
[32] R. S. Adve, M. C. Wicks, T. B. Hale, and P. Antonik, "Ground moving target indication using knowledge based space time adaptive processing," in *Proc. IEEE Int. Radar Conf.*, May 2000, pp. 735–740.
[33] H. Deng, B. Himed, and M. C. Wicks, "Image feature-based space-time processing for ground moving target detection," *IEEE Signal Process. Lett.*, vol. 13, no. 4, pp. 216–219, Apr. 2006.
[34] A. El Khatib, K. Assaleh, and H. Mir, "Learning-based space-time adaptive processing," in *Proc. Int. Conf. Commun., Signal Process., Appl.*, Feb. 2013, pp. 1–4.
[35] A. E. Khatib, K. Assaleh, and H. Mir, "Space-time adaptive processing using pattern classification," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 766–779, Feb. 2015.
[36] 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.
[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. NIPS*, 2012, pp. 1097–1105.
[38] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. (2014). "CNN features off-the-shelf: An astounding baseline for recognition." [Online]. Available: https://arxiv.org/abs/1403.6382
[39] G. W. Titi and D. F. Marshall, "The ARPA/NAVY mountaintop program: Adaptive signal processing for airborne early warning radar," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, May 1996, pp. 1165–1168.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[41] K. He and J. Sun. (2014). "Convolutional neural networks at constrained time cost." [Online]. Available: https://arxiv.org/abs/1412.1710
[42] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 1, pp. 208–216, Jan. 1992.
[43] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009.



ZHE LIU received the B.S. degree in communication engineering from the Chengdu Institute of Meteorology, Chengdu, China, in 1999, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2002 and 2009, respectively. Since 2017, she has been a Visiting Scholar with the University of Missouri, Columbia, MO, USA. She is currently an Associate Professor with UESTC. Her research interests include radar signal processing and machine learning.

DOMINIC K. C. HO (S'89–M'91–SM'00–F'09) was born in Hong Kong. He received the B.Sc. degree (Hons.) in electronics in 1988 and the Ph.D. degree in electronic engineering in 1991. From 1991 to 1994, he was a Research Associate with the Royal Military College of Canada. In 1995, he joined Bell-Northern Research, Montreal, QC, Canada, as a Member of Scientific Staff. From 1996 to 1997, he was a Faculty Member of the Department of Electrical Engineering, University of Saskatchewan, Saskatoon, SK, Canada. Since 1997, he has been with the University of Missouri, Columbia, MO, USA, where he is currently a Professor with the Department of Electrical Engineering and Computer Science. He is an inventor of 22 patents in U.S., Canada, Europe, and Asia on geolocation and signal processing for wireless communications. He was active in the development of the ITU-T Standard Recommendations from 1995 to 2012. He was the Rapporteur of ITU-T Q15/SG16: Voice Gateway Signal Processing Functions and Circuit Multiplication Equipment/Systems from 2009 to 2012 and the Associate Rapporteur of ITU-T Q16/SG16: Speech Enhancement Functions in Signal Processing Network Equipment in 2012. He was the Editor of the ITU-T Recommendations G.160: Voice Enhancement Devices from 2006 to 2012, G.168: Digital Network Echo Cancellers from 2000 to 2012, and G.799.2: Mechanism for Dynamic Coordination of Signal Processing Network Equipment from 2004 to 2009. He was a Technical Co-Chair of the IEEE International Conference on Acoustics, Speech and Signal Processing 2016. His research interests include sensor array processing, source localization, subsurface object detection, wireless communications, and adaptive processing. He received the Senior Faculty Research Award in 2009 and 2014, the Junior Faculty Research Award in 2003, and the Teaching Award in 2006 from the College of Engineering, University of Missouri. He was the Past Chair in 2015, the Chair from 2013 to 2014, and the Vice-Chair from 2011 to 2012 of the Sensor Array and Multichannel (SAM) Technical Committee of the IEEE Signal Processing Society. He participated in the organizing committees of the IEEE SAM 2008 Workshop and the IEEE CAMSAP 2011. He was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2003 to 2006 and from 2009 to 2013 and the IEEE SIGNAL PROCESSING LETTERS from 2004 to 2008.

XIAOQING XU was born in 1993. She is currently pursuing the master's degree with the University of Electronic Science and Technology of China. Her research interests include moving target indication.

JIANYU YANG received the B.S. degree in electronic engineering from the National University of Defense Technology, Changsha, China, in 1984, and the M.S. and Ph.D. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1987 and 1991, respectively. He is currently a Professor with the School of Electronic Engineering, UESTC. His research interests include synthetic aperture radar and statistical signal processing. He is a Senior Editor of the *Journal of Systems Engineering and Electronics*.