IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 18, 2019, accepted October 9, 2019, date of publication October 21, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948618

SAR Target Recognition Based on Cross-Domain and Cross-Task Transfer Learning

KE WANG[1], GONG ZHANG[2], (Member, IEEE), AND HENRY LEUNG[3], (Fellow, IEEE)

[1]School of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China
[2]Key Laboratory of Radar Imaging and Microwave Photonics, Ministry of Education, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China
[3]Department of Electrical and Computer Engineering, University of Calgary, Calgary, T2N 1N4, Canada

Corresponding author: Gong Zhang (gzhang@nuaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871218, Grant 61801211, Grant 61471191, Grant 61501233, Grant 61701046, and Grant 61501228, in part by the Fundamental Research Funds for the Central University, China, under Grant 3082017NP2017421, in part by the Base Research Foundation under Grant NS2015040, and in part by the Funding of Key Laboratory of Radar Imaging and Microwave Photonics, Nanjing University of Aeronautics and Astronautics, Ministry of Education.

ABSTRACT Inspired by their tremendous success in optical image detection and classification, convolutional neural networks (CNNs) have recently been used in synthetic aperture radar automatic target recognition (SAR-ATR). Although CNN-based methods can achieve excellent recognition performance, it is difficult to collect a large number of real SAR images available for training. In this paper, we introduce simulated SAR data to alleviate the problem of insufficient training data. To address domain shift and task transfer problems caused by differences between simulated and real data, we propose a model that integrates meta-learning and adversarial domain adaptation. We use sufficient simulated data and a few real data to pre-train the model. After fine-tuning, the pre-trained model can quickly adapt to new tasks in real data. Extensive experimental results obtained in the real SAR dataset demonstrate that our model effectively solves the cross-domain and cross-task transfer problem. Compared with conventional SAR-ATR methods, the proposed model can achieve better recognition performance with a small amount of training data.

INDEX TERMS Synthetic aperture radar (SAR), target recognition, convolutional neural network (CNN), meta-learning, adversarial domain adaptation.

I. INTRODUCTION

Synthetic aperture radar (SAR) is an active sensor mounted on moving platforms such as aircraft, satellites, and spaceships. SAR provides two-dimensional high-resolution images by receiving the electromagnetic echoes of targets. Benefiting from its unique imaging mechanism, SAR can operate day and night, independent of weather conditions, and has specific surface penetration capability. The SAR system has unique advantages in many applications, ranging from disaster monitoring and resource exploration to military inspection, and it plays an unreplaceable role in both military and civilian fields.

Automatic target recognition (ATR) is an essential topic in the field of SAR application research. According to different implementation methods, classic ATR methods can be classified into feature-based and model-based approaches. Feature-based methods extract discriminative features, such as binary regions [1], target contours [2], monogenic signals [3], [4], projection features [5], [6], and tensor decomposition features [7] from images. Classifiers such as K-nearest neighbor (KNN) [8], support vector machine (SVM) [9], the Bayesian classifier [10], and the sparse representation classifier [11] have been developed to classify the extracted features. Both feature extraction and classification require careful selection by experienced researchers. Model-based methods [12]–[14] focus on the electromagnetic scattering features of a target, which are related to the physical characteristics of the target. A computer-aided design (CAD) model is developed for the target and is then processed through electromagnetic simulation software to obtain the attribute scattering centers. The similarity between the attribute scattering centers of the test target and the training target is evaluated by specifically designed matching methods. However, building a CAD model that stores all the details of the target is time-consuming. Electromagnetic simulation software also

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

has difficulty in simulating diffuse scattering and ground intersection in real scenes. These difficulties degrade the recognition performance of model-based methods.

In recent years, CNN [15] has been widely used in image segmentation, image detection, and image classification. Inspired by these successful applications, CNN has also been introduced to the SAR-ATR field. Chen *et al.* [16] proposed an all-convolutional network (A-ConvNets) for SAR target recognition, and convolutional layers were used to replace the fully connected layers. Kwak *et al.* [17] proposed a noise-invariant CNN to prevent the degradation of recognition performance caused by speckle noise. In [18], CNN was employed to extract the features of SAR images. Then the extracted features were sent to an SVM classifier. Min *et al.* [19] proposed a micro-CNN (MCNN) to reduce the network computational cost and memory consumption so that the network can be deployed in a real-time recognition system. To minimize the influence of noise in SAR images, Cho and Park [20] proposed a multiple feature aggregation CNN (MFCNN). They extracted the feature maps of different CNN layers and aggregated them into a column vector for classification. Deng *et al.* [21] employed a multilayer autoencoder (AE) combined with a Euclidean distance constraint to extract the latent features of SAR images. The extracted features are subsequently classified by a linear SVM classifier.

CNN can achieve state-of-the-art performance with a significant amount of labeled training data. However, due to cost and confidentiality factors, collecting real SAR images for training is difficult and time-consuming. Previous work has attempted to increase training data via simulated SAR image generation, image augmentation, transfer learning, etc. Ding *et al.* [22] produced virtual SAR images through augmentation operations such as translation, speckle noise addition, and pose synthesis, effectively extending the training dataset. Pei *et al.* [23] proposed a multiview deep learning framework to reduce the number of required training samples. The input of the framework is a combination of multiview SAR images rather than a single SAR image. With the recent advances in generative models, researchers have applied the generative adversarial network (GAN) [24] and Wasserstein autoencoder (WAE) [25] to generate simulated SAR images.

In addition to generating simulated images, another approach to expand training data is transfer learning. Transfer learning can transfer prior knowledge from a source domain with large amounts of data to a target domain of limited data. Previous researchers have applied network-based transfer learning to SAR-ATR [26]–[28]. They used optical image data or simulated SAR data as the source domain and real SAR data as the target domain. Network-based transfer learning attempts to use images in the source domain to pre-train the network. The pre-trained network needs only a few real SAR images for fine-tuning, thus alleviating the need for real data in the target domain. However, the differences in target pose, view angle, scene illumination, and background clutter between the source domain and target domain lead to data distribution differences between the two domains. We call this phenomenon the domain shift [29]. Moreover, different image classes in the source domain and the target domain will cause task transfer problems [30]. Domain shift and task transfer make the transfer of knowledge from simulated SAR data to real SAR data a cross-domain and cross-task transfer problem, which will degrade the performance of network-based transfer learning methods.

In this paper, we utilize transfer learning between simulated SAR data [31] and real SAR data to solve the problem of insufficient training samples. We propose a unified model to address the cross-domain and cross-task transfer problem. Recently, model-agnostic meta-learning (MAML) [32] is one of the best methods for few-shot learning. MAML builds a variety of tasks to train a network that can quickly adapt to new tasks in testing. However, MAML assumes that the training tasks and the test tasks are from the same distribution; otherwise, its performance will degrade. Following the training strategy of MAML, we improve the adaptability of the model on new tasks to address cross-task problems.

We also introduce adversarial domain adaptation [33], [34] to solve domain shift. We use the adversarial loss to measure the distance between source and target domain mappings. By optimizing adversarial loss, we minimize the distance between the source and target features. Our model maps image features across domains into a shared space, thus solving the problem of domain shift. To solve cross-domain and cross-task transfer problems simultaneously, we propose a unified network that combines meta-learning with adversarial domain adaptation. Meta-learning enables our model to adapt to new tasks in the source domain quickly. Adversarial domain adaptation reduces the distance between the source domain and the target domain. Hence, our model is capable of quickly learning new tasks in the target domain.

The rest of this paper is organized as follows. Section II introduces the framework, network structure, and optimization objectives of our model. In section III, we present the detailed experimental results based on the real SAR dataset and verify the effectiveness of the model. We conclude this paper in Section IV.

II. METHODOLOGY

We first use the meta-learning and adversarial domain adaptation to pre-train the model. The prior knowledge learned during training initializes the network parameters of our model. Next, we fine-tune the model with a small amount of real data. The fine-tuned model can produce good recognition results in the real dataset.

A. MODEL FRAMEWORK

As illustrated in Fig. 1, our model consists of a classifier, a domain discriminator, and a feature extractor. All three components are parameterized by deep neural networks. During training, we collect simulated images and their class labels from the simulated dataset to form various source tasks. The feature extractor extracts source features from input tasks. Next, the classifier identifies these features to

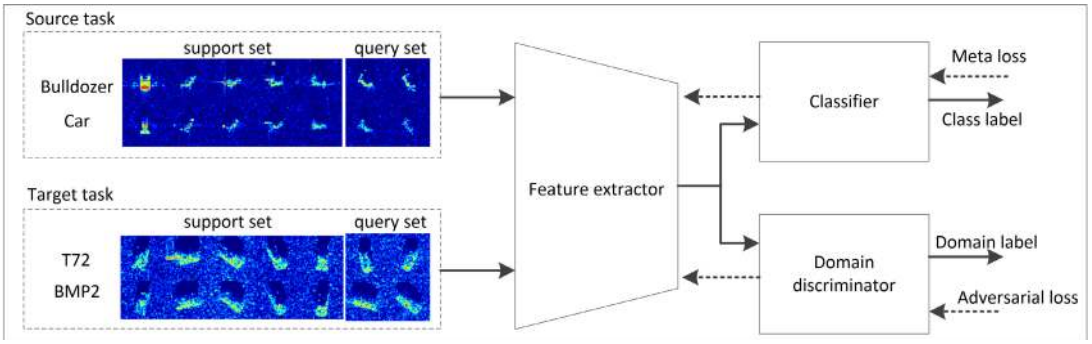


FIGURE 1. The framework of our model. Simulated images and real images are organized into source tasks and target tasks, respectively. The feature extractor extracts image features from the tasks. The classifier identifies the classes of image features, and the discriminator determines whether the image features are from the source domain or the target domain. Dashed arrows indicate the backward gradient flow.

obtain the corresponding class labels and to calculate the meta-learning loss. By minimizing the meta-learning loss on various training tasks, the adaptability of the model to new tasks is improved. The gradients of meta-learning loss are backward to the feature extractor and classifier to update their network parameters. Meanwhile, we collect a few real images to form the target tasks and send them to the feature extractor to obtain target features. Target features and source features are fed into the domain discriminator to predict their domain labels. We minimize adversarial loss to confuse the domain discriminator. When the domain discriminator cannot distinguish between the source features and the target features, the distance between the source domain and target domain is minimal. We calculate the gradients of adversarial loss and back-propagate them in the path indicated in Fig. 1. We optimize meta-loss and adversarial loss jointly so that our model can achieve excellent generalization performance on new tasks in the target domain. This adaptability helps solve cross-domain and cross-task transfer problems.

After training, we discard the domain discriminator and fine-tune the network parameters of the feature extractor and classifier with real data. Using only a small amount of real training data, our model can produce good recognition performance in the real SAR dataset.

B. OPTIMIZATION OBJECTIVE

In this section, we formulate the optimization objectives of the model. We optimize the feature extractor f_θ and classifier f_ϕ through meta-learning while optimizing f_θ and domain discriminator f_ω through adversarial domain adaptation. θ, ϕ and ω represent the network parameters of the three model components. For each training iteration, we sample a source task τ_s from the source domain distribution $p_s(\tau)$, as well as a target task τ_t from the target domain distribution $p_t(\tau)$.The source task τ_s consists of M training samples; thus, $\tau_s = \left\{\left(x_j, y_j\right)\right\}_{j=1}^{M}$, where x and y represent images and the corresponding class labels, respectively. Following the standard meta-learning protocol, we split the samples in the source task τ_s into disjoint training samples and test samples, and call them the support set τ_s^v and the query set τ_s^q, respectively. Thus, $\tau_s = \tau_s^v \cup \tau_s^q$, where $\tau_s^v = \left\{\left(x_j, y_j\right)\right\}_{j=1}^{P}$ and $\tau_s^q = \left\{\left(x_j, y_j\right)\right\}_{j=P+1}^{M}$. The target task τ_t is defined similarly.

1) META-LEARNING

The critical assumption of meta-learning is that tasks drawn from the same domain distribution share a common structure that can be used for fast adaptation of new tasks. Let us denote the parameters set of feature extractor f_θ and classifier f_ϕ as (θ, ϕ). We use one or more gradient descent steps on the support set to optimize (θ, ϕ) so that it achieves good performance on the query set. For each task τ_s, we use the gradient of classification loss of τ_s^v to update (θ, ϕ). (θ, ϕ) is updated as follows:

$$(\theta, \phi) = (\theta, \phi) - \alpha \nabla_{\theta,\phi} L_{\tau_s^v}\left(f_{\theta,\phi}\right) \tag{1}$$

where $\nabla_{\theta,\phi}$ denotes a gradient operation for θ and ϕ.The gradient step size α is a hyperparameter. In practice, we repeat (1) several times to obtain a better result. $L_{\tau_s^v}\left(f_{\theta,\phi}\right)$ is the classification loss of τ_s^v calculated by the cross-entropy function; thus, it is defined as:

$$L_{\tau_s^v}\left(f_{\theta,\phi}\right) = -\sum_{x_j, y_j \sim \tau_s^v} y_j * log\left(f_{\theta,\phi}\left(x_j\right)\right) \tag{2}$$

where x_j and y_j denote the j^{th} image and class label in τ_s^v, respectively, $f_{\theta,\phi}\left(x_j\right)$ is the predicted class label of x_j, and the operator $*$ represents the inner product between vectors. Our goal is to improve the recognition performance of the model updated by (1) on the query set τ_s^q. We set the classification loss of τ_s^q as our meta-objective and minimize it in meta-learning. The meta-objective L_{meta} is as follows:

$$\min_{\theta,\phi} L_{meta} = E_{\tau_s \sim p_s(\tau)}\left[L_{\tau_s^q}\left(f_{(\theta,\phi)-\alpha\nabla_{\theta,\phi} L_{\tau_s^v}(f_{\theta,\phi})}\right)\right] \tag{3}$$

where L_{meta} is minimized by training θ and ϕ across tasks sampled from $p_s(\tau)$. $L_{\tau_s^q}$ denotes the classification loss of τ_s^q after using (1) to update the network parameters.

2) ADVERSARIAL DOMAIN ADAPTATION

The distance between the source domain and the target domain is reduced by minimizing an adversarial loss in the feature space. We design two independent objectives to minimize the adversarial loss, one for the feature extractor f_θ and one for the domain discriminator f_ω. The adversarial loss used in [34] is the GAN loss, but the training of GAN is unstable. To improve the stability of training, we introduce the Wasserstein distance [35] as the adversarial loss.

First, we train f_ω to identify whether the image features are from the source domain or the target domain. The discriminator marks the source feature with a positive label and the target feature with a negative label. The training objective of f_ω takes the following form:

$$\min_{\omega} L_{advD} = E_{\tau_t \sim p_t(\tau)}\left[f_\omega\left(f_\theta\left(\tau_t\right)\right)\right] - E_{\tau_s \sim p_s(\tau)}\left[f_\omega\left(f_\theta\left(\tau_s\right)\right)\right] + \lambda \cdot E_\mu\left[\left(\left\|\nabla_\mu f_\omega\left(\mu\right)\right\|_2 - 1\right)^2\right] \quad (4)$$

where $f_\theta(\tau_t)$ denotes the extracted image features of τ_t, $f_\omega(f_\theta(\tau_t))$ represents the domain labels of target images produced by f_ω, and μ is a linear interpolation of the source and target features, which is defined as:

$$\mu = \epsilon f_\theta\left(\tau_s\right) + (1-\epsilon) f_\theta\left(\tau_t\right) \quad (5)$$

where $\epsilon \sim U[0,1]$ is a uniformly distributed random variable. The difference between the first two terms of (4) is the original loss of f_ω. The smaller the original loss is, the stronger the discrimination ability of f_ω. The last term of (4) is a regularized term constraining the gradient norm of $f_\omega(\mu)$. λ is a hyperparameter that balances the original loss and gradient penalty. Details can be found in [35].

Second, we train f_θ to produce domain invariant features to fool f_ω. The training objective of f_θ is:

$$\min_{\theta} L_{advF} = -E_{\tau_t \sim p_t(\tau)}\left[f_\omega\left(f_\theta\left(\tau_t\right)\right)\right] \quad (6)$$

where f_θ is trained to minimize L_{advF} while leaving f_ω fixed. When L_{advF} has not been optimized, the target feature should be labeled as negative by f_ω. By minimizing L_{advF}, the output of f_ω changes from negative to positive, that is, f_ω mistakes the target feature for the source feature. The feature extracted by f_θ can be regarded as domain invariant.

In a training iteration, we begin by sampling τ_s and τ_t from $p_s(\tau)$ and $p_t(\tau)$. Next, we use τ_s and τ_t to evaluate L_{meta}, L_{advD}, and L_{advF} sequentially. Finally, we use the gradient of the losses to update the network parameters. The detailed training process is outlined in TABLE 1.

TABLE 1. Training algorithm.

Input: source task distribution $p_s(\tau)$, target task distribution $p_t(\tau)$, gradient step size α, learning rate β, γ, feature extractor f_θ, classifier f_ϕ,discriminator f_ω
1: randomly initialize parameters θ, ϕ, ω
2: while not converged do
3: sample a source task $\tau_s \sim p_s(\tau)$ and a target task $\tau_t \sim p_t(\tau)$
4: evaluate $L_{\tau_s^v}$ of the support set τ_s^v
5: compute adapted parameters θ, ϕ
$(\theta, \phi) \leftarrow (\theta, \phi) - \alpha \nabla_{\theta,\phi} L_{\tau_s^v}\left(f_{\theta,\phi}\right)$
6: evaluate L_{meta} of τ_s by equation (3)
7: evaluate L_{advD} of τ_s and τ_t by equation (4)
8: evaluate L_{advF} of τ_t by equation (6)
9: update parameters with gradient decent:
10: $\theta \leftarrow \theta - \beta \nabla_\theta L_{meta} - \gamma \nabla_\theta L_{advF}$
11: $\phi \leftarrow \phi - \beta \nabla_\phi L_{meta}$
12: $\omega \leftarrow \omega - \gamma \nabla_\omega L_{advD}$
13: end while

C. NETWORK STRUCTURE

Fig. 2 illustrates the network structure of our model. The feature extractor consists of 4 convolutional modules. Each convolutional module consists of a convolutional (CONV) layer, a batch normalization (BN) layer [36], a rectified linear unit (ReLU) layer, and a 2×2 max-pooling layer. BN plays a vital role in our network. It can accelerate the convergence of the network and effectively improve the stability of training

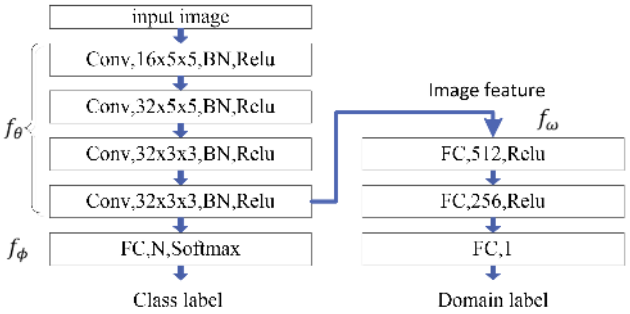


FIGURE 2. Network structure of our model.

when there are few training samples. The classifier is implemented by a single fully connected (FC) layer. The domain discriminator is a cascaded network with three FC layers.

We preprocess the raw SAR amplitude images to a log scale and normalize the pixels to $[-1, 1]$. The images are first cropped into a size of 64 × 64 to reduce clutter interference while preserving the target characteristics. Next, the cropped images are fed into the feature extractor. In the first convolutional module, inputs are filtered by 16 filters with a kernel size of 5×5. Since the convolutional layer uses spatial zero paddings and a convolutional stride of 1 pixel, the outputs of this layer are 16 feature maps of size 64 ×64. After being processed by the BN layer and the ReLU layer, the outputs are downsampled at the max-pooling layer to obtain 16 feature maps of size 32 ×32. The outputs of the last convolutional module are 32 feature maps of the size 4 ×4. We flatten the output feature maps into feature vectors that can be subsequently processed by the classifier or discriminator. The classifier reduces the dimension of features to N and uses the softmax function to produce class labels. N represents the number of classes included in a single task. In the discriminator, the features are processed by three FC layers sequentially to produce domain labels. We use Xavier [37] to initialize the parameters of the network as uniformly distributed random

variables. That is,

$$\Theta \sim U\left(-\sqrt{\frac{6}{n_i+n_o}}, \sqrt{\frac{6}{n_i+n_o}}\right) \tag{7}$$

where Θ denotes the initialized parameters, U represents a uniform distribution, n_i is the input dimension of a layer in the network, and n_o is the output dimension of this layer.

III. EXPERIMENTAL RESULTS

A. DATASETS

During testing, we use a small amount of real SAR data to fine-tune the model and test it on the remaining real SAR data. We use the moving and stationary target acquisition and recognition (MSTAR) dataset, which was collected and released by the U.S. Air Force Research Laboratory, as the real SAR dataset. A variety of ground targets are imaged by an HH-polarized X-band SAR system operating in spotlight mode. All images have a resolution of 0.3 m $\times$ 0.3 m and a size of 128 $\times$128 pixels. These target images are acquired at full azimuth angles ($0° - 360°$) and limited depression angles (15°, 17°, 30°, and 45°). The publicly released MSTAR dataset contains ten classes of military vehicles, such as battle tanks and infantry fighting vehicles. Because of the authority and completeness of its data, the MSTAR dataset is widely used to compare various SAR-ATR approaches. We summarize the classes, numbers, and depression angles of the targets available for training and testing in TABLE 2.

TABLE 2. Description of the real SAR dataset.

		Training		Test	
Class	Target	Depression	Number	Depression	Number
1	T72	17°	232	15°	196
2	BMP2	17°	233	15°	196
3	BTR60	17°	256	15°	195
4	BTR70	17°	233	15°	196
5	2S1	17°	299	15°	274
6	BRDM2	17°	298	15°	274
7	T62	17°	299	15°	273
8	D7	17°	299	15°	274
9	ZSU234	17°	299	15°	274
10	ZIL131	17°	299	15°	274

The simulated SAR images were provided by researchers at the Technical University of Denmark. They built a CAD model of the target and used commercial electromagnetic simulation software to obtain the radar cross-section (RCS) of the target. A self-developed postprocessing software program was used to generate simulated images containing the target's RCS, speckle noise, and ground clutter. The simulation software parameters were set by referring to the imaging parameters of the MSTAR dataset. The simulated dataset contains seven types of targets, including cars, trucks, and bulldozers. Each type of target is imaged using two different CAD models. Hence, there are a total of fourteen classes of targets in the simulated dataset. The simulated images are acquired at various depression angles (15°, 17°, 25°, 30°, 35°, 40°, and 45°) over different azimuth angles ($0° - 360°$, with a step of 5°). Compared with the MSTAR dataset, the target classes and angles of the simulated dataset are more diverse. TABLE 3 lists the classes of simulated images, the CAD model numbers used, and the number of images per class.

TABLE 3. Description of the simulated SAR dataset.

Target	Class	Model	Number
Bulldozer	1	8020	504
	2	13013	504
Bus	3	30726	504
	4	55473	504
Car	5	Toyota	504
	6	Peugeot	504
Humvee	7	3663	504
	8	9657	504
Motorbike	9	3651	504
	10	3972	504
Tank	11	65047	504
	12	86347	504
Truck	13	2096	504
	14	2107	504

B. TRAINING DETAILS

According to the standard meta-learning setup, the inputs of our model are organized into N-way, K-shot tasks. To form a task, we randomly select N classes in the dataset and then extract L images in each class. The N*K images in a task form a support set, and the remaining N*(L-K) images form a query set. It is worth noting that the image labels in the tasks are randomly assigned, which forces the model to learn the task-general features. During testing, we randomly extract training samples from the real dataset to form input batches. Then, we use stochastic gradient descent to fine-tune the model with a learning rate of ρ and a batch size of σ. The image labels used during fine-tuning correspond to the classes of targets rather than being randomly assigned. The hyperparameters of our model are searched through cross-validation. We list the detailed hyperparameter settings in TABLE 4.

TABLE 4. Hyperparameters settings.

hyperparameters	values
Number of classes selected in one task (N)	10
Number of images per class (L)	20
Number of images per class in the support set (K)	5
Gradient step size (α)	0.01
Meta-learning learning rate (β)	0.002
Domain adaptation learning rate (γ)	0.001
Gradient penalty coefficient (λ)	10
Fine-tuning learning rate (ρ)	0.01
Fine-tuning batch size (σ)	50

C. TARGET RECOGNITION UNDER SOC

In this experiment, we evaluate the ten-class target recognition performance of our model under standard operating conditions (SOC). SOC refers to the acquisition of training and test images under similar conditions. The serial number, configuration, and depression angle of the same class of images are similar for the training set and test set. After training, images at a depression angle of 17° are used to fine-tune the model, and the images at a depression angle of 15° are used to test it. Detailed classes and numbers of images are shown in TABLE 2.

We compare the recognition performance of three methods under the same network structure and the same number of training samples. These methods are CNN without pre-training (denoted CNN_ORG), CNN pre-trained with network-based transfer learning (denoted CNN_TF) [28], and CNN pre-trained with meta-learning and domain adaptation (our model). The accuracy curves of these methods are plotted in Fig. 3. When using only 10% of the training set, the recognition accuracies of our model and CNN_TF are 88.9% and 81.3%, respectively, which is much higher than the 76.9% of CNN_ORG. When using the complete training set, the accuracy of our method is 98.5%, while the accuracies of CNN_TF and CNN_ORG are 95.6% and 95.2%, respectively. Compared with training from scratch, pre-training improves the recognition performance of CNN. The smaller the amount of data used for training, the higher the advantage of the pre-training method.

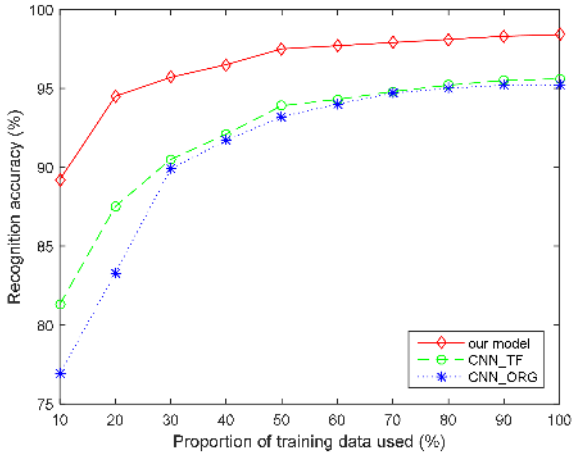


FIGURE 3. Recognition accuracy of three methods under the SOC test.

The pre-training on the simulated data allows CNN to learn some general features, such as edges and textures. These transferable features are also applicable to real data, which provides a good starting point for fine-tuning on real data. Fig. 3 shows that as the quantity of training data increases, the accuracies of CNN_TF and CNN_ORG tend to be consistent. The accuracy of our model is always higher than those of CNN_TF and CNN_ORG. Our model uses meta-learning and domain adaptation for pre-training. Hence it achieves better recognition results in such cross-domain and cross-task transfer scenarios.

TABLE 5. Comparison of different methods under the SOC test.

Method	Accuracy
Our model	98.5±0.1%
CNN_TF	95.6%
MCNN	98.2%
A-ConvNet	99.1%
KSR	93.9%
TJSR	93.4%
CDSPP	91.0%
KRLDP	95.1%

We also compare our model with the recognition methods proposed in recent years. The methods used for comparison include CNN-based methods (A-ConvNet [16] and MCNN [19]), sparse representation-based methods (KSR [3] and TJSR [4]), and locality-preserving projection-based methods (CDSPP [5] and KRLDP [6]). The recognition accuracies of all methods are obtained on the complete training dataset. As shown in TABLE 5, our model is superior to most methods according to the SOC test. The ± shows 95% confidence intervals of accuracy. Moreover, the average accuracy of these methods is approximately 95%. As shown in Fig.3, our model can achieve this accuracy, with only 30% of the training dataset. Our model performs well, even if the real training data are scarce, which is quite practical in the SAR-ATR field.

D. DTARGET RECOGNITION UNDER DEPRESSION VARIATIONS

In this experiment, we evaluate the recognition performance of the model under significant depression angle variations, which is one of the extended operating conditions (EOC) tests. Due to the unique imaging mechanism of SAR, SAR images are sensitive to the depression angle variations of targets. The change in the depression angle has a significant influence on the appearance of the target image. We compare the images of the same target at different depression angles in Fig. 4. When the depression angle of the target changes from 17° to 30°, the intensity and position of the scattering centers of the images change slightly. When the depression angle changes to 45°, the shape, shadow, and even the noise intensity of the image change drastically. To explore the robustness of our model, we use target images collected at a depression angle of 17° for training and use the images collected at depression angles of 30° and 45° for testing. Referring to the experimental setup of [3], [4], we select three classes of targets, 2S1, BRDM2, and ZSU234, for training and testing. The class and numbers of target images used for training and testing are summarized in TABLE 6.

We compare the recognition performance of three methods at a depression angle of 30° in Fig. 5. When only 10% of the

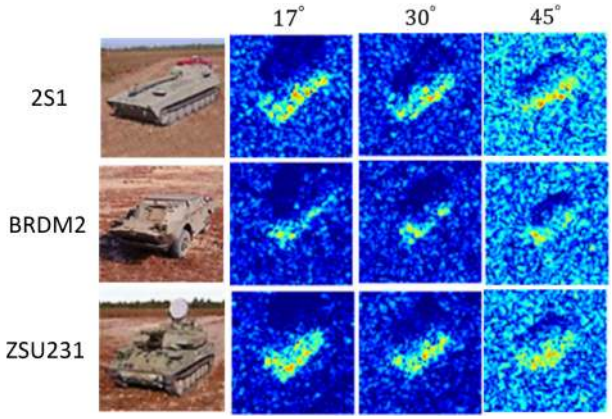


FIGURE 4. Examples of three targets at different depression angles.

TABLE 6. Dataset for recognition under various depression angles.

Target	2S1	BRDM2	ZSU234
Training(17°)	299	298	299
Test(30°)	288	287	288
Test(45°)	303	303	303

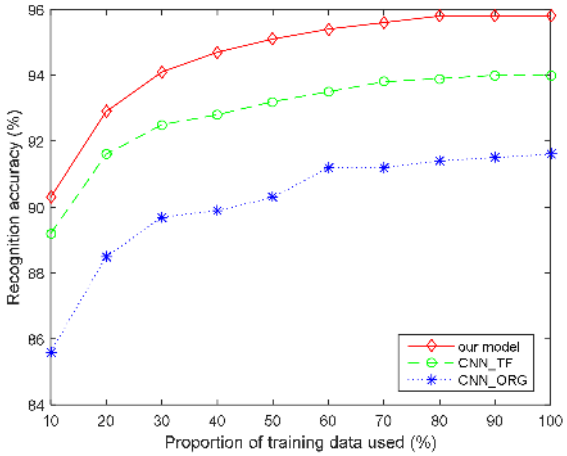


FIGURE 5. Recognition accuracy of three methods at a depression angle of 30.

training data is used, the recognition accuracies of our model, CNN_TF, and CNN_ORG are 90.3%, 89.2%, and 85.6%, respectively. If all the training data is used, the recognition accuracies of the three methods increase to 95.8%, 94.0%, and 91.6%, respectively. The recognition accuracy increases with the number of training samples. Our model is superior to the other two methods. The recognition performance of our model is similar to that under the SOC test. When the depression angle of the training data and the test data do not differ considerably (17° versus 30°), the recognition performance of our model will not decrease significantly.

As shown in Fig.6, when the depression angle of the test images changes from 30° to 45°, the recognition accuracies of the three methods decrease significantly. When using 10% of the training data, the recognition accuracies of our model, CNN_TF, and CNN_ORG are 65.6%, 65.3%, and 52.3%,

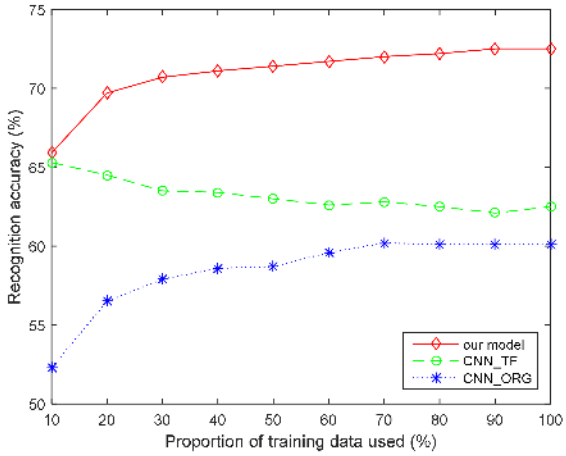


FIGURE 6. Recognition accuracy of three methods at a depression angle of 45.

respectively. The accuracies of the three methods based on 100% of the training data are 72.5%, 62.5%, and 60.1%, respectively. Significant variations in depression angles lead to dramatic changes in the appearance of the target, which results in a considerable difference between the training images and the test images. In this case, fine-tuning has a limited effect on recognition accuracy. We observe that with the increase in the quantity of training data used, the recognition accuracy of CNN_TF decreases. It is possible that network-based transfer learning is not suitable for scenarios in which the training images are very different from the test images.

We also compare the model with several SAR-ATR methods at different depression angles. The recognition accuracies based on all training data are shown in TABLE 7. Compared with the SOC test results, the recognition accuracy of each method in this EOC test has decreased. The considerable difference between the training images and the test images will degrade recognition performance. The accuracies of our model at 30° and 45° are 95.8% and 72.5%, respectively, which are better than those of most methods. In addition, the average accuracies of the SAR-ATR methods at 30° and 45° are 94.8% and 71.0%, respectively. Our model can obtain these accuracies with only 50% of the training data. Even if the depression angle of the target varies dramatically, our model can achieve reasonable performance without a large quantity of training data.

TABLE 7. Comparison of methods under various depression angles.

Method	Accuracy	
	30°	45°
Our model	95.8±0.2%	72.5±0.5%
CNN_TF	94.0%	62.5%
MCNN	95.7%	--
KSR	93.8%	72.8%
TJSR	95.2%	70.7%
CDSPP	92.3%	77.2%
KRLDP	98.1%	--

E. TARGET RECOGNITION UNDER CONFIGURATION VARIATIONS

Differences in the configuration of the same target, such as whether the target equips with fuel barrels, fenders, or smoke grenade launchers, will lead to variances in the corresponding SAR images. These variances will degrade the performance of recognition methods. In this EOC test, several targets under different configurations are used for recognition. We select four targets, T72, T62, BMP2, and BTR60, for training and testing, where T72 and BMP2 contain three variants denoted by different serial numbers. The details of the training and test samples are summarized in TABLE 8.

TABLE 8. Dataset for recognition under configuration variations.

Class	Training(17°)		Test(15°)	
	Serial Number	Number	Serial Number	Number
T72	132	232	S7, 812	386
T62	A51	299	A51	273
BMP2	9563	233	C21, 9566	392
BTR60	k10yt7532	256	k10yt7532	195

The difference between the test image and training image is mainly due to two factors. First, their depression angles are slightly different, and second, their appearance varies due to configuration differences. Compared with the SOC test, these two differences will result in a decrease in the recognition accuracy. When using all the training data, we compare our model with other recognition methods. The recognition results are shown in TABLE 9. Benefiting from pre-training using meta-learning and domain adaptation, our model is robust to the variations in the target configuration. Under the same network structure, our model is superior to CNN_TF and CNN_ORG. Compared with other traditional SAR-ATR methods, our model achieves state-of-the-art results.

TABLE 9. Comparison of methods under configuration variations.

Method	Accuracy
Our model	94.4±0.3%
CNN_TF	91.7%
CNN_ORG	90.0%
KSR	93.9%
TJSR	91.2%
KRLDP	92.2%
MSRC [11]	87.5%

F. COMPARISON OF THE TRAINING CONVERGENCE

In this experiment, we compare the convergence speed of the proposed model with those of CNN_TF and CNN_ORG. We evaluate the convergence speed in two cases, one using 100% of the training data and the other using 50% of the training data. As shown in Fig. 7, when using 50% of the training data, the three methods need to train 14, 25, 40 epochs to converge. When using 100% of training data, the convergence epochs of the three methods decrease to 10, 20, and 30, respectively. Compared with training from scratch, the pre-training-based methods converge much faster. Benefiting from the excellent features of meta-learning, our model converges faster than CNN_TF. Our model can quickly adapt to new tasks with a few fine-tuning steps, which is a considerable advantage in practical applications in which there are many more targets than in the MSTAR dataset. Moreover, the number of targets in practical applications is not fixed but will continue to increase. Instead of training from scratch, our model can adapt to the expanded dataset by fine-tuning with the new target images.

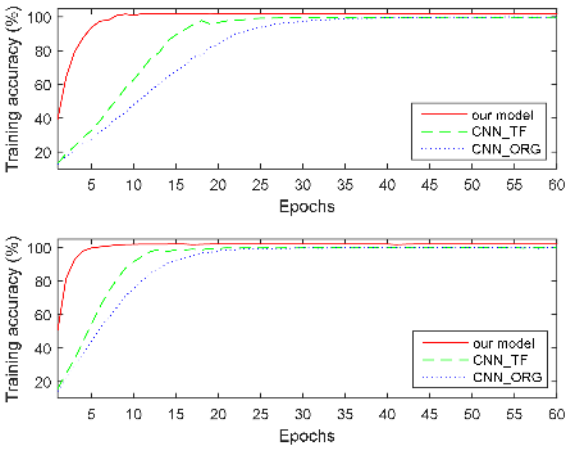


FIGURE 7. Comparison of the training convergence using (top) 50% of the training data (bottom) 100% of the training data.

IV. CONCLUSION

In practice, CNN-based SAR-ATR methods lack sufficient real SAR images to train networks. To alleviate this problem, we employ simulated SAR data to expand the training dataset. We propose a unified model integrating meta-learning and adversarial domain adaptation to transfer prior knowledge from simulated SAR data to real SAR data. By introducing meta-learning and domain adaptation into the application scenario of transfer learning, our model addresses the cross-domain and cross-task transfer problem. Experimental results on the MSTAR dataset verify that our model is superior to the network-based transfer learning method and non-pre-training method. Compared with traditional SAT-ATR methods, our model achieves better performance with fewer training data.

REFERENCES

[1] B. Ding, G. Wen, C. Ma, and X. Yang, "Target recognition in synthetic aperture radar images using binary morphological operations," *Proc. SPIE*, vol. 10, no. 4, Oct. 2017, Art. no. 046006.
[2] R. Zhang and M. Zhang, "SAR target recognition based on active contour without edges," *J. Syst. Eng. Electron.*, vol. 28, no. 2, pp. 276–281, Apr. 2017.
[3] G. Dong and G. Kuang, "SAR target recognition via sparse representation of monogenic signal on Grassmann manifolds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 3, pp. 1308–1319, Mar. 2016.
[4] G. Dong, G. Kuang, N. Wang, L. Zhao, and J. Lu, "SAR target recognition via joint sparse representation of monogenic signal," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 8, no. 7, pp. 3316–3328, Jul. 2015.

[5] M. Liu, S. Chen, J. Wu, F. Lu, J. Wang, and T. Yang, "Configuration recognition via class-dependent structure preserving projections with application to targets in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 6, pp. 2134–2146, Jun. 2018.

[6] M. Yu, S. Zhang, G. Dong, L. Zhao, and G. Kuang, "Target recognition in SAR image based on robust locality discriminant projection," *IET Radar Sonar Navig.*, vol. 12, no. 11, pp. 1285–1293, Jun. 2018.

[7] X. Y. Huang, H. Qiao, and B. Zhang, "SAR target configuration recognition using tensor global and local discriminant embedding," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 222–226, Feb. 2016.

[8] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[9] Q. Zhao and J. C. Principe, "Support vector machines for SAR automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 2, pp. 643–654, Apr. 2001.

[10] C. Nilubol, R. M. Mersereau, and M. J. T. Smith, "A SAR target classifier using radon transforms and hidden Markov models," *Digital Signal Processing*, vol. 12, nos. 2–3, pp. 274–283, 2002.

[11] 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.

[12] J. R. Diemunsch and J. Wissinger, "Moving and stationary target acquisition and recognition (MSTAR) model-based automatic target recognition: Search technology for a robust ATR," *Proc. SPIE*, vol. 3370, pp. 481–491, Sep. 1998.

[13] Z. Jianxiong, S. Zhiguang, C. Xiao, and F. Qiang, "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.

[14] B. Ding and G. Wen, "A region matching approach based on 3-D scattering center model with application to SAR target recognition," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4623–4632, Jun. 2018.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[16] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

[17] Y. Kwak, W.-J. Song, and S.-E. Kim, "Speckle-noise-invariant convolutional neural network for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 549–553, Apr. 2019.

[18] S. A. Wagner, "SAR ATR by a combination of convolutional neural network and support vector machines," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 6, pp. 2861–2872, Dec. 2016.

[19] R. Min, H. Lan, Z. Cao, and Z. Cui, "A gradually distilled CNN for SAR target recognition," *IEEE Access*, vol. 7, pp. 42190–42200, 2019.

[20] J. H. Cho and C. G. Park, "Multiple feature aggregation using convolutional neural networks for SAR image-based automatic target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 12, pp. 1882–1886, Dec. 2018.

[21] S. Deng, L. Du, C. Li, J. Ding, and H. Liu, "SAR automatic target recognition based on Euclidean distance restricted autoencoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 7, pp. 3323–3333, Jul. 2017.

[22] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.

[23] J. Pei, Y. Huang, W. Huo, Y. Zhang, J. Yang, and T.-S. Yeo, "SAR automatic target recognition based on multiview deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2196–2210, Apr. 2018.

[24] J. Guo, B. Lei, C. Ding, and Y. Zhang, "Synthetic aperture radar image synthesis by using generative adversarial nets," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1111–1115, Jul. 2017.

[25] K. Wang, G. Zhang, Y. Leng, and H. Leung, "Synthetic aperture radar image generation with deep generative models," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 912–916, Jun. 2019.

[26] C. Kang and C. He, "SAR image classification based on the multilayer network and transfer learning of mid-level representations," in *Proc. Int. Geosci. Remote Sens. Symp. (IGARSS)*, Beijing, China, Jul. 2016, pp. 1146–1149.

[27] C. Zhong, X. Mu, X. He, J. Wang, and M. Zhu, "SAR target image classification based on transfer learning and model compression," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 412–416, Mar. 2019.

[28] D. Malmgren-Hansen, A. Kusk, J. Dall, A. A. Nielsen, R. Engholm, and H. Skriver, "Improving SAR automatic target recognition models with transfer learning from simulated data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1484–1488, Sep. 2017.

[29] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2014, pp. 2481–2488.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3320–3328.

[31] A. Kusk, A. Abulaitijiang, and J. Dall, "Synthetic SAR image generation using sensor, terrain and target models," in *Proc. EUSAR*, Hamburg, Germany, Jun. 2016, pp. 1–5.

[32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, vol. 70, Aug. 2017, pp. 1126–1135.

[33] G. Yaroslav and V. Lempitsky, "Unsupervised domain adaptation by backpropagation.," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, Lille, France, vol. 37, Jul. 2015, pp. 1180–1189.

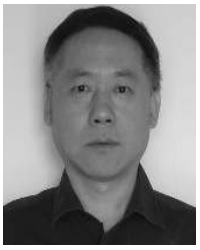
[34] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 7167–7176.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.* Long Beach, CA, USA, 2017, pp. 5767–5777.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, vol. 37, Jul. 2015, pp. 448–456.

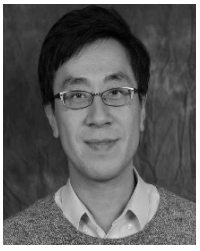
[37] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

KE WANG received the B.S. and M.S. degrees in electronic and information engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003 and 2006, respectively, where he is currently pursuing the Ph.D. degree in communication and information system. His research interests include SAR image detection, SAR image recognition, and machine learning.



GONG ZHANG (M'07) received the Ph.D. degree in electronic engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2002.

From 1990 to 1998, he was a member of Technical Staff with the No724 Institute of China Shipbuilding Industry Corporation, Nanjing. Since 1998, he has been with the College of Electronics and Information Engineering, NUAA, where he is currently a Professor. His research interests include radar signal processing and compressive sensing. He is a member of the Committee of Electromagnetic Information, Chinese Society of Astronautics, and a Senior Member of the Chinese Institute of Electronics.



HENRY LEUNG (F'15) was a Defense Scientist with the Department of National Defense (DND), Canada. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Calgary. His current research interests include information fusion, machine learning, the IoT, nonlinear dynamics, robotics, and signal and image processing. He is a Fellow of SPIE. He is also an Associate Editor of the *IEEE Circuits and Systems Magazine* and the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS. He is a topic Editor on Robotic Sensors of the *International Journal of Advanced Robotic Systems*. He is also an Editor of the *Information Fusion and Data Science* (Springer book series).