

Electromagnetic Simulation Aided SAR Target Classification Via Deep Domain Adaptation

Xiaoling Lv

Aerospace Information Research
Institute, Chinese Academy of Sciences
School of Electronic, Electrical and
Communication Engineering, University
of Chinese Academy of Sciences
Beijing, China
lvxiaoling19@mails.ucas.ac.cn

Xiaolan Qiu

Key Laboratory of Technology in
Geo-Spatial Information Processing
and Application System
Suzhou Aerospace Information
Research Institute
Suzhou, China
xlqiu@mail.ie.ac.cn

Wenming Yu

State key Laboratory of Millimeter Waves
Southeast University
Nanjing, China
wmyu@seu.edu.cn

Abstract—Convolutional neural networks (CNNs) have made tremendous success in optical images classification recently. However, in synthetic aperture radar (SAR) target classification, it is difficult to annotate a large amount of real SAR images to train CNNs. Sufficient annotated images can be easily obtained through simulation, but the disparity between the simulated images and the real images makes them difficult to directly apply to the real images classification. In this paper, we propose a model that integrates multi-kernel maximum mean discrepancy (MK-MMD) and domain-adversarial training to alleviate this problem. Simulated SAR images with annotation and unlabeled real SAR images are used to train our model. First, we use domain-adversarial training to prompt the model to extract domain-invariant features. Then, the MK-MMD between the hidden representations of simulated images and real images is reduced to narrow domain discrepancy. Experimental results on the real SAR dataset demonstrate that our method effectively solves the domain shift problem and improves the classification accuracy.

Index Terms—Synthetic aperture radar (SAR), unsupervised domain adaptation (UDA), target classification, convolutional neural network (CNN)

I. INTRODUCTION

Synthetic aperture radar (SAR) automatic target recognition (ATR) is an important part of SAR image interpretation, mainly composed of three stages: detection, discrimination, and classification. With the rapid development of deep learning in recent years, many excellent algorithms based on convolutional neural network (CNN) have also emerged in SAR target classification. Due to the limited SAR data and the sensitivity to observation conditions, training CNN directly with SAR data is easy to cause overfitting. Chen *et al.* proposed an all-convolutional network (A-ConvNets), using convolutional layers instead of fully connected layers to reduce the number of free parameters, and achieved an average accuracy of 99% on a ten-category classification task [1]. Pan *et al.* proposed a Siamese network based on metric learning for SAR target classification with few training samples, which increases the amount of training data since the input is the sample pair [2]. But these methods still require a number of labeled real

images. Obtaining annotated real SAR data under different imaging conditions is very expensive and time-consuming.

To solve the problem of insufficient measured data, using simulated SAR images as the training samples may be a good solution. Through electromagnetic simulation technology with 3-D models of different targets, the full-aspect SAR images can be acquired. But in many cases, even if visually it is difficult to distinguish between simulated SAR images and real SAR images, CNN trained on simulated SAR images is hard to classify real SAR targets accurately. Song *et al.* proposed a method of nonessential factor suppression to realize simulation-aided zero-shot learning for SAR ATR [3]. A series of preprocessing, including non-maximum suppression, style adjustment, and segmentation, is operated on the input to narrow the disparity between simulation data and measured data. However, only the T72 target, one of the ten categories in the dataset, is replaced by simulation images in this experiment, which is limited in verifying the support effect of simulated images on real images classification.

To achieve performance improvement in simulation-aided SAR target classification, it is necessary to narrow the distance between the feature distributions of simulated images and real images. Unsupervised domain adaptation (UDA) may be an effective approach, which focuses on the information transfer from the labeled source domain to the unlabeled target domain and common feature extraction methods between different domains. Existing UDA methods can be divided into four main types: discrepancy-based, adversarial-based, reconstruction-based, and sample-generation-based. For example, deep adaptation networks (DAN) can learn transferable features with statistical guarantees by using an optimal multi-kernel selection method for mean embedding match [4]. Ganin *et al.* proposed a new back-propagation method based on the gradient reverse layer (GRL), by which traditional CNNs can learn discriminative and domain-invariant features [5].

Mention the application of domain adaptation in SAR target classification task, Huang *et al.* elaborately discussed three issues: what to transfer, where to transfer, and how to transfer and proposed a transitive transfer method based on multi-

This work was supported by the National Natural Science Foundation of China under Grant 61991421.

source data with domain adaptation [6]. Zhang *et al.* proposed a multi-level domain adaptation method to accomplish multi-band SAR image classification. These indicate that domain adaptation methods have great application prospects in SAR image classification. But the research of simulation-aided SAR target classification using UDA hasn't been reported yet to the authors' knowledge.

In this paper, we conducted simulation-aided SAR target classification via domain adaptation and proposed a model integrating domain-adversarial training and discrepancy metric, which effectively improved the accuracy of classification.

The rest of this paper is organized as follows. Section II introduces the proposed method in detail, including the model framework, the network architecture, and the optimization objective. The experimental data and results are depicted in Section III. Section IV gives the conclusion.

II. METHODS

In this paper, we propose a simulation-aided SAR target classification method via unsupervised domain adaptation, which combines domain-adversarial training and multi-kernel maximum mean discrepancy (MK-MMD) [9]. As is shown in Fig. 1, the overall framework can be divided into three parts: feature extractor, target classifier, and domain discriminator.

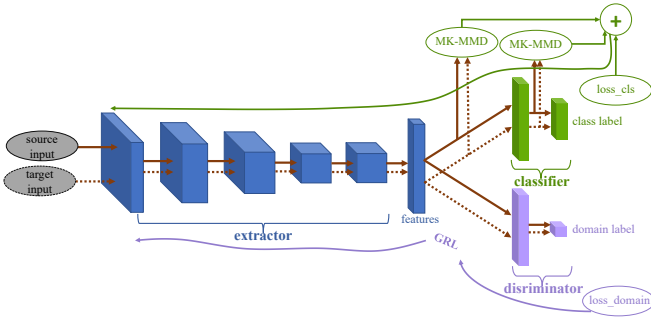


Fig. 1. The overall framework of our model.

A. Model Framework

In supervised learning, the model receives n labeled samples $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$ from $X \times Y$, where X is the input space and Y is the output space, which is $\{1, 2, \dots, N\}$ in classification task. N represents the number of categories. In unsupervised domain adaptation, there are two different data distributions, the source domain S and the target domain T . The model is trained on labeled samples $\hat{S} = \{(\mathbf{x}_i^s, y_i^s)_{i=1}^n\}$ from the source domain and unlabeled samples $\hat{T} = \{(\mathbf{x}_i^t)_{i=1}^m\}$ from the target domain. To distinguish the data from the source domain or the target domain, we use d_i to represent the domain label of the i -th sample, that is, $d_i = 0$ if $\mathbf{x}_i \sim S$ and $d_i = 1$ if $\mathbf{x}_i \sim T$.

For each input $\mathbf{x}$, the model predicts its category label $\hat{y}$ and its domain label $\hat{d}$. θ_e , θ_c , and θ_d stand for the parameters of extractor, classifier and discriminator respectively. The input $\mathbf{x}$ is mapped by the mapping G_e to a D-dimensional feature

vector $\mathbf{f}$, i.e. $\mathbf{f} = G_e(\mathbf{x}; \theta_e)$. Then the predicted label $\hat{y}$ is obtained as $\hat{y} = G_c(\mathbf{f}; \theta_c)$. Likewise, $\hat{d} = G_d(\mathbf{f}; \theta_d)$.

During the training procedure, the simulation data and the real data are fed into the model simultaneously. The classification loss $Loss_{cls}$ is calculated based on the truth label y_i^s and the output $\hat{y}_i^s$. Meanwhile, the MK-MMD is calculated at the hidden layers of the classifier. To help the extractor get domain-invariant features, we reverse the gradient from the domain discriminator before passing it to the extractor in back-propagation. In summary, we optimize classification loss of the source training data, the MK-MMD, and domain-adversarial loss jointly to achieve excellent performance on the task transferred from simulation data to real data.

B. Network Architecture

Fig. 2 illustrates the network architecture used in this paper. The feature extractor refers to the feature module design of Alexnet [8], including five convolutional layers. The discriminator and the classifier are both composed of two fully connected layers.

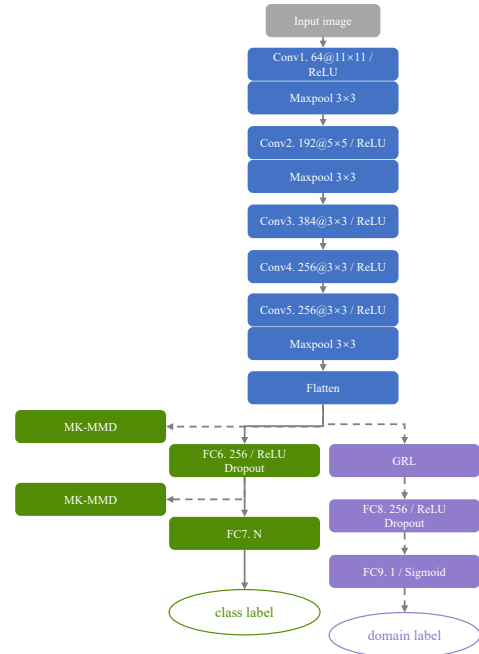


Fig. 2. The network architecture of our model. The approaches shown in the dotted line are only activated in training.

In the experiment, we first normalize the raw SAR amplitude value into $[0, 1]$. Subsequently, the images are centrally cropped into 88×88 and resized into 224×224 . The outputs of the extractor are 256 feature maps with the size of 6×6 . The feature maps are flattened before they are fed into the classifier and discriminator. The classifier converts the flattened feature into an N-dimensional output, representing the scores for each category. Meanwhile, the discriminator reduces the dimension to 1, which indicates the predicted domain label.

C. Optimization Objective

The optimization objective of our model mainly consists of three parts: classification loss, MK-MMD, and domain-adversarial loss, which are formulated in this section.

Classification loss. Since only the data from the source domain are labeled, the classification loss is calculated on training samples of the source domain. It is defined as:

$$Loss_{cls} = \frac{1}{n} \sum_{\substack{i=1 \\ d_i=0}}^n J_y(G_c(G_e(\mathbf{x}_i; \theta_e); \theta_c), y_i) \quad (1)$$

where $J_y(\cdot)$ is the softmax cross-entropy loss function. We try to minimize $Loss_{cls}$ to get an effective classification model on the source domain.

MK-MMD. In a nutshell, utilizing MK-MMD to measure the distance between the distributions s and t is to calculate the distance between the mean embeddings of s and t in reproducing kernel Hilbert space (RKHS). Denote by $d_k(s, t)$ be the MK-MMD between s and t . The squared formulation is defined as (2).

$$d_k^2(s, t) \triangleq \|E_s[\phi(\mathbf{x}^s)] - E_t[\phi(\mathbf{x}^t)]\|_{H_k}^2 \quad (2)$$

where H_k denotes the RKHS endowed with the kernel k , and $\phi(\cdot)$ is the feature mapping associated with kernel k . The multiple kernels K are formulated as (3), in which the constraints on coefficients $\{\beta_u\}$ are imposed to guarantee that k is characteristic.

$$K \triangleq \{k = \sum_{u=1}^p \beta_u k_u : \sum_{u=1}^p \beta_u = 1, \beta_u \geq 0, \forall u\} \quad (3)$$

By adding an MK-MMD-based multi-layer adaptation regularizer to (1), the distributions of the source and target will become similar under the hidden representations of $fc6 - fc7$. The optimization objective can be described as:

$$\min_{\theta_e, \theta_c} \frac{1}{n} \sum_{\substack{i=1 \\ d_i=0}}^n J_y(G_c(G_e(\mathbf{x}_i; \theta_e); \theta_c), y_i) + \gamma \sum_{l=l_1}^{l_2} d_k^2(D_s^l, D_t^l) \quad (4)$$

where $\gamma > 0$ is a penalty parameter, l_1 and l_2 are the index of layers between which the MK-MMD regularizer is activated, D_*^l is the hidden representation for the input in l -th layer. In our implementation, $\gamma = \frac{2}{1 + \exp(-10 * \frac{j}{epoch})} - 1$, where j is the current epoch number.

Domain-adversarial training. To obtain domain-invariant features, we seek to find an extractor that can “cheat” the domain discriminator. So we add a GRL [5] between the extractor and the discriminator. In forward propagation, the GRL acts as an identity transform. In back-propagation though, it gets the gradient from the subsequent layer, multiplies it by a negative constant, and passes it to the preceding layer. The GRL can be described as (5)-(6).

$$R_\lambda(\mathbf{x}) = \mathbf{x} \quad (5)$$

$$\frac{dR_\lambda}{d\mathbf{x}} = -\lambda \mathbf{I} \quad (6)$$

where $\mathbf{I}$ is an identity matrix. $\lambda = 1$ in our experiment. After introducing the GRL, the optimization objective of domain-adversarial training is shown in (7).

$$\min_{\theta_e, \theta_d} \frac{1}{q} \sum_{i=1}^q J_d(G_d(R_\lambda(G_e(\mathbf{x}_i; \theta_e)); \theta_d), d_i) \quad (7)$$

where $J_d(\cdot)$ is the binary cross-entropy loss function, and q is the number of training samples for both source and target.

In a training iteration, (4) and (7) are alternately executed to ensure that each part of the optimization objective converges smoothly.

III. EXPERIMENTAL RESULTS

A. Data Description and Implementation Details

The moving and stationary target acquisition and recognition (MSTAR) dataset is used as the target in our experiment. Table I shows the number and categories of the target data. The simulation data are generated by the software developed by JCSST Co. LTD and SEU of China using the shooting and bouncing ray tracing method with clutter models. To expand the diversity of samples, we produce simulation images with not only two depression angles but also three backgrounds: grassland automatically generated by the software, grassland and soil simulated according to the optical images provided in the MSTAR dataset, which are denoted by G1, G2, and S respectively in Fig. 3. There are 721 simulation images for each category in each background and each depression angle. The training set and the test set of simulation data are divided according to the ratio of 4:1. Fig. 3 gives the real images and simulated images under the same azimuth and depression.

TABLE I
DESCRIPTION OF THE REAL SAR DATASET

Class	Target	Training		Test	
		Depression	Number	Depression	Number
0	BMP2	17°	233	15°	195
1	BTR70	17°	233	15°	196
2	T72	17°	232	15°	196

Our experiment was carried out in Pytorch 1.4.0, Linux 3.10.0. The hardware is based on Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz CPU and NVIDIA Tesla V100S GPU. The learning rate lr is initially set as $lr_0 = 0.01$ and decrease with the training epochs, i.e., $lr = \frac{lr_0}{(1 + 10 * \frac{j}{epoch})^{0.75}}$.

B. Experimental Results and Analysis

In this experiment, we compare the effects of the MK-MMD-based multi-layer adaptation regularizer and domain-adversarial training on improving classification performance. The results shown in Table II indicate that both MK-MMD and domain-adversarial training work in improving the accuracy of simulation-aided SAR target classification. The confusion matrixes of the MSTAR test set given in Fig. 4 prove that the proposed method improves the classification accuracy of each category.

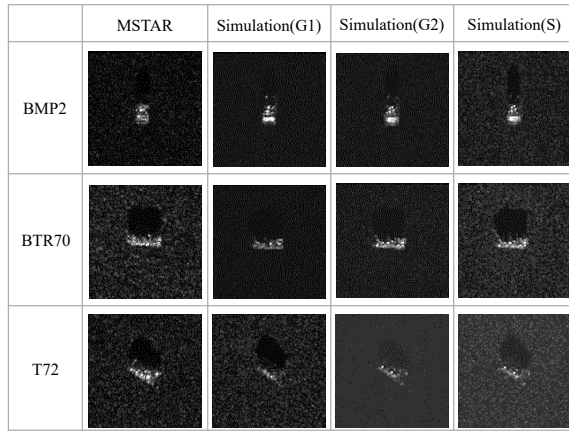


Fig. 3. The real SAR images and simulated images under the same azimuth and depression.

TABLE II
ACCURACY OF THE MSTAR TEST SET USING DIFFERENT METHODS

Methods		Accuracy
MK-MMD	Domain-Adversarial Training	
✗	✗	0.7462
✓	✗	0.7599
✗	✓	0.8092
✓	✓	0.9012

To further demonstrate and analyze the effectiveness of the proposed method, we visualized the features generated by the extractor utilizing t-SNE [10]. By comparing Fig. 5(a) and Fig. 5(b), it can be concluded that the proposed method effectively makes the distribution of similar targets from different domains closer.

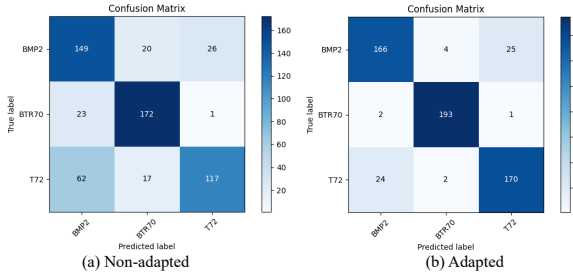


Fig. 4. The confusion matrixes of the MSTAR test set.

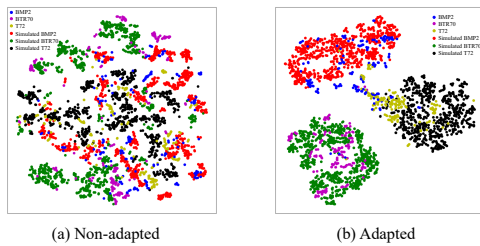


Fig. 5. Visualization of extracted features using t-SNE.

In order to verify the convergence of our model, the trend of loss and accuracy during the training procedure is reported in Fig. 6. We can see that the classification accuracy of the MSTAR test set increased with the decline of the MK-MMD and domain classification accuracy. The trend of the curves proves that the training process of our model is smooth and convergent.

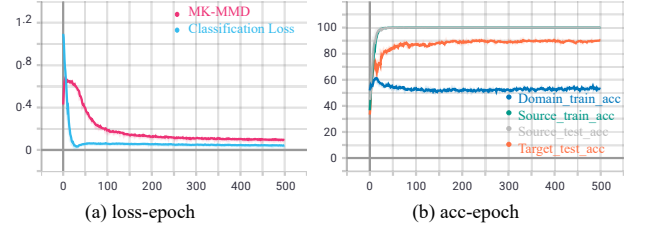


Fig. 6. The loss and accuracy curves during the training procedure.

IV. CONCLUSION

In practice, CNN-based SAR target classification methods are limited due to the lack of labeled real data. To alleviate the problem, we propose a unified model integrating MK-MMD and domain-adversarial training. It transfers the knowledge learned from simulation data to the real SAR data with totally unsupervised settings. Experimental results on the MSTAR dataset demonstrate that our method can effectively boost the performance of electromagnetic-simulation-aided SAR target classification.

REFERENCES

- [1] S. Chen, H. Wang, F. Xu and Y. Jin, "Target Classification Using the Deep Convolutional Networks for SAR Images," in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 8, pp. 4806-4817, Aug. 2016.
- [2] Z. Pan, X. Bao, Y. Zhang, B. Wang, Q. An and B. Lei, "Siamese Network Based Metric Learning for SAR Target Classification," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 1342-1345.
- [3] Q. Song, H. Chen, F. Xu and T. J. Cui, "EM Simulation-Aided Zero-Shot Learning for SAR Automatic Target Recognition," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 6, pp. 1092-1096, June 2020.
- [4] M. Long, Y. Cao, Z. Cao, J. Wang and M. I. Jordan, "Transferable Representation Learning with Deep Adaptation Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 12, pp. 3071-3085, 1 Dec. 2019.
- [5] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," JMLR, vol. 17, no. 1, pp. 2096C2030, 2016.
- [6] Z. Huang, Z. Pan and B. Lei, "What, Where, and How to Transfer in SAR Target Recognition Based on Deep CNNs," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 4, pp. 2324-2336, April 2020.
- [7] W. Zhang, Y. Zhu and Q. Fu, "Adversarial Deep Domain Adaptation for Multi-Band SAR Images Classification," in IEEE Access, vol. 7, pp. 78571-78583, 2019.
- [8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems 25 (2012): 1097-1105.
- [9] Gretton, A., Sriperumbudur, B., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., and Fukumizu, K. "Optimal kernel choice for large-scale two-sample tests," In NIPS, 2012b.
- [10] Laurens Van Der Maaten and Geoffrey Hinton, "Visualizing Data using t-SNE," Journal of Machine Learning Research 9.2605(2008):2579-2605.