Multiscale Deep Neural Network With Two-Stage Loss for SAR Target Recognition With Small Training Set

Jian Guan, *Member, IEEE*, Jiabei Liu, Pengming Feng, *Member, IEEE*, and Wenwu Wang, *Senior Member, IEEE*

***Abstract*— Deep learning models have been used recently for target recognition from synthetic aperture radar (SAR) images. However, the performance of these models tends to deteriorate when only a small number of training samples are available due to the problem of overfitting. To address this problem, we propose a two-stage multiscale densely connected convolutional neural networks (TMDC-CNNs). In the proposed TMDC-CNNs, the overfitting issue is addressed with a novel multiscale densely connected network architecture and a two-stage loss function, which integrated the cosine similarity with the prevailing softmax cross-entropy loss. Experiments were conducted on the MSTAR data set, and the results show that our model offers significant recognition accuracy improvements as compared with other state-of-the-art methods, with severely limited training data. The source codes are available at https://github.com/Stubsx/TMDC-CNNs.**

***Index Terms*— Deep learning, limited data, synthetic aperture radar (SAR), target recognition.**

I. Introduction

SYNTHETIC aperture radar (SAR) has been used in various applications, such as marine surveillance, climate monitoring, and disaster relief [1], for identifying targets from the SAR images collected in the target area [2]. Unlike conventional methods for SAR target recognition, such as [3], deep learning-based methods require a large amount of training data to achieve high performance [4], [5]. However, data labeling is a time-consuming and labor-intensive process and often requires expertise from the application domain. It is often hard to obtain a large amount of labeled data to meet the requirement of conventional deep learning-based methods in real applications. On the other hand, due to the absence of a large-scale labeled SAR data set, and the difficulty of learning with optical images, it is difficult to directly apply the prevalent transfer learning-based methods (e.g., meta-learning [6]) with limited samples to the problem of SAR target recognition. Therefore, research efforts are still needed for SAR target recognition with a small training set, although learning with a limited number of samples is a common problem arising in much wider applications.

Three main ideas have been developed to improve SAR recognition performance with limited training samples. The first idea is to use customized transfer learning. For example, an auto-encoder was used in [7] to learn knowledge from sufficient unlabeled SAR images and transfer it to a labeled SAR data set. The second idea is to focus on model improvement. For example, in [8], features from optimally selected convolutional layers are cascaded and an ensemble learning-based classifier is then used to replace the original softmax layer to achieve more accurate recognition with limited samples. In [9], a light-weight and robust structure based on highway convolutional neural networks (CNNs) is used to reduce the over-fitting problem. In [10], a multiple feature-based lightweight CNNs (MFCNNs) model is presented for SAR target recognition with different ratios of training data. By reducing the complexity of the model, a smaller number of free parameters need to be optimized, and thus the model is less prone to overfitting. The third idea is to use data augmentation techniques. Examples in this category include multilevel reconstruction [11], super-resolution [12], and simulation [13] where the amount of training data is increased via techniques such as mix-up.

However, these methods still have limitations. For example, transfer learning from unlabeled data is less effective than that from labeled data. As shown in [7], this method still needs a fair amount of data, e.g., 50 samples for each category, to achieve acceptable performance. Although model improvement [8]–[10] and data augmentation [11], [13] methods can improve the recognition accuracy, their performance falls behind the transfer learning-based methods. In addition, when the number of available training samples is very small (e.g., less than 20 samples per category), the performance of all these methods drops significantly.

Manuscript received October 28, 2020; revised January 15, 2021 and February 22, 2021; accepted March 4, 2021. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 3072020CFT0602, in part by the Open Research Fund of State Key Laboratory of Space-Ground Integrated Information Technology under Grant 2018_SGIIT_KFJJ_AI_01, and in part by the National Natural Science Foundation of China under Grant 61806018. *(Jian Guan and Jiabei Liu contributed equally to this work.) (Corresponding authors: Jian Guan; Pengming Feng.)*

Jian Guan and Jiabei Liu are with the Group of Intelligent Signal Processing (GISP), College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China (e-mail: j.guan@hrbeu.edu.cn; liu_jabber@outlook.com).

Pengming Feng is with the State Key Laboratory of Space-Ground Integrated Information Technology, China Academy of Space Technology, Beijing 100095, China (e-mail: p.feng.cn@outlook.com).

Wenwu Wang is with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: w.wang@surrey.ac.uk).

Color versions of one or more figures in this letter are available at https://doi.org/10.1109/LGRS.2021.3064578.

Digital Object Identifier 10.1109/LGRS.2021.3064578

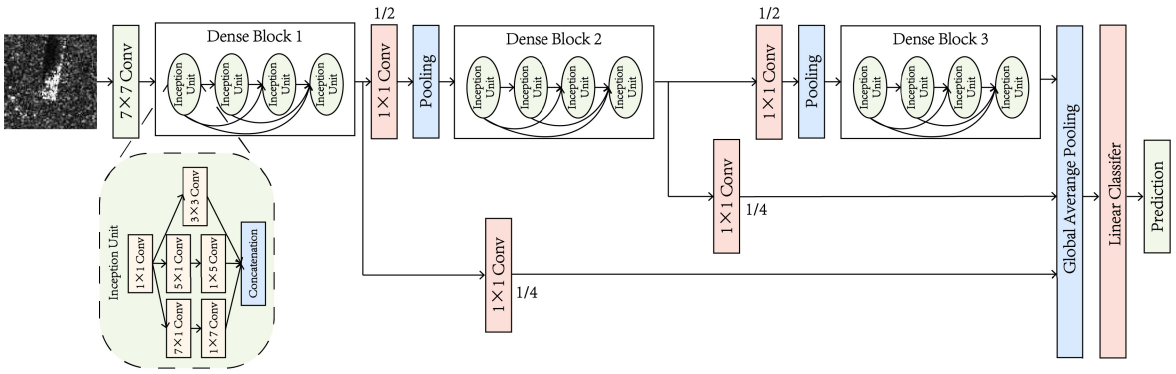


Fig. 1. Framework of the proposed TMDC-CNNs. In this model, each dense block contains 16 Inception units, and each Inception unit outputs 12 channels of the feature maps. The ratio of output number of feature map channels among different convolutional kernels (i.e., 3, 5, 7) is 1:3:8. The 1×1 Conv layers after the first two dense blocks reduce the number of feature map channels to a certain ratio, which is annotated in the figure. Pooling denotes a 2×2 average pooling with stride 2. Only the first 7×7 Conv stride is 2, the others are 1. The linear classifier is a FC layer with an output of K classes.

In this letter, we introduce a new method for SAR target recognition, namely two-stage multiscale densely connected CNNs (TMDC-CNNs), to improve the recognition accuracy with a limited number of training samples. There are two novel aspects in our TMDC-CNNs. First, based on DenseNet [14] and Inception [15], we propose a novel network structure, which provides an improved ability in mitigating the overfitting problem by utilizing features of different scales. Second, we introduce a new two-stage loss function by integrating the cosine similarity loss with the softmax cross-entropy loss. This can help avoid overconfident prediction in the early stage of training and improve the generalization ability of the network.

II. Related Works

A. Network Architecture

Inception [15] and DenseNet [14] are widely used network architectures for target recognition tasks. Inception introduces a network-in-network architecture that can extract features with different scales, whereas DenseNet connects each layer to every other layer in a feedforward manner. However, the gradient vanishing problem still exists in Inception. In DenseNet, since the size of the feature maps in each dense block (i.e., the convolution kernel) is fixed, it is difficult to extract features with different scales. To address these issues, we present a novel network architecture i.e., TMDC-CNNs, as shown in Fig. 1, which can exploit the advantages of both Inception and DenseNet.

B. Deep Learning Loss Function for Target Recognition

Softmax cross-entropy is one of the most commonly used loss functions in deep neural networks. However, it is not quite suitable for small data sets because of the overconfident prediction it may give [16]. In contrast, the cosine loss function is more inclined to minimize the directional error of the model prediction by reparameterizing the cosine similarity [16], [17]. It allows the model to give a smoother prediction [16]. However, since the differences between different SAR targets are not as significant as those among natural images, the direction-based cosine loss often leads to under-fitting [16]. To address these problems, we propose a two-stage loss function that offers advantages in the case of limited training data.

III. Proposed Method

A. TMDC-CNNs Network Architecture

Inspired by Huang *et al.* [14] and Szegedy *et al.* [15], we present a novel network architecture for SAR target recognition. We use the Inception unit rather than the normal convolutional layer in the backbone of DenseNet, such that the proposed architecture can not only alleviate the vanishing gradient problem and strengthen feature propagation but also exploit features with different scales. Moreover, skip connections are added to enhance the ability of the network in utilizing features of different scales in the final prediction. Such a network architecture can alleviate the overfitting problem due to its reduced number of learnable parameters in the model.

The proposed network architecture is given in Fig. 1, which contains three dense blocks, and each block is constructed by several densely connected Inception units. The grid size of the feature map and its channel number are reduced to half of its original size via the convolution layer and pooling layer between adjacent dense blocks. Three kernels with different sizes (i.e., 3, 5, 7) are used to extract the features from different scales in an Inception unit. Denoting $\mathbf{H}_l$ as the output of the lth layer in a dense block, then the output of $(l+1)$th layer $\mathbf{H}_{l+1}$ can be expressed as

$$\mathbf{H}_{l+1} = [[f_7(\mathbf{H}_l), f_5(\mathbf{H}_l), f_3(\mathbf{H}_l)], \mathbf{H}_{l-1}, \ldots, \mathbf{H}_1] \quad (1)$$

where $f_n(\cdot)$ denotes the Conv layer with kernel size of n, and $[\cdot]$ is the concatenation operation on channels. Here, the Conv layer consists of four basic operations, namely, batch normalization, convolution without bias, rectified linear unit (ReLU) activation, and dropout with 0.2 probability. The Inception unit consists of several Conv layers, and its first 1×1 Conv is introduced as the bottleneck layer that produces four times of feature-map dimensions. Then, the feature maps are processed by Conv layers of different kernel sizes. For a kernel of large size, e.g., 7, two Conv layers with a kernel size of 7×1 and 1×7 are used to replace this 7×7 kernel. The bottleneck layer and replacement operations can help reduce model complexity and alleviate over-fitting. Moreover, to utilize features from different grid sizes, the feature maps generated by the first two dense blocks are added to the final fully connected (FC) layer via a 1×1 Conv layer and global average pooling.

B. Two-Stage Loss Function

In this section, we present a two-stage loss function for model training, which can mitigate the overfitting problem caused by insufficient training data. Unlike single-stage softmax cross-entropy loss, we divide the training process into two parts: warm-up and fine-tuning. In the warm-up stage, a new log cosine loss is proposed for model training to alleviate overfitting by reparameterizing the cosine similarity with negative log, which ensures the loss value to be above 0. After this stage, the loss function is switched to the weighted sum of the log cosine loss and softmax cross-entropy to fine-tune the model for better performance. The proposed two-stage loss function can be described as follows:

$$\mathcal{L}_{\text{two-stage}} = \begin{cases} \mathcal{L}_{\text{Cosine}}, & s < \alpha E \\ \mathcal{L}_{\text{Softmax}} + \beta \mathcal{L}_{\text{Cosine}}, & \alpha E \leq s < E \end{cases} \tag{2}$$

where E denotes the total number of epochs, s is the number of trained epochs, α and β are the parameters that control the time to switch between the loss functions and the weight of the log cosine loss in the second stage, respectively. Here, in a single training batch with N instances, the conventional softmax loss $\mathcal{L}_{\text{Softmax}}$ and our reparameterized log cosine loss $\mathcal{L}_{\text{Cosine}}$ can be formulated, respectively, as follows:

$$\mathcal{L}_{\text{Softmax}} = \frac{1}{N}\sum_{i=1}^{N} -\boldsymbol{y}_i^{\top} \log \frac{e^{\ f_\theta(\mathbf{X}_i)}}{\|e^{\ f_\theta(\mathbf{X}_i)}\|_1} \tag{3}$$

$$\mathcal{L}_{\text{Cosine}} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\sigma\left(f_\theta(\mathbf{X}_i), \boldsymbol{y}_i\right) + 1}{2} \tag{4}$$

where $\mathbf{X}_i$ denotes the training SAR image sample, and $f_\theta(\cdot)$ denotes the CNNs which are parameterized by the learnable parameters θ. $f_\theta(\mathbf{X}_i)$ is a column vector that represents the corresponding prediction scores of sample $\mathbf{X}_i$ over K classes. Denote $c_i \in \{1, \ldots, K\}$ as the corresponding category of $\mathbf{X}_i$, and $\boldsymbol{y}_i$ as the one-hot vector of c_i. Then, the cosine similarity between $f_\theta(\mathbf{X}_i)$ and $\boldsymbol{y}_i$ can be formulated as

$$\sigma\left(f_\theta(\mathbf{X}_i), \boldsymbol{y}_i\right) = \frac{f_\theta(\mathbf{X}_i)^{\top} \cdot \psi\left(\boldsymbol{y}_i\right)}{\|f_\theta(\mathbf{X}_i)\|_2 \left\|\psi\left(\boldsymbol{y}_i\right)\right\|_2} \tag{5}$$

where $\psi(\cdot)$ denotes the label shift operation on $\boldsymbol{y}_i$ used to further reduce the over-confident prediction. The shifted label $\boldsymbol{y}_i'$ of $\boldsymbol{y}_i$ can be cast as

$$\boldsymbol{y}_i' = \psi(\boldsymbol{y}_i) = \boldsymbol{y}_i + \tau\, \boldsymbol{y}_i - \tau \tag{6}$$

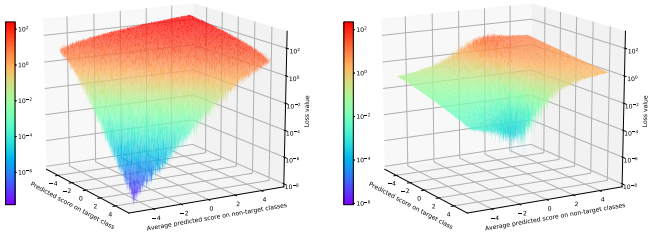
where τ is the shift value, then the standard one-hot label $\boldsymbol{y}_i$ can be expressed as

$$\boldsymbol{y}_i = \varphi_{\text{onehot}}\ (c_i) = \left[\underbrace{0 \cdots 0}_{c_i - 1}\ 1\ \underbrace{0 \cdots 0}_{K - c_i}\right]^{\top} \tag{7}$$

such that $\boldsymbol{y}_i'$ can be recast as

$$\boldsymbol{y}_i' = \psi\left(\boldsymbol{y}_i\right) = \left[\underbrace{-\tau \cdots -\tau}_{c_i - 1}\ 1\ \underbrace{-\tau \cdots -\tau}_{K - c_i}\right]^{\top}. \tag{8}$$

To visually show the difference between softmax and the proposed log cosine loss, and illustrate the characteristics of log cosine loss against overfitting, the value heatmaps of both



(a) Softmax Cross-Entropy (b) Log Cosine Loss

Fig. 2. 3-D heatmaps of the log cosine loss ($\tau = 0.3$) and softmax cross-entropy in a ten-class classification task. Here, the x- and y-axis denote the predicted score of the correct labels and the average predicted score of the other nine wrong labels, respectively. The range of the predicted score for the correct labels is $[-5, 5]$, and predicted for wrong labels is drawn from a normal distribution with a standard deviation of 1 and a mean range of $[-5, 5]$. (a) Softmax cross-entropy. (b) Log cosine loss.

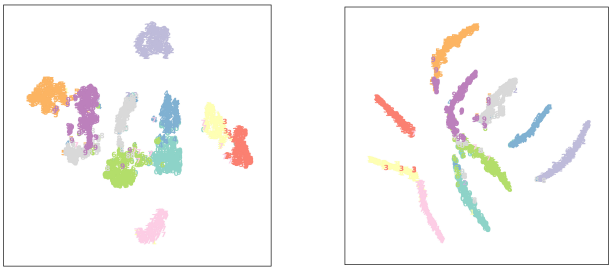
log cosine loss and softmax cross-entropy are given in Fig. 2. From the heat maps, we summarize the two reasons that our proposed log cosine loss can mitigate the over-fitting problem.

First of all, compared with softmax, our proposed log cosine loss can avoid overconfident predictions caused by limited training samples, thereby reducing the possibility of overfitting. In a deep learning paradigm with insufficient samples, each sample will be repeatedly trained. As can be seen from Fig. 2, the gradient of softmax will not decrease during optimization, so the softmax loss will finally be reduced to a very small value [i.e., the dark blue area of Fig. 2(a)], which causes overconfident predictions. However, our proposed log cosine loss can avoid this overconfident prediction. During optimization, the log cosine loss will enter a smoother area, i.e., the low plateau area (light blue) observed in Fig. 2(b). Such smooth and stable gradients can alleviate the overconfident prediction caused by continuous optimization for the same sample.

Second, with the log cosine loss, the deep learning model can avoid continuing optimization of the loss when it has a particularly small value (i.e., near zero), which has been verified to cause severe overfitting [18]. As can be seen from the two heatmaps in Fig. 2, the proposed log cosine loss has a larger minimum loss value, and it will eventually enter a relatively stable platform area, so the loss value will not continue to decrease to a near-zero value, which helps our log cosine loss to resist overfitting.

To further demonstrate the effectiveness of the log cosine loss and explicitly show the difference between our proposed two-stage loss and the softmax cross-entropy, we provide t-distributed stochastic neighbor embedding (t-SNE) cluster visualization of the output before the final FC layer of TMDC-CNNs (softmax cross-entropy) and TMDC-CNNs (two-stage loss), respectively. Both networks are trained by the same 5% of all training samples in MSTAR, the results are shown in Fig. 3.

As can be seen from Fig. 3, the proposed two-stage loss function can not only gather the features of different types of samples by locations but also according to their directions. This is because the proposed log cosine loss can recognize the target by the vector angle between predicted scores and labels. As a result, for samples with blurred features, the network trained by the two-stage loss can recognize them better according to their directions.



(a) (b)

Fig. 3. t-SNE cluster figures of output features before the final FC layer of models trained by different loss functions. The numbers denote the labels of different types of targets. Both models are trained by the same 5% of all training samples, where TMDC-CNNs trained by softmax cross-entropy achieved recognition accuracy of 90.80%, and TMDC-CNNs trained by two-stage loss achieved recognition accuracy of 92.91%. (a) TMDC-CNNs (softmax cross-entropy). (b) TMDC-CNNs (two-stage loss).

TABLE I

NUMBER OF SAMPLES IN EACH CATEGORY IN THE MSTAR DATA SET

	2S1	BMP2	BRDM2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU234
Train (17°)	299	233	298	256	233	299	299	232	299	299
Test (15°)	274	196	274	195	196	274	273	196	274	274

IV. SIMULATIONS AND RESULTS

A. Data Set

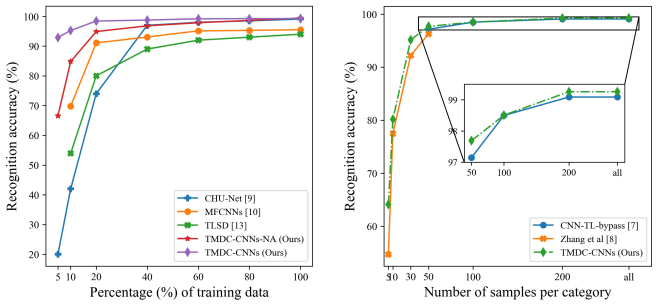
We evaluate our methods on the MSTAR data set. It is an SAR image data set with 0.3 m resolution, which consists of SAR targets at different depression angles and aspect angles [19]. For fair comparison, we select ten typical classes from MSTAR for classification [19]. The detail about each class is given in Table I.

B. Experimental Setup

In all the experiments, an aggressive but simple data augment strategy is applied to make full use of each available sample. Each image is randomly rotated by $0° - 360°$, flipped horizontally and vertically with a probability of 0.5. In addition, we also center-crop the image into a 64×64 patch to accelerate the training and reduce computation. All the networks are trained using stochastic gradient descent (SGD) with 10^{-4} weight decay and 0.9 Nesterov momentum without dampening. The initial learning rate is set to 0.3, and then scaled by 0.3 after every 200 epochs during a total of 500 epochs training. In our two-stage loss training process, the hyperparameters are empirically set as $\alpha = 0.6$, $\beta = 1$, and $\tau = 0.3$. According to our experiments, at these values and their vicinity, the model can obtain a better performance.

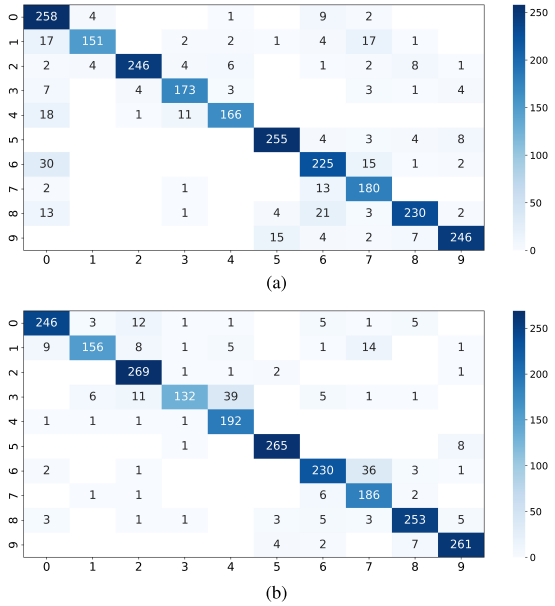
C. Performance Comparison

Our model is compared with state-of-the-art methods, including MFCNNs [10], CHU-Net [9], CNN-TL-bypass [7], transfer learning from simulated data (TLSD) [13], and Zhang *et al.* [8]. The results of these baseline methods are taken from their original letters, and our models are evaluated according to the data selection strategies as used in these methods, i.e., selecting different percentages of the training data as in [9], [10], and [13], and using different numbers of samples in each class as in [7] and [8]. For each setting, ten independent tests are conducted for our proposed models, where the data



(a) (b)

Fig. 4. Performance comparison in terms of accuracy under different data selection strategies. (a) Different percentages of the training data from the data set. (b) Different numbers of samples in each category. Note that the results for some test cases are not shown in the figure if they are not provided in the corresponding references.



(a)

(b)

Fig. 5. Confusion matrices of Inception and TMDC-CNNs with 15 training samples in each class, the number on the horizontal and vertical coordinates represents the numeric label for each target category. (a) Confusion matrix of inception. (b) Confusion matrix of TMDC-CNNs.

used in each test are selected randomly from the training set. The baseline methods [7]–[9] used similar data augmentation strategies, while no data augmentation was used in [10] and [13]. For a fair comparison, we also show the performance of our TMDC-CNNs without using data augmentation, namely, TMDC-CNNs-NA. The results are given in Fig. 4, where we can see that our model can achieve better performance than the baseline methods. More specifically, our proposed TMDC-CNNs perform significantly better than [8] (i.e., with five samples per category) and [9] (i.e., with 5% and 10% training data). Even without using data augmentation, our TMDC-CNNs-NA still outperforms [10] and [13], such performance improvement becomes larger when the amount of training data decreases.

D. Ablation Studies

As our TMDC-CNNs are derived from DenseNet and Inception, we also perform ablation studies with DenseNet, Inception, and the most commonly used architecture: ResNet [20], to show further the improvement of our proposed architecture.

TABLE II
ABLATION STUDIES OF DIFFERENT NETWORK ARCHITECTURES AND LOSS FUNCTIONS IN TERMS OF RECOGNITION ACCURACY (%)

Model	Number of samples in each class							
	5	10	15	30	50	100	200	All
ResNet-18 (baseline) [20]	55.92	72.01	80.59	90.44	93.89	97.53	98.72	98.72
Inception [15]	58.52	74.64	82.14	91.32	94.21	98.47	99.09	99.09
DenseNet [14]	59.52	75.16	82.78	92.05	95.06	98.64	99.24	99.26
TMDC-CNNs (softmax cross-entropy)	61.31	77.79	85.88	94.47	95.91	98.97	99.34	99.48
TMDC-CNNs (two-stage loss, without data augmentation)	46.52	61.73	71.55	86.93	**97.81**	**99.09**	**99.55**	**99.55**
TMDC-CNNs (two-stage loss)	**64.11**	**80.17**	**88.62**	**95.14**	97.69	98.51	99.26	99.26

In addition, we evaluate our TMDC-CNNs by using different loss functions, i.e., softmax cross-entropy and the proposed two-stage loss function. Different numbers of samples in each category are used to demonstrate the performance of different models, and the results are given in Table II. It can be seen that, compared to the baseline, i.e., ResNet [20], the Inception, DenseNet, and our network architecture TMDC-CNNs give better performance for a small number of training samples, and our proposed TMDC-CNNs model offers the largest improvement. In addition, the proposed two-stage loss function improves the performance when only limited training data (i.e., 50 and less samples per category) is available, although the proposed loss results in a small decrease in recognition accuracy when sufficient data are available (i.e., 100 and more samples per category). Moreover, the results demonstrate that our data augmentation strategy can significantly improve the recognition accuracy when data are severely limited, i.e., 30 and less samples in each class.

To further illustrate the improvement and effectiveness of our TMDC-CNNs, confusion matrices of Inception and TMDC-CNNs on the test set are given in Fig. 5(a) and (b), respectively. Both models are trained using a data set with 15 samples per category. It can be seen that TMDC-CNNs can achieve high recognition accuracy when the number of training samples is small. In addition, the recognition accuracy of most categories can be improved when compared with the Inception.

V. CONCLUSION

We have introduced a TMDC-CNNs model to improve the SAR target recognition performance with limited training samples, which included a novel network architecture and a two-stage loss function. With the proposed network architecture, multiscale features can be extracted. With the two-stage loss function, the generalization ability of the proposed model was further improved when only limited training samples were available. Experiments conducted on MSTAR demonstrated the improved performance of our proposed model, as compared with several state-of-the-art methods, with a severely limited amount of training data. In the future, we will also study intelligent data augmentation techniques for this problem.

REFERENCES

[1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013.

[2] M. Cha, A. Majumdar, H. T. Kung, and J. Barber, "Improving SAR automatic target recognition using simulated images under deep residual refinements," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2606–2610.

[3] C. Clemente, L. Pallotta, D. Gaglione, A. De Maio, and J. J. Soraghan, "Automatic target recognition of military vehicles with Krawtchouk moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 493–500, Feb. 2017.

[4] S. Chen and H. Wang, "SAR target recognition based on deep learning," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2014, pp. 541–547.

[5] A. El Housseini, A. Toumi, and A. Khenchaf, "Deep learning for target recognition from SAR images," in *Proc. Seminar Detection Syst. Archit. Technol. (DAT)*, 2017, pp. 1–5.

[6] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–16.

[7] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.*, vol. 9, no. 9, p. 907, 2017.

[8] F. Zhang, Y. Wang, J. Ni, Y. Zhou, and W. Hu, "SAR target small sample recognition based on CNN cascaded features and AdaBoost rotation forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1008–1012, Jun. 2020.

[9] Z. Lin, K. Ji, M. Kang, X. Leng, and H. Zou, "Deep convolutional highway unit network for SAR target classification with limited labeled training data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1091–1095, Jul. 2017.

[10] 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.

[11] B. Ding, G. Wen, X. Huang, C. Ma, and X. Yang, "Data augmentation by multilevel reconstruction using attributed scattering center for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 6, pp. 979–983, Jun. 2017.

[12] L. M. Novak, G. R. Benitz, G. J. Owirka, and L. A. Bessette, "ATR performance using enhanced resolution SAR," *Proc. SPIE*, vol. 2757, pp. 332–337, Apr. 1996.

[13] 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.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-V4, inception-Resnet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[16] B. Barz and J. Denzler, "Deep learning on small datasets without pre-training using cosine loss," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1371–1380.

[17] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[18] I. Takashi, Y. Ikko, S. Tomoya, N. Gang, and S. Masashi, "Do we need zero training loss after achieving zero training error," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 4604–4614.

[19] 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.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.