MdpCaps-Csl for SAR Image Target Recognition With Limited Labeled Training Data

YUCHAO HOU\textsuperscript{1,2}, TING XU\textsuperscript{2}, HONGPING HU\textsuperscript{2}, PENG WANG\textsuperscript{2}, HONGXIN XUE\textsuperscript{3}, AND YANPING BAI\textsuperscript{1,2}

\textsuperscript{1}School of Information and Communication Engineering, North University of China, Taiyuan 030051, China
\textsuperscript{2}School of Science, North University of China, Taiyuan 030051, China
\textsuperscript{3}School of Data Science and Technology, North University of China, Taiyuan 030051, China

Corresponding author: Yanping Bai (baiyp666@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61774137, in part by the Key Research and Development Projects of Shanxi Province under Grant 201903D121156, in part by the Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi under Grant 2019L0929, in part by the Shanxi Natural Science Foundation under Grant 201801D120026, in part by the Shanxi Graduate Education Innovation Projects under Grant 2020SY387, in part by the Science Foundation of North University of China under Grant 2017027, and in part by the Postgraduate Scientific and Technological Innovation Project of North University of China under Grant 20181549.

ABSTRACT Although convolutional neural networks (CNN) have shown excellent performance in many image recognition tasks, it commonly requires a lot of labeled data, and the recognition effect is frequently unsatisfied due to the limited labeled training data. In recent years, capsule network (CapsNet) has been shown to achieve a high recognition accuracy with a small group of training samples. In this study, a class separable loss based on cosine similarity is suggested to enhance the distinguishability of the extracted network. It is added as a regularization term to the original loss function to train the network, narrowing the intra-class difference and increasing the inter-class difference in each iteration. Meanwhile, a multi-dimensional parallel capsule module is established to obtain robust features and spatial relationships from the original images. Feature maps from convolution of different levels are extracted as the input of this module. Structural features derived from low-level convolution and semantic features derived from high-level convolution are used for low-dimensional capsule coding and high-dimensional capsule coding, respectively. In our experiment, the general moving and stationary target acquisition and recognition (MSTAR) database is used. We find that the accuracy of the multi-dimensional parallel capsule network with class separable loss (MdpCaps-Csl) is 99.79\% using all training samples, which is higher than most current recognition methods. More importantly, the accuracy is up to 97.73\% even if only 10\% training samples are applied, indicating MdpCaps-Csl can make excellent performance upon limited training samples.

INDEX TERMS Capsule network, class separable loss, image target recognition, limited labeled training data, multi-dimensional parallel capsule module, synthetic aperture radar.

I. INTRODUCTION

As an active microwave remote sensing imaging system, synthetic aperture radar (SAR) can penetrate clouds and vegetation to identify covered information with high resolution and is little influenced by weather conditions [1]. SAR has been widely used in battlefield reconnaissance [2], environmental monitoring [3], geological survey [4], disaster assessment [5] and other fields. However, compared with optical images, there are too much speckle noises in SAR images, so it is difficult for human eyes to accurately and effectively interpret SAR image targets [6], [7]. Therefore, it is of great significance to achieve SAR image automatic target recognition (SAR-ATR) with more effective techniques.

The previous work on SAR image target recognition is mainly based on template matching [8], [9]. They determine the type of the tested targets by comparing them with a series of templates generated by training samples. However, increasing recognition types and small deformation will often produce obvious scattering, significantly reducing the recognition accuracy. Hence some investigators attempt to put forward the method based on pattern recognition [10], which actually transforms the original data into the appropriate feature vector in a certain way and then uses the extracted
feature training classifier to identify the targets. The feature dimension of SAR images can be effectively reduced by using feature extraction, and its quality directly affects the classification of subsequent classifiers. The generally used feature extraction methods include principal component analysis (PCA) [11], [12], linear decision analysis (LDA) [13], independent component analysis (ICA) [14], fast fourier transform (FFT) [15] and ratio detector (RD) [16], etc. It is difficult to satisfactorily classify SAR image targets by using general classifiers directly. Therefore, the classifier design is generally based on the extracted features. Support vector machine (SVM) [17], [18], adaptive boosting (AdaBoost) [19], k-nearest neighbor (KNN) [20] and sparse representation-based classifier (SRC) [21], [22], etc., are widely used classifiers. Although these methods are effective in SAR image target recognition, they typically require extracting features manually, which easily affected by human subjectivity. At the same time, it is difficult to ensure the effectiveness of the algorithm since the classifier has no feedback.

As a representation learning method that can facilitate feature classification by using automatically learning with large amounts of data, deep learning has been successfully applied in many fields, e.g., image classification [23], speech recognition [24], natural language processing [25]. Different from traditional methods, deep learning can automatically extract more powerful features as well as more abstract and distinctive features through deep structures [26]. Recently, deep learning, especially methods based on convolutional neural networks (CNN) [27], has also made great achievement in SAR image target recognition. Chen et al. [28] used an unsupervised sparse autoencoder to initialize the convolution kernel and fasten the CNN’s feature learning process, achieving an accuracy of 84.7% in 10 categories of moving and stationary target acquisition and recognition (MSTAR) target classification. Chen et al. [29] proposed a new all-convolutional networks (A-ConvNet) consisting of merely sparsely connected layers without fully-connected layers, the accuracy of which is 99.13%. Jiang et al. [30] designed a SAR image target recognition method based on hierarchical fusion of CNN and ASC (attributed scattering center) matching, which can not only inherit the CNN’s excellent performance in recognition but also maintain the model robustness through ASC. Wang et al. [31] proposed an updated squeeze and excitation network (ESENNet) to reduce the impact of feature maps with little information being automatically obtained by CNN on the SAR-ATR performance. The enhanced-SE module can suppress these feature maps through allocating and computing different weights to the corresponding maps. Shao et al. [32] designed a lightweight CNN model for SAR image target recognition, which greatly reduced the iteration time and effectively alleviated the negative impact of data imbalance on recognition performance, receiving an accuracy of 99.54%.

Although the classification accuracy has been greatly improved, the most noticeable CNN-based recognition method often needs multiple convolution kernels to perform the same convolution operation, requiring a large amount of labeled training data. The data that can be used for SAR image target recognition is limited compared with optical images, since accurately acquiring SAR image data is high price as well as time-consuming. Insufficient data volume brings difficulty to effectively train the network, thereby further limits the development of SAR-ATR.

The data enhancement method has been proposed to solve this problem. It changes parts of the data structures and their combination methods in the existing data set. The method also creates an “expanded data set” and adds it to the original training set, increasing the number of training samples [33].

However, it is a time-consuming manual task to build and optimize a more complex data augmentation combination in practice to achieve higher recognition performance. At the same time, some studies have found that some linear composite images could reduce recognition accuracy, rather than increase it [34]. Capsule network (CapsNet) [35] was proposed by Sabour et al. in 2017, and it is shown that CapsNet has better recognition performance than CNN when using a small number of training samples. The main difference between CapsNet and CNN is that CNN constantly adds layers for creating deep networks, while CapsNet embeds neural layers in another layer. A capsule is a group of neurons that introduces more structures into the network and generates a vector to represent an object in the images [36], [37]. The most important thing is that CapsNet integrates pose information and spatial attributes, allowing it can learn well using a small group of data. This method is closer to the human brain’s thinking mode, and it can better show the hierarchy of the internal knowledge representation in neural networks. As shown in Fig. 1, in terms of the SAR image with several objects, the length and orientation of the activity vector, the output of each capsule, represent the possibility of the object and its instantiation parameters. In this example, the blue capsule attempts to find out the outline of the vehicle, and the red capsule attempts to seek the shadow of the vehicle. Importantly, different from these visual ones, the image contains another object being implicitly defined, including typical pose, accurate position, lighting conditions, deformation and other information, etc.

Although CapsNet was proposed not long ago, it has been successfully applied in SAR images. Shah et al. [38] introduced a CapsNet structure composed of a convolution layer, two capsule layers and a decoder network for SAR image target recognition. Its accuracy reached 98.14% in the tests of 10 kinds of MSTAR database.
Schwegmann et al. [39] applied CapsNet in the SAR ship detection task, which stimulated its ability to detect smaller adjacent ships. Comer et al. [40] proposed a principal CapsNet (PCN) architecture for SAR image classification in the context of self-service learning (S3L). This architecture used information invariant clustering (IIC) and auto encoding (AE) to learn from the unlabeled data. Ma et al. [41] proposed an improved detection method for SAR images based on image mapping and CapsNet. In this work, the two heterogeneous images are firstly transformed and compared in the feature space. Then the classified images are sampled. Finally, the classification results can be obtained by inputting the sample results into CapsNet.

Therefore, CapsNet is deeply studied in the present study based on the previous studies mentioned above. The original architecture is modified to work well in the SAR image target recognition. The main contributions of our work can be summarized in the following three aspects:

1) In order to enhance the CapsNet’s extraction capability for SAR image data, a class separable loss based on cosine similarity is added to the improved loss function. It is used as a regularization term, which can reduce intra-class differences and increase the inter-class differences during feature extraction.

2) A multi-dimensional parallel capsule module is proposed to learn the spatial features of the images at different dimensions and enhance the CapsNet’s robustness by limited training data. The feature maps obtained by convolution at different levels are taken as inputs and capsule-encoded in this module to improve the recognition performance of the network.

3) The proposed multi-dimensional parallel capsule network with class separable loss (MdpCaps-Csl) is used to perform a large number of algorithm experiments on MSTAR database. The results show that MdpCaps-Csl works better than most existing methods. It demonstrates good recognition performance, whether using all training samples, a part of training samples, or even an extremely small number of training samples.

The proposed method is evaluated using MSTAR database and is validated using the experiments. The remainder of this paper is organized as follows. The basic structure of CapsNet is introduced in Section II. The proposed class separable loss, multi-dimensional parallel capsule module, and MdpCaps-Csl for SAR image target recognition are described in Section III. The experimental results and discussion for all training samples, partial training samples, and fewer training samples are presented in Section IV. Finally, the conclusions are drawn in Section V.

II. CAPSULE NETWORK

CapsNet, a new capsule structure-based neural network, is robust to affine transformation. In CapsNet, capsule represents various features of a specific entity in the images, e.g., position, size, orientation, hue, texture, etc., and exists as a single logical unit. The data learned and predicted by the capsule itself can be passed to the higher-level capsule through the dynamic routing mechanism. The higher-level capsule can kept active when the prediction is consistent. CapsNet has three advantages compared with the widely used CNN:

- Output of CapsNet is vector with directions, while output of CNN is scalar. Not only can CapsNet use statistical information to detect features, it can also understand them well. It can detect the same object in different directions, thereby learning the basic thought of the object.
- CNN requires the superposition of multiple convolution kernels to perform the same convolution operation, requiring a large number of training samples. In terms of CapsNet, the model can learn characteristic variables in the capsule to maximize the retention of valuable information, which can use fewer training samples to infer the possible variables and get the same generalization as CNN.
- The pooling operation of CNN can loss many important characteristic information. As a result, the output is not sensitive to small variations in the input. Different from that, capsules of CapsNet can carry different attributes and each capsule carries a large amount of target information, where the detailed posture information will be saved in the network. Therefore, CapsNet has better performance in extracting target feature information compared with CNN.

Fig. 2 shows the process transferring information from the low-level to the high-level through dynamic routing mechanism in CapsNet. Similar to the general neuron, weighted summation and non-linear activation are applied during the calculation of CapsNet. However, an additional matrix transformation operation is needed to transform the model from low-level features to high-level features in CapsNet, considering the spatial and hierarchical relationships between objects. The working principle of CapsNet is described in details as following.

Firstly, the output $u_i$ of the low-level feature capsule $i$ is transformed into a prediction $\hat{u}_{ij}$ of the high-level feature capsule $j$ using the spatial transformation matrix $W_{ij}$.
The conversion equation is:

$$\hat{u}_{ji} = W_{ij} \cdot u_i$$  \hspace{1cm} (1)

Next, performing weighted summation with the prediction capsule $\hat{u}_{ji}$ to obtain the input vector $s_j$ of the high-level feature capsule $j$:

$$s_j = \sum_i c_{ij} \cdot \hat{u}_{ji}$$  \hspace{1cm} (2)

where: $c_{ij}$ is the coupling coefficient between the two capsules.

Finally, following CNN, an activation function is required to non-linearly activate its output during the capsule computation, while the difference is that the capsule is a vector rather than a scalar. The activation function used in general neural networks is not suitable for capsule activation. The squash function can keep the activated output vector $v_j$ varying between 0-1 and ensure that $v_j$ and the input vector $s_j$ have the same direction. The equation is as follows:

$$v_j = \frac{\|s_j\|}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|}$$  \hspace{1cm} (3)

The first term of (3) is normalized according to the different length of $s_j$, it is about 1 for long $s_j$ and is about 0 for short $s_j$. $s_j$ is united by the second term of (3), that is, keeping the $s_j$ direction unchanged and its length to 1. The length of the output vector $v_j$ will also be taken as the probability of occurrence of a specific object.

Unlike the weighted summation of general neurons, the weight in the weighted summation of capsule is determined by the dynamic routing mechanism, rather than the back propagation algorithm, which can change the coefficient $c_{ij}$ to update the connection weights between the low-level feature capsules and the high-level feature capsules. The algorithm flow is as following:

Step 1: The connection probability $b_{ij}$ between the low-level feature capsule $i$ and the high-level feature capsule $j$ is initialized to 0;

Step 2: The coupling coefficient $c_{ij}$ varies with capsule connection, indicating that the low-level feature capsules contribute differently to the high-level feature capsules. The coupling coefficient $c_{ij}$ can be calculated by the Softmax function:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$  \hspace{1cm} (4)

Step 3: All the predicted capsules are weighted and summed using (4) and the output capsule $v_j$ is obtained by activation of the squash function;

Step 4: A dot multiplication operation is performed on the converted input capsule $\hat{u}_{ji}$ and output capsule $v_j$ in each capsule in the lower layer, updating the connection probability $b_{ij}$. The equation is as following:

$$b_{ij} = b_{ij} + \hat{u}_{ji} \cdot v_j$$  \hspace{1cm} (5)

Step 5: The loop is ended when reaching a predetermined route iterative number, otherwise skipping to the second step.

Except the coupling coefficient $c_{ij}$, other parameters are updated using the back-propagation algorithm to minimize the loss function. Assuming that a task contains $P$ samples and $K$ labels, the margin loss function for the $p$ sample of $k$ capsule (the number of high-level feature capsules is same as that of image labels) is as following:

$$L_{\text{margin}}^{p,k} = T_k^p \cdot \max(0, m^+ - \|v_k^p\|^2) + \lambda_{\text{margin}} \cdot (1 - T_k^p) \cdot \max(0, \|v_k^p\| - m^-)^2$$  \hspace{1cm} (6)

where: $T_k^p$ is 1, if and only if the $k$-th capsule is consistent with the true label of the sample. Otherwise, it is 0. $m^+$ and $m^-$ are used to control the learning intensity of the network. Ideally, the network not only learns the correct labels, but also can enable that the output probability of corresponding capsule is not less than $m^+$ when the network’s input is the correct label. Similarly, the value is not higher than $m^-$ when the sample does not belong to the current label. The parameter $\lambda_{\text{margin}}$ is used to adjust the loss of the network when the sample is not in the true label, which can ensure the model stability.

In addition, the network takes a decoder consisting of multiple fully connected layers to reconstruct the input images. The euclidean norm between the input image $x_p$ and the reconstructed image $x'_p$ is defined as reconstruction loss. The accuracy of the final classification can be improved by fine-tuning the parameters used in the network [42]. The reconstruction loss is as following:

$$L_{\text{reconstruction}}^p = \|x_p - x'_p\|^2$$  \hspace{1cm} (7)

### III. IMPROVEMENT OF CAPSULE NETWORK

The recognition performance of CapsNet is directly determined by its network structure. In this section, a class separable loss based on cosine similarity and a multi-dimensional parallel capsule module are proposed to enhance the feature extraction ability and improve the recognition performance of the network.

#### A. CLASS SEPARABLE LOSS

Original loss function primarily focuses on the overall information of the SAR images and ignores their difference, while a large amount of common information with small information difference occurs in the SAR images. Consequently, the original loss function enables CapsNet to learn the approximate distribution of training samples, but makes it difficult to increase the difference between the sub-distributions of different image category, which may finally affect the classification performance of CapsNet.

Some scientists tried to strengthen the class separability of feature extraction from models by using contrastive loss [43], [44], triple loss [45]–[48] and multi-class n-pair loss [49]. Although these methods could improve the accuracy of model classification to a certain extent, they did not make full...
use of the types and level information of different images. In fact, many image target recognition tasks can be divided into parent classes and subclasses. The parent class refers to the initial classification of image targets, and the subclass refers to the accurate classification of image targets. The same parent class can be subdivided into multiple subclasses. Here, we construct a class separable loss based on cosine similarity by using the hierarchical relationship between the parent classes and the subclasses of SAR images, and add it to the original loss function to enhance the recognition performance of CapsNet.

The cosine similarity $C(x_i, x_j)$ of image $x_i$ and $x_j$ is:

$$C(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\|_2 \cdot \|x_j\|_2}$$  

(8)

The closer the value is to 1, the more similar the two images are. In order to determine the sample center of a subclass, we firstly calculate the theoretical average of all images in this subclass, then select the image whose the theoretical average has the largest cosine similarity as the sample center. As shown in Fig. 3, for the input SAR image $x_i$, the sample center of its subclass is $c_p^{il}$, $c_p^{i2}$ is the sample center of the closest subclass to $x_p$ and in the same parent class with $x_p$. The cosine similarity $C(x_p, c_p^{il})$ can be expressed as follows:

$$C(x_p, c_p^{il}) = \max \left\{ C(x_p, c_p^{i1}), C(x_p, c_p^{i2}), \cdots, C(x_p, c_p^{i_{n_l}-1}) \right\}$$  

(9)

where: $n_l$ is the number of subclasses which have the same parent class as $x_p$. Note that the sample center of $x_p$’s subclass, $c_p^{il}$, does not involved in the calculation. $c_p^{il}$ is the sample center of a subclass that is closest to $x_p$ but has a different parent class. The cosine similarity $C(x_p, c_p^{il})$ can be expressed as follows:

$$C(x_p, c_p^{il}) = \max \left\{ C(x_p, c_p^{i1}), C(x_p, c_p^{i2}), \cdots, C(x_p, c_p^{il}) \right\}$$  

(10)

where: $n_l$ is the number of subclasses, originating from a different parent class from $x_p$.

Ideally, the similarity relationship among the four different SAR images can be expressed as follows:

$$-1 \leq C(x_p, c_p^{il}) + m_l < C(x_p, c_p^{i0}) + m_0 < C(x_p, c_p^{i'}) \leq 1$$  

(11)

where: $m_l$ and $m_0$ are hyper-parameters used to control the boundary of the feature space, and $m_l > m_0 > 0$. We hope $C(x_p, c_p^{il}) > C(x_p, c_p^{i'})$, and keep at least an interval of $m_l - m_0$. At the same time, we hope $C(x_p, c_p^{i0}) > C(x_p, c_p^{i'})$, and keep at least an interval of $m_0$.

Hinge loss is a common loss function, which is often used to solve the problem of maximum margin [50], [51]. Hinge loss has the characteristics of sparsity, which can reduce the computational expense and enhance the feature extraction ability of the model in nonlinear problems. In order to strengthen the constraint of the similarity relationship in the process of network training, we construct a class separable loss $L_{class}^p$ based on the hinge loss:

$$L_{class}^p = \frac{1}{2} \max(0, (C(x_p, c_p^{il}) + m_l) - (C(x_p, c_p^{i0}) + m_0))$$

$$+ \frac{1}{2} \max(0, (C(x_p, c_p^{i'}) + m_0) - C(x_p, c_p^{i'})$$  

(12)

The loss can be utilized to model the multi-level similarity relationship between the parent class and the subclass of the training samples. The new loss function $L_{total}$ is based on the weighted sum of the three loss functions, which can be obtained by adding $L_{class}^p$ to the original loss function, which is presented as:

$$L_{total} = \frac{1}{P} \sum_{p=1}^{P} \left\{ \sum_{k=1}^{K} L_{margin}^{p,k} + \lambda_r \cdot L_{reconstruction}^p + \lambda_c \cdot L_{class}^p \right\}$$  

(13)

where: $K$ is the number of sample labels. $\lambda_r$ is set 0.0005 to reduce the reconstruction loss. Its value hardly affects the overall loss function during the training.

Note that, in a few recognition tasks, image objects can only be divided into subclasses. They can be considered as special cases of the abovementioned tasks, in which only exists a subclass in each parent class.

B. MULTI-DIMENSIONAL PARALLEL CAPSULE MODULE

Low-level convolution and high-level convolution are used to extract low-level structural features and high-level semantic features in CNN, respectively. In order to save the spatial information of the images, only one scalar convolution layer
is used for feature extraction in the original CapsNet, while the absence of high-level semantic information makes it perform poorly in complex classification. Xiang et al. [52] proposed a multi-scale capsule network (MS-CapsNet) to obtain robust features from the original images. In the first stage of the multi-scale capsule coding unit, there are three feature extraction routes, which are independent of each other. Although it can improve the recognition accuracy to a certain extent, the feature extraction routes of MS-CapsNet also brings many additional convolution operations. Besides, the first layer of MS-CapsNet also retains a large number of huge scalar volume convolution layers used in the original CapsNet, which may lead to over fitting with the increasing number of network parameters.

The multi-dimensional parallel capsule module proposed in our study can fix this problem. As shown in Fig. 4, the dotted part on the left is the three scalar convolution layers in front of the module. $branch_1$, $branch_2$ and $branch_3$ represent the low-, medium-, and high- dimensional capsule coding branch, respectively. $s_1$, $s_2$ and $s_3$ represent the step size of each branch. $c_1 \times c_1$, $c_2 \times c_2$ and $c_3 \times c_3$ represent the convolution kernel size of each branch. $n_1$, $n_2$ and $n_3$ represent the number of feature maps of each branch. The inputs of $branch_1$ are the feature maps obtained by Conv1, Conv2 and Conv3. The inputs of $branch_2$ are the feature maps obtained by Conv1 and Conv2. The inputs of $branch_3$ are the feature maps only obtained by Conv1. Each branch has different step size and the number of feature maps. Three dimensions of the features obtained by convolution of different layers can be encoded in this module, enabling network to further obtain the multi-dimensional feature representation. Compared with the multi-scale capsule encoding unit of MS-CapsNet, the multi-dimensional feature representation can be encoded in this module, enabling network to further obtain the multi-dimensional feature representation. Compared with the multi-scale capsule encoding unit of MS-CapsNet, the multi-dimensional parallel capsule module does not need to add the convolution operation. $branch'$ can be expressed as a concatenation of three branches:

$$branch' = concat (branch_1, branch_2, branch_3) \quad (14)$$

C. MdpCaps-CSL STRUCTURE

MdpCaps-CSL for SAR image target recognition is proposed in this paper. As shown in Fig. 5, the network includes input layer, three scalar convolution layers, multi-dimensional parallel capsule module, digitcaps layer, and output layer. In addition, a decoder network with three fully-connected layers attempts to reconstruct the input images by using the instantiation parameters of the digitcaps layer.

Each input image with size of $96 \times 96$ can generate 8 feature maps with the size of $48 \times 48$, 16 feature maps with the size of $24 \times 24$ and 32 feature maps with the size of $12 \times 12$ after passing through the first, second and third zero-filled convolutions with a step size of 2 with ReLU [53] as activation function. These feature maps of different sizes will be used as the input of the next layer. The multi-dimensional parallel capsule module passes three zero-filled capsule branches with convolution kernel sizes of $c_1 \times c_1$, $c_2 \times c_2$ and $c_3 \times c_3$, respectively, and steps of 1, 2 and 4, respectively and can generate 8, 4 and 4 vector feature maps with size of $12 \times 12$ and vector length of 8. The connected feature maps are used as the input of the digitcaps layer with capsule length of 16 and the number of image labels as capsule number. The weights between the multi-dimensional parallel capsule module and the digitcaps layer are updated by dynamic routing mechanism. The output layer determines the input image type based on the length of the digitcaps layer output vector $v_j$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. MSTAR DATABASE

The proposed method has been verified on the universal MSTAR database [54]. The data was collected using the Sandia National Laboratory (SNL) x-band SAR sensor platform in spotlight imaging mode and was co-funded by the defense advanced research projects agency (DARPA) and air force research laboratory (AFRL). This resolution of database is $0.3m \times 0.3m$ and the omnidirectional angle coverage is $0^\circ - 360^\circ$. It has been widely used in the examination and evaluation of SAR image target recognition algorithms.

The MSTAR database mainly consists of ground military vehicle and civilian vehicle images of different target types, aspect angles, depression angles, serial numbers, articulation and version variants. The database has 3 parent classes with 10 different subclasses (Artillery: 2S1 and ZSU234;
FIGURE 5. MdpCaps-Csl structure.

Truck: BRDM2, BTR60, BMP2, BTR70, D7, and ZIL131; Tank: T62 and T72. Their SAR images and corresponding optical images are shown in Fig. 6. In our experiments, vehicle targets with depression angle of 17° and 15° are used as training samples and test samples, respectively. The detailed information of training set and test set are shown in Table 1.

B. EXPERIMENTAL SETTINGS

Input images of the same size are required by most recognition models, most of which are 128 × 128 in size, but part of them are larger than that, e.g., the image sizes of type 2S1 and type T-62 are 158 × 158 and 172 × 173, respectively. To avoid the influence of background noise, we directly cut the center of all the SAR images into the size of 96 × 96 input models.

A computer with Intel (R) Core (TM) i7 9800X @ 3.8GHz CPU, NVIDIA GeForce RTX 2080Ti GPU, and 16GB of memory is used in the experiment. The experiments are carried out with the software environment of 64-bit Ubuntu 16.04 operating system, CUDA 10.0.130, CuDNN 7.5.1, Tensorflow deep learning framework and Python 3.6.5 programming language. Adam optimization method [55] was used as the gradient descent algorithm for training. The hyper-parameters after multiple trial and error experiments in training can be expressed as followings: the parameter $\lambda_{\text{margin}}$ is 0.1, $m^+$ is 0.9, $m^-$ is 0.1, $m_l$ is 0.3, $m_0$ is 0.1, algorithm iteration times is 50, batch number is 16, weight attenuation coefficient is 0.0001 and the dropout rate is 0.2.

The recognition accuracy and the confusion matrix are used as evaluation indicators in these experiments, which are compared with the current most advanced methods. The expression of recognition accuracy $R_{\text{a}}$ is:

$$R_{\text{a}} = \frac{N_{\text{cor}}}{N_{\text{sum}}} \quad (15)$$

where: $N_{\text{sum}}$ is the total number of test samples with same type, and $N_{\text{cor}}$ is the total number of correctly-identified samples with same type. The larger the value $R_{\text{a}}$, the better the classification performance is.

The confusion matrix is applied as a visualization method, where the row represents instances in the actual class, and column represents instances in the predicted class. It is calculated by comparing the position and classification of actual images with corresponding predicted ones.
C. ANALYSIS OF EXPERIMENTAL PARAMETERS

In this section, we leverage the method in [21], [56] and [57], randomly selecting a part of the images with 17° depression angle as training samples in proportion, and all images with 15° depression angle as test samples, to test the parameter $\lambda_c$, the convolution kernel size combination $(c_1, c_2, c_3)$ of different capsule branches and the route iteration number under three representative training sample ratios of 10%, 50% and 100%. The optimal parameters are selected and applied in the subsequent algorithm experiments.

1) PERFORMANCE UNDER DIFFERENT PARAMETER $\lambda_c$

The initial value of parameter $\lambda_c$ has a significant impact on the final recognition performance of the network. The loss function is the original one when the parameter $\lambda_c$ is 0. The class separable loss will weaken the influence of the original loss function on the overall parameters and the recognition performance will decline when the parameter $\lambda_c$ is higher. As shown in Fig. 7, the recognition accuracies of MdpCaps-Csl increase first, then decrease with increasing parameter $\lambda_c$ under three different training sample ratios and reach the maximum when the parameter $\lambda_c$ is 0.5. Therefore, the value of parameter $\lambda_c$ is fixed as 0.5 and applied to the subsequent algorithm experiments.

2) PERFORMANCE UNDER DIFFERENT CONVOLUTION KERNEL SIZE COMBINATIONS $(c_1, c_2, c_3)$ OF DIFFERENT CAPSULE BRANCHES

The capsule branches of the multi-dimensional parallel capsule module are the core components of MdpCaps-Csl. These branches are all learned from the previous scalar convolution layers, and they can represent small entities in the SAR

| Types     | Serial no. | Training set | Test set |
|-----------|------------|--------------|----------|
|           |            | Depression angle | No.images | Depression angle | No.Images |
| Artillery | 2S1        | 17°           | 299       | 15°           | 274       |
|           | ZSU-234    | 17°           | 299       | 15°           | 274       |
|           | BRDM-2     | 17°           | 298       | 15°           | 274       |
|           | BTR-60     | 17°           | 296       | 15°           | 195       |
|           | BMP-2      | 17°           | 233       | 15°           | 195       |
|           | C71        | 17°           | 233       | 15°           | 196       |
|           | D7         | 17°           | 299       | 15°           | 274       |
|           | ZIL-131    | 17°           | 299       | 15°           | 274       |
| Truck     | T-62       | 17°           | 299       | 15°           | 273       |
|           | T-72       | 17°           | 232       | 15°           | 196       |
images. The convolution kernel size of each capsule branch has an important effect on its performance. When the convolution kernel is small, the feature extraction capability of the network is weak, while when the convolution kernel is large, the network may contain a lot of redundant information. We chose the five groups of (3, 3), (3, 5), (3, 5), (5, 5), and (7, 5) to evaluate their impact on performance. As shown in Fig. 8, the recognition accuracies of network are all the highest when the scheme (5, 5) is selected under three training samples ratios. This scheme is also applied in the subsequent algorithm experiments.

3) PERFORMANCE UNDER DIFFERENT ROUTE ITERATION NUMBERS
Coupling coefficient $c_{ij}$, an important parameter of the network, is updated by dynamic routing mechanism. Choosing a right route iteration number is helpful to get the best coupling coefficient. Specifically, the small number can not enable the parameters to be trained effectively, while the large number will lead to over fitting and increase the extra training time. Fig. 9 shows the variation of recognition accuracies of MdpCaps-Csl with the route iteration numbers under three training samples ratios. The recognition accuracies all have peak values under three training samples ratios. When the route iteration number is 3, the accuracies all reach the maximum values. Therefore, the routing iteration number is set to 3 in the following experiments to achieve better recognition performance.

D. EXPERIMENTS ON ALL TRAINING SAMPLES
Firstly, all the training samples are used to perform algorithm experiments. CapsNet, CapsNet-Csl and MdpCaps-Csl are used to compare and verify the class separable loss and the multidimensional parallel capsule module, where the CapsNet, the baseline, includes input layer, three scalar convolution layers, primarycaps layer, digitcaps layer, and output layer, as shown in Fig. 10. The loss function is the combination of the margin loss function and the reconstruction loss function used in the initial proposal [35]. CapsNet-based CapsNet-Csl is revised through adding the class separable loss to the original loss function, as shown in (13). The multidimensional parallel capsule module based on CapsNet-Csl is used in MdpCapsCsl. All these models use the same experiment configuration and are fully trained.

The detailed test results of CapsNet, CapsNet-Csl and MdpCaps-Csl in Tables 2—4 show that the overall accuracy of MdpCaps-Csl is 99.79%, which is improved 0.90% and 0.37%, respectively, compared with CapsNet and CapsNet-Csl. The test results of MdpCaps-Csl using all training samples show that 4 types of target recognition error occur among the 10 types of vehicle targets, and the accuracy of each type exceeds 99%. Compared with CapsNet, the parent classes in CapsNet-Csl and MdpCaps-Csl have no errors, which have added class separable loss classification methods. The misidentified samples are all from one of parent classes, which are misclassified as the subclass of a certain type in the corresponding parent class. It also shows that the class separable loss based on cosine similarity can greatly enhance the class separability. The feature vectors representing input instances in MdpCaps-Csl can be easily observed through image reconstruction. A reconstructed SAR image derived
TABLE 2. Confusion matrix of CapsNet using all training samples.

| Class | Artillery | Truck | Tank | Recognition accuracy |
|-------|-----------|-------|------|----------------------|
|       | 2S1       | ZSU-234 | BRDM-2 | BTR-60 | BMP-2 | BTR-70 | D7 | ZIL-131 | T-62 | T-72 |       |
| Artillery | 271       | 1       | 0     | 0     | 1      | 0      | 0  | 0        | 0    | 0    | 98.91% |
| ZSU-234 | 1         | 273     | 0     | 0     | 1      | 0      | 1  | 0        | 0    | 0    | 98.91% |
| BRDM-2  | 0         | 1       | 271   | 0     | 0      | 2      | 0  | 0        | 0    | 0    | 98.91% |
| BTR-60  | 0         | 0       | 0     | 191   | 0      | 0      | 2  | 0        | 0    | 2    | 97.95% |
| BMP-2   | 0         | 0       | 0     | 0     | 193    | 0      | 1  | 0        | 2    | 0    | 98.97% |
| BTR-70  | 0         | 1       | 0     | 2     | 193    | 0      | 0  | 269      | 1    | 0    | 98.47% |
| D7      | 0         | 0       | 0     | 2     | 0      | 2      | 669| 1        | 0    | 0    | 98.18% |
| ZIL-131 | 0         | 1       | 1     | 0     | 1      | 0      | 1  | 271      | 0    | 0    | 98.91% |
| Tank    | T-62      | 0       | 0     | 0     | 0      | 1      | 0  | 1        | 0    | 195  | 99.27% |
| T-72    | 0         | 0       | 0     | 0     | 0      | 0      | 0  | 0        | 0    | 0    | 99.49% |
| Total   |           |         |       |       |        |        |    |          | 98.89% |      |

TABLE 3. Confusion matrix of CapsNet-Csl using all training samples.

| Class | Artillery | Truck | Tank | Recognition accuracy |
|-------|-----------|-------|------|----------------------|
|       | 2S1       | ZSU-234 | BRDM-2 | BTR-60 | BMP-2 | BTR-70 | D7 | ZIL-131 | T-62 | T-72 |       |
| Artillery | 273       | 1       | 0     | 0     | 0      | 0      | 0  | 0        | 0    | 0    | 99.64% |
| ZSU-234 | 1         | 273     | 0     | 0     | 0      | 0      | 0  | 0        | 0    | 0    | 99.64% |
| BRDM-2  | 0         | 0       | 272   | 0     | 0      | 1      | 1  | 0        | 0    | 0    | 99.27% |
| BTR-60  | 0         | 0       | 0     | 192   | 0      | 1      | 0  | 2        | 0    | 0    | 98.46% |
| BMP-2   | 0         | 0       | 0     | 1     | 191    | 1      | 2  | 0        | 0    | 0    | 97.95% |
| BTR-70  | 0         | 0       | 0     | 1     | 0      | 195    | 0  | 0        | 0    | 0    | 99.49% |
| D7      | 0         | 0       | 0     | 0     | 0      | 272    | 0  | 0        | 0    | 0    | 99.27% |
| ZIL-131 | 0         | 0       | 0     | 0     | 0      | 274    | 0  | 0        | 0    | 0    | 100%  |
| Tank    | T-62      | 0       | 0     | 0     | 0      | 0      | 0  | 0        | 273  | 0    | 100%  |
| T-72    | 0         | 0       | 0     | 0     | 0      | 0      | 0  | 0        | 0    | 196  | 100%  |
| Total   |           |         |       |       |        |        |    |          | 99.42% |      |

FIGURE 10. The CapsNet structure using as baseline.

from the reconstruction of multiple fully-connected layers is shown in Fig. 11. It can be seen that the reconstructed image well retains the main features of the original image target with good robustness. The reconstructed image is smoothed on local noise compared with the original image.

Table 5 compares the capsule structure-based MdpCaps-Csl with the pattern recognition-based traditional methods and the deep learning-based methods proposed in recent years. The accuracy of MdpCaps-Csl can be improved by 11.79%, 4.79%, 6.19%, 3.19%, 0.99%, 7.09%, 2.79% and 7.19%, respectively, compared with EMACH (extended maximum average correlation height) + PDCCF (polynomial distance classifier correlation filter) [58], IGT (iterative graph thickening) [59], SRC [22], MSS (monogenic scale space) [60], MPMC (modified polar mapping classifier) [61], AdaBoost [19], CGM (conditionally gaussian model) [62], and BCS (bayesian compressive sensing) + scattering centers [63]. Also, MdpCaps-Csl is slightly higher in accuracy than other deep learning-based methods e.g., CNN [28], ComplexNet [64], A-ConvNet [29], CNN + SVM [65], DCHUN [56], CNN-TL-bypass [66], CNN + ASC [30], LCNN + Visual Attention [32] and APCRLNet [57], etc. The above experiments show that MdpCaps-Csl can perform well in recognition without data enhancement.
TABLE 4. Confusion matrix of MdpCaps-Csl using all training samples.

| Class   | Artillery | Truck | Tank | Recognition accuracy |
|---------|-----------|-------|------|----------------------|
|         | 2S1       | ZSU-234 |      |                      |
| Artillery |          |       |      |                      |
|          | 274       | 0      | 0    | 100%                 |
| BRDM-2   | 0         | 0      | 0    | 0%                   |
| BMP-2    | 0         | 0      | 0    | 0%                   |
| D7       | 0         | 0      | 0    | 0%                   |
| ZIL-131  | 0         | 0      | 0    | 0%                   |
| Tank     | T-62      | T-72   |      |                      |
| T-62     | 0         | 0      | 0    | 100%                 |
| T-72     | 0         | 0      | 0    | 100%                 |

TABLE 5. Comparison of recognition performance between MdpCaps-Csl and the state-of-the-art methods.

| Method                          | Recognition accuracy |
|--------------------------------|----------------------|
| EMACH + PDCCF [58]             | 88%                  |
| IGT [59]                       | 95%                  |
| SRC [22]                       | 93.6%                |
| MSS [60]                       | 96.6%                |
| MPMC [61]                      | 98.8%                |
| Adaboost [19]                  | 92.7%                |
| CGM [62]                       | 97%                  |
| BCS + scattering centers [63]  | 92.6%                |
| CNN [28]                       | 84.7%                |
| ComplexNet [64]                | 97.00%               |
| A-ConvNet [29]                 | 99.13%               |
| CNN+SVM [65]                   | 99.15%               |
| DCHUN [56]                     | 99.09%               |
| CNN-TL-bypass [66]             | 99.09%               |
| CNN+ASC [30]                   | 99.41%               |
| LCNN+Visual Attention [32]     | 99.54%               |
| APCRLNet [57]                  | 99.75%               |
| MdpCaps-Csl                     | 99.79%               |

FIGURE 12. Recognition accuracy comparison of different methods using partial training samples.

E. EXPERIMENTS ON PARTIAL TRAINING SAMPLES

Most SAR image target recognition methods reported commonly use all training samples in algorithm experiments, while training data that is manually labeled by professionals is typically limited. We use the same method as in Section IV-C. A part of the images (10% to 100%) with 17° depression angle are randomly selected as training samples and all images with 15° depression angle are used as test samples, to compare and test the recognition performance of MdpCaps-Csl using partial training samples.

Fig. 12 shows the experiment results of MdpCaps-Csl, two traditional machine learning-based methods, e.g., SVM [21] and SRC [21], and six deep learning-based methods proposed in recent years, e.g., A-ConvNet [29], DCHUN [56], APCRLNet [57], probabilistic meta-learning (PML) [67], CapsNet and CapsNet-Csl. Notably, the results of A-ConvNet in this section are derived from recurrent code cited from previous study [57]. Overall, the accuracies of SVM and SRC are much lower than other deep learning-based methods. The fewer training samples are used, the larger difference among them. The accuracy of MdpCaps-Csl is much higher than that of other methods, e.g., A-ConvNet, APCRLNet, PML, CapsNet and CapsNet-Csl, using only 10% training samples, specifically, the values are 97.73%, 73.44%, 78.10%, 89.0%, 89.11% and 93.48%, respectively. In addition, the accuracy of CNN-TL-bypass [66] is 97.15% using
Y. Hou et al.: MdpCaps-Csl for SAR Image Target Recognition With Limited Labeled Training Data

### TABLE 6. Confusion matrix of CapsNet using 10% training samples.

| Class | Artillery | Truck | Tank | Recognition accuracy |
|-------|-----------|-------|------|----------------------|
|       | 2S1 | ZSU-234 | 2S1 | ZSU-234 | BRDM-2 | BTR-60 | BMP-2 | BTR-70 | D7 | ZIL-131 | T-62 | T-72 |
| Artillery | 266 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 266 | 2 |
| BRDM-2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| BTR-60 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BMP-2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BTR-70 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| ZIL-131 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| Total | 266 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 266 | 2 |

### TABLE 7. Confusion matrix of CapsNet-Csl using 10% training samples.

| Class | Artillery | Truck | Tank | Recognition accuracy |
|-------|-----------|-------|------|----------------------|
|       | 2S1 | ZSU-234 | 2S1 | ZSU-234 | BRDM-2 | BTR-60 | BMP-2 | BTR-70 | D7 | ZIL-131 | T-62 | T-72 |
| Artillery | 273 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 270 | 2 |
| BRDM-2 | 2 | 271 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BTR-60 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BMP-2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BTR-70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| ZIL-131 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| Total | 273 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 270 | 2 |

### TABLE 8. Confusion matrix of MdpCaps-Csl using 10% training samples.

| Class | Artillery | Truck | Tank | Recognition accuracy |
|-------|-----------|-------|------|----------------------|
|       | 2S1 | ZSU-234 | 2S1 | ZSU-234 | BRDM-2 | BTR-60 | BMP-2 | BTR-70 | D7 | ZIL-131 | T-62 | T-72 |
| Artillery | 273 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 272 | 0 |
| BRDM-2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BTR-60 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BMP-2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| BTR-70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| D7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| ZIL-131 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| Total | 273 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 272 | 0 |

A total of 500 training samples that are randomly selected from 50 samples in 10 types of training data. With only 10% of the training data (only 275 images are selected as training samples), the accuracy of MdpCaps-Csl is 97.73%. The comparison shows that MdpCaps-Csl can recognize more accurately with fewer training samples than CNN-TL-bypass. The accuracies of hierarchical fusion of CNN and ASC matching method [30] and semi-supervised transfer learning model [68] are 87% and 91.36%, respectively, when using 20% of the training data. The accuracy of MdpCaps-Csl under the same condition is 98.80%, far higher than the above two methods.

Tables 6–8 show the detailed test results of CapsNet, CapsNet-Csl and MdpCaps-Csl using only 10% training samples. For different subclass targets within the same parent class, MdpCaps-Csl is not confused in “Tank” (yellow area) and only one artillery of type 2S1 is incorrectly classified as type ZSU-234 in the ‘Artillery’ (blue area), the main confusion is in the parent class “Truck” (green area). In addition, we found that among the different parent classes, CapsNet...
recognized incorrectly 49 SAR images, while CapsNet-Csl and MdpCaps-Csl with class separable loss only failed 13 and 8 SAR images, respectively. In this case, CapsNet’s recognition accuracy of coarse level is 97.98%, while CapsNet-Csl and MdpCaps-Csl are as high as 99.46% and 99.67%, respectively. This also proves that the class separable loss built with the types and level information of different SAR images helps to extract identification information of different classes more efficiently. This loss has an important role especially in reducing the confusion among different parent classes.

F. EXPERIMENTS ON FEWER TRAINING SAMPLES
In order to further verify the recognition performance of MdpCaps-Csl with fewer training samples, we randomly select 5, 10 and 20 images with 17° depression angle from each type training data as training samples and all images with 15° depression angle as test samples to conduct algorithm experiments.

Fig. 13 shows the experiment results of A-ConvNet, APCRLNet, CNN cascaded features and AdaBoost RoF (CCFAR) [69], CapsNet, CapsNet-Csl and MdpCaps-Csl. The accuracies of MdpCaps-Csl are 63.05%, 78.56% and 90.19%, respectively, which are much higher than those of the other five methods. Although lacking training data with different rotation angles for this case [70], MdpCaps-Csl can still perform well.

V. CONCLUSIONS
CapsNet is very effective in collecting pose information and spatial attributes of the images compared with CNN and it can learn well with a small group of data. This method is similar to the human’s thinking and can effectively express the hierarchical relationship of internal knowledge in neural networks. A cosine similarity-based class separable loss is introduced as a regularization term of the original loss function. A multi-dimensional parallel capsule module is used as well to improve CapsNet, which greatly enhances the feature extraction capability and robustness of the network. To verify the recognition performance of the method, the universal MSTAR database is used. As a result, the accuracy of MdpCaps-Csl is 99.79% using all training samples, which is higher than most pattern recognition-based traditional methods and deep learning-based methods. Even if only 10% training samples is used, the accuracy of MdpCaps-Csl can reach 97.73%, which is much higher than other methods. The results demonstrate that MdpCaps-Csl can still perform well using fewer training samples, despite not all rotation angles are examined. The application of CapsNet in the SAR image target recognition shows a broad prospect, while it is still at the early stage and needs further exploration. We will focus on the network performance of deep-level capsule structures in the future.

REFERENCES
[1] F. Zhou, L. Wang, X. Bai, and Y. Hui, “SAR ATR of ground vehicles based on LM-BN-CNN,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 12, pp. 7282–7293, Dec. 2018.
[2] W. Ye, H. Ruan, S.-X. Zhang, and L. Yan, “Study of noise jamming based on convolution modulation to SAR,” in Proc. Int. Conf. Comput., Mechatronics, Control Electron. Eng., Changchun, China, Aug. 2010, pp. 169–172.
[3] V. C. Koo, Y. K. Chan, V. Gobi, M. Y. Chua, C. H. Lim, C.-S. Lim, C. C. Thum, T. S. Lim, Z. B. Ahmad, K. A. Mahmood, M. H. B. Shahid, C. Y. Ang, W. Q. Tan, P. N. Tan, K. S. Yee, W. G. Cheaw, H. S. Boey, A. L. Choo, and B. C. Sew, “A new unmanned aerial vehicle synthetic aperture radar for environmental monitoring,” Prog. Electromagn. Res., vol. 122, pp. 245–268, 2012.
[4] F. Chen, H. Lin, W. Zhou, T. Hong, and G. Wang, “Surface deformation detected by ALOS PAL SAR small baseline SAR interferometry over permafrost environment of Beiluhe section, Tibet Plateau, China,” Remote Sens. Environ., vol. 138, pp. 10–18, Nov. 2013.
[5] T. Strozzi, C. Ambrosi, and H. Raetzo, “Interpretation of aerial photographs and satellite SAR interferometry for the inventory of landslides,” Remote Sens., vol. 5, no. 5, pp. 2554–2570, May 2013.
[6] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, “High-resolution SAR image classification via deep convolutional autoencoders,” IEEE Geosc. Remote Sens. Lett., vol. 12, no. 11, pp. 2351–2355, Oct. 2015.
[7] Z. Cui, S. Dang, Z. Cao, S. Wang, and N. Liu, “SAR target recognition in large scene images via region-based convolutional neural networks,” Remote Sens., vol. 10, no. 5, p. 766, May 2018.
R. Huan, R. Liang, and Y. Pan, “SAR target recognition with the fusion of
Y. Sun, Z. Liu, S. Todorovic, and J. Li, “Adaptive boosting for SAR
J.-I. Hwang and H.-S. Jung, “Automatic ship detection using the artificial
C. Zheng, X. Jiang, and X. Liu, “Generalized synthetic aperture radar
176230
VOLUME 8, 2020
Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep
T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in
Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and
A. R. Pathak, M. Pandey, and S. Rautaray, “Deep learning approaches
G. E. Hinton, S. Sabour, and N. Frosst, “Matrix capsules with EM routing,”
[36] A. Lin, J. Li, and Z. Ma, “On learning and learned data representation by
capsule networks,” IEEE Access, vol. 7, pp. 50088–50082, Apr. 2019.
R. Shah, A. Soni, V. Mall, T. Gadiyari, and A. Roy, “Automatic target recognition from SAR images using capsule networks,” in Proc. 8th Int. Conf. Pattern Recognit. Mach. Intell., Tezpur, India, 2019, pp. 377–386.
C. P. Schwegmann, W. Kleyhans, B. P. Salmon, L. W. Muldaken, and R. G. Meyer, “Synthetic aperture radar ship detection using capsule networks,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Valencia, Spain, Jul. 2018, pp. 725–728.
J. F. Comer, R. W. Andrews, N. Naderializadeh, S. Kolouri, and H. Hoffman, “SAR automatic target recognition with less labels,” in Proc. SPIE, Autom. Target Recognit., CA, USA, vol. 11394, 2020, Art. no. 113940Q.
W. Ma, Y. Xiong, Y. Wu, H. Yang, X. Zhang, and L. Jiao, “Change detection in remote sensing images based on image mapping and a deep capsule network,” Remote Sens., vol. 11, no. 6, p. 626, Mar. 2019.
A. Mobiny, H. Lu, H. V. Nguyen, B. Roysam, and N. Varadarajan, “Automated classification of apoptosis in phase contrast microscopy using capsule network,” IEEE Trans. Med. Imag., vol. 39, no. 1, pp. 1–10, Jan. 2020.
S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR), San Diego, CA, USA, Jun. 2005, pp. 539–546.
R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), New York, NY, USA, Jun. 2006, pp. 1735–1742.
K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” J. Mach. Learn. Res., vol. 10, pp. 2077–2096, Dec. 2009.
F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Boston, MA, USA, Jun. 2015, pp. 815–823.
A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, “Working hard to know your neighbor’s margins: Local descriptor learning loss,” in Proc. Adv. Neural Inf. Process. Syst., Long Beach, CA, USA, 2017, pp. 4826–4837.
H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 4004–4012.
K. Sohn, “Improved deep metric learning with multi-class N-pair loss objective,” in Proc. Adv. Neural Inf. Process. Syst., Barcelona, Spain, 2016, pp. 1857–1865.
L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, “Are loss functions all the same?” Neural Comput., vol. 16, no. 5, pp. 1063–1076, May 2004.
Y. Wu and Y. Liu, “Robust truncated hinge loss support vector machines,” J. Amer. Stat. Assoc., vol. 102, no. 479, pp. 974–983, Sep. 2007.
C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, “MS-CapsNet: A novel multi-scale capsule network,” IEEE Signal Process. Lett., vol. 25, no. 12, pp. 1850–1854, Dec. 2018.
X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proc. 14th Int. Conf. Artif. Intell. Stat. (AISTATS), Fort Lauderdale, FL, USA, 2011, pp. 315–323.
Yuchao Hou received the M.S. degree from the School of Science, North University of China, Taiyuan, Shanxi, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering. His research interests include synthetic aperture radar (SAR) target recognition and image processing.

J.-I. Park and K.-T. Kim, “Modified polar mapping classifier for SAR automatic target recognition,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 7, pp. 1091–1095, Jul. 2017.

R. Singh and B. V. Kumar, “Performance of the extended maximum average correlation height (EMACH) filter and the polynomial distance classifier correlation filter (PDCCF) for multi-class SAR detection and classification,” in Proc. SPIE, Int. Soc. Opt. Eng., Orlando, FL, USA, 2002, pp. 265–276.

U. Srinivas, V. Monga, and R. G. Raj, “SAR automatic target recognition using discriminative graphical models,” IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 1, pp. 591–606, May 2014.

G. Dong and G. Kuang, “Classification on the monogenic scale space: Application to target recognition in SAR image,” IEEE Trans. Image Process., vol. 24, no. 8, pp. 2527–2539, Aug. 2015.

J. A. O’Sullivan, M. D. DeVore, V. Kedia, and M. I. Miller, “SAR ATR performance using a conditionally Gaussian model,” IEEE Trans. Aerosp. Electron. Syst., vol. 37, no. 1, pp. 91–108, Jan. 2001.

X. Zhang, J. Qin, and G. Li, “SAR target classification using Bayesian compressive sensing with scattering centers features,” Prog. Electromagn. Res., vol. 136, pp. 385–407, Jan. 2013.

R. Chakraborty, J. Wang, and S. X. Yu, “Sur-real: Frenet mean and distance transform for complex-valued deep learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 889–897.

F. Gao, T. Huang, J. Sun, J. Wang, A. Hussain, and E. Yang, “A new algorithm for SAR image target recognition based on an improved deep convolutional neural network,” Cognit. Comput., vol. 11, no. 6, pp. 809–824, Jun. 2018.

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, Aug. 2017.

K. Wang, G. Zhang, X. Yu, and H. Leung, “SAR target recognition based on probabilistic meta-learning,” IEEE Geosci. Remote Sens. Lett., early access, Apr. 13, 2020, doi: 10.1109/LGRS.2020.2983988.

W. Zhang, Y. Zhu, and Q. Fu, “Semi-supervised deep transfer learning-based on adversarial feature learning for label limited SAR target recognition,” IEEE Access, vol. 7, pp. 152412–152420, Oct. 2019.

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.

Y. Xue, J. Pei, Y. Huang, J. Yang, and Y. Zhang, “Target recognition for SAR images based on heterogeneous CNN ensemble,” in Proc. IEEE Radar Conf. (RadarConf), Oklahoma City, OK, USA, Apr. 2018, pp. 507–512.

Hongping Hu received the Ph.D. degree from the North University of China, Taiyuan, Shanxi, China, in 2009. She is currently an Associate Professor and a Master’s Tutor with the Department of Mathematics, North University of China. Her research interests include combinatorial mathematics, artificial intelligence, and image processing.

Peng Wang received the Ph.D. degree from the North University of China, Taiyuan, Shanxi, China, in 2013. He visited the School of Computing and Information Sciences, Florida International University, as a Visiting Scholar, from 2014 to 2015. He is currently an Associate Professor, a Master’s Tutor, and the Director of the Mathematics Experiment Center, Department of Mathematics, North University of China. His research interests include intelligent algorithm and signal processing.

Hongxin Xue received the Ph.D. degree from the School of Information and Communication Engineering, North University of China, Taiyuan, Shanxi, China, in 2019, where she is currently a Lecturer with the School of Data Science and Technology. Her research interests include artificial intelligence, signal processing, and intelligent optimization algorithm.