Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Deep feature extraction and combination for synthetic aperture radar target classification

Moussa Amrani
Feng Jiang

SPIE.

Moussa Amrani, Feng Jiang, "Deep feature extraction and combination for synthetic aperture radar target classification," *J. Appl. Remote Sens.* **11**(4), 042616 (2017), doi: 10.1117/1.JRS.11.042616.

Deep feature extraction and combination for synthetic aperture radar target classification

Moussa Amrani* **and Feng Jiang**
Harbin Institute of Technology, School of Computer Science and Technology, Harbin, China

Abstract. Feature extraction has always been a difficult problem in the classification performance of synthetic aperture radar automatic target recognition (SAR-ATR). It is very important to select discriminative features to train a classifier, which is a prerequisite. Inspired by the great success of convolutional neural network (CNN), we address the problem of SAR target classification by proposing a feature extraction method, which takes advantage of exploiting the extracted deep features from CNNs on SAR images to introduce more powerful discriminative features and robust representation ability for them. First, the pretrained VGG-S net is fine-tuned on moving and stationary target acquisition and recognition (MSTAR) public release database. Second, after a simple preprocessing is performed, the fine-tuned network is used as a fixed feature extractor to extract deep features from the processed SAR images. Third, the extracted deep features are fused by using a traditional concatenation and a discriminant correlation analysis algorithm. Finally, for target classification, K-nearest neighbors algorithm based on LogDet divergence-based metric learning triplet constraints is adopted as a baseline classifier. Experiments on MSTAR are conducted, and the classification accuracy results demonstrate that the proposed method outperforms the state-of-the-art methods. [DOI: 10.1117/1.JRS.11.042616]

Keywords: synthetic aperture radar; target classification; deep features; feature fusion; discriminant correlation analysis.

Paper 170499SS received Jun. 8, 2017; accepted for publication Sep. 22, 2017; published online Oct. 17, 2017.

1 Introduction

Synthetic aperture radar (SAR) is a high resolution, mostly airborne and spaceborne strip-map, spotlight, or scan remote sensing system, which uses the path traversed by the platform to simulate a reasonable size antenna for imaging distant targets on a scene. SAR is unique in its imaging capability in that it can operate proficiently in all-weather day-and-night conditions and provide microwave images of extremely high resolution. This paper considers a spotlight mode SAR, where the system transmits electromagnetic pulses with high power from radar mounted on a moving platform to a fixed particular area of interest on the target and receives the echoes of the backscattered signal in a sequential way. This type of SAR collects data from multiple viewing angles and combines them coherently to illuminate and attain a very high resolution of the scene. SAR has been primarily employed for many potential applications on a target[1,2] such as military surveillance, reconnaissance, classification, etc. However, searching targets of interest in massive SAR images are time-consuming and difficult to carry on manual interpretation compared to optical images, which describe a good appearance of a target. This leads up to develop SAR automatic target recognition (ATR) algorithms.

Similar to the progress in speech and object recognition, feature extraction plays a key role in the classification performance of synthetic aperture radar automatic target recognition (SAR-ATR). Several feature extraction methods have been proposed for SAR images including geometric descriptors, such as peak locations, edges, corners, and shapes,[3] and transform-domain

*Address all correspondence to: Moussa Amrani, E-mail: amrani.lmcice@hotmail.com, amrani.moussa@hit.edu.cn

1931-3195/2017/$25.00

coefficients, such as wavelet coefficients.[4] Although the above-mentioned methods may have some advantages, most of these methods are mostly hand-designed and relatively simple, containing dense sampling of local image patches, and representing them by means of visual descriptors, such as histogram of oriented gradients and scale invariant feature transform. In addition, they failed to achieve the promising classification performance (i.e., accuracy). More recently, deep features[5] have been adopted in several applications such as computer vision and recognition tasks.[6–8] The top layers of convolutional neural networks (CNNs) contain more semantic information and describe the global feature of the images, whereas the intermediate layers describe the local features, and the bottom layers contain more low-level information for the description of texture, edges, etc. Hence, comparing the handcrafted features with deep features, deep features have more powerful discriminative and robust representation ability, which can represent images on various factors (illumination, pose, expression, corruption, and occlusion) and achieve better performance. Therefore, in this paper, we exploit the deep features for SAR images and apply the feature level fusion due to the feature level provides more information and details, which lead to a better classification performance. Moreover, our approach uses a discriminant correlation analysis (DCA) algorithm to combine relevant features and overcome the curse of dimensionality problem. This paper proposes an efficient feature extraction method, which can solve the multitarget SAR images classification effectively by exploiting the deep features, selects adaptive feature layers for targets, and fuses the estimated features based on DCA a algorithm. The distributions of the proposal method mainly include four aspects.

- VGG-S net is fine-tuned on moving and stationary target acquisition and recognition (MSTAR) public release database.
- A simple preprocessing is performed. Then, to improve the representation ability of SAR images, the proposed approach exploits the deep features for multimodal features based on the fine-tuned network.
- To combine the selective adaptive layer's features for SAR images, the proposed method adopts a DCA algorithm[9] for features fusion by concatenation.
- For better performance, *K*-nearest neighbors (K-NN) algorithm based on LogDet divergence-based metric learning triplet constraints (LDMLT)[10] is used as a baseline classifier for target classification. In addition, the fusion between K-NN classifier and the discriminant features is studied, which brings higher classifier precision than the common methods.

The proposed method is evaluated on the real SAR images from MSTAR and the experimental results validate the effectiveness of the proposed method. The rest of this paper is organized as follows. Section 2 describes the proposed method that consists of deep features, discriminative correlation analysis algorithm, and K-NN for classification. Section 3 briefly introduces the overall VGG-S1 architecture that is used as fixed feature extractor. The experimental results are presented in Sec. 4. Finally, Sec. 5 gives the concluding remarks of this paper.

2 Proposed Method

This paper presents a robust and efficient feature extraction method for SAR target classification using deep features. Figure 1 shows the structure of the proposed method, which we used in SAR target classification, includes the SAR images, the extracted deep features based on VGG-S1 net, DCA algorithm for fusion layers' features, and K-NN classifier for classification. In the following sections, each aspect of the proposed method (data preprocessing, feature extraction, feature fusion, and classification) is described with more details.

2.1 *Data Preprocessing*

Before feature extraction, all training SAR target images are first processed in two steps, the first step takes the sample SAR images with pixel values in the range of 0 to 255. In the second step of preprocessing, the integer mean value is subtracted across every individual feature in the SAR images and has the geometric interpretation of centering the cloud of the SAR target image

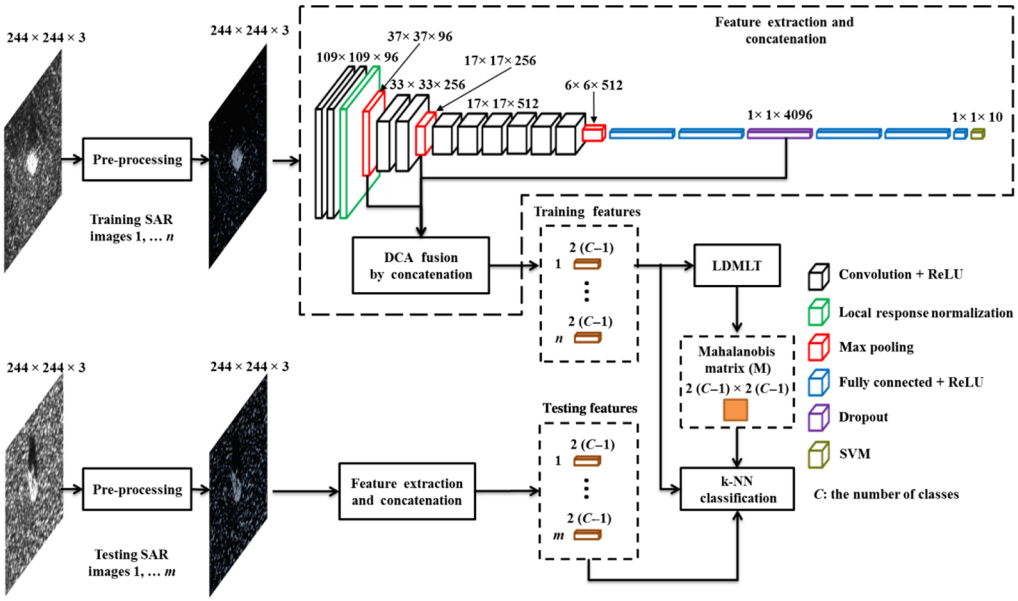
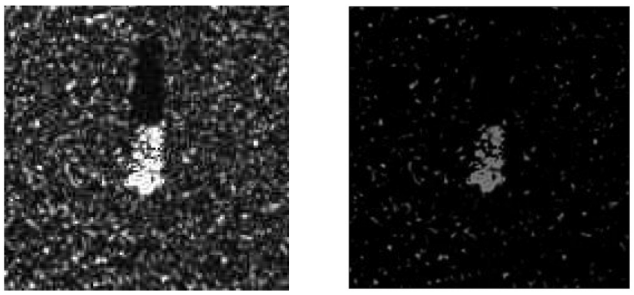


Fig. 1 Overall architecture of the proposed method.



(a) (b)

Fig. 2 Data preprocessing: (a) original SAR image and (b) processed SAR image.

around the origin along every dimension. Let $I(x, y)$ be a sample SAR image, the mean substraction of $I(x, y)$ is denoted as

$$N(x, y) = [I(x, y) - \overline{I}(x, y)], \tag{1}$$

$$\overline{I}(x, y) = |\text{mean}[I(x, y)]|_{\text{Round}}, \tag{2}$$

where $\overline{I}(x, y)$ is the integer mean value. Figure 2 shows the preprocessing result of a sample SAR image.

2.2 *Feature Extraction*

VGG-S1 net is used to extract the deep features from SAR images. The proposed VGG-S1 is described with details in Sec. 3. The network is trained on the MSTAR database to classify an image into one of the C categories, where C is the number of classes. This determines all the parameters of the CNN, such as the weights of the convolutional filters. For an SAR image, the trained network is used to generate discriminative features from the response of layers in the architecture with the input SAR image. Then, to combine the discriminative layer features, a traditional fusion based on a simple concatenation is adopted. In particular, we considered the following discriminative feature vectors that are extracted from the net to represent the SAR

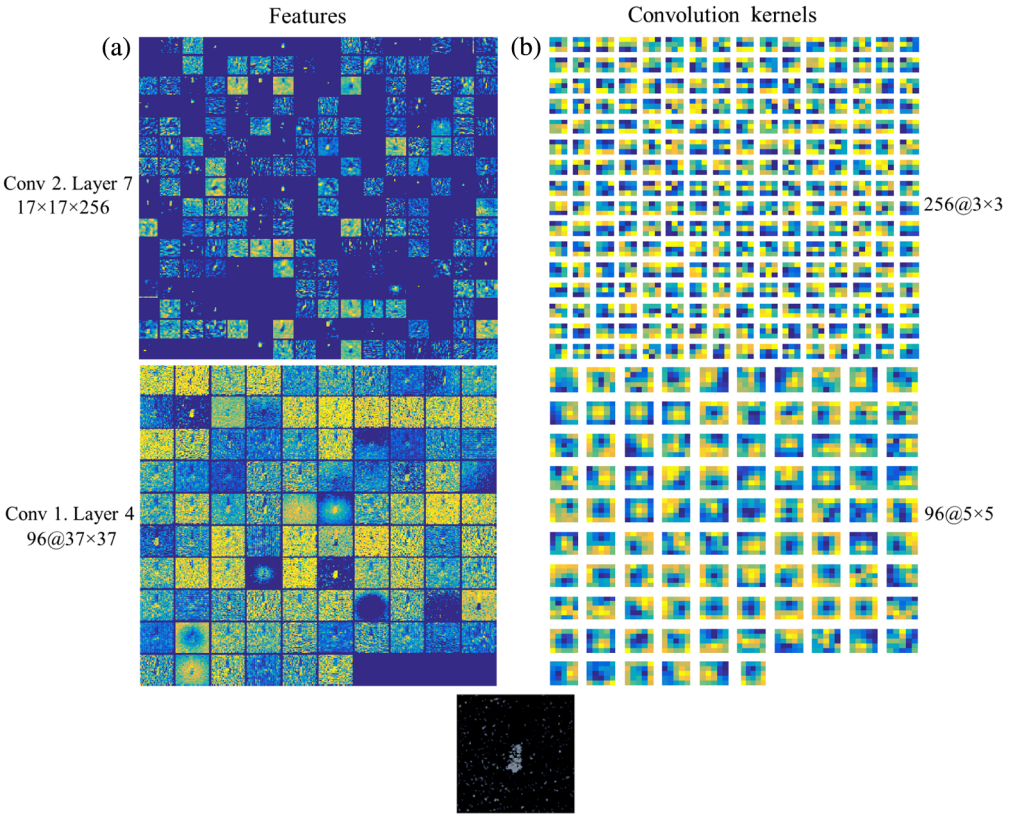


Fig. 3 Deep features visualization: (a) output feature maps and (b) convolution kernels of the considered.

images: (1) the first max-pooling layer output with 131,424 feature channels, which contain more semantic information and describe the global feature of the images; (2) the second max-pooling layer output with 73,984 feature channels, respectively, that describe more the local features; and (3) the first fully connected (fc6) layer output with 4096 feature channels, which contain more low-level information for the description of texture and edges. In addition, the combination of the selected layers improves the accuracy of the classification. Due to the limitation of high dimension, feature fusion by a DCA algorithm is performed that achieves a better performance beside its reduction of vectors dimension. Figure 3 shows the outputs from the considered convolutional layers corresponding to a sample SAR image from MSTAR dataset.

2.3 *Feature Fusion*

A DCA algorithm[9] is a feature level fusion algorithm that includes the class associations into the correlation analysis for the feature sets. We adopted a DCA algorithm to fuse the extracted deep features from SAR images. Feature level fusion based on DCA is performed by using concatenation for the transformed layer features. Moreover, DCA is used to reduce the dimension of the fused feature vectors. Haghighat et al.[9] generalized the DCA algorithm and called it the multiset discriminant correlation analysis (MDCA) to use it for more than two feature sets; furthermore, the MDCA algorithm applies a DCA for reducing dimensions. To illustrate the MDCA approach, we assume that we have five feature vectors to represent the image where each feature vector has a rank. If five feature vectors are V1, V2, V3, V4, and V5 with rank (V1) > (V2), rank (V2) > (V3), rank (V3) > (V4), and rank (V4) equal (V5). Then, MDCA starts with fusing the two feature vectors that have the highest ranks (i.e., V1 and V2), after that, the fusion result of V1 and V2 is fused with next highest rank feature vector (V3) and so on. When the ranks of two feature vectors are equals as V4 and V5, it can be combined at any time; as shown in Fig. 4. After the network is applied on the MSTAR database, the SAR image targets are represented by new feature vectors. The new feature vectors are then fused as follows.

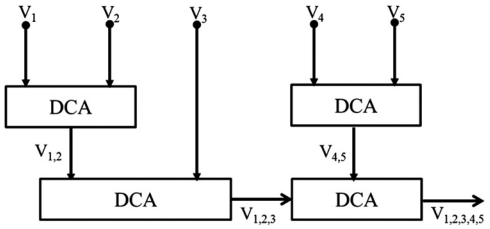


Fig. 4 A diagram of MDCA for five features sets. s.t: rank (V1) > rank (V2) > rank (V3) > rank (V4), and rank (V4) = rank (V5).

Let $X, Y, Z \in \Re^{p_i \times n}$, $i = 1,2,3$, be the training feature sets, and let $X_{\text{test}}, Y_{\text{test}}, Z_{\text{test}} \in \Re^{q_i \times n}$, $i = 1,2,3$, be the testing feature sets, where X and X_{test} are extracted using the first max-pooling layer output, Y and Y_{test} are extracted using the second max-pooling layer output, and Z and Z_{test} are extracted using the first fully connected layer output, p_i is the dimensionality of feature sets, and n is the number of training samples. MDCA is used to fuse the three sets of features by concatenation. Based on MDCA paradigm, only two features sets with the highest ranks will be fused together, and the maximum length of the fused feature vector is $\min[2(C-1), \text{rank}(X), \text{rank}(Y)]$. First, the feature sets are computed and found that rank $(X) = \text{rank}(Y) = \text{rank}(Z) = p_i$. Then, due to the feature sets with the same rank, X and Y are fused together, and the result of the fusion of X and Y is fused with Z. The fusion of the extracted feature sets is done into two principal phases: in the first phase, we calculated the transformation matrices Wx and Wy as well as projected the training feature sets into the DCA subspace, and then fused the two transformed training feature sets by concatenation. In the second phase, we projected the testing feature sets into the DCA subspace and fused the two transformed testing feature sets by concatenation.

2.3.1 *Transformation matrix computation and project the training feature sets into the DCA subspace*

First, we computed the mean vectors of the training feature sets for each class. Therefore, the n columns of the data matrix are divided into c separate classes, where n_i columns belong to the i'th class $(n = \sum_{i=1}^{c} n_i)$. Let $x_{ij} \in X$ denote the feature vector corresponding to the j'th sample in the i'th class $\overline{x_i}$ and $\overline{x}$ denote the means of the x_{ij} vectors in the i'th class and the whole feature set, respectively. Thus, $\overline{x_i} = \frac{1}{n}\sum_{j=1}^{n_i} x_{ij}$ and $\overline{x} = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n_i} x_{ij} = \frac{1}{n}\sum_{i=1}^{c} n_i \overline{x}_i$. Second, we diagonalized the between-class scatter matrix (S_b) for the two training feature sets X and Y as follows:

$$S_{bx_{(p\times p)}} = \sum_{i=1}^{c} n_i(\overline{x_i} - \overline{x})(\overline{x_i} - \overline{x})^{\mathrm{T}} = \Phi_{bx}\Phi_{bx}^{\mathrm{T}}, \tag{3}$$

$$\Phi_{bx_{(p\times c)}} = \left[\sqrt{n_1}(\overline{x_1} - \overline{x}), \sqrt{n_1}(\overline{x_1} - \overline{x}), \ldots, \sqrt{n_1}(\overline{x_1} - \overline{x})\right]. \tag{4}$$

The number of features is higher than the number of classes $(p \gg c)$ so that for low complexity the covariance matrix $(\Phi_{bx}^{\mathrm{T}}\Phi_{bx})_{cc}$ is diagolalized as[11]

$$(\Phi_{bx}Q)^{\mathrm{T}} S_{bx}(\Phi_{bx}Q) = \Lambda_{(r\times r)}, \tag{5}$$

where $Q_{(c\times r)}$ contains the first r eigenvectors from the orthogonal matrix, and Λ is the diagonal matrix of real nonnegative eigenvalues. Third, we projected the training feature sets in the between-class scatter matrices space by using the transformation $W_{bx} = \Phi_{bx}Q\Lambda^{-1/2}$ to reduce the dimensionality of X from p_1 to r as

$$W_{bx}^{\mathrm{T}} S_{bx} W_{bx} = I, \tag{6}$$

$$X'_{(r\times n)} = W_{bx_{(r\times p_1)}}^{\mathrm{T}} X_{(p_1\times n)}, \tag{7}$$

where I is the between-class scatter matrix, and X' is the projection of X in the space of I. Similar to the above step, the between-class scatter matrix for the second modality S_{by} is used to compute the transformation matrix W_{by}, which reduce the dimensionality of the training feature set Y from p_2 to r as

$$W_{by}^{\mathrm{T}} S_{by} W_{by} = I, \tag{8}$$

$$Y'_{(r\times n)} = W_{by_{(r\times p_2)}}^{\mathrm{T}} Y_{(p_2\times n)}. \tag{9}$$

Fourth, after using the between-class scatter matrices to transform X and Y to X' and Y', respectively, the singular value decomposition is utilized to diagonalize the between-set covariance matrix of the transformed feature sets $S'_{xy} = X'Y'^{\mathrm{T}}$ as

$$S'_{xy_{(r\times r)}} = U \sum V^{\mathrm{T}} \Rightarrow U^{\mathrm{T}} S'_{xy} V = \sum, \tag{10}$$

where $\sum$ is a nonzero diagonal matrix. Let $W_{cx} = U \sum^{-1/2}$ and $W_{cy} = V \sum^{-1/2}$, therefore

$$\left(U \sum^{-1/2}\right)^{\mathrm{T}} S'_{xy} \left(V \sum^{-1/2}\right) = I. \tag{11}$$

Then, the between-set covariance matrix S'_{xy} is used to transform the training feature sets matrices as follows:

$$X'' = W_{cx}^{\mathrm{T}} X' = W_{cx}^{\mathrm{T}} W_{bx}^{\mathrm{T}} X = W_x X, \tag{12}$$

$$Y'' = W_{cy}^{\mathrm{T}} Y' = W_{cy}^{\mathrm{T}} W_{by}^{\mathrm{T}} Y = W_y Y, \tag{13}$$

where $W_x = W_{cx}^{\mathrm{T}} W_{bx}^{\mathrm{T}}$ and $W_y = W_{cy}^{\mathrm{T}} W_{by}^{\mathrm{T}}$ are the final transformation matrices for X and Y, respectively. Finally, the transformed training feature sets X'' and Y'' are fused by concatenation.

2.3.2 *Project the testing feature sets into the DCA subspace*

In this phase, the final transformation matrices are used to transform the testing feature sets as follows:

$$X''_{\mathrm{test}} = W_x X_{\mathrm{test}}, \tag{14}$$

$$Y''_{\mathrm{test}} = W_y Y_{\mathrm{test}}. \tag{15}$$

Then, the projected testing feature sets X''_{test} and Y''_{test} are fused by concatenation.

2.4 *Learn the Mahalanobis Distance Metric and K-NN Classification*

After the feature extraction process, learning a suitable similarity metric plays a key role in measuring the similarity of the input feature space of SAR target samples. First, the Mahalanobis distance is adopted as the similarity metric to measure the feature space of the training SAR samples. The Mahalanobis distance is a standard distance metric parameterized by a positive semidefinite matrix M. Let $x_i \in \Re^d$, $i = 1,2,\ldots,n$ be the training feature samples, the square Mahalanobis distance between samples x_i and x_j is defined as

$$d_M(x_i, x_j) == (H^{\mathrm{T}} x_i - H^{\mathrm{T}} x_j)^{\mathrm{T}} \sum{}' (H^{\mathrm{T}} x_i - H^{\mathrm{T}} x_j), \tag{16}$$

where H is a unitary matrix, which satisfies $HH^{\mathrm{T}} = I$ and $\sum'$ is a diagonal matrix, which contains all the singulars values. Second, to select and weight the features, an efficient online metric learning algorithm LDMLT[10] is applied to solve the following metric learning problem:[12]

$$M_{t+1} = \arg\ \min\ D(M, M_t) + n_t \ell(M, \hat{\gamma}_t, \gamma_t), \qquad M \geq 0, \tag{17}$$

$$\ell(M, \hat{\gamma}_t, \gamma_t) = \max[0, \rho + d_M(x_i, x_j) - d_M(x_i, x_k)], \tag{18}$$

$$\hat{\gamma}_t = d_M(x_i, x_k) - d_M(x_i, x_j), \tag{19}$$

where M_t represents the iterative Mahalanobis matrix at iteration t, $\eta_t > 0$ is a regularization parameter, which equalizes the regularization function $D\ (M, M_t)$ and the loss function $\ell(M, \hat{\gamma}_t, \gamma_t)$, $\hat{\gamma}_t$ is the prediction distance, and $y_t = \rho$ is the target distance.

When receiving a new triplet $\{x_i, x_j, x_k\}$ at t: if $d_{M_t}(x_i, x_k) - d_{M_t}(x_i, x_j) \geq \rho$, so there is no loss by utilizing the current M_t to clarify the relation between these three samples, if $d_{M_t}(x_i, x_k) - d_{M_t}(x_i, x_j) < \rho$, M_t should be updated to a better Mahalanobis distance to minimize the loss as

$$M_{t+1} = \gamma_t + \frac{\eta_t \gamma_t q_t q_t^{\mathrm{T}} \gamma_t}{1 - \eta_t q_t^{\mathrm{T}} \gamma_t q_t}, \tag{20}$$

$$\gamma_t = M_t - \frac{\eta_t M_t - p_t p_t^{\mathrm{T}} M_t}{1 + v^{\mathrm{T}} A^{-1} u}, \tag{21}$$

where $p_t = x_i - x_j$ and $q_t = x_i - x_k$. During the updating of the Mahalanobis matrix, the regularization parameter $\eta_t = \frac{\alpha}{\mathrm{tr}[(I - M_t)^{-1} M_t q_t q_t^{\mathrm{T}}]}$ is used to tradeoff between the regularization function and the loss function, where I is the unit matrix and $0 \leq \alpha \leq 1$ is the learning rate parameter. Third, the matrix M that minimizes the loss function in Eq. (20) is utilized as a Mahalanobis distance metric for K-NN classification. Finally, K-NN is trained to be able to classify unknown SAR targets into one of the learned class labels in the training set. More precisely, the classifier calculates the similarity of all trained classes and assigns the unlabeled targets to the class with the highest similarity measure by a majority vote of its K-NN (k is a positive integer). The overall SAR target classification framework is given in Algorithm 1.

Algorithm 1 Deep feature extraction and classification.

1: **Input:** $I_{\text{train}} = \{I_{\text{tr1}} \ldots I_{\text{trN}}\}$ training set, $I_{\text{test}} = \{I_{\text{tst1}} \ldots I_{\text{tstN}}\}$ testing set, $t = 0$, iteration number L, margin ρ.

2: **Output:** overall accuracy.

3: **for** $i = 1$ to n **do**

4: Compute N% preprocessing step.

5: Extract deep features using VGG-S1 net: $X_{\text{pool1}} = \{x_{\text{pool11}}, x_{\text{max 12}}, \ldots, x_{\text{pool1}n}\}$, $Y_{\text{pool2}} = \{y_{\text{pool21}}, y_{\text{pool22}}, \ldots, y_{\text{pool2}n}\}$, $Z_{\text{full1}} = \{z_{\text{full11}}, z_{\text{full12}}, \ldots, z_{\text{full1}n}\}$.

6: **end for**

7: $(X'', Y'') = DCA(X_{\text{pool1}}, Y_{\text{pool2}})$.

8: $T_{X,Y} = concatenation(X'', Y'')$.

9: $(T''_{X,Y}, Z'') = DCA(T_{X,Y}, Z_{\text{full1}})$.

10: $T_{X,Y,Z} = concatenation(T''_{X,Y}, Z'') = x_i \in \Re^d$, $i = 1,2,\ldots,n$

11: **repeat**

$t = t + 1$;

12: **if** $d_{M_t}(x_i, x_k) - d_{M_t}(x_i, x_j) < \rho$ **then**

Calculate M_{t+1} using Eq. (20).

until $t = L$

13: Perform K-NN classification

Table 1 The proposed VGG-S1 architecture.

Conv1	Conv2	Conv3	Conv4	Conv5	Full6	Full7	Full8
$96 \times 7 \times 7$	$256 \times 5 \times 5$	$512 \times 3 \times 3$	$512 \times 3 \times 3$	$512 \times 3 \times 3$	4096-dropout	4096	C
st = 2, pad = 0	st = 1, pad = 0	st = 1, pad = 1	st = 1, pad = 1	st = 1, pad = 1	st = 1, pad = 0	st = 1, pad = 0	SVM
ReLU 1	ReLU 2	ReLU 3	ReLU 4	ReLU 5	ReLU 6	ReLU 7	
LRN							
pool 3×3	pool 2×2			pool 3×3			
st = 3, pad = 2	st = 2, pad = 1			st = 3, pad = 2			

3 Proposed VGG-S1 Architecture

In practice, only few researchers train an entire CNN from the beginning, the reason being that they rarely have a sufficient size dataset. Instead, it is common to remove the last fully connected layer from a pretrained network and treat the rest of the network as a fixed feature extractor for the new dataset classification. For better performance, the pretrained VGG-S[13] is fine-tuned from scratch on MSTAR database and then the fine-tuned network is treated as a fixed feature extractor. The network is fine-tuned as shown in Table 1: (1) for better normalization, local response normalization (LRN) layer parameters are modified; (2) to prevent over-fitting, a dropout[14] is adopted in the first fully connected layer (Full6); (3) for simplicity and better classification accuracy, the linear support vector machine (SVM) is used as a baseline classier instead of soft-max classifier.[15]

The first five layers of VGG-S1 are convolutional layers, and the remaining three are fully-connected layers. The convolutional layers' details are given in the subrows: the first indicates the number of convolution filters and their size; the second specifies the convolution stride "st" and the spatial padding "pad;" the third indicates rectified linear unit (ReLU);[16] the fourth indicates LRN if it is applied; and the fifth and the sixth indicate the max-pooling subsampling factor and the max-pooling stride and spatial padding, respectively, if max-pooling is applied. Specifically, the first convolutional layer is followed by their response normalization layer and max-pooling layer. Furthermore, the ReLU is applied to the output of every convolutional and fully connected layer. Full6 is regularized using dropout technique while the last layer (full8) acts as a C-class SVM classifier that is trained for.

3.1 *Local Response Normalization*

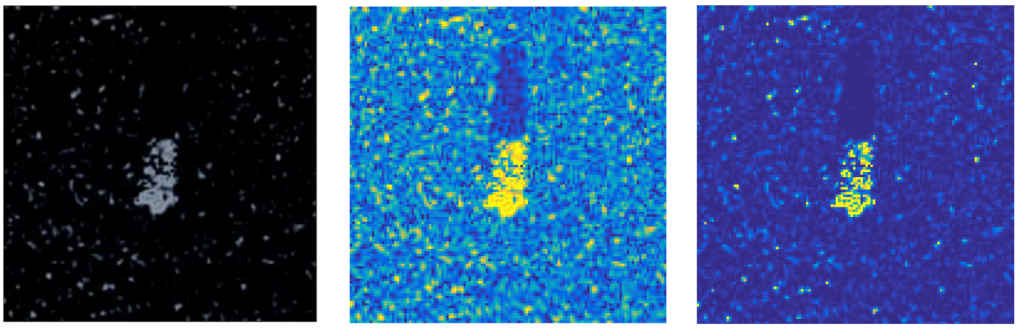
LRN is an important VGG-S building block. This operator is applied independently at each spatial location in the input map x to normalize the vector of feature channels as follows:

$$y_{ij\kappa} = x_{ij\kappa}\left[\kappa + \sum\nolimits_{\iota \in G(\kappa)} x_{ij\iota}^2\right]^{-\beta}, \tag{22}$$

where $G(k) = [k - \lfloor \frac{\rho}{2} \rfloor, k + [\frac{\rho}{2}]] \cap 1,2,\ldots,k$ is a group of ρ consecutive feature channels in the input map. In our fine-tuned VGG-S1, the parameters $\rho = 5$, $\kappa = 2$, $\alpha = 10^{-4}$, and $\beta = 2$ of LRN are changed to be 5, 2, -1.35, and 0.5, respectively. Figure 5 depicts LRN normalization of a processed SAR image.

3.2 *Dropout Regularization*

The best solution to improve the performance of machine learning methods is to combine the predictions obtained by large neural network architectures. However, averaging the outputs of many independently trained networks is computationally expensive. There are several



(a) (b) (c)

Fig. 5 LRN normalization: (a) processed SAR image, (b) normalized SAR image using VGG-S, and (c) normalized SAR image using VGG-S1.

techniques that are used to control the capacity of CNNs from over-fitting such as: L1, L2 regularizations and max norm constraints (max-norm). In our proposed VGG-S1, dropout technique[14] is applied, which is simple, effective, and complements L1, L2, and max-norm. During the training set, dropout is performed by randomly omitting each hidden unit with a probability of 0.5. It can be seen as sampling a different architecture of neural network and updating the sampled network parameters for each training case. During the testing set, it can be interpreted that there is no dropout implemented; only the outgoing weights of the hidden units are multiplied by 0.5. Convolutional layers suffer less from over-fitting because they have a smaller number of parameters compared to the number of activations. In addition, adding dropout to convolutional layers slows down the training. For these reasons, dropout is used in the fully connected layer.

3.3 *SVM Classifier*

Recently, most of the deep learning models utilize the soft-max classifier for prediction and minimize cross-entropy loss for classification.[17–19] In this paper, for better performance and parameter optimization, L2-SVM[15] is adopted to train VGG-S1 for SAR target classification, when the learning minimizes a margin-based loss by backpropagating the gradients from the top layer linear SVM.

4 Experimental Results and Analysis

In this paper, two different MSTAR public release datasets are used for the experiments, which are available to be downloaded.[20] The datasets are described and then the results attained throughout the classification are discussed.

4.1 *Datasets*

MSTAR public database is a standard database for testing SAR-ATR algorithms efficiency, in which this database consists of SAR images of ground vehicle targets from different types. All SAR images are with one-foot resolution collected by Sandia National Laboratory. They are collected using the STARLOS X-band SAR sensor in a spotlight mode with a circular flight path from diverse depression angles. The first dataset chosen to do our experiment is MSTAR public mixed target dataset, which represents mixed targets containing some military vehicle targets and a few civilian vehicle targets. Ten targets are used for classification including: armored personnel carrier: BMP-2, BRDM-2, BTR-60, and BTR-70; tank: T-62, T-72; rocket launcher: 2S1; air defense unit: ZSU-234; truck: ZIL-131; bulldozer: D7. The optical images of the 10 targets and their relative SAR images are shown in Fig. 6. The second dataset is the MSTAR public T-72 variants dataset. In this dataset, the training and testing sets are composed of eight variants of T-72: A04, A05, A07, A10, A32, A62, A63, and A64. Optical images and the corresponding SAR images of the eight T-72 targets are shown in Fig. 7.

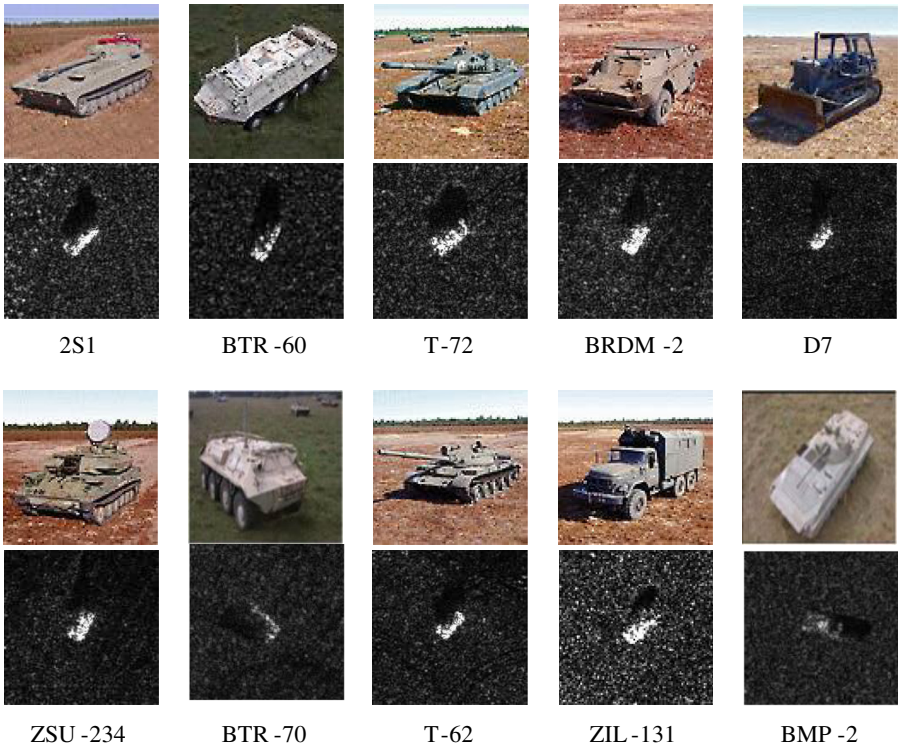


Fig. 6 Types of military targets: optical images in the top associated with their relative SAR images in the bottom.

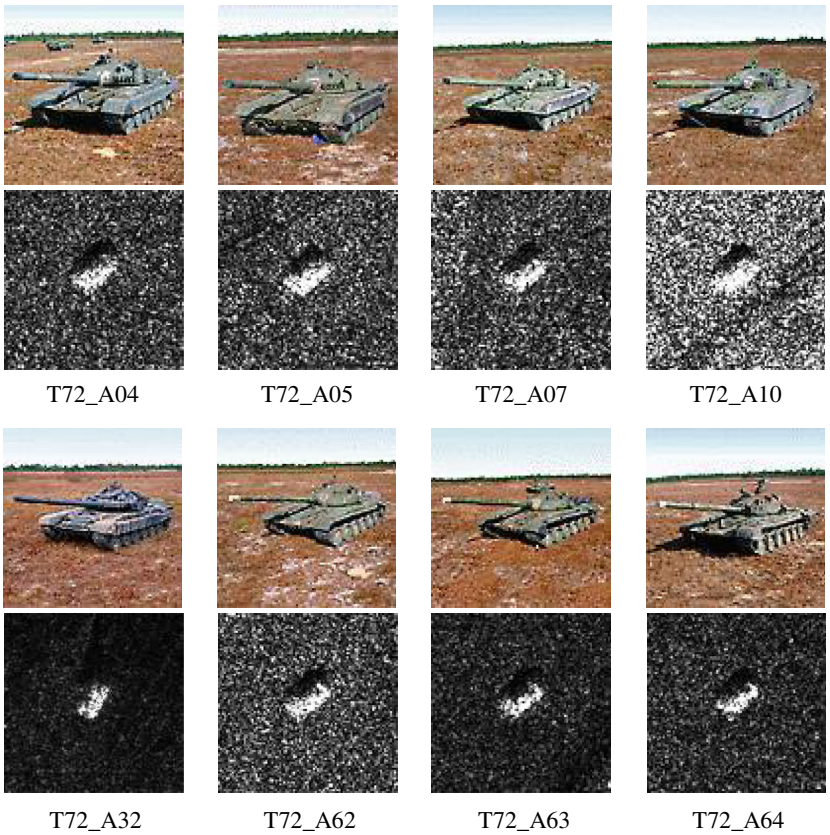


Fig. 7 T-72 multivariants: optical images in the top associated with their relative SAR images in the bottom.

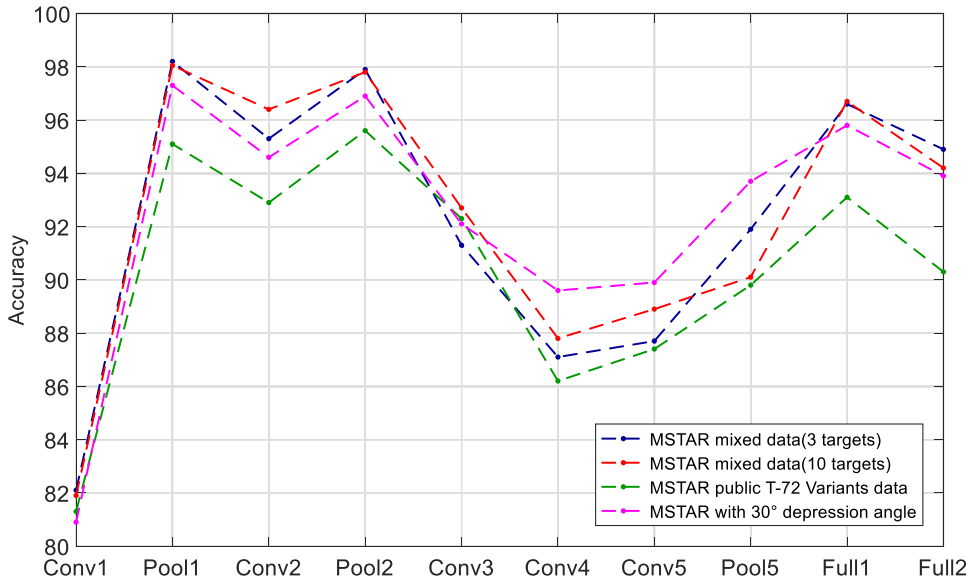


Fig. 8 Effect of the distinctive layer features on the classification accuracy with MSTAR database.

4.2 *Experimental Results*

First, VGG-S1 is used for SAR target classification, which can be divided into two parts: feature extractor and SVM classifier. The former part consists of five convolution layers, each followed by a ReLU layer, LRN layer, two max-pooling layers, and three fully connected layers. The size of the used convolutional filters are 7×7, 5×5 followed by three 3×3, the first fully connected layer is regularized using dropout technique, and the number of units in the last fully connected layer is C. The parameters of the VGG-S1 are trained with stochastic gradient descent and back-propagation. It usually costs 1 day on our graphics processing unit (GPU) card for training and about several minutes for testing. Our implementation has referred to the public available code provided by Vedaldi et al.[13] The evaluation is conducted on a 2.7-GHz CPU with 32 GB of memory and a moderate GPU card. All methods have been implemented using Microsoft Windows 10 Pro 64-bit and MATLAB R2016a. Second, the trained network is used to generate descriptor vectors from the response of a layer in the architecture for each input SAR image. In order to study the sensitivity of the distinctive layer features, we measured the overall classification accuracy on MSTAR database for the layer features and the experimental results showed that the first and the second max-pooling layer features attained the highest accuracy as shown in Fig. 8. In our method, we selected the first and the second max-pooling layers output as well as the first fully connected layer output. Moreover, DCA is utilized to fuse the descriptor vectors by concatenation forming new discriminant features of size $2(C-1)$. Then, LDMLT is used to learn the similarity metric and measure the feature space of the training samples. In the proposed method, a dimension matrix with $2(C-1) \times 2(C-1)$ is used to represent the Mahalanobis matrix and the cycle of dynamic triplets process is set as 7. In each cycle, the quantity of triplets is $N = 5n$, where n is the number of training samples. The parameter α is set as $\alpha = 1/N$. Finally, the classification of the targets with respect to its training set is done according to the classification error rates using two different classifiers K-NN and SVM, and the classification performance with K-NN induced the highest accuracy rates on MSTAR database as shown in Tables 3, 4, 6, and 8, respectively, where the variable k in K-NN classification is chosen as the number of classes as summarized in Fig. 9. The samples from each target class for the training and the testing sets are randomly selected and the final statistical results are evaluated after 5 runs.

The performance of the preprocessing step is also assessed and it shows that there is an improvement in the classification accuracy compared to the proposed method without preprocessing. The classification accuracies are compared in Tables 3, 4, 6, and 8, respectively. According to the results, it can be noticed that the proposed method with the preprocessing step achieves better performance compared with the others without preprocessing.

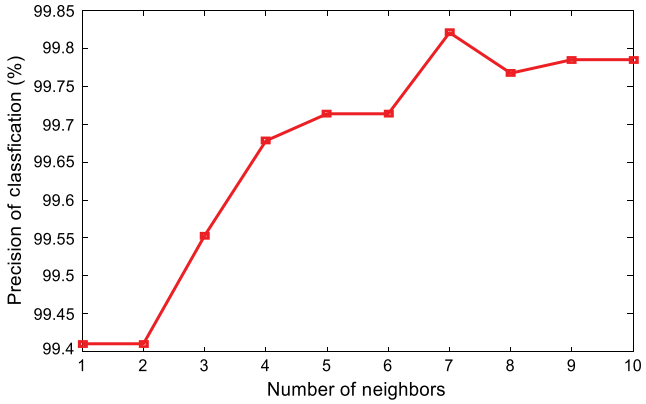


Fig. 9 The classification accuracy of the 10 targets dataset using K-NN with $k = 10$.

4.3 *MSTAR Public Mixed Target Dataset*

The classification performance of the proposed method is first evaluated on three targets: BMP-2, T-72, and BTR-70; then the robustness of the proposed method is evaluated on the more challenging 10 targets classification problem. In this experimental setup, the 15-deg depression angle data are used for testing and the 17-deg depression angle data are used for training purpose. Table 2 obviously lists all target types and the number of training and testing sets of each target type.

The classification performance of the proposed method is compared with several paradigms such as: A-convnets,[21] CNN,[22] CNN–SVM,[23] BCS + scattering centers,[24] and DWT + real-adaboost.[25] As shown in Tables 3 and 4, our proposed method produced the highest accuracy rates. The overall accuracies and the confusion matrices are clarified in Figs. 10–12, respectively.

Table 2 Number of training and testing samples used in the experiments for MSTAR public mixed target dataset.

		Train		Test	
Target		Depression (deg)	No. images	Depression (deg)	No. images
BMP-2	SN-C21	17	233	15	196
	SN-C9563	17	233	15	195
	SN-C9566	17	232	15	196
BTR-70		17	233	15	196
T-72	SN-132	17	232	15	196
	SN-812	17	231	15	195
	SN-S7	17	228	15	191
BTR-60		17	256	15	196
2S1		17	299	15	274
BRDM-2		17	299	15	274
D7		17	299	15	274
T-62		17	299	15	274
ZIL-131		17	299	15	274
ZSU-234		17	299	15	274

Table 3 The performance comparison between the proposed method and the state-of-the-art methods on three targets dataset.

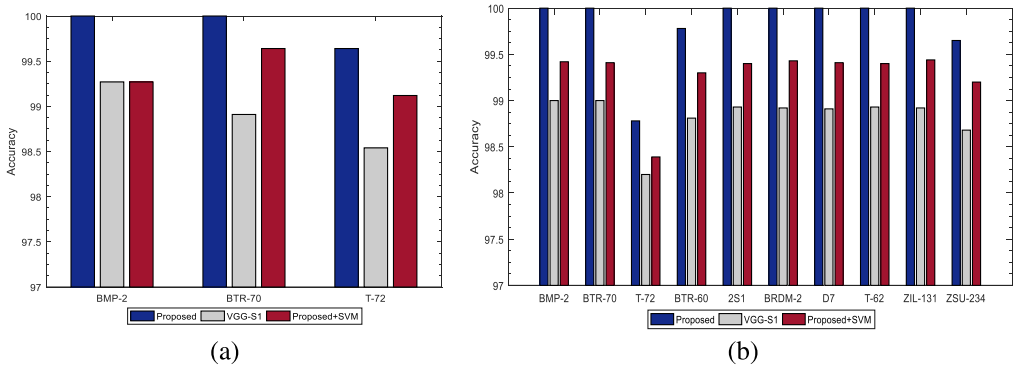
Method	A-convnets[21]	CNN[22]	CNN–SVM[23]	BCS + scattering centers[24]	VGG-S1
Accuracy (%)	99.3	90.1	99.6	97.3	98.9
Method	DWT + real-adaboost[25]		Proposed + SVM	Proposed without preprocessing	Proposed
Accuracy (%)	99.5		99.34	99.76	**99.88**

Note: Bold values provide the classification accuracies of the proposed method.

Table 4 The performance comparison between the proposed method and the state-of-the-art methods on 10 targets dataset.

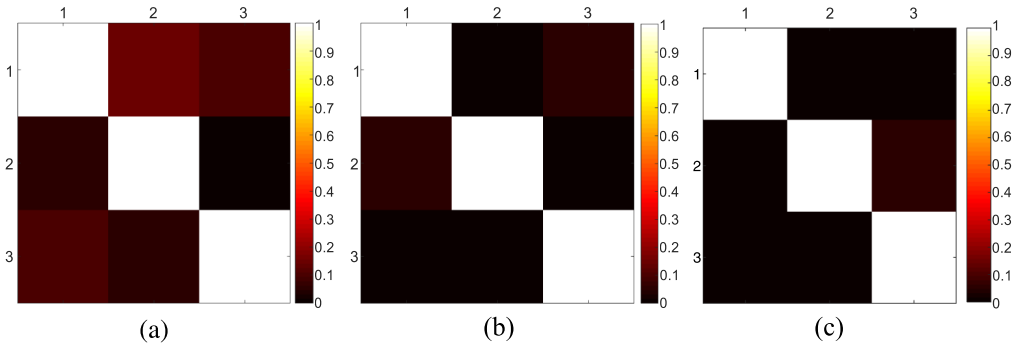
Method	A-convnets[21]	CNN[22]	CNN–SVM[23]	BCS + scattering centers[24]	VGG-S1
Accuracy (%)	99.13	84.7	99.5	92.6	98.83
Method	DWT + real-adaboost[25]		Proposed + SVM	Proposed without preprocessing	Proposed
Accuracy (%)	99.3		99.28	99.75	**99.82**

Note: Bold values provide the classification accuracies of the proposed method.



(a) (b)

Fig. 10 Produced accuracies on MSTAR public mixed target dataset for: (a) three targets dataset and (b) ten targets dataset.



(a) (b) (c)

Fig. 11 Confusion matrix of the classification performance on three targets dataset for: (a) VGG-S1; (b) the proposed with SVM; and (c) the proposed method. The rows and columns of the matrix indicate the actual and predicted classes, respectively. The class labels range from 1 to 3.

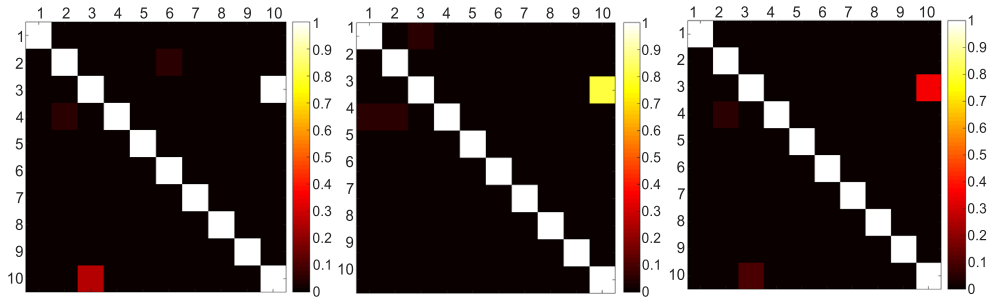


Fig. 12 Confusion matrix of the classification performance on 10 targets dataset for: (a) VGG-S1; (b) the proposed with SVM; and (c) the proposed method. The rows and columns of the matrix indicate the actual and predicted classes, respectively. The class labels range from 1 to 10 as shown in Fig. 6.

4.4 *MSTAR Public T-72 Variants Dataset*

In the same way, the MSTAR public T-72 variants are tested to evaluate the credibility of the proposed method more, since all the tank targets are almost indistinguishable. The depression angles and the number of images for the training and the testing are listed in Table 5.

The performance rate of the proposed method is compared with the same approaches as shown above. As displayed in Table 6, the proposed method attained the highest accuracy rate. The overall accuracies and the confusion matrices are shown in Figs. 13 and 14, respectively.

Table 5 Number of training and testing samples used in the experiments for the MSTAR public T-72 variants dataset.

	Train		Test	
Target	Depression (deg)	No. images	Depression (deg)	No. images
A04	17	299	15	274
A05	17	298	15	274
A07	17	299	15	274
A10	17	296	15	271
A32	17	298	15	274
A62	17	299	15	274
A63	17	299	15	273
A64	17	298	15	274

Table 6 The performance comparison between the proposed method and the state-of-the-art methods on the MSTAR public T-72 variants.

Method	A-convnets[21]	CNN[22]	CNN–SVM[23]	BCS + scattering centers[24]	VGG-S1
Accuracy (%)	95.43	81.08	95.75	88.76	94.15
Method	DWT + real-adaboost[25]		Proposed + SVM	Proposed without preprocessing	Proposed
Accuracy (%)	95.52		95.38	96	**96.16**

Note: Bold values provide the classification accuracies of the proposed method.

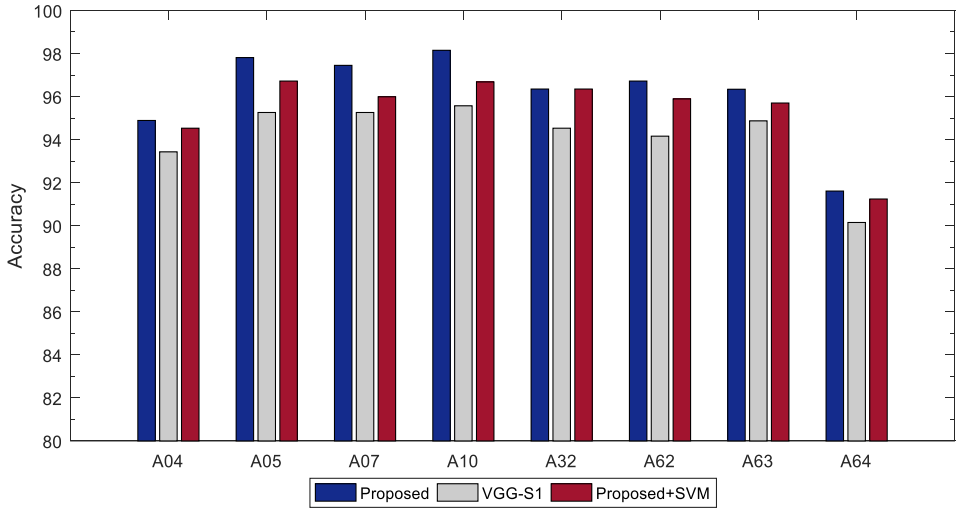


Fig. 13 Produced accuracies on MSTAR public T-72 variants dataset.

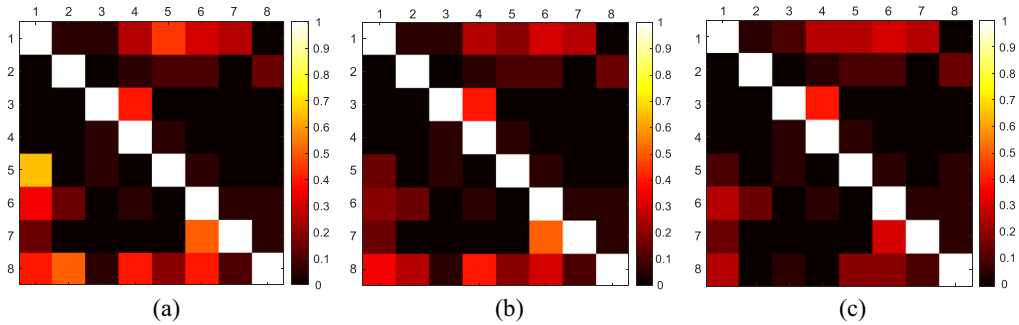


Fig. 14 Confusion matrix of the classification performance on the MSTAR publicT-72 variants for: (a) VGG-S1; (b) the proposed with SVM; and (c) the proposed method. The rows and columns of the matrix indicate the actual and predicted classes, respectively. The class labels range from 1 to 8 as shown in Fig. 7.

4.5 *MSTAR Public Database for Large Depression Angle Variations*

In another scenario test, as SAR images are very sensitive to depression angle variations, the proposed method is evaluated with large depression angle. In these experiments, only four types of targets (2S1, BRDM-2, T-72, and ZSU-234) from the MSTAR database are considered. The sample SAR images are obtained at 30 deg of depression angle. The types and the number of the adopted SAR images are shown in Table 7.

Table 7 Number of training and testing samples used in the experiments for the MSTAR public database with 30-deg depression angle.

Target	Type	Depression (deg)	No. images
2S1	B01	30	288
BRDM-2	E71	30	287
T-72	A64	30	288
ZSU-234	D08	30	288

Table 8 The performance comparison between the proposed method and the state-of-the-art methods on the MSTAR public database with 30-deg depression angle.

Method	A-convnets[21]	CNN[22]	CNN–SVM[23]	BCS + scattering centers[24]	VGG-S1
Accuracy (%)	96.12	81.56	96.6	92.6	95.83
Method	DWT + real-adaboost[25]		Proposed + SVM	Proposed without preprocessing	Proposed
Accuracy (%)	96.28		96.70	98.26	**98.44**

Note: Bold values provide the classification accuracies of the proposed method.

The experimental results as indicated in Table 8 clearly demonstrate that our method produced the highest accuracy rate through the comparison of the final classification accuracy with the same different approaches as shown above. The overall accuracies and the confusion matrices on the MSTAR public dataset with respect to large depression angle are well explained in Figs. 15 and 16, respectively.

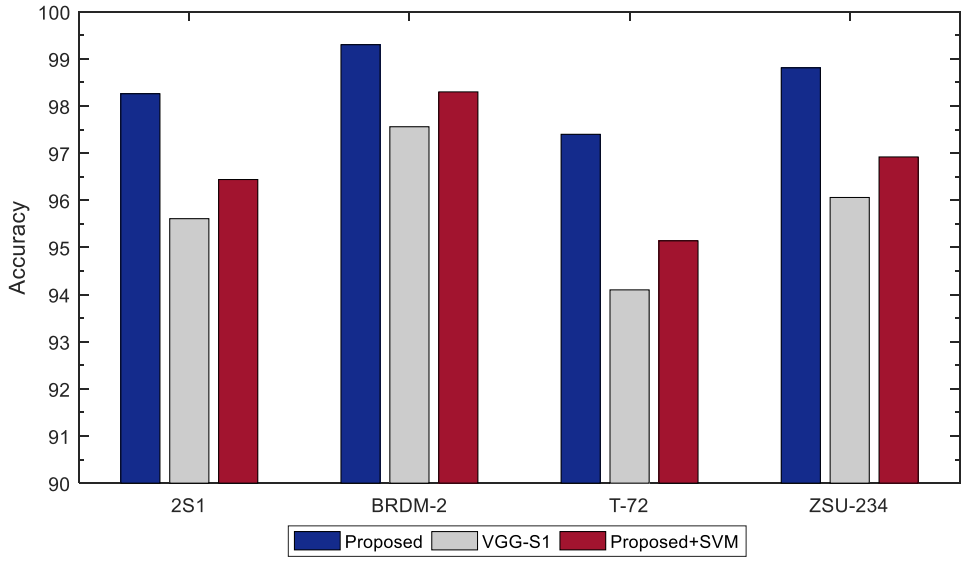


Fig. 15 Produced accuracies on MSTAR public database for large depression angle variations.

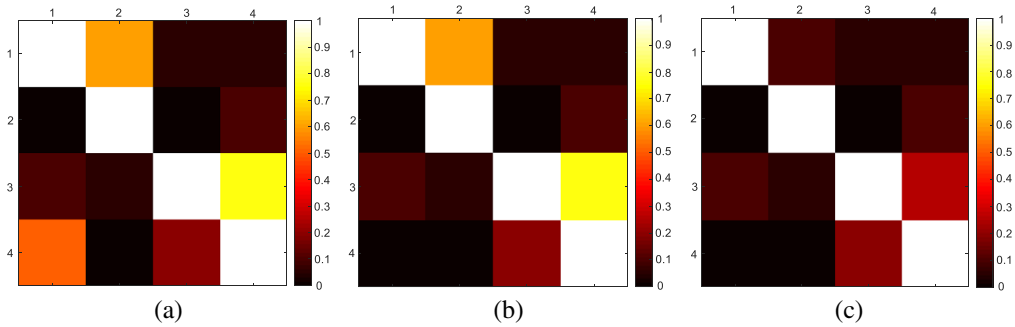


Fig. 16 Confusion matrix of the classification performance on the MSTAR public dataset with 30-deg depression angle for: (a) VGG-S1; (b) the proposed with SVM; and (c) the proposed method. The rows and columns of the matrix indicate the actual and predicted classes, respectively. The class labels range from 1 to 4 as shown in Table 7.

5 Conclusion

Feature extraction plays a dominated role in the classification performance of SAR-ATR. On this basis, this paper proposes an efficient feature extraction method, which takes advantages of exploiting deep features from CNNs to precisely represent the targets. First, VGG-S1 net is used to extract deep features from SAR images. Afterward, the DCA algorithm is adopted to combine the relevant features together obtaining discriminative features. Finally, K-NN is applied for matching and classification. Experimental results on the MSTAR database demonstrate the effectiveness of the proposed method compared with the state-of-the-art methods. Performance wise, the main contribution of the proposed method is the constructed features, which are characterized by the high discrimination through a variety of feature values, the independence through a low correlation between features, and the downscaling through the feature redundancy reducing.

Acknowledgments

This work was partially funded by the MOE–Microsoft Key Laboratory of Natural Language, Processing, and Speech, Harbin Institute of Technology, the Major State Basic Research Development Program of China (973 Program 2015CB351804), and the National Natural Science Foundation of China under Grant Nos. 61572155, 61672188, and 61272386. The author would like to thank Dr. Ahmad Hagag and Dr. Souleyman Chaib for their support.

References

1. C. Ozdemir, *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*, Vol. **210**, John Wiley & Sons, Hoboken, New Jersey (2012).
2. C. V. Jakowatz et al., *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach: A Signal Processing Approach*, Springer Science & Business Media, Berlin, Germany (2012).
3. C. F. Olson and D. P. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Trans. Image Process.* **6**(1), 103–113 (1997).
4. N. M. Sandirasegaram, *Spot SAR ATR Using Wavelet Features And Neural Network Classifier*, Defence Research and development Canada, Ottawa, Ontario, Canada, DRDC Ottawa TM 2005-154, Tech. Memorandum (2005).
5. Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.* **1**(4), 541–551 (1989).
6. A. Ulu et al., "Convolutional neural network-based representation for person re-identification," in *24th Signal Processing and Communication Application Conf. (SIU '16)*, pp. 945–948, IEEE (2016).
7. Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:14085882 (2014).
8. Ş. Karahan and Y. S. Akgül, "Eye detection by using deep learning," in *24th Signal Processing and Communication Application Conf. (SIU '16)*, pp. 2145–2148, IEEE (2016).
9. M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, "Discriminant correlation analysis: real-time feature level fusion for multimodal biometric recognition," *IEEE Trans. Inf. Forensics Secur.* **11**(9), 1984–1996 (2016).
10. J. Mei et al., "Logdet divergence-based metric learning with triplet constraints and its applications," *IEEE Trans. Image Process.* **23**(11), 4920–4931 (2014).
11. M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.* **3**(1), 71–86 (1991).
12. J. V. Davis et al., "Informationtheoretic metric learning," in *Proc. of the 24th Int. Conf. on Machine learning*, pp. 209–216 (2007).
13. K. Chatfield et al., "Return of the devil in the details: delving deep into convolutional nets," arXiv preprint arXiv:14053531 (2014).
14. N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).

15. Y. Tang, "Deep learning using linear support vector machines," arXiv preprint arXiv:1306.0239 (2013).
16. X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proc. of the Fourteenth Int. Conf. on Artificial Intelligence and Statistics*, pp. 315–323 (2011).
17. G. E. Dahl et al., "Phone recognition with the mean-covariance restricted Boltzmann machine," in *Advances in Neural Information Processing Systems* (2010).
18. S. Rifai et al., "The manifold tangent classifier," in *Advances in Neural Information Processing Systems*, pp. 2294–2302 (2011).
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1106–1114 (2012).
20. SDMS, "MSTAR data," https://www.sdms.afrl.af.mil/index.php?collection=mstar (23 November 2016).
21. S. Chen et al., "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.* **54**(8), 4806–4817 (2016).
22. S. Chen and H. Wang, "SAR target recognition based on deep learning," in *Int. Conf. on Data Science and Advanced Analytics (DSAA '14)*, pp. 541–547, IEEE (2014).
23. S. A. Wagner, "SAR ATR by a combination of convolutional neural network and support vector machines," *IEEE Trans. Aerosp. Electron. Syst.* **52**(6), 2861–2872 (2016).
24. X. Zhang, J. Qin, and G. Li, "SAR target classification using Bayesian compressive sensing with scattering centers features," *Prog. Electromagn. Res.* **136**, 385–407 (2013).
25. X. Zhao and Y. Jiang, "Extracting high discrimination and shift invariance features in synthetic aperture radar images," *Electron. Lett.* **52**(11), 958–960 (2016).

Moussa Amrani received his BSc degree in computer science from the Faculty of Engineering, University of Frères Mentouri, Constantine, Algeria, in 2009, and his MSc degree from the same university in 2012. He is pursuing his MSc degree at the School of Computer Science and Technology, Harbin Institute of Technology (HIT). His research interests are in the field of pattern recognition and classification of remote sensing images, distributed source coding, and image processing.

Feng Jiang received his BS, MS, and PhD degrees in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 2001, 2003, and 2008, respectively. He is currently an associate professor in the Department of Computer Science, HIT, China. His research interests include computer vision, pattern recognition, and image and video processing.