BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 3
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0035

One-vs-All Convolutional Neural Networks for Synthetic Aperture Radar Target Recognition

Bileesh Plakkal Babu, Swathi Jamjala Narayanan

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
E-mails: bileeshpbabu@gmail.com swathi.jns@gmail.com

***Abstract*:** *Convolutional Neural Networks* (*CNN*) *have been widely utilized for Automatic Target Recognition* (*ATR*) *in Synthetic Aperture Radar* (*SAR*) *images. However, a large number of parameters and a huge training data requirements limit CNN's use in SAR ATR. While previous works have primarily focused on model compression and structural modification of CNN, this paper employs the One-Vs-All* (*OVA*) *technique on CNN to address these issues. OVA-CNN comprises several Binary classifying CNNs* (*BCNNs*) *that act as an expert in correctly recognizing a single target. The BCNN that predicts the highest probability for a given target determines the class to which the target belongs. The evaluation of the model using various metrics on the Moving and Stationary Target Acquisition and Recognition* (*MSTAR*) *benchmark dataset illustrates that the OVA-CNN has fewer weight parameters and training sample requirements while exhibiting a high recognition rate.*

***Keywords*:** *Automatic target recognition, synthetic aperture radar, convolutional neural network, deep learning, computer vision.*

1. Introduction

Synthetic Aperture Radar (SAR) is widely used in the military for surveillance and reconnaissance due to its all-weather, day-and-night working capability. Despite the all-time working capability of SAR, the presence of speckle noise in SAR images has imposed a difficulty for domain experts to interpret the image. This paved the way for designing an Automatic Target Recognition (ATR) system for analyzing SAR images [1]. In the beginning, Lincoln laboratories have depicted the SAR ATR system as a three-stage sequential architecture that consists of detection, discrimination, and classification [2]. The accuracy in classifying the SAR images becomes crucial in determining the quality of the SAR ATR system. As a result, enhancing the recognition rate of the SAR ATR classification stage has become a research focus.

Image classification is a two-stage process that consists of feature extraction and classification based on the extracted features. The accuracy of such systems mainly

rely on the quality of the features used for classification. Therefore, the majority of researchers have focused on the development of novel feature extraction techniques. With the development of Convolutional Neural Networks (CNNs), the traditional two-stage classification approach has been replaced by a single CNN unit capable of feature extraction and classification. CNN has been successfully applied to several computer vision tasks such as image denoising, object detection, and classification [3-5]. The availability of a vast labelled dataset, especially the ImageNet dataset and the high computational GPUs, have fuelled the growth of deeper CNN architectures [6].

Recently, the breakthrough results in ImageNet classification competitions have drawn CNN's attention to the SAR community. The adoption of CNN has produced a ground-breaking result in SAR ATRs with minimum or no pre-processing requirements [7]. The significant improvement of SAR ATR's classification accuracy has attracted more researchers in this field to use CNN. For instance, G a o et al. [8] propose an enhanced deep CNN to extract features and use an SVM classifier to map the features to output labels. T i a n et al. [9] achieve above 99% classification accuracy by integrating weighted kernels into the CNN architecture to improve the feature extraction capability of the model. Z h a n g, X i n g and X i e [10] achieve good target recognition in various operating conditions by combining the CNN extracted features with the attributed scattering centre features. Furthermore, G u o et al. [11] have achieved state-of-the-art results in SAR ATR by modifying Hinton's capsule network.

While CNN produces impressive results, two significant issues impede the growing interest in CNN's application to SAR ATR. The first issue is that deep CNNs, particularly Alexnet, VGG-16, GoogLeNet, and ResNet-50, are memory-intensive models [12-15]. These models have 61M, 138M, 6.9M, and 12.2M weight parameters, which resembles the amount of memory required to deploy them. As a result, it is critical to minimize the number of weight parameters when using CNN in a memory-limited environment. The second issue is that deep CNNs require a large amount of labelled data to avoid overfitting. However, labelling SAR images is a time-consuming and expensive process.

Numerous techniques have been used in recent works to reduce CNN weight parameters. These techniques can be classified into two categories: structural modification and model compression [16]. While structural modification reduces parameters by reshaping the CNN's architecture, model compression enables the deployment of more sophisticated CNN models on low-power and resource-constrained devices through techniques such as pruning, quantization, low-rank factorization, and knowledge distillation. Recently, researchers have investigated the effect of structural modification of CNN in SAR ATR. The structural re-parameterization of CNN improves the performance by replacing convolutional layers of trained networks with FC layers, such as in the works [17, 18]. Likewise, C h e n et al. [19] have conducted experiments in SAR ATR using a modified A-convnet in which the convolutional layer replaces the Fully Connected (FC) layers. Similarly, Wagner demonstrates that replacing FC layers with Support Vector Machines (SVMs) could reduce the weight parameters of the CNN model [20].

However, the structural modification method, such as in [19], have an adverse effect on the computational complexity of the resulting CNN.

In CNN model compression, pruning removes the redundant parameters in CNN layers, and quantization reduces the number of bits required to represent the weight parameters. Zhong et al. [21] apply pruning and quantization techniques for the first time in the SAR ATR domain. Liu et al. [22] apply structured pruning to reduce parameter space of convolutional neural network for multitask target recognition in SAR. Similarly, Chen et al. [23] apply network pruning to yield a compact CNN model. To overcome the performance decline caused by network pruning, the author applies the knowledge distillation technique. Although pruning compresses CNN, the model's accuracy degrades when the connections are wrongly pruned [24]. Similarly, quantization is sensitive to the loss of the precision of the model. Yu et al. [25] note that the low-rank factorization decomposes CNN kernels into sparse. However, low-rank factorization requires a unique hardware setup to deploy on an embedded platform. Min et al. [26] recently use the knowledge distillation method to create a micro-CNN with only two layers, a compressed form of a CNN with eighteen layers. In another study, Zhang et al. [27] apply the knowledge distillation technique and achieve about 65 times the compression rate than A-convnets. On the other hand, the knowledge distillation technique only supports training the CNN model from scratch and needs a master network trained with a bulk quantity of data samples.

In recent years, SAR ATR have adopted transfer learning and data augmentation techniques to address the overfitting issues of CNN. Data augmentation simulates the data by applying image transformation techniques. Several studies have revealed that the CNN model better represents the targets while being trained with the simulated images. Peng et al. [28] simulate various targets such as tanks and aircraft and generated targets with various backgrounds to produce a CNN with more robustness. Ding et al. [29] have demonstrated the effectiveness of using augmented training data to reduce overfitting and improve the model's robustness. Pei et al. [30] propose a Multiview deep convolutional neural network trained on Multiview SAR data generated using augmentation techniques. The proposed framework has achieved superior performance and requires a small number of raw SAR images for network training. Yan [31] have enhanced the robustness of CNN to various operating conditions by using noisy samples at different signal-to-noise ratios, multiresolution representations, and partially occluded images. Furthermore, researchers have studied the effects of transfer learning on addressing the overfitting issues of CNN. Transfer learning utilizes CNN models pre-trained on a huge dataset and fine-tune the final layers using the target data. Zhong et al. [21] have trained a CNN by applying transfer learning on the ImageNet data set and then apply model compression techniques to reduce the weight parameters. Wang, Zhang and Leung [32] have trained a CNN model with sufficient simulated data and few real SAR data. Similarly, Malmgren-Hansen et al. [33] have pre-trained a CNN model using simulated data and have applied transfer learning to fine-tune the model with real SAR data. Furthermore, Wang et al. [34] apply the sub-aperture decomposition technique to convert a single-channel SAR image to three-channel SAR data to fine-tune a CNN model pre-trained on ImageNet data. It appears from

the investigations mentioned above that the idea of fine-tuning a pre-trained CNN model and the inclusion of augmented SAR images for training has a great potential to create a better-generalized CNN model. However, transfer learning requires a larger network trained with a large dataset and a source task similar to target recognition [35-36]. Therefore, such pre-trained models are not suitable for embedded applications like SAR ATR.

More recent attention has focused on the provision of using convolutional neural network ensembles for multiclass image classification. Combining several classifiers can enlarge the effective hypothesis [37]. In [38], the author proposes a lightweight CNN architecture with multiple streams that extract local and global features of the SAR target and achieves a high recognition accuracy. H a f i z and H a s s a b a l l a h [39] explore the potential of the One-Vs-All (OVA) ensemble technique and improves the performance of digit classification. P o l a t and K o c [40] apply the OVA technique on CNN for skin disease classification. Although there are few researches on ensemble CNN, no attempt has been made to investigate the effects of an ensemble of CNN on weight parameter reduction and overfitting of a CNN-based SAR ATR. This paper exhibits the strength of the OVA ensemble technique on reducing CNN's parameters and overfitting issues. To demonstrate the potential of this approach and its suitability for the SAR ATR application, OVA-CNN is investigated on the MSTAR dataset. The main contributions of this paper are specified below:

- The SAR ATR classifier is designed as an OVA ensemble of CNNs, which is the first work of its kind in this domain to our knowledge.
- The proposed OVA-CNN architecture consists of minimum weight parameters.
- The overfitting problem of CNN is reduced by using OVA technique.

The rest of this paper is organized as follows. Section 2 introduces the preliminary concepts of CNN. Section 3 describes the theory, architecture, and training algorithm of OVA-CNN. Section 4 presents the experimental results and compares the OVA-CNN with state-of-the-art SAR image classification methods. Finally, Section 5 concludes this paper.

2. Preliminaries

This section provides a brief overview of synthetic aperture radar based automatic target recognition system and the theoretical foundations of convolutional neural network.

2.1. Synthetic aperture radar-automatic target recognition system

SAR ATR is a computer processing system that predicts the target classes of SAR data without the need for human participation. ATR technology is widely employed in both military and civilian purposes. SAR ATR is traditionally divided into three stages: detection, discrimination, and classification. The first two stages of SAR ATR are combinedly known as the Focus-Of-Attention (FOA) module. The FOA module performs intensive computer processing on the SAR images. The FOA module serves

as an interface between the input SAR image and a group of targets, effectively processing the input SAR image to identify potential targets.

The detection stage in FOA removes visible clutters by employing a simple computational technique that filters out regions with an area smaller than a certain threshold. The discriminating unit in FOA is a binary classifier that selects only potential targets. A multiclass classifier is used as the final stage of a SAR ATR system. The potential targets from the preceding discrimination stage are fed into the classification stage. Based on the implementation method, the classification stage of a SAR ATR is divided into three categories: template matching based, model-based, and machine learning methods.

In template matching methods, the template-based systems store the SAR images under different conditions, such as different view angles and backgrounds, to describe the characteristics of the targets. In contrast, the model-based methods use the physical or conceptual models to describe the targets, such as CAD models or 3D scattering centre models. In a machine learning-based method, the target's features are extracted using feature extraction methods, and a classifier is trained based on these extracted features. The template matching-based method needs a huge storage requirement to store each template. Similarly, the model-based method requires separate mathematical modeling for each new target. However, huge storage requirements and mathematical modeling are not required in the machine learning-based method. Moreover, the advent of deep learning techniques such as convolutional neural networks unified the feature extraction unit and the classifier into a single unit to produce state-of-the-art results.

2.2. Convolutional neural network

The convolutional neural network proposed by LeCun is a modified feed-forward neural network with neurons in the initial layers are convolution operations [41]. CNN comprises a two-stage architecture that includes a feature extractor and a classifier, combinedly provides automatic feature extraction and end-to-end training with minimum pre-processing requirements.

Conventionally, CNN is composed of three layers viz. convolutional layer, pooling layer, and Fully Connected (FC) layer. The main component of the convolutional layer is a convolution operation in which a two-dimensional kernel K of size $k \times k$ convolved over a two-dimensional input data I of size $N \times N$ with a stride S to produce a feature map F_l of size $((N-k)/s)+1$ on each axis and the computation in one neuron is expressed as

$$F_l = \sum_{i=0}^{N-k} \sum_{j=0}^{N-k} I(i,j) \odot K + b_l \tag{1}$$

where '$\odot$' denotes the convolution operation, which is a pixel-wise dot product and addition process, and b_l is the bias vector of the l-th layer.

The addition of a non-linear activation function, such as ReLu placed after the convolution operation enhances the network's ability to represent highly complex data and speeds up training [34]. The ReLu activation function on a feature map F_l is defined as

$$F_l^{\text{new}} = \max(0, F_l), \tag{2}$$

where F_l^{new} is the non-linear feature map. The max() function returns the highest non-zero pixel value in the convolution window.

Batch normalization is a technique used to standardize the training input and stabilize the learning process of deep neural networks [43]. By applying to a small batch of training data, batch normalization reduces the number of training iterations significantly. This layer is composed of two weight parameters learned during the training process: mean and standard deviation.

The pooling layer adds translational invariance to feature maps and reduces their dimension. Pooling reduces the feature map of size $N \times N$ into $(N/k) \times (N/k)$ over a $k \times k$ size window. Maxpooling is a pooling operation that determines the maximum value contained within each patch of a feature map.

After an arbitrary number of convolution and pooling layers, the feature maps are transformed into a one-dimensional vector and stacked with one or more FC layers. The FC layers are similar to the layers in a multi-layer perceptron. The number of neurons in the final FC layer is the same as the number of classes to be predicted. In the final layer of CNN, a softmax function is used which is computed as

$$P(X_i) = \frac{e^{X_i}}{\sum_{j=0}^{M} e^{X_j}}, \tag{3}$$

where $P(X_i)$ is the probability of a target to be in the *i*-th class.

An optimization algorithm is used to minimize a loss function during the training of a CNN. Some of the most frequently used loss functions in deep neural networks are binary cross entropy, categorical cross entropy, and hinge loss. The cross-entropy function quantifies the dissimilarity of the actual and predicted probabilities and it is formulated as

$$L(y, P) = -\sum_i y_i \log(P_i), \tag{4}$$

where $L(\cdot)$ is the loss function, y_i is the actual label of *i*-th sample, and P_i is the predicted probability of *i*-th sample.

The optimization function such as Adam, optimizes the weight parameters and bias variable using gradient descent principle [44]. The total number of parameters in CNN equals the sum of all weights and biases in convolutional layers and fully connected layers. The number of parameters in the convolution layer is determined as

$$W_{\text{c}} = \sum_{l-1}^{L} (F_{l-1}^2 N_l k^2) + N_l, \tag{5}$$

where W_{c} denotes the total number of parameters in convolutional layers, F_{l-1} denotes the feature map size of previous layer, F_l denotes the feature map of the current layer, N_{l-1} and N_l represent the number of neurons in previous layer and current layer, respectively. Similarly, the number of parameters in fully connected layers is calculated as

$$W_{\text{f}} = \sum_{l-1}^{L} N_{l-1} N_l, \tag{6}$$

where W_{f} denotes the number of parameters in fully connected layers. The total number of parameters in CNN is formulated as

(7) $$W = W_{\mathrm{c}} + W_{\mathrm{f}} + (2 * N_{\mathrm{batch}}),$$

where W denotes the total number of CNN parameters and N_{batch} denotes the number of batch normalization layers. Due to the weight sharing mechanism, the number of weight parameters in convolutional layers are fewer when compared to the number of parameters in fully connected layers, i.e., $W_{\mathrm{f}} \gg W_{\mathrm{c}}$. The conventional CNN has an excessive amount of weight parameters and is therefore prone to overfitting. It is inapplicable to the SAR ATR domain due to the application's low memory requirements and limited availability of labelled images.

3. Proposed methodology

To overcome the issues of conventional CNN, this paper proposes a novel OVA-CNN for SAR ATR of ground military vehicles. This section will first introduce the theoretical aspects of the OVA technique. The architecture of OVA-CNN is then described, followed by the OVA-CNN training algorithm.

3.1. Theoretical aspects of OVA-CNN

While optimizing a single error function in a multiclass classification problem, the training algorithm gets stuck at the local minima. Hence, finding an optimal hypothesis using a single classifier is difficult for applications with limited data availability.

In a multiclass classification problem, a CNN classifier C trained on the samples $S = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where n is the number of the training samples, x_i belongs to a d -dimensional vector, and y_i is the label from $Y = 1, \ldots, k$, maps an unknown sample X to Y_j where $1 \leq j \leq k$. In OVA-CNN, the multiclass classification problem is split into multiple binary classification subproblems. Each subproblem s is solved using the binary classifier BCNN, that maps an unknown sample X to $Y = \{-1, 1\}$.

Vapnik-Chervonenkis (VC) dimension measures the capacity of a set of functions that a classification algorithm can learn [45]. It is defined as the largest set of points that the algorithm can shatter. The generalization ability of a classifier is often related to its VC dimension. A model with a higher VC dimension can identify more complex relationships in the data but requires more training data. The relationship between the error rate ε and the VC dimension VCdim of a classifier is given in [46] as

(8) $$\frac{2e\varepsilon \mid S \mid}{\mathrm{VCdim} - 1} = 1,$$

where $|S|$is the number of training samples, and e denotes the base of the natural logarithm. That is, for a fixed value of $|S|$, ε is in proportion to VCdim. Hence, for a constant number of training samples, the classifier's error rate can be reduced by reducing the VC dimension of the classifier. Moreover, the VCdim of a classifier is directly proportional to the model complexity P. The relationship between the number of classes K and the model complexity P is defined in [45] as

(9) $$K = 2^{P/d},$$

where d is the dimension of the input data. Equation (9) can be rearranged as

$$P = \log_2 K * d. \tag{10}$$

According to (10), the classifier's complexity is directly proportional to the number of classes to be predicted for a fixed dimension of training samples. As a result, by reducing the VC dimension of the classifier, a classifier with reduced model complexity can be designed without compromising the error rate for classifying a smaller number of classes. The VC dimension can be reduced in two ways: by reducing the depth of the CNN or by reducing the width of each layer in the CNN. As CNN depth has a greater impact on classification accuracy, we designed BCNNs with fewer parameters by reducing the width of each layer in the CNN given in [31].

3.2. Architecture of OVA-CNN

The architecture of the OVA-CNN illustrated in Fig. 1 is composed of two major components: an ensemble of binary classifying CNNs (BCNNs) and a maximum-value selector. The ensemble module consists of ten BCNNs. Although the architecture of every BCNN is similar, each BCNN gives a different probability for the same target. The BCNNs in OVA-CNN act as an expert in classifying a single target correctly. That is, the BCNN, which is designed to recognize the target "T72"; could not recognize any other targets correctly. Each BCNN predicts a probability for an input SAR image during the testing phase. The max-value selector determines the final class based on the BCNN with the highest predicted probability.

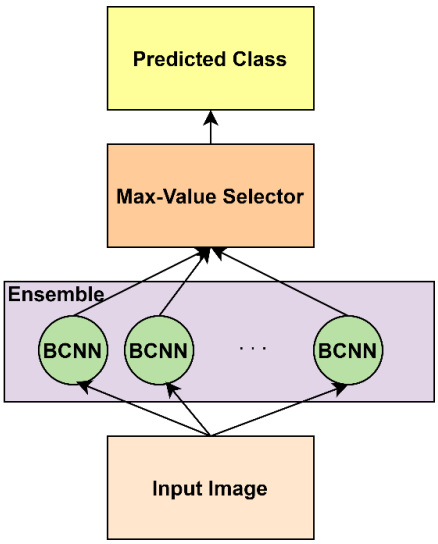

Fig. 1. OVA-CNN architecture

The BCNN's structure is depicted in Fig. 2. The structure of BCNN is analogous to the conventional CNN described in [31]. The term "CONV" in Fig. 2 refers to the convolution layers with 5×5 kernels. "Pooling" is a two-by-two maxpooling operation. Batch normalization layer, non-linear activation function, fully connected layer, and softmax layer are denoted by BN, ReLU, FC, and softmax, respectively.

After the first “CONV” and “Pooling” layer, 16 feature map of size 10×10 is produced. Similarly, after the second “CONV” and “Pooling” layer, 32 feature maps of size 3×3 are produced, which is flattened into a 1×288 sized feature vector. The feature vector is then fed into the classifier part of the BCNN that consists of multiple FC layers. The final FC layer consists of two neurons, and each produces a probability for the target into the positive class and in the negative class.

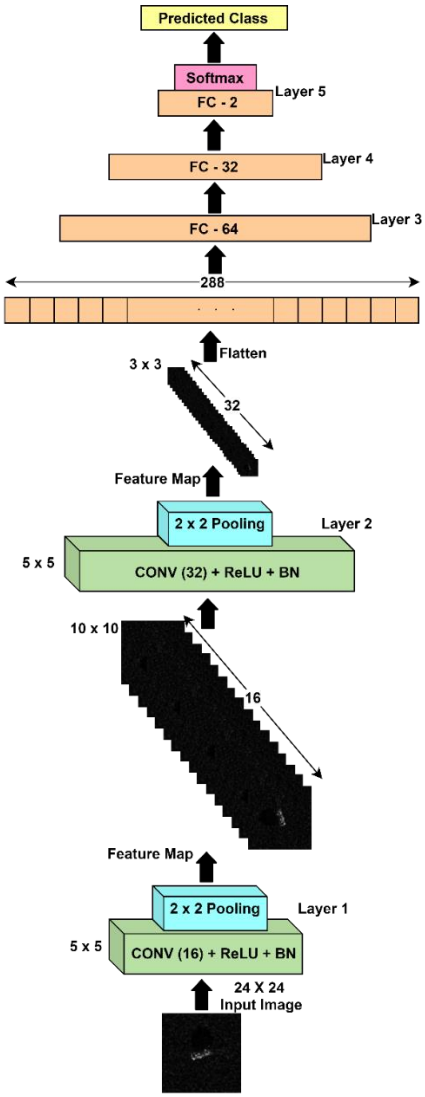

Fig. 2. Structure of BCNN

The number of neurons in each layer of BCNN is reduced by reducing the VC dimension of the model presented in [31]. While the CNN in [31] has {64, 64, 384, 192, 10} neurons in each of its layers from one to five, the BCNN has

{16, 32, 64, 32, 2} neurons in its respective layers. BCNN's weight parameters and VC dimension have been significantly reduced as a result of this modification to the design. Thus, even when training with fewer training samples, the design of BCNN avoids overfitting.

3.3. Training of OVA-CNN Training Algorithm

The OVA-CNN Training Algorithm depicted in Table 1. has three major sections: a method to split the dataset, a method to train each BCNN, and a method to combine the BCNNs' predictions. Theoretical and experimental studies have demonstrated that improved generalization can be achieved by combining the output of negatively correlated classifiers. In OVA-CNN, each BCNN is trained using a different training set and produces uncorrelated classifiers. Steps 1-3 in the training algorithm define the OVA splitting process to create distinct subsets to train the BCNNs. Each of these subsets contains one target as a positive class and the rest of the targets as a negative class. For instance, the subset used to train the BCNN for classifying the "T72" target contains the "T72" target with positive labels and the remaining targets with negative labels.

Table 1. Pseudocode of OVA-CNN Training Algorithm

Input: Training dataset D_{train}, augmentation techniques, L, and the number of classes k
Output: An ensemble C
Step 1. Split D_{train} to $\{D_1, D_2, \ldots, D_k\}$ such that $D_i \neq D_j$ where $0 \leq i, j \leq k$
Step 2. for each augmentation τ in L
Step 3. for each sample s in D_i
a) $S_{\text{aug}} \leftarrow \text{augmentation}(s, \tau)$
b) $D_{i_{\text{modified}}} \leftarrow D_i \cup S_{\text{aug}}$
c) $i \leftarrow i + 1$
Step 4. While $i \leq k$
a) $\text{BCNN}_i \leftarrow \text{train}(D_{i_{\text{modified}}})$
b) $i \leftarrow i + 1$
Step 5. return $C = \bigcup_{i=1}^{k} \text{BCNN}_i$

The training of BCNNs in OVA-CNN is defined in Step 4 in Table 1. Each BCNN is designed as described in Section 3.2. BCNNs are trained by optimizing the weight parameters and bias values using the Adam optimization algorithm. Although each BCNN is designed similarly, they are trained using distinct sets of training data. As a result, the weight parameters and bias values of each BCNN are unique.

During test phase, the results of the BCNNs are combined to predict the final class of an unknown sample. The combining strategy used in OVA-CNN is formulated as

$$\text{class} = \text{argmax}_{i=0,\ldots,K} c_i, \tag{11}$$

where c_i is the *i*-th classifier's probability. The final class predicted by OVA-CNN is the output class of the classifier that gives the largest positive value for a given SAR image.

4. Result and discussion

In this section, we firstly introduce the dataset and the relevant experimental settings followed by the findings and outcomes of the experiment.

4.1. Dataset and experimental setup

The effectiveness of OVA-CNN has been validated using the MSTAR dataset, collected as part of the MSTAR program [47]. This dataset consists of publicly available ten types of military vehicles captured as *X*-band spotlight SAR images with 128×128-pixel size, 0.3 m resolution, and measured over entire azimuth angles. The military vehicles in this dataset are armoured personnel carriers: BMP-2, BRDM-2, BTR-60, and BTR-70; rocket launcher 2S1; truck ZIL-131; tank T-62 and T-72; air defence unit ZSU-234; and bulldozer D7.

In the MSTAR dataset, the experiments have been carried out in two different scenarios. The algorithm is tested under Standard Operating Conditions (SOC) and Extended Operating Conditions (EOC) to evaluate the performance effectiveness. SOC refers to the fact that the serial numbers and target configuration is similar to the training set but with varying aspect angles and depression angles. There are significant differences in depression angle, target articulation, and version variants between the training and test sets in EOC test scenarios.

The experiments have been conducted on a laptop with the Windows 10 Pro operating system, 2.2 GHz Intel i3-3210 CPU, and 4 GB RAM. The program codes are written in Python, and the Keras API is used to develop the BCNN models. Each BCNN is trained for 100 epochs using the Adam optimization algorithm with a fixed learning rate of 0.01.

4.2. Results under SOC

The training and test data for experiments under SOC is listed in Table 2. The data measured at a 17-degree depression angle constitutes the training set, and a 15-degree depression angle is used as the test set.

Table 2. Distribution of training set and test set

Class	Serial No	Training images	Test images
BMP2	9563	233	195
BTR70	C71	233	196
T72	132	232	196
T62	A51	299	273
BRDM2	E71	298	274
BTR60	K10yt7532	256	195
ZSU23/4	D08	299	274
D7	92v13015	299	274
ZIL131	E12	299	274
2S1	B01	299	274

Table 3 presents the confusion matrix of the ten-class classification problem in SAR- ATR. The rows of the confusion matrix correspond to the target's actual class, and the columns indicate the classifier's prediction. It can be seen from Table 3 that OVA-CNN has a recognition rate of more than 97% for all the targets. Furthermore, the overall classification accuracy of OVA-CNN under SOC is 98.81 percent, which indicates the strength of the OVA technique.

Table 3. Confusion matrix and PCC of ten class classification

Class	BMP 2	BTR70	T72	T62	BRDM2	BTR60	ZSU23/4	D7	ZIL131	2S1	PCC (%)
BMP 2	**193**	0	1	0	1	0	0	0	0	0	98.97
BTR70	0	**194**	0	0	0	2	0	0	0	0	98.97
T72	0	0	**196**	0	0	0	0	0	0	0	100
T62	0	0	0	**267**	0	1	1	1	3	0	97.80
BRDM2	0	0	1	0	**271**	1	0	0	1	0	98.90
BTR60	0	2	1	0	0	**191**	0	0	1	0	97.94
ZSU23/4	1	0	0	0	0	0	**272**	1	0	0	99.27
D7	0	0	0	0	2	0	0	**272**	0	0	99.27
ZIL131	0	0	0	0	0	0	1	0	**273**	0	99.63
2S1	0	0	0	1	1	2	1	0	2	**267**	97.44
Avgerage	98.81										

Next, we evaluate the OVA-CNN using the error rate, number of weight parameters, and training samples. The error rate measures the defectiveness of the model on the classification task and gives the overall percentage of wrongly classified targets. The number of weight parameters directly implies the size of the CNN model. A smaller CNN model can train faster and store on-chip directly while deploying. The number of training samples used to train the CNN indicates the amount of data required to converge the model during training. Training a CNN model with fewer images is essential in SAR ATR, where the availability of labeled data is limited. Table 4 compares the results of OVA-CNN with state-of-the-art SAR ATR models.

Table 4. Comparison of OVA-CNN and other CNN models under SOC

CNN	Error rate	No of parameters	No of training samples
In [31]	1.94	1.06	247231
In [19]	0.87	0.30	4620464
In [26]	1.8	0.06	46699
In [21]	1.61	0.22	4620464
In [11]	0.21	6.01	175808
OVA-CNN	1.19	0.34	24723

In addition to CNN in [19] and [11], OVA-CNN has the lowest error rate, as shown in Table 4. It is to be noted that CNN in [19] needs 188 times more training data than OVA-CNN and both models contain almost the same number of weight parameters. Similarly, when compared to CNN in [11], OVA-CNN has 17 times fewer weight parameters. From the data given in Table 4, it can be seen that OVA-CNN needs fewer training samples than other networks under comparison, and hence proves that OVA-CNN is superior to most CNNs. The results from Table 4 reaffirms that the ensemble of multiple binary-classifying CNNs has more impact on the recognition rate than a single multi-class CNN.

4.3. Results under EOC

SAR images are highly vulnerable to variations in depression angles. Hence, evaluating the robustness of a SAR ATR system with respect to the difference in the depression angle is crucial for specific applications. Table 5 provides an overview of training and test images for experiments based on the depression angle variation (EOC-1).

As shown in Table 5, the EOC-1 experiment consists of four target categories: 2S1, BRDM-2, T-72, and ZSU23/4. The images with a depression angle of 17 degrees have been used as training samples, while the images with a depression angle of 30 degrees have been used as test samples. The confusion matrix for EOC-1 is given in Table 6.

Table 5. EOC-1 data distribution

Class	Serial No	Training images	Test images
2S1	b01	299	288
T72	A64	232	288
BDRM2	E71	298	287
ZSU23/4	D08	299	288

Table 6. Confusion matrix of EOC-1

Class	2S1	T72	BRDM2	ZSU23/4	Accuracy (%)
2S1	**288**	0	0	0	100
T72	8	**279**	1	0	96.87
BRDM2	5	2	**280**	0	97.56
ZSU23/4	4	0	0	**284**	98.61
Average					98.26%

It can be seen from Table 6 that all targets are classified with high accuracy, which implies the robustness of the model to variation of depression angle. However, the targets T72, BRDM2, and ZSU23/4 are occasionally misclassified as 2S1. All these target vehicles consist of a turret mounted above the fighting compartment. This might be the cause for the misclassification of these targets. We also assess the robustness of the proposed method for various target configurations and versions. The training set in this experimental setup consists of four targets (BMP2, BRDM2, BTR70, and T72) at a 17-degree depression angle, as shown in Table 7. The test set comprises two groups, as given in Table 8 (EOC-2) and Table 9 (EOC-3).

Table 7. Training data of EOC-2 and EOC-3

Class	Serial No	Depression	No of samples
T72	132	17	232
BMP2	9563	17	233
BRDM2	E71	17	298
BTR70	C71	17	233

Table 8. Test data of EOC-2

Class	Serial No	Depression	No of samples
T72	S7	15 and 17	419
	A32	15 and 17	572
	A62	15 and 17	573
	A63	15 and 17	573
	A64	15 and 17	573

Table 9. Test data of EOC-3

Class	Serial No	Depression	No of samples
T72	S7	15 and 17	419
	A32	15 and 17	572
	A62	15 and 17	573
	A63	15 and 17	573
	A64	15 and 17	573
BMP2	9566	15 and 17	428
	C21	15 and 17	429

As listed in Table 8, the EOC-2 test set contains five configuration variants of the target T72 that are not included in the training set. Similarly, the EOC-3 test set in Table 9 contains two BMP2 version variants and five T72 version variants. The targets at depression angle 15 degrees and 17 degrees are included in the EOC-2 and EOC-3 test sets.

The confusion matrices for EOC-2 and EOC-3 are provided in Table 10 and Table 11, respectively. From Table 10, it can be seen that the proposed network achieves a recognition rate of 98.62%. Table 11 shows that the recognition rate for EOC-3 test data is 98.33%. The results presented in these tables illustrate that the proposed network has an excellent performance in recognizing the targets with different versions and configurations. According to the results in Table 10 and Table 11, BMP2 and T72 are frequently confused. It seems possible that these results are due to the geometrical similarities between these two targets.

Table 10. Confusion matrix of EOC-2

Class	BMP2	BRDM2	BTR70	T72	Accuracy (%)
A32	7	0	0	565	98.77
A62	3	0	0	570	99.47
A63	7	0	0	566	98.77
A64	10	0	0	563	98.25
S7	9	0	0	409	97.84
Avgerage					98.62

Table 11. Confusion matrix of EOC-3

Class	Serial No	BMP2	BRDM2	BTR70	T72	Accuracy (%)
BMP2	9566	411	1	2	14	96.02
	C21	422	1	0	6	98.36
T72	A04	6	2	0	565	98.60
	A05	1	0	0	572	99.82
	A07	3	0	0	570	99.48
	A10	1	0	0	566	99.82
	812	15	1	0	410	96.24
Avgerage						98.33

Next, we compare the performance of OVA-CNN with the recently proposed CNN models in the literature. The average classification accuracies of these algorithms are provided in Table 12. While comparing OVA-CNN with CNN in [11] and [19], it can be seen that the OVA-CNN has a competitive result in SOC and EOC experiments. The results indicate that the OVA ensemble of CNN has a good recognition capability, fewer weight parameters and training samples.

Table 12. Comparison of OVA-CNN with other CNN models

Model	SOC (%)	EOC-1 (%)	EOC-2 (%)	EOC-3 (%)
CNN [31]	98.06	-	-	-
CNN [19]	99.14	9.12	99.83	98.60
CNN [26]	98.20	95.7	-	-
CNN [21]	98.39	-	-	-
CNN [11]	99.79	98.78	99.80	99.35
OVA-CNN	98.81	98.26	98.62	98.33

4.4. Visualization of OVA-CNN

Although CNN gives promising results, the working mechanism of CNN remains a black box. Hence, visually interpreting CNN has drawn wide attention [48, 49]. In this section, the feature maps of the first and second convolutional layers of OVA-CNN are visualized and interpreted for SAR images. Eight feature maps of the first and second convolutional layers of OVA-CNN are depicted in Figs 3 and 4, respectively. The analysis of the feature maps gives insight into the transformations of the target image at classified objectives.

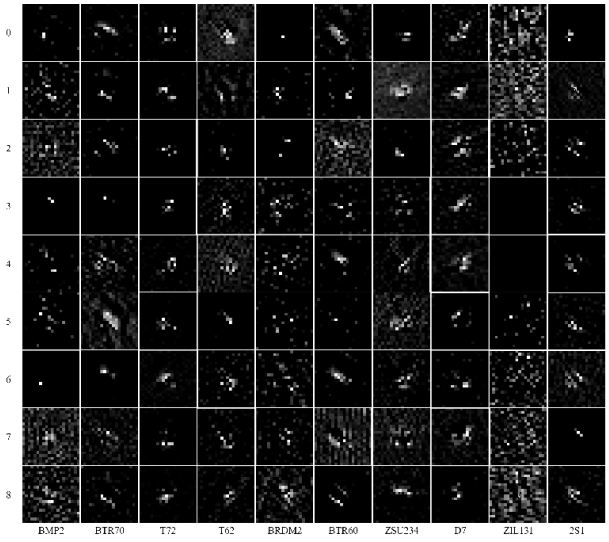

Fig. 3. Feature map of first convolutional layer of OVA-CNN

From Fig. 3, we analyze that the filters in the first layer of OVA-CNN extract generic features of the target. In contrast, the second layer of convolutional filters extracts more abstract features, as illustrated in Fig. 4. Although the BCNN for each target has a similar architecture, the features learned by these filters differ, as one BCNN's training is independent of the other. Hence, the features extracted by one feature map for a particular target differ from a similar feature map of another target. This is more evident in Fig. 3. While feature map three of BMP2 and BTR70 consists of a few pixels, feature map three of other targets contain more information about the target. Moreover, the dissimilarity between the feature maps of a target exhibits that the redundant feature maps are not produced during the training phase. Similarly, the feature map of one target varies from all other targets, which shows the effectiveness

of the combination of classifiers to predict the final class. Instead of the generic features, the feature maps of the second convolutional layer extract more abstract features specific to each target. The filter corresponding to feature map three of Fig. 4 extracts the lower-right features of the target, while the filter of feature map eight extracts the lower-left features of the target.

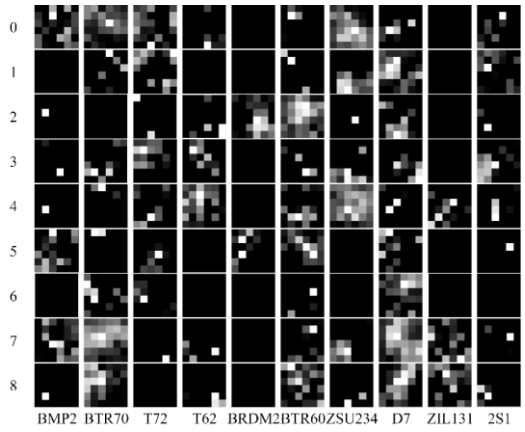

Fig. 4. Feature map of second convolutional layer of OVA-CNN

To analyze the effect of spatial parameters of the SAR target based on the augmentation techniques applied, the feature maps corresponding to each augmentation technique of the target T72 is visualized in Fig. 5.

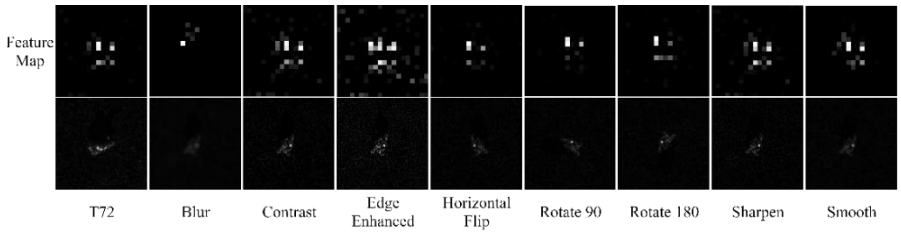

Fig. 5. Feature map of T72 target after applying various augmented techniques.

From Fig. 5, it can be seen that the feature maps of the target T72 slightly varies based on the augmentation technique applied to the target. When the blurring effect is applied on the target T72, a few of the original image features disappear in the blurred target's feature map. Similarly, the feature map of the edge-enhanced version of T72 extracts more information compared to the original image. Hence, we interpret that the usage of augmented images for training improves the robustness of the CNN model and causes an improvement in the generalization of the model.

5. Conclusion

CNN has been widely used in SAR ATR to produce state-of-the-art results, but it is challenging to train CNN with limited labeled images. Moreover, the huge number

of weight parameters in CNN impedes its deployment on embedded platforms. This paper explains the effects of the one-vs-all ensemble technique on reducing parameters and the training samples of a CNN classifier. Specifically, the OVA-CNN splits the multiclass classification problem among multiple binary classifying CNNs. The OVA- CNN predicts the label by choosing the class corresponding to the binary classifier that gives the highest probability for any unknown target. The extensive studies on the benchmark MSTAR dataset illustrate the effectiveness of the OVA-CNN. Based on our experimental results, we conclude that applying the OVA technique on CNN can improve the model's accuracy while reducing the number of parameters and training sample requirements. The findings in this work are subject to at least two limitations. First, the current study is a rough approach that has only examined on MSTAR dataset. Second, the study does not evaluate the effects of the OVA technique on the computational complexity of the model, and cannot guarantee the quality of recognition close to optical image-based SAR ATR.

References

1. Novak, L. M., G. J. Owirka, W. S. Brower, A. L. Weaver. The Automatic Target Recognition System in SAIP. – Lincoln Laboratory Journal, Vol. **10**, 1983, No 2.
2. Owirka, G. J., A. L. Weaver, L. M. Novak. Performance of a Multiresolution Classifier Using Enhanced-Resolution SAR Data. – Radar Sensor Technology II, Vol. **3066**, 1997, pp. 90-100.
3. Zhang, K., W. Zuo, L. Zhang. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. – IEEE Transactions on Image Processing, Vol. **27**, 2018, No 9, pp. 4608-4622.
4. Mu, N., X. Xu, X. Zhang, H. Zhang. Salient Object Detection Using a Covariance-Based CNN Model in Low-Contrast Images. – Neural Computing and Applications, Vol. **29**, 2018, No 8, pp. 181-192.
5. Xie, F., Q. Gao, C. Jin, F. Zhao. Hyperspectral Image Classification Based on Superpixel Pooling Convolutional Neural Network with Transfer Learning. – Remote Sensing, Vol. **13**, 2021, No 5, pp. 930.
6. Deng, J., W. Dong, R. Socher, L. J. Li, K. Li, L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255.
7. Morgan, D. A. E. Deep Convolutional Neural Networks for ATR from SAR Imagery. – In: Algorithms for Synthetic Aperture Radar Imagery XXII. Vol. **9475**. 2015, p. 94750F.
8. Gao, F., T. Huang, J. Sun, J. Wang, A. Hussain, E. Yang. A New Algorithm for SAR Image Target Recognition Based on an Improved Deep Convolutional Neural Network. – Cognitive Computation, Vol. **11**, 2019, No 6, pp. 809-824.
9. Tian, Z., L. Wang, R. Zhan, J. Hu, J. Zhang. Classification via Weighted Kernel CNN: Application to SAR Target Recognition. – International Journal of Remote Sensing, Vol. **39**, 2018, No 23, pp. 9249-9268.
10. Zhang, J., M. Xing, Y. Xie. FEC: A Feature Fusion Framework for SAR Target Recognition Based on Electromagnetic Scattering Features and Deep CNN Features. – IEEE Transactions on Geoscience and Remote Sensing, Vol. **59**, 2020, No 3, pp. 2174-2187.
11. Guo, Y., Z. Pan, M. Wang, J. Wang, W. Yang. Learning Capsules for SAR Target Recognition. – IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. **13**, 2020, pp. 4663-4673.
12. Krizhevsky, A., I. Sutskever, G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. – Advances in Neural Information Processing Systems, Vol. **25**, 2017, pp. 1097-1105.

13. S i m o n y a n, K., A. Z i s s e r m a n. Very Deep Convolutional Networks for Large-Scale Image Recognition. – arXiv Preprint arXiv:1409.1556, 2014.
14. S z e g e d y, C., W. L i u, Y. J i a, P. S e r m a n e t, S. R e e d, D. A n g u e l o v, D. E r h a n, V. V a n h o u c k e, A. R a b i n o v i c h. Going Deeper with Convolutions. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.
15. H e, K., X. Z h a n g, S. R e n, J. S u n. Deep Residual Learning for Image Recognition. – In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
16. C h e n g, Y., D. W a n g, P. Z h o u, T. Z h a n g. A Survey of Model Compression and Acceleration for Deep Neural Networks. – arXiv Preprint arXiv:1710.09282, 2017.
17. D i n g, X., X. Z h a n g, J. H a n, G. D i n g. RepMLP: Re-Parameterizing Convolutions into Fully-Connected Layers for Image Recognition. – arXiv Preprint arXiv:2105.01883, 2021.
18. D i n g, X., X. Z h a n g, N. M a, J. H a n, G. D i n g, J. S u n. RepVGG: Making VGG-style Convnets Great Again. – In: Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733-13742.
19. C h e n, S., H. W a n g, F. X u, Y. Q. J i n. Target Classification Using the Deep Convolutional Networks for SAR Images. – IEEE Transactions on Geoscience and Remote Sensing, Vol. **54**, 2016, No 8, pp. 4806-4817.
20. W a g n e r, S. A. SAR ATR by a Combination of Convolutional Neural Network and Support Vector Machines. – IEEE Transactions on Aerospace and Electronic Systems, Vol. **52**, 2016, No 6, pp. 2861-2872.
21. Z h o n g, C., X. M u, X. H e, J. W a n g, M. Z h u. SAR Target Image Classification Based on Transfer Learning and Model Compression. – IEEE Geoscience and Remote Sensing Letters, Vol. **16**, 2021, No 3, pp. 412-416.
22. L i u, Y., F. Z h a n g, F. M a, Q. Y i n, Y. Z h o u. Incremental Multitask SAR Target Recognition with Dominant Neuron Preservation. – In: Proc. of IEEE International Geoscience And Remote Sensing Symposium (IGARSS'20), 2020, pp. 754-757.
23. C h e n, S., R. Z h a n, W. W a n g, J. Z h a n g. Learning Slimming SAR Ship Object Detector through Network Pruning and Knowledge Distillation. – IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2020, No 14, pp. 1267-1282.
24. C h e n, H., F. Z h a n g, B. T a n g, Q. Y i n, X. S u n. Slim and Efficient Neural Network Design for Resource-Constrained SAR Target Recognition. – Remote Sensing, Vol. **10**, 2018, No 10, pp. 1618.
25. Y u, M., G. D o n g, H. F a n, G. K u a n g. SAR Target Recognition via Local Sparse Representation of Multi-Manifold Regularized Low-Rank Approximation. – Remote Sensing, Vol. **10**, 2018, No 2, pp. 211.
26. M i n, R., H. L a n, Z. C a o, Z. C u i. A Gradually Distilled CNN for SAR Target Recognition. – IEEE Access, Vol. **7**, 2019, pp. 42190-42200.
27. Z h a n g, F., Y. L i u, Y. Z h o u, Q. Y i n, H. C. L i. A Lossless Lightweight CNN Design for SAR Target Recognition. – Remote Sensing Letters, Vol. **11**, 2020, No 5, pp. 485- 494.
28. P e n g, L., M. L i u, X. L i u, L. D o n g, M. H u i, Y. Z h a o. SAR Image Classification Based on CNN in Real and Simulation Datasets. – In: Proc. of 9th International Conference on Graphic and Image Processing, 2018, p. 106152V.
29. D i n g, J., B. C h e n, H. L i u, M. H u a n g. Convolutional Neural Network with Data Augmentation for SAR Target Recognition. – IEEE Geoscience and Remote Sensing Letters, Vol. **13**, 2016, No 3, pp. 364-368.
30. P e i, J., Y. H u a n g, W. H u o, Y. Z h a n g, J. Y a n g, T. S. Y e o. SAR Automatic Target Recognition Based on Multiview Deep Learning Framework. – IEEE Transactions on Geoscience and Remote Sensing, Vol. **56**, 2017, No 4, pp. 2196-2210.
31. Y a n, Y. Convolutional Neural Networks Based on Augmented Training Samples for Synthetic Aperture Radar Target Recognition. – Journal of Electronic Imaging, Vol. **27**, 2018, No 2, pp. 023024.
32. W a n g, K., G. Z h a n g, H. L e u n g. SAR Target Recognition Based on Cross-Domain and Cross-Task Transfer Learning. – IEEE Access, Vol. **7**, 2019, pp. 153391-153399.

33. M a l m g r e n-H a n s e n, D., A. K u s k, J. D a l l, A. A. N i e l s e n, R. E n g h o l m, H. S k r i v e r. Improving SAR Automatic Target Recognition Models with Transfer Learning from Simulated Data. – IEEE Geoscience and Remote Sensing Letters, Vol. **14**, 2017, No 9, pp. 1484-1488.
34. W a n g, Z., L. D u, J. M a o, B. L i u, D. Y a n g. SAR Target Detection Based on SSD with Data Augmentation and Transfer Learning. – IEEE Geoscience and Remote Sensing Letters, Vol. **16**, 2018, No 1, pp. 150-154.
35. H u a n g, Z., C. O. D u m i t r u, Z. P a n, B. L e i, M. D a t c u. Classification of Large-Scale High-Resolution SAR Images with Deep Transfer Learning. – IEEE Geoscience and Remote Sensing Letters, Vol. **18**, 2020, No 1, pp. 107-111.
36. H u a n g, Z., Z. P a n, B. L e i. What, Where, and How to Transfer in SAR Target Recognition Based on Deep CNNs. – IEEE Transactions on Geoscience and Remote Sensing, Vol. **58**, 2019, No 4, pp. 2324-2336.
37. L e e, S., S. P u r u s h w a l k a m, M. C o g s w e l l, D. C r a n d a l l, D. B a t r a. Why M Heads Are Better Than One: Training a Diverse Ensemble of Deep Networks. – arXiv Preprint arXiv:1511.06314, 2015.
38. H u a n g, X., Q. Y a n g, H. Q i a o. Lightweight Two-Stream Convolutional Neural Network for SAR Target Recognition. – IEEE Geoscience and Remote Sensing Letters, Vol. **18**, 2020, No 4, pp. 667-671.
39. H a f i z, A. M., M. H a s s a b a l l a h. Digit Image Recognition Using an Ensemble of One-Versus-All Deep Network Classifiers. – arXiv Preprint arXiv:2007.01192, 2020.
40. P o l a t, K., K. O. K o c. Detection of Skin Diseases from Dermoscopy Image Using the Combination of Convolutional Neural Network and One-Versus-All. – Journal of Artificial Intelligence and Systems, Vol. **2**, 2020, No 1, pp. 80-97.
41. L e C u n, Y., L. B o t t o u, Y. B e n g i o, P. H a f f n e r. Gradient-Based Learning Applied to Document Recognition. – Proceedings of the IEEE, Vol. **86**, 1998, No 1, pp. 2278-2324.
42. N a i r, V., G. E. H i n t o n. Rectified Linear Units Improve Restricted Boltzmann Machines. – In: Proc. of 27th International Conference on Machine Learning (ICML'10), Haifa, Israel, 2010.
43. I o f f e, S., C. S z e g e d y. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. – In: Proc. of International Conference on Machine Learning, 2015, pp. 448-456.
44. K i n g m a, D. P., J. B a. Adam: A Method for Stochastic Optimization. – arXiv preprint arXiv:1412.6980, 2014.
45. V a p n i k, V. N., A. Y. C h e r v o n e n k i s. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. – In: Measures of Complexity. Springer Cham, 2015, pp. 11-30.
46. Z h a n g, K., W. Z u o, L. Z h a n g. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. – IEEE Transactions on Image Processing, Vol. **27**, 2018, No 9, pp. 4608-4622.
47. R o s s, T. D., S. W. W o r r e l l, V. J. V e l t e n, J. C. M o s s i n g, M. L. B r y a n t. Standard SAR ATR Evaluation Experiments Using the MSTAR Public Release Data Set. – In: Algorithms for Synthetic Aperture Radar Imagery V. Vol. **3370**. 1998, pp. 566-573.
48. F e n g, Z., H. J i, L. S t a n k o v i c, J. F a n, M. Z h u. SC-SM CAM: An Efficient Visual Interpretation of CNN for SAR Images Target Recognition. – Remote Sensing, Vol. **13**, 2021, No 20, pp. 4139.
49. Z h a n g, A., X. Y a n g, S. F a n g, J. A i. Region Level SAR Image Classification Using Deep Features and Spatial Constraints. – ISPRS Journal of Photogrammetry and Remote Sensing, Vol. **163**, 2020, pp. 36-48.

Received: 13.12.2021; Second Version: 01.05.2022; Third Version: 20.06.2022; Accepted: 02.08.2022