Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 09, No. 04, December 2023.

FUSION OF DEEP LEARNING ARCHITECTURES FOR ENHANCED TARGET RECOGNITION ON SAR IMAGES

K. Cheikh[1], R. Aitahcene[1], A. Toumi[2] and Z. Hammoudi[1]

(Received: 26-Aug.-2023, Revised: 24-Oct.-2023, Accepted: 2-Nov.-2023)

ABSTRACT

In various applications of radar imagery, one of the fundamental problems is mainly linked to the analysis and interpretation of the images provided, in particular the recognition of moving and/or fixed targets. This task has become more difficult due to the large volume of radar data. This led to the use of automatic-processing and target-recognition methods. The aim of this study is to explore data fusion in SAR (Synthetic Aperture Radar) image classifiers. To this end, we propose a new approach to combine three CNN (Convolutional Neural Network) architectures with several fusion rules. First, we perform a training process of three deep-learning architectures; namely, the basic CNN, the Xception and the AlexNet architectures. Then, two fusion techniques are proposed. The first one deals with the majority rule and the second uses a neural network to combine the decision outputs obtained from three elementary classifiers to achieve the final decision. To evaluate and validate the proposed approach, the MSTAR (Moving and Stationary Target Acquisition and Recognition) dataset is used. The obtained performances of the fusion techniques improve the recognition rate with a final accuracy of 99.59% for the majority rule and 99.51% for the neural network-based rule, which surpasses the accuracy of each individual CNN.

KEYWORDS

Automatic target recognition, Synthetic aperture radar images, Deep learning, Decision fusion.

1. INTRODUCTION

One of the fundamental problems in radar imaging is mainly related to the analysis and interpretation of images acquired in various applications, including recognition of moving and fixed targets. This task becomes more difficult to achieve when automatic processing methods and decision-making are concerned in regard to the large volume of radar data and speckle measurements [1]-[6]. Regarding its global coverage as well as its weather independence, SAR (Synthetic Aperture Radar) imagery has great potentials for military and/or civilian surveillance. In order to leverage these potentials, ML (Machine Learning) can be used to automatically process large amounts of data for different goals, such as ATR (Automatic Target Recognition). For instance, target recognition in radar images presents an essential task for monitoring and surveillance of sensitive areas, such as military or/and civilian zones.

Several methods for target recognition and classification from radar images have been developed in the literature. Due to the absence of efficient feature presentation, interpretation and understanding of radar images, classification of radar images is a significant challenging task. An adequate feature-extraction approach, which can abstract spatial information from radar images and improve classification accuracy, is required. Feature extraction and target recognition in images have been of great interest for many years. Many methods have been proposed by many researchers [1], [7]. These include classical classification methods, such as Bayesian methods, SVM (Support Vector Machine) [8], AdaBoost (Adaptive Boosting) [9], decision trees, WSC (Weighted Sparse Classification), …etc. [10]-[11]. Although, these methods perform well in some situations, their performance degrades significantly in other situations and conditions. This has led researchers to adopt other more sophisticated approaches, such as ANNs (Artificial Neural Networks), which have a very efficient learning capability in a multitude of system-modeling problems [1]. In recent years, we have seen the emergence of DL (Deep Learning) methods with several variants. These latter have given very satisfactory results in several application areas, such as cybersecurity, text analysis, visual and image recognition [12]-[16] and many more. Several research studies have explored the use of RNNs (Recurrent Neural Networks) in the context of SAR image classification, with particular emphasis on

1. K. CHEIKH (Corresponding Autor), R. Aitahcene and Z. Hammoudi are with SISCOM Laboratory, Department of Electronics, University of Frères Mentouri, Constantine 1, Algeria. Email: khaireddine.cheikh@umc.edu.dz
2. A.Toumi is with Lab STICC UMR CNRS 6285, ENSTA Bretagne, Brest, France. Email: abdelmalek.toumi@ensta-bretagne.fr

LSTM (Long Short-Term Memory) networks. LSTMs are specifically designed to handle sequences of data or time series, a feature particularly relevant to the sequential nature inherent in SAR data [17]. To further enhance SAR classification performance, other researchers have delved into hybrid architectures that combine both CNNs (Convolutional Neural Networks) and RNNs (Recurrent Neural Networks). These hybrid approaches showed promising results in SAR image classification by enabling the simultaneous capture of spatial features and temporal dependencies [18]-[19]. With the constant evolution of deep-learning network architectures and their maturity, many researchers have turned to the use of pre-trained DNNs (Deep Neural Networks) in the context of SAR image classification. This approach explores how features pre-learned by these networks can be judiciously employed to significantly increase the accuracy of SAR image classification [20].

Other researchers have attempted to expand the scope of the training dataset by implementing various operations on the images. These operations include rotations, translations, changes in scale and the introduction of noise [21]-[23]. Furthermore, several super-resolution approaches based on DCNNs (Deep Convolutional Neural Networks) have been widely adopted to enhance image resolution. This increase in resolution has directly contributed to an improvement in classification accuracy [24]. Additionally, there are studies that have employed data-augmentation techniques utilizing GANs (Generative Adversarial Networks) to generate synthetic images [25]. Furthermore, in last years, advanced approaches have emerged, combining a DNN in order to improve and achieve more robust and accurate classification results. Furthermore, in recent years, advanced approaches have emerged, utilizing data fusion to significantly enhance SAR image classification. Among these methods, data fusion from multiple sensors has demonstrated a clear improvement compared to using data from a single sensor [26]. Another promising approach lies in the use of model ensembles for SAR image classification, which explores how the aggregation of multiple models can substantially increase classification accuracy. Commonly adopted methods include ensemble learning, where multiple classifiers are trained. Their predictions are combined using various techniques, including voting, weighting and stacking methods [27]-[28].

In this context, the main objective of this research work is to explore new combined architectures of DL to accomplish the classification task, using two methods. The first method relies on fusing the output data from each classifier to make a final decision using the majority fusion rule. The second approach involves the use of ANNs (Artificial Neural Networks) to build a data fusion model, thus exploiting the advantages of each DLN (DL Network) architecture used in this study. Both methods will be evaluated and validated on the MSTAR (Moving and Stationary Target Acquisition and Recognition) dataset [29].

The contribution of this work deals with the study of two fusion techniques. The first one is based on the majority rule and the second one relies on the NN-based rule, in order to improve the recognition performance.

The remainder of this paper is organized as follows. In Section 2, we introduce the CNNs and their variants. In section 3, we present the proposed method. In Section 4, we detail the training, describe and discuss some experimental results performed on the MSTAR dataset. In Section 5, we conclude our work and give some future perspectives.

2. CONVOLUTIONAL NEURAL NETWORKS

CNNs are to date the most efficient models for classifying images and particularly radar images. An input image is provided in the form of a matrix of pixels. It has a two-dimensional array for SAR image for each channel/layer. For a multi-layer image, such as color images (3 layers for Red, Green and Blue) or multi-spectral images, the input image is provided as a multi-dimensional arrays.

The first part of a CNN is the convolutive step, which operates as an image-feature extractor. The image is passed through a series of filters or convolution kernels, creating new images called convolution maps. Some intermediate filters reduce the size of the input data by a maximum pooling operation. Finally, the last convolution maps are laid flat and concatenated into a feature vector, called the CNN code. This CNN code is the input of a layer called the FC (Fully Connected) layer which is a multi-layer perceptron. Its role is the combination of the characteristics of the CNN code to classify the image. The output is the last layer, called SoftMax layer, which uses a SoftMax function as an

activation function [5]. In this work, we consider three CNN architectures. The first is the basic CNN, the second is the AlexNet and the third is the Xception.

2.1 Basic CNN

The proposed basic CNN architecture is illustrated in Figure 1. It is formed by a successive independent layer of the convolution layer and subsampling layer. Theses layers are followed by an FC layer.

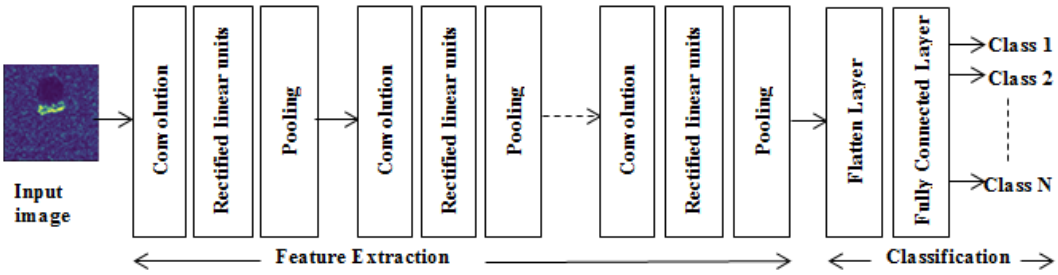


Figure 1. Architecture of a basic CNN model.

To optimize the training process, transfer learning is adopted from the classification of the ImageNet as a source domain. Transfer learning is commonly used in learning applications in order to optimize the learning process and improve the performance of recognition tasks. For the common public architectures, a pre-trained network is used as a starting learning point for a destination task. Thus, the number of classes at the Soft-max layer of the pre-trained CNN network is adjusted.

2.2 AlexNet

AlexNet is a pivotal pre-trained CNN model introduced by Alex Krizhevsky in 2012 [30]. It played a crucial role in advancing computer vision by learning hierarchical representations from 224x224 pixel RGB images. This deep architecture includes convolutional and max-pooling layers for feature extraction, starting from basic patterns and progressing to higher-level abstractions. Extracted features are flattened and processed through three FC layers to facilitate classification; with the final layer having units corresponding to dataset classes and using Softmax activation for probability outputs. In this study, we adapt the structure of the pre-trained network AlexNet, so that it can classify SAR images composed of 10 classes. We therefore act on the FC layer to keep only 10 neurons for the output layers.

2.3 Xception

Xception, derived from "Extreme Inception" is a DCNN (Deep CNN) architecture inspired by inception. It excels at capturing features of different sizes by isolating the learning of spatial and channel-wise features [31]. Instead of combining these types of learning in a single convolution, Xception uses depthwise separable convolutions. It starts with spatial convolutions on each channel and then integrates information across channels using 1x1 convolutions. This design optimizes efficiency while maintaining depth, enabling Xception to efficiently capture both local and global features. It ends with FC layers for feature relationships and uses Softmax activation for multi-class classification, making it highly efficient and accurate for computer vision tasks like image classification. That is why in our study we adapt the output layer to align with the number of classes in the MSTAR dataset.

3. DATA FUSION

To obtain information that is as reliable and precise as possible in all observation conditions, by taking advantage of the complementarity and redundancy of elementary information, as shown in Figure 2, data fusion is used to improve the results. For the data-fusion rule of the different classifiers, the simplest and most popular approach is the majority-vote rule [26]. We will also explore the possibility of applying NNs for the fusion of data from different classifiers. It should be noted that, unlike the

majority rule which aggregates the different elementary decisions to reach the final decision, NNs are instead fed by the distinct outputs of the fully connected layer from each classifier. These outputs represent the results of the Softmax function, effectively providing us with the probabilities for each class. This approach is employed to construct a capable fusion rule, leveraging the capabilities of NNs to model any rule through learning.

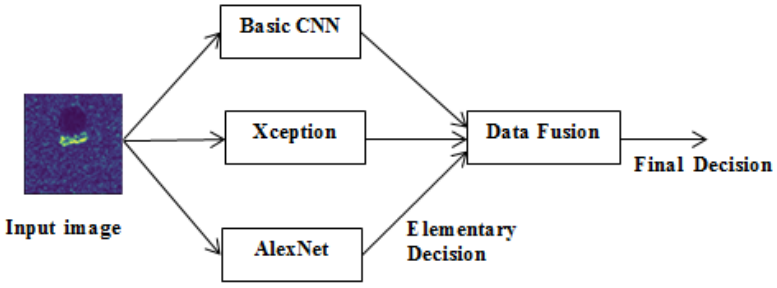


Figure 2. Classification by fusion rule.

3.1 Majority Vote

The majority vote consists in choosing the decision taken by the maximum number of methods. If there are m monitoring methods $\{S_1, S_2, \dots S_m\}$ such that the method $S_{j \in \{1 \dots m\}}$ assigns the decision D_i to the observation x, noted $S_j(x) = i$, where' i' is the class number and associates an indicator function M_j to each method, such as:

$$M_j^i(x) = \begin{cases} 1 & if \quad S_j(x) = i \\ 0 & else \end{cases} \tag{1}$$

The data fusion from 'm' classifiers is carried out according to the following expression:

$$M_K(x) = \sum_{j=1}^{m} M_K^j \tag{2}$$

The majority-vote rule consists, therefore, in choosing the maximum of M_K. The correct class is the one that is most often chosen by the majority of the methods.

3.2 Rule Based on NNs

Exploring the possibilities offered by NNs for constructing classification rules through learning is a central challenge in the field of image classification. In this study, our focus lies in the classification of SAR images using three distinct approaches. The first involves a basic CNN classifier, while the other two leverage pre-trained architectures; namely, the Xception and AlexNet, renowned for their ability to capture high-level features. A critical step in this methodology involves the use of the Softmax output from each classifier to feed an MLP (Multi-layer Perceptron) NN. This approach skillfully combines the strengths of each classifier while minimizing potential errors. The Softmax function takes as input a vector of scores *S*, where each element S_i represents the degree of belongingness of an example to a particular class. The output of the Softmax function, denoted as *P*, is also a vector of the same dimension as *S*, but each element P_i represents the probability of the example belonging to class *i*.
The mathematical formula for the Softmax function applied to a score vector *S* is as follows:

$$P_i = \frac{e^{S_i}}{\sum_{j=1}^{N} e^{S_j}} \tag{3}$$

where, P_i is the probability that the example belongs to class *i*, S_i the score associated with class *i* and M the total number of classes.

In our approach, we have chosen to utilize the Softmax outputs generated by the classifiers (Basic, AlexNet, Xception) in place of class labels. This decision was made to provide the MLP network with richer and more informative data. Consequently, in our approach, our training dataset comprises input-output vectors. The input vector is composed of the Softmax outputs from each classifier:

$$X = [P_{1,1}\, P_{1,2} \dots . P_{1,10}\, P_{2,1} P_{2,2} \dots . P_{2,10}\, P_{3,1} P_{3,2 \dots \dots} P_{3,10}] \tag{4}$$

where, $P_{i,j}$ is the output of the Softmax function of classifier i for class j, with i ranging from 1 to 3 and j from 1 to 10.

The output vector Y represents the actual outputs of the classes in the MSTAR training dataset, which represents a vector of 10 numerical values, with one value equal to 1, corresponding to the true class, while the others are equal to 0.

One of the learning algorithms for the MLP network is the BP (Backpropagation) algorithm, which is one of the supervised learning algorithms. The principle of this algorithm is based on modifying synaptic weights by propagating the error from the output layer back to the input layer through the intermediate layers.

Therefore, if we consider an MLP network with three layers, the BP algorithm can be listed as follows:

Step 1 - Initialization

We start by initializing the weights for the input layer to the hidden layer in matrix W_1 and the weights from the hidden layer to the output layer in matrix W_2. We also initialize the biases in the vectors b_1 and b_2.

Step 2 - Forward Propagation

We calculate the output of the neurons in the hidden layer using the sigmoid activation function:

$$Z_1 = X.W_1 + b_1 \tag{5}$$

$$A_1 = \frac{1}{1+e^{-Z1}} \tag{6}$$

We calculate the output of the neurons in the output layer using a linear activation function:

$$Z_2 = A_1.W_2 + b_2 \tag{7}$$

$$A_2 = Z_2 \tag{8}$$

Step 3 - Prediction

The output A_2 contains predictions for the 10 classes.

Step 4 - Error Calculation

We calculate the error by comparing predictions A_2 with the true values Y, using the MSE (Mean Squared Error), as follows:

$$Erreur = MSE(A2, Y) \tag{9}$$

Step 5 - Backpropagation

We use backpropagation to calculate the gradients of the error with respect to weights W_1 and W_2, as well as biases b_1 and b_2.

$$\nabla W_1 = \frac{\partial Erreur}{\partial W_1} \tag{10}$$

$$\nabla W_2 = \frac{\partial Erreur}{\partial W_2} \tag{11}$$

$$\nabla b_1 = \frac{\partial Erreur}{\partial b_1} \tag{12}$$

$$\nabla b_2 = \frac{\partial Erreur}{\partial b_2} \tag{13}$$

Step 6 - Weight Update

We use the calculated gradients to update the network's weights

$$W_1 = W_1 - \eta \nabla W_1 \tag{14}$$

$$W_2 = W_2 - \eta \nabla W_2 \tag{15}$$

$$b_1 = b_1 - \eta \nabla b_1 \tag{16}$$

$$b_2 = b_2 - \eta \nabla b_2 \tag{17}$$

Each time, we present a new input vector to the network with its associated output and repeat the calculation process from Step 2. Once we have presented all the examples in the training dataset, we calculate the sum of all the errors.

$$E = \sum_{K=1}^{M} Erreur_K \quad (18)$$

where, M is the total number of examples in the training dataset.

If we reach the desired error, the learning process ends. Otherwise, we repeat the algorithm from Step 2 with all the examples in the training dataset.

To make the learning algorithm faster, a term called Momentum, which takes into account the weight changes between two successive iterations, is added to Equations 14-17.

Thus, our goal is to improve the overall performance of SAR image classification, by judiciously combining the results of different classifiers, using the power of an MLP network. Subsequently, this MLP network will be trained on the MSTAR training dataset. The architecture chosen will be the one that gives the best performance in terms of correct-classification rates, obtained through meticulous tuning of hyperparameters to achieve an optimal configuration.

4. RESULTS AND DISCUSSION

In this work, we classified the SAR images from the MSTAR database, according to 10 class categories. In other words, we investigated the different architectures of DL for the classification of SAR satellite images. That is, we started with a simple architecture (Basic CNN) in which we looked for the optimal one. Next, based on the concept of transfer learning, we examined the pre-training architectures to use them in this work. Subsequently, we integrated the outcomes from the three CNNs through a dual approach. The initial method involves the use of the majority fusion rule, while the second method employs the capabilities of NNs to learn and implement the fusion rule *via* supervised learning. To begin, we introduce the software and hardware tools employed in our study. Subsequently, we detail the dataset utilized in this paper. Lastly, we assess the performance of the individual architectures, along with their combined counterpart, using the confusion matrix as a key reference. Specifically, we will focus on the accurate classification rate

4.1 Software and Hardware Tools

To conduct our experiments and effectively categorize all SAR images, we employed the Matlab software due to its user-friendly programming environment and extensive repository of image processing and ML resources. This encompasses artificial NNs and DL capabilities. For the DL framework, we utilized the DL toolbox. It is worth noting that the version of CUDA used in our experiments is 11.2. This combination allowed us to efficiently implement and train our CNN models, including the Basic CNN, earning AlexNet and Xception architectures.

Regarding the MLP network, we employed the NN toolbox to design and implement it. This approach provided the necessary tools for our network's development and evaluation; contributing to the overall success of our methodology.

Regarding our hardware setup, the configuration employed for training and assessing CNN models consisted of an Intel® Core i7 microprocessor running at 3.5GHz, coupled with 64GB of RAM. Additionally, a NVIDIA RTX 2080 TI GPU has been integrated into the system.

4.2 Database of SAR Images

To implement the proposed classification methodology utilizing various DL techniques, the MSTAR database of SAR images was employed. Widely utilized in the literature due to its public availability, the MSTAR dataset encompasses a collection of SAR images with a resolution of 0.3 m × 0.3 m, captured using an X-band spotlight SAR sensor [29].

Figure 3 illustrates the MSTAR database, housing SAR images of military vehicles and organized into 10 distinct class categories. Within the dataset, a total of 5165 images exist, with 2740 images allocated for training and 2425 images designated for the test dataset. The distribution of images across each target (class) can be found in Table 1. These images depict ten distinct classes of ground targets, encompassing entities, like tanks (T62 and T72), rocket launchers (2S1), trucks (ZIL131), armored personnel carriers (BTR70, BTR60, BRDM2 and BMP2), air-defense units (ZSU23/4) and bulldozers (D7). Notably, the images were captured under diverse conditions, spanning different

aspect angles, depression angles and serial numbers. To evaluate the effectiveness of our approach, the MSTAR base, captured under SOC (Standard Operating Conditions), was considered [29].

Table 1. Training and testing MSTAR dataset.

Targets	Train	Test
2s1	299	274
BMP2	233	195
BRDM2	298	274
BTR60	256	195
BTR70	233	196
D7	299	274
T62	299	273
T72	232	196
ZILI131	299	274
ZSU234	299	274

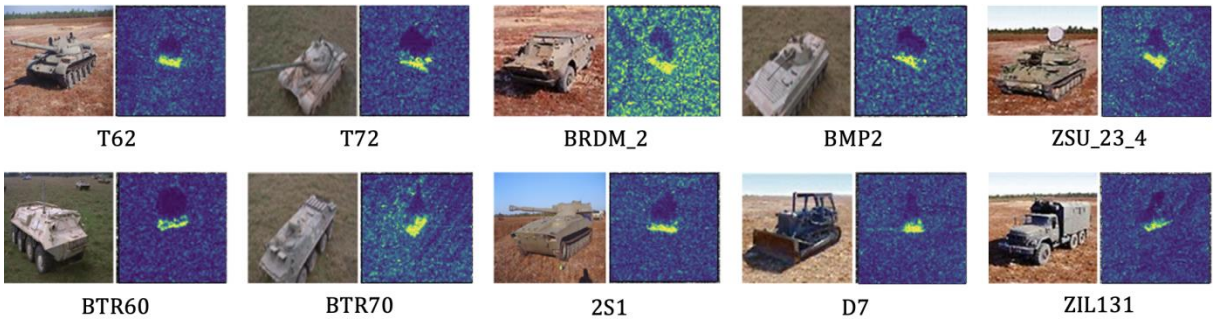


Figure 3. Military vehicles from MSTAR database and their corresponding optical images.

4.3 Simulation Results for Basic CNN

Based on the basic-CNN architecture, the proposed approach determines the number of convolution layers, the number of filters, the size of the convolution filters as well as the subsampling step. As this is an optimization problem the inherent CNN hyper-parameters of which are difficult to get, we remedy it by giving initial values for all CNN hyper-parameters. Therefore, the training stage is to change a single hyper-parameter until the optimal value yields a better classification. The process is repeated for the other hyper-parameters until all optimal hyper-parameters are obtained.

Here, we used 158×158 input images and proposed a basic CNN architecture with initial hyper-parameters as given in Table 2.

Table 2. Initial hyper-parameter values.

Hyper-parameters	
Number of convolutional layers	4
Size of filters (stride=2)	3x3
Number of filters	5
Size of max-pooling (stride=2)	5x5
Size of epoch	5
Initial learning rate	10^{-1}
Percentage of the learning/validation database	0.5

Note that these hyper-parameters were obtained separately from each other. Therefore, this allowed us to locate the range of hyper-parameters to be investigated. After successive tests, we found the optimal hyper-parameters, shown in Table 3. These latter, which allowed us to achieve the best performances, are discussed in the next sub-section.

Table 3. Optimal hyper-parameters.

Optimal hyper-parameters	
convolutional layers Number	3
Size of filters (stride=2)	7x7
Filters number	16
Size of max-pooling (stride=2)	2x2
Epoch number	10^3
Initial learning rate	10^{-2}
Percentage of the learning/validation database	0.9

Table 4 shows the performance evaluation of the basic CNN in terms of correct classification rate on the basis of SAR imaging test (MSTAR). The results were obtained through the training of the basic CNN, with the optimal hyper-parameters, achieving an overall correct-classification rate of 97.3%. It should also be noted that for the ZSU234 military machine, the correct-classification rate reached 100%.

Table 4. Confusion matrix of the proposed basic CNN model.

Classes	**2S1**	**BMP 2**	**BRDM2**	**BTR60**	**BTR70**	**D7**	**T62**	**T72**	**ZILI131**	**ZSU234**	**RR%**
2S1	263	0	0	0	0	0	0	0	1	10	95.99
BMP 2	9	167	2	6	3	0	0	7	1	0	85.64
BRDM2	0	0	269	0	0	0	0	4	1	0	98.18
BTR60	6	1	5	181	2	0	0	0	0	0	92.82
BTR70	0	0	1	2	193	0	0	0	0	0	98.47
D7	0	0	2	0	0	271	0	0	1	0	98.91
T62	2	0	0	0	0	0	269	0	2	0	98.53
T72	1	1	1	1	0	0	0	192	0	0	97.96
ZILI131	1	0	0	0	0	0	0	0	273	0	99.64
ZSU234	0	0	0	0	0	0	0	0	0	274	100.00
Average Accuracy = 96.61											

In order to determine the architecture that yields a good classification of the SAR images, the next sub-section is devoted to the investigation of the two other pre-trained deep-learning architectures.

4.4 Simulation Results for the AlexNet and Xception Networks

AlexNet and Xception are transfer-learning networks that are based on pre-trained models. The model that we used has already been previously presented.

To classify new SAR images, we recycle a pre-trained network by replacing the three final layers with new layers adapted to the new dataset. Here, we need to change the number of classes to 10 and retain the parameters for the other layers.

Finally, a training operation is launched, so that the new network adapts with the new learning data. For a better classification, this is accomplished through the adjustment of the network parameters.

Table 5. Optimal hyper-parameters for AlexNet.

Optimal hyper-parameters	
MiniBatchSize	18
ValFrequency	10
Maximum epoch	70
Initial learning rate	1e-2
Percentage of the learning/validation database	0.6

Table 6. Optimal hyper-parameters for Xception.

Optimal hyper-parameters	
MiniBatchSize	36
ValFrequency	32
Maximum epoch	30
Initial learning rate	1e-4
Percentage of the learning/validation database	0.9

It should be noted that the optimal hyper-parameters of the two pre-trained networks were found by successive tests. Tables 5 and 6 show the optimal hyper-parameters that were obtained via their

adjustment, element by element.

Simulation results on the testing MSTAR dataset are shown in Tables 7 and 8.

Table 7. Confusion matrix of AlexNet.

Classes	2S1	BMP 2	BRDM2	BTR60	BTR70	D7	T62	T72	ZILI131	ZSU234	RR%
2S1	215	0	4	2	0	0	50	0	0	3	78.47
BMP 2	3	175	2	2	0	0	0	13	0	0	89.74
BRDM2	0	0	274	0	0	0	0	0	0	0	100.00
BTR60	0	0	0	191	0	0	0	4	0	0	97.95
BTR70	0	1	0	4	191	0	0	0	0	0	97.45
D7	1	0	0	0	0	266	2	0	0	5	97.08
T62	1	0	0	0	0	0	251	0	0	21	91.94
T72	0	0	1	1	0	0	0	194	0	0	98.98
ZILI131	0	0	0	0	0	0	0	0	274	0	100.00
ZSU234	0	0	1	0	0	0	0	0	0	273	99.64
Average Accuracy=95.12											

Table 8. Confusion matrix of Xception.

Classes	2S1	BMP 2	BRDM2	BTR60	BTR70	D7	T62	T72	ZILI131	ZSU234	RR%
2S1	269	0	1	0	0	0	1	0	0	3	98.18
BMP 2	0	178	3	2	1	0	0	11	0	0	91.28
BRDM2	0	0	273	1	0	0	0	0	0	0	99.64
BTR60	0	0	0	193	2	0	0	0	0	0	98.97
BTR70	0	0	0	0	196	0	0	0	0	0	100.00
D7	3	0	0	0	0	266	0	0	1	4	97.08
T62	1	0	0	0	0	0	271	0	0	1	99.27
T72	0	0	0	1	0	0	0	195	0	0	99.49
ZILI131	0	0	0	0	0	0	0	0	274	0	100.00
ZSU234	0	0	0	0	0	0	0	0	0	274	100.00
Averarage Accuracy=98.39											

Finally, based on Tables 4, 7 and 8, we achieved better performance with the Xception classifier, achieving a correct-classification rate of 98.39%. This was followed by the Basic classifier, which achieved a rate of 96.61% and lastly came the AlexNet classifier, which achieved a classification rate of approximately 95.12%.

After having trained several DL architectures for the classification of SAR images, it is important to consider the possibility of exploring the theory of data fusion for the classification of SAR images. To achieve this, we use a parallel architecture, in which we apply the fusion rule via the majority vote as well as the use of NNs for the construction of a new data-fusion rule.

4.5 Simulation Results with Decision Fusion Techniques

In this sub-section, we use a parallel architecture consisting of three classifiers; namely, the basic CNN, the two pre-trained networks, AlexNet and Xception, which are passed to the fusion center. In the following part, we will evaluate the performance of the two data fusion approaches; namely, the majority rule and the rule based on NNs. The two techniques will be evaluated *via* the MSTAR testing dataset.

4.5.1 Majority Voting Rule

In this technique, the basic decisions from each classifier are transmitted to the fusion center for a final determination using a majority voting rule for data fusion. It is worth noting that in instances where the three classifiers yield differing classes, the rule will yield a random decision. The performance of this architecture was appraised using the MSTAR test database. The resulting confusion matrix is detailed in Table 9, showing an average global accuracy rate of 99.53%; indicating a gain of 1.14%.

4.5.2 Neural Network

The second technique which we used is data fusion using ANNs (Artificial NNs), exploiting the ability of NNs to model any rule by the supervised learning. Therefore, we investigated several architectures of NNs; the MLP architecture has been retained. In our investigation, we used the Levenberg-Marquardt learning algorithm to train the network. Levenberg-Marquardt algorithm is a widely used

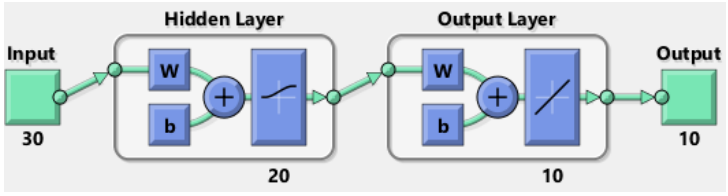
optimization algorithm for training NNs. It adjusts the network's weights to minimize the error between the predictions and the actual values.

Table 9. Confusion matrix of majority-voting rule.

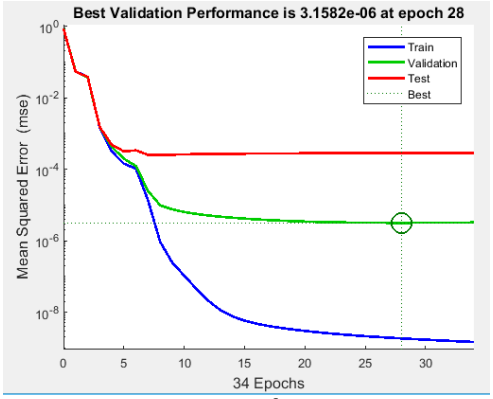
Classes	2S1	BMP 2	BRDM2	BTR60	BTR70	D7	T62	T72	ZILI131	ZSU234	RR%
2S1	272	0	0	0	0	0	0	0	0	2	99.27
BMP 2	0	189	1	1	0	0	0	4	0	0	96.92
BRDM2	0	0	274	0	0	0	0	0	0	0	100.00
BTR60	0	0	0	195	0	0	0	0	0	0	100.00
BTR70	0	0	0	0	196	0	0	0	0	0	100.00
D7	0	0	0	0	0	274	0	0	0	0	100.00
T62	0	0	0	0	0	0	272	0	0	1	99.63
T72	0	0	0	1	0	0	0	195	0	0	99.49
ZILI131	0	0	0	0	0	0	0	0	274	0	100.00
ZSU234	0	0	0	0	0	0	0	0	0	274	100.00
Average Accuracy=99.53											

The configuration of this network was iteratively fine-tuned through multiple experiments, involving the minimization of a cost function and adjustments of hyper-parameters, such as the number of hidden layers, the number of neurons in the hidden layers, the activation functions, the error metrics and the Momentum parameter, using the MSAR training dataset. This iterative process continued until achieving optimality, as demonstrated in Figures 4 and 5, which illustrate the optimal structure of the MLP network and the loss function.

Upon completing the training phase, simulation results on the MSAR test dataset yielded an average overall accuracy rate of 99.44%, as depicted in the confusion matrix of Table 10, indicating a significant improvement.



Figue 4. NN-based data-fusion center.



Figue 5. Loss function.

Table 10. Confusion matrix of NN-based data-fusion center.

Classes	2S1	BMP 2	BRDM2	BTR60	BTR70	D7	T62	T72	ZILI131	ZSU234	RR%
2S1	272	0	0	0	0	0	0	0	0	2	99.27
BMP 2	0	191	0	1	0	0	0	3	0	0	97.95
BRDM2	0	0	274	0	0	0	0	0	0	0	100.00
BTR60	2	0	0	193	0	0	0	0	0	0	98.97
BTR70	0	0	0	1	195	0	0	0	0	0	99.49
D7	1	0	0	0	0	273	0	0	0	0	99.64
T62	0	0	0	0	0	0	272	0	0	1	99.63
T72	0	0	0	1	0	0	0	195	0	0	99.49
ZILI131	0	0	0	0	0	0	0	0	274	0	100.00
ZSU234	0	0	0	0	0	0	0	0	0	274	100.00
Average Accuracy=99.44											

4.6 Execution-time Analysis for the Proposed Methods

We have taken into consideration the execution time of the two proposed methods, recognizing the crucial importance of speed in radar detection and SAR image-classification applications, which often operate in real time. Firstly, it is important to note that the three used DL architectures require significant time during the training phase. However, once this phase is completed "in offline" mode, we measured the execution time of the three architecture; namely, the basic CNN, the AlexNet and the

Xception. The execution times that we obtained for classifying a single image are 0.908 ms, 2.76 ms and 11.48 ms, respectively.

It is essential to note that these execution times are calculated for classifying a single image. In a parallel implementation, we consider the longest time among these three times, which is that of the Xception architecture, as the next step can only occur when the three outputs from the three classifiers are ready.

Regarding the execution time for data fusion, we observed that using the majority rule requires approximately 15.8 µs, while an MLP network requires about 20.86 µs. It is worth noting that these times are practically comparable, with a slight advantage for the majority rule.

In summary, while the three used DL architectures require a significant amount of time during the training phase, once this phase is completed, our fusion methods present execution times that are compatible with the requirements of real-time radar detection and SAR image-classification applications; thus demonstrating their practicality and efficiency.

4.7 Performance Comparison of Different Methods

In order to facilitate the comparison of our findings with other DL methodologies assessed on the MSATR test dataset, we present a summary of methods and their corresponding average classification rates in Table 11. Particularly, our approach, employing a parallel architecture with a fusion center, achieves the highest average accuracy in correct classification. This outcome underscores how our strategy of data fusion effectively leverages the unique strengths of individual classifiers while mitigating classification errors. Note that our majority voting-based fusion rule exhibits a superior performance. Additionally, it is worth highlighting that the data-fusion technique centered on ANNs also delivers highly commendable results.

Table 11. Performance comparison of different methods

METHOD	AVERAGE ACCURACY (%)
FCNN [33]	98.10
ENHANCED-SHAPE CNN [34]	99.29
MFFD-CNN [35]	99.30
FEF-NET [36]	99.31
CNN & SVM [37]	99.50
PROPOSED METHOD	99.53

5. CONCLUSION

The primary goal of this study was to employ a combined architecture consisting of three classifiers and a data-fusion center for the purpose of classifying SAR satellite images. To achieve this, a basic CNN architecture was utilized. This architecture underwent optimization by fine-tuning the network parameters, including hyper-parameters, until reaching the optimal configuration. In addition to this, two other pre-trained CNN networks were employed. For these networks, the final three layers were adapted to facilitate SAR image classification. A similar optimization process was applied to determine the best architecture. To merge the outputs of the distinct classifiers, a fusion center was implemented. Two techniques were employed for this fusion; namely, the majority-voting rule and the utilization of NNs to construct a learning decision rule. Through simulations conducted on SAR images, with a focus on the MSTAR database, the effectiveness of data fusion in SAR image classification was demonstrated. Notably, the majority fusion rule technique exhibited an exceptional performance, achieving an accurate-classification rate of 99.53%. As future works and in order to validate the performance of the proposed approach, extensive experiments on other datasets should be implemented with other fusion techniques.

REFERENCES

[1] L. M. Novak, "State-of-the-art of SAR Automatic Target Recognition," Proc. of Record of the IEEE 2000 Int. Radar Conf., pp. 836-843, Alexandria, VA, USA, 2000.

[2] A. Karine, A. Toumi, A. Khenchaf and M. E. Hassouni, "Saliency Attention and Aift Key-points Combination for Automatic Target Recognition on MSTAR Dataset," Proc. of the Int. Conf. on

Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-5, Fez, Morocco, 2017.

[3] D. R. Hush and B. G. Horne, "Progress in Supervised Neural Networks," IEEE Signal Processing Magazine, vol. 10, no. 1, pp. 8-39, 1993.

[4] T. D. Ross et al., "Standard SAR ATR Evaluation Experiments Using the MSTAR Public Release Data Set," Proc. SPIE, Algorithms for Synthetic Aperture Radar Imagery V, vol. 3370, pp.566–573, 1998.

[5] A. Toumi and A. Khenchaf, "Aircrafts Recognition Using Convolutional Neurons Network," Proc. of the Int. Conf. on Radar Systems (RADAR'17), pp.23-26, UK, 2017.

[6] A. Masoomi, R. Hamzehyan and N.C. Shirazi, "Speckle Reduction Approach for SAR Image in Satellite Communication," Int. Journal of Machine Learning and Computing, vol. 2, no. 1, 2012

[7] A. Toumi, A. Alhousseini and A. Khenchef, "Aircraft Recognition Using Convolutional Neurons Network," Proc. of the IEEE 7th Seminar on Detection Systems: Architectures and Technologies (DAT 2017), Algeria, 20-22 February 2017.

[8] Q. Zhao and J. C. Principe, "Support Vector Machines for SAR Automatic Target Recognition," IEEE Transactions on Aerospace and Electronic Systems, vol. 37, pp. 643–654, 2001.

[9] Y. Sun, Z. Liu, S. Todorovic and J. Li, "Adaptive Boosting for SAR Automatic Target Recognition," IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no.1, pp. 112–125, 2007.

[10] A. Karine, A. Toumi, A. Khenchaf and M. Hassouni, "Multivariate Copula Statistical Model and Weighted Sparse Classification for Radar Image Target Recognition," Computers & Electrical Engineering, vol. 84, p. 106633, 2020.

[11] J. C. Cexus, A. Toumi and M. Riahi, "Target Recognition from ISAR Image Using Polar Mapping and Shape Matrix," Proc. of the 2020 IEEE 5th Int. Conf. on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1-6, Sousse, Tunisia, September 2020.

[12] C. Coman and R. Thaens, "A Deep Learning SAR Target Classification Experiment on MSTAR Dataset," Proc. of the 19th IEEE Int. Radar Symposium (IRS 2018), Bonn, Germany, 20-22 June 2018.

[13] S. A. Wagner, "SAR ATR by a Combination of Convolutional Neural Network and Support Vector Machines," IEEE Trans. on Aerospace and Electronic Systems, vol. 52, no. 6, pp. 2861–2872, 2016.

[14] A. David and E. Morgan "Deep Convolutional Neural Networks for ATR from SAR Imagery," Proc. of SPIE, vol. 9475, DOI: 10.1117/12.2176558, 2015.

[15] A. Ameta, V. Singh1 and V. Devi, "A Convolutional Neural Network Based Approach for SAR Image Classification of Vehicles," Int. Journal of Engineering Research & Technology (IJERT), vol. 9, no. 06, pp. 544-547, 2020.

[16] A. T. Al-Taani and I. T. Al-Dagamesh, " Automatic Detection of Pneumonia Using Concatenated Convolutional Neural Network", Jordanian Journal of Computers and Information Technology (JJICIT), vol. 09, no. 02, pp. 118-136, 2023.

[17] R. Hang, Q. Liu, D. Hong and P. Ghamisi, "Cascaded Recurrent Neural Networks for Hyper-spectral Image Classification," IEEE Trans. Geoscience and Remote Sensing, vol. 57, pp. 5384–5394, 2019.

[18] C. Wang, X. Liu, J. Pei, Y. Huang, Y. Zhang and J. Yang, "Multi-view Attention CNN-LSTM Network for SAR Automatic Target Recognition," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 12504-12513, 2021.

[19] X. Fan, L. Chen, X. Xu, C. Yan, J. Fan and X. Li, "Land Cover Classification of Remote Sensing Images Based on Hierarchical Convolutional Recurrent Neural Network," Forests, vol. 14, no. 9, p. 1881, 2023.

[20] Z. Huang, Z. Pan and B. Lei, "Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data," Remote Sensing, vol. 9, no. 9, p. 907, 2017.

[21] Y. Zhai, H. Ma et al., "SAR ATR with Full-angle Data Augmentation and Feature Polymerization," IET, vol. 19, pp. 6226-6230, 2019

[22] Z. Geng, Y. Xu, B.-N. Wang, X. Yu, D.-Y. Zhu and G. Zhang, "Target Recognition in SAR Images by Deep Learning with Training Data Augmentation" Sensors, vol. 23, no. 2, p. 941, 2023.

[23] J. Ding, B. Chen, H. Liu and M. Huang, "Convolutional Neural Network with Data Augmentation for SAR Target Recognition," IEEE Geoscience and Remote Sensing Letters, vol. 13, no.3, pp. 364–368, 2016.

[24] L. Lin, J. Li, Q. Yuan and H. Shen, "Polarimetric SAR Image Super-resolution *via* Deep Convolutional Neural Network", Proc. of the IEEE Int. Geoscience and Remote Sensing Symposium (IGARSS 2019), DOI: 10.1109/IGARSS.2019.8898160, Yokohama, Japan, 2019.

[25] Z. Cui, M. Zhang, Z. Cao and C. Cao, "Image Data Augmentation for SAR Sensor *via* Generative Adversarial Nets," IEEE Access, vol. 7, pp. 42255 – 42268, 2019.

[26] G. Joshi, R. Natsuaki and A. Hirose "Multi-sensor Satellite-imaging Data Fusion for Earthquake Damage Assessment and the Significant Features," Proc. of the 7th Asia-Pacific Conf. on Synthetic Aperture Radar (APSAR), DOI: 10.1109/APSAR52370.2021.9688438, 2021.

[27] Y. Khenchaf and A. Toumi, "Siamese Neural Network for Automatic Target Recognition Using Synthetic Aperture Radar," Proc. of the Int. Geoscience and Remote Sensing Symposium (IGARSS), Pasadena, USA, Jully 2023.

[28] C. Zhang, Y. Wang, H. Liu, Y. Sun and L. Hu, " SAR Target Recognition Using Only Simulated Data for Training by Hierarchically Combining CNN and Image Similarity," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022.
[29] Master Public Targets, The Sensor Data Management System, [Online], Available: https://www.sdms.afrl.af.mil/index.php?collection=mstar&page=targets.
[30] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), pp. 1-9, 2012.
[31] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1251-1258, 2017.
[32] G. Niu, S. S. Lee, B. S. Yang and S. J. Lee, "Decision Fusion System for Fault Diagnosis of Elevator Traction Machine," Journal of Mechanical Science and Technology, vol. 22, no.1, pp. 85-95, 2008.
[33] L. Yu et al., "SAR ATR Based on FCNN and ICAE," Journal of Radars, vol. 7, no.5, pp. 622-631, 2018.
[34] M. Huang, F. Liu and X. Meng, "Few Samples of SAR Automatic Target Recognition Based on Enhanced-shape CNN," Journal of Mathematics, vol. 2021, pp.1-16, 2021.
[35] L. Guo, "SAR Image Classification Based on Multi-feature Fusion Decision Convolutional Neural Network," IET Image Process, vol. 16, pp.1–10, 2022.
[36] J. Pei, Z. Wang et al., "FEF-Net: A Deep Learning Approach to Multi-view SAR Image Target Recognition, " Remote Sensing, vol. 13, no.17, p. 3493, 2021.
[37] S. A. Wagner, "SAR ATR by a Combination of Convolutional Neural Network and Support Vector Machines," IEEE Transactions on Aerospace and Electronic Systems, vol. 52, no. 6, pp. 2861-2872, 2016.

ملخص البحث:

في تطبيقاتٍ متعدّدة للتّصوير بالرّادار، ترتبط إحدى المشكلات الأساسية بتحليل الصُّور وتفسيرِها، وخصوصاً عند تمييز الأهداف المتحركة والثابتة. وقد ازداد هذا الأمر صعوبةً بسبب الكمّ الهائل من البيانات، وأدّى ذلك الى استخدام طرقٍ أوتوماتيكية لمعالجة البيانات وتمييز الأهداف.

يهدف هذا البحث الى استقصاء دمْج البيانات في مصنّفات صُور الرّادار، وفيه نقترح طريقةً جديدة لدمْج ثلاثٍ من بِنى الشّبكات العصبية الالتفافية، وهي الشّبكة العصبية الالتفافية الأساسية، وبِنية (Xception)، وبِنية (AlexNet). ويتمّ ذلك باستخدام طريقتين من طُرق الدّمْج، تعتمد أولاهما على قاعدة انتخاب الأغلبية، بينما تستخدم الثانية شبكةً عصبية لدمج القرارات من ثلاثة مصنِّفات أولية للوصول الى القرار النّهائي.

ولتقييم الطّريقة المقترحة، تمّ تجريبها على مجموعة بيانات (MSTAR) الخاصّة بكشف وتمييز أهداف الرّادار المتحركة والثابتة. وقد ادّى استخدام تقنية الدّمج المقترحة في هذا البحث الى تحسين معدّل التّمييز بدّقة نهائية بلغت 99.59% لطريقة الدّمج الخاصّة بقاعدة انتخاب الأغلبية و 99.51% لطريقة الدّمج باستخدام الشّبكة العصبية، حيث تجاوزت هذه المعدّلات تلك التي تمّ الحصول عليها باستخدام شبكة عصبية التفافية مفرَدة.