Target classification using convolutional deep learning and auto-encoder models

Sarra Zaied, Abdelmalek Toumi, Ali Khenchaf

Target Classification Using Convolutional Deep Learning and Auto-Encoder Models

Sarra Zaied, Abdelmalek Toumi and Ali Khenchaf
Lab-STICC UMR CNRS 6285, ENSTA Bretagne,
2 rue François Verny, 29806 Brest Cedex 9, France.
Email: {toumiab, khenchal}@ensta-bretagne.fr

Abstract*—Targets recognition in radar images presents an essential task for monitoring and surveillance of sensitive areas such as military zones. The fundamental problem in radar imaging is related to the recognition of objects in radar images, that its needs a whole chain of treatment. To classify radar images a feature extraction method is used to detect an appropriate subspace in the original feature space, which is based on transformation of the original feature. This subspace should be big enough to maintain minimal loss of information and small enough to minimize the complexity of classifier. Since the feature extractor is difficult to build in manual mode and needs to be redesigned for each application, a Deep Learning in automatic mode is used with a training process subdivided into several modules. In this paper, we lay out an approach to classify Synthetic aperture radar (SAR) and Inverse Synthetic aperture radar (ISAR) images using Deep learning techniques. At first, in order to evaluate the effect of convolution layers and the number of hidden layers of the perceptron we thought of implementing 4 configurations of CNN (Convolutional neural network). In the second time, we use the CAE (Convolutional auto-encoder) to learn the optimal filters that minimize the reconstruction error, after we use these filters to feed the CNN retained and evaluate the effect on performance's system.

Keywords*—Target recognition, Deep learning, ISAR images, SAR images, Convolutional neural network, Convolutional auto-encoder.

I. INTRODUCTION

Synthetic Aperture Radar produces two-dimensional (2D) images where one dimension in the image is a range and the other dimension is an azimuth (or cross range). The SAR remote sensing presents numerous advantages over optical remote sensing, due to its independence on atmospheric and sunlight condition [1]. It has been successfully applied in many fields such as military, and oceanology. Similar to the SAR, Inverse Synthetic Aperture Radars (ISAR) also produce a two-dimensional image (range and cross range, and intensity). However, the ISARs synthesize a large aperture antenna by using the motion of the target. Thus, the notation "inverse synthetic aperture" is generally applied to movement of the target which permits derivation of information about the shape and size of the target.

The classification of radar images is a significant challenging task yet [18]. Due to the presence of speckle noise and the absence of effective feature presentation, the interpretation and the understanding of radar images are always much different from optical images [2]. An adequate feature extraction approach, which can abstract spatial information from radar images as well as improve classification accuracy, is required. Feature extraction of these images has been researched for many years, and a lot of methods have been proposed by many researchers. Recently, inspired from artificial intelligence, Deep Learning methods [6,11,19,20] are used to classify SAR and ISAR images.

In our framework, ISAR and SAR databases are used, the first is composed of aircrafts images and the second is a benchmark of SAR images representing military vehicles. At first we evaluated the impact of the convolutions layers and the hidden layers of the perceptron on system performances, by the implementation of 4 architectures of CNN. Then we proposed the use of a Convolutional Auto-Encoder (CAE) to generate optimal filters, which will be wormed in the convolution layers.

The remainder of this paper is organized as follows. Section II gives an introduction of different deep learning techniques. In section III, details of training are described and some experimental results on the MSTAR and Aircraft datasets are presented. Finally a conclusion will be given in the section IV.

II. DEEP LEARNING METHODS

A. Convolutional Neural Network (CNN)

For pattern recognition and image classification tasks, variants of CNNs have emerged as a robust supervised feature learning and classification tools, especially when combined with pooling operations [5].

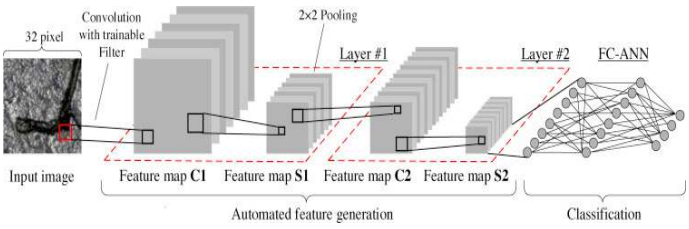

Fig. 1. CNN architecture.

As shown in Fig 1, the CNNs are multi-layered NNs which is composed of several alternations of convolution and pooling layers, followed by fully connected layers on the top. They are specialized on recognition visual patterns directly from image pixels.

1) Convolutional layer

This layer is parameterized by the number of feature maps J and the size of these maps also the kernel sizes $F \times F$. Each layer has many maps of equal size [7]. The kernel $k_{ji}^{(l)}(u,v)$ is shifted over the valid region of the input feature map, a trainable bias $b_i^{(l)}$ is added and a sigmoid function is applied to activate this layer. Thus, in a Convolutional layer, each map $M_j^{(l)}(j = 1 \dots J)$ is connected to all of its preceding feature maps $M_i^{(i-1)}(x,y)$, where $M_i^{(i-1)}(x,y)$ is the unit of the ith input feature map at position (x,y), and $M_j^{(l)}(x,y)$ is the unit of the j^{th} output feature map at position (x,y). The convolution operation is defined by (1) as:

$$V_j^{(l)}(x,y) = \sum_{i=1}^{I} \sum_{u,v=1}^{F-1} k_{ji}^{(l)}(u,v).\, M_i^{(l-1)}(x-u, y-v) + b_j^{(l)} \quad (1)$$

$$M_j^{(l)}(x,y) = f\left(V_j^{(l)}(x,y)\right) \quad (2)$$

Where $f(x)$ is the nonlinear activation function, and $V_j^{(l)}(x,y)$ denotes the weighted sum of the j^{th} inputs to the output feature map at position(x,y).

To improve classification tasks, a nonlinear activation function should be added at each convolution layer. In traditional ConvNets, a sigmoid $f(x) = \frac{1}{(1+e^{-x})}$ or a hyperbolic tangent function$f(x) = \tanh(x)$ is applied to each unit on the output feature maps on the convolutional layer. Recent research has found a non-saturating nonlinearity, i.e., the rectified linear unit (ReLU), which often works better in practice. The ReLU activation function is given by (3):

$$f(x) = max(0,x) \quad (3)$$

2) Pooling layer

The purpose of the pooling layers is to achieve spatial invariance by reducing the resolution of the feature maps and to control the overfitting [3]. Each map in the pooling layer corresponds to one map of the previous layer. Their units are combined from a small patch (local region) of pixels; this patch can be of arbitrary size. There are too different pooling operations the average-pooling which computes the average over the inputs and max pooling which takes the maximum of the neighborhood. The max pooling operation is defined by (4):

$$M_i^{(l)} = \max_q M_i^{(l)}(x.s + u, y.s + v) \quad (4)$$

Where q is the pooling size and s is the stride which determines the intervals between neighbor pooling windows.

3) Classification layer

After stacking convolution and pooling layers, a shallow Multilayer Perceptron (PMC) is used to complete the architecture. The output layer has one neuron per class and a sigmoid is used as a nonlinear activation function. Thus, each neuron's output in between 0 and 1, which makes it possible to consider this output as a probability. Other than the supervised methods, there are the unsupervised ones, which aim to extract generally useful features from unlabeled data, to detect and remove input redundancies, and to preserve essential aspects of the data in robust and discriminative representations.

Unsupervised methods have been used in many scientific and industrial applications. In the context of neural network architectures, un-supervised layers can be stacked on top of each other to build deep architectures [8]. These methods permits unsupervised initializations which tend to avoid local minima, which increase the network's performance stability [9].

II. CONVOLUTIONAL AUTO-ENCODER (AE)

The Convolutional Auto-encoder is an unsupervised method for hierarchical feature extraction; it forms a convolutional neural network (CNN). The CAE discovers good CNNs initializations that avoid the numerous distinct local minima of highly non-convex functions arising in virtually all deep problems. While the CAE's representation makes learning even harder than for standard auto-encoders [12], good filters emerge if we use pooling layer, an elegant way of enforcing sparse codes without any regularization parameters to be set by trial and error [13].

Fully connected AEs ignore the 2D image structure, which can cause problems like introducing redundancy in the parameters, with forcing each feature to be global. The CAE differs from conventional AEs as their weights are shared among all locations in the input, preserving spatial locality. The reconstruction is hence due to a linear combination of basic image patches [14].

For a mono-channel inputs x the latent code of the j^{th} feature map is given by (5):

$$h^j = \sigma(x * w^j + b^j) \quad (5)$$

Where the bias b^j is broadcasted to the whole map, σ is the activation function, and * denotes the 2D convolution. A single bias per maps is used in [14] because one bias per pixel would introduce too many degrees of freedom.
The reconstruction is obtained using (6)

$$y = \sigma(h^j * \widehat{w^j} + c) \quad (6)$$

Where again c is a bias per input channel and $\widehat{w^j}$ represent the flip operation over both dimensions of the weights. In this CAE the back-propagation is applied to compute the gradient of error function with respect to the parameters. The convolution of a $k \times k$ matrix with a $n \times n$matrix may in fact result in an$(m - n + 1) \times (m - n + 1)$. The cost function to minimize is the mean squared error (MSE) given by (7):

$$E(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(x_i - y_i)^2 \quad (7)$$

III. EXPERIMENTS AND DISCUSSION

In this part, we report and analyze the implementations and classification results obtained on ISAR and SAR images. We are interested in this section both in classical methods and in Deep Learning techniques by combining the auto-encoder and the convolutional auto-encoder with a network of convolutional neurons to enhance system performances. Note that the tests are performed on an Intel processor, CPU 3.10 GHZ with 8 GB of RAM. We introduce in the first of this section the datasets used for different simulations.

A. *Datasets*

For target recognition, we utilize two datasets. The first one is MSTAR database [15] which include ten different categories of ground targets (armed personnel carrier: BMP-2, BRDM-2, BTR-60, and BTR-70; tank: T-62, T-72; rocket launch: 2S1; air defense unit: ZSU-234; truck: ZIL-131; bulldoze: D7) as shown in Fig. 2. They were collected using an X-band SAR sensor, in a 1-ft resolution spotlight mode with a full aspect coverage (in the range 0° to 360°).

The second datasets, SynISAR [16], is composed of 7 classes of aircrafts as illustrated in Fig. 3, which each class contains 181 images. We have reduced the initial size of the images which is 300×300 to 32×32, so that the center of the image obtained is the same as that of the initial image. We note also that for the ISAR database we have mapped the images in 70% (840 images) for the learning base and 30% (427 images) for the test base.

Base d'apprentissage	Nombre d'images	Base de test	Nombre d'images
BMP2	233	BMP2	195
BRDM2	298	BRDM2	274
BTR60	256	BTR60	195
BTR70	233	BTR70	196
D7	299	D7	274
2S1	299	2S1	274
T62	299	T62	273
T72	232	T72	196
ZILI131	299	ZILI131	274
ZSU234	299	ZSU234	274

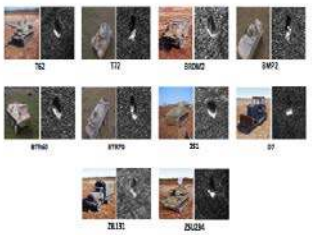

Fig. 2. MSTAR Dataset.

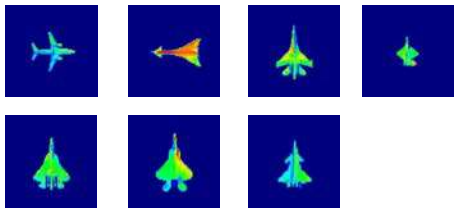

Fig. 3. SynISAR dataset [16].

B. *Standard classification versus deep learning*

In this section, we proposed the use of the multilayer perceptron (MLP) and the K nearest neighbors (KNN) as classifier, after that we implement a CNN which composed of 2 convolutional layers and a perceptron of two layers, to evaluate and analyze the results obtained over these methods. For the KNN, first of all we tried with different number of K, we noticed that we obtain a minimal error when k is high but the time of training here is too long. So we chose a k=9 which we have acquired 95.71 % for ISAR images and 72.53% with the SAR ones. For the PMC, each pixel of the image is connected to an input neuron, so 1024 neurons are used for the input layer (32 × 32 Input size). The number of hidden layer neurons is chosen according to the equation cited in [17]where the number of neurons in this layer must be equal to the square root of the product of the number of neurons in the input and output layer, that's why we have configured the hidden layer with 85 neurons. In the output layer, we have 7 classes which correspond to 7 neurons. For the minimization of the error and the optimization of the network, a learning rate of 0.5 was chosen. After learning and updating the weights and biases, we obtain a recognition rate of 92.97% for the ISAR database. For classification of SAR images (10 classes), we have configured the network with 102 neurons according to [17]. Since there are 1024 neurons in the input layer and 10 neurons in the output layer. The overall recognition rate is 62.24%. With the CNN architecture; we adopted the approach used for the networks of deep neurons, which requires a judicious choice of the number of epochs and the size of the block. These two parameters make it possible to define an optimal number of iterations to arrive at a minimum error rate. The first architecture of deep neural network is the convolutional neural network, there are different models of this network, depending on the size of the data inputs, the configuration of the layers, the number of filters, the size of kernels and the sub-sampling steps. The architecture of this network is given in Fig. 4.

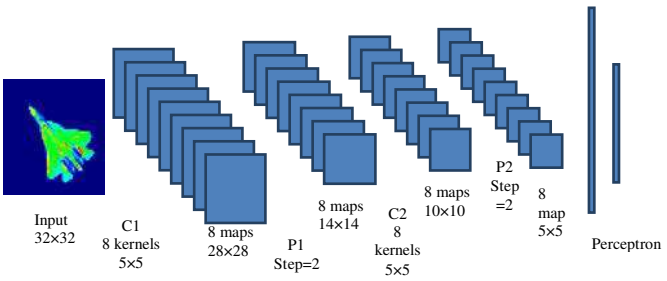

Fig. 4. CNN architecture.

At each convolution layer, a bias is added to the convolved images before passing through an activation function. In this architecture 1 of CNN (Fig. 4), we used a sigmoid function which presents the correction layer. The last 8 images obtained are sized 5×5 and concatenated into a vector, which is the input of the neural network. The input layer contains 200 neurons and the output layer contains neurons as the number of classes (7 neurons for ISAR Dataset and 10 neurons for MSTAR Dataset) with a sigmoid function as activation function. Note that the neural network in this architecture has no hidden layer. Using this CNN, we obtained 96.48% with ISAR images and 72.82% with SAR images. The obtained results on SAR and ISAR datasets are given in Table I.

TABLE I. CLASSIFICATION ACCURACIES

	K-NN	*PMC*	*CNN*
ISAR database	95.71%	92.97%	96.48%
SAR database	72.53%	62.24%	72.82%

The CNN use relatively little pre-processing, unlike the other classical algorithms such as KNN and MLP.. The absence of initial parameterization and human intervention for the extraction of the characteristics is a major asset of the CNN, in addition to these advantages; the recognition rate is better by applying the CNN.

To evaluate the performance of CNN architecture, the convolution layers and the hidden layer on the classification step (see Fig.1) are studied. The influence of these layers are presented and discussed in the section below.

C. Evaluation of convolution layers and hidden layer

In order to evaluate the effect of convolution layers and the number of hidden layers of the perceptron we thought of implementing 3other configurations of CNN. As a proposition of amelioration, we used a CAE to learn the optimal filters that minimize the reconstruction error, after that we used these filters to feed the CNN retained among the 3 configurations. To optimize the weights of the perceptron we integrated an AE. Since everything is optimal in the architecture we proposed the minimization of the number of filters used to reduce the learning time and improve the recognition rate.

- 1st architecture

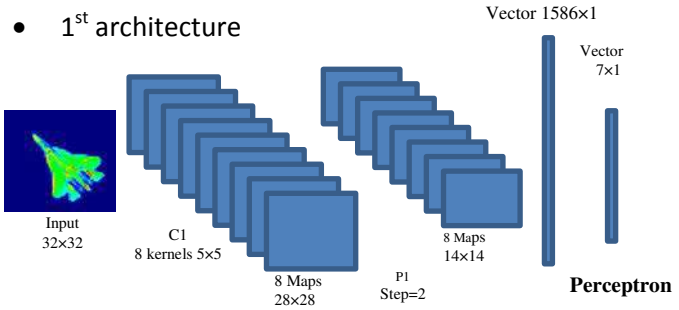

Fig. 5. CNN with one convolutionnel layer and without hidden layer.

For this configuration, 8 kernels were used with one convolutionnel layer. After sub-sampling of the founded images, we obtained 14 × 14 images. The input vector of the perceptron has 1568 neurons with an output layer of 7 neurons. With this configuration, we obtained a recognition rate of 92.27% with an estimated learning time of 21 seconds for ISAR dataset. We also used this architecture for the MSTAR database; the perceptron in this case contains 1586 neurons in the input layer and 10 neurons for the output layer (10 classes). We found 62.27% as the recognition rate, with a learning time of 55 seconds. The obtained recognition results on MSTAR dataset show that this fist architecture fails to recognize the 3 classes (recognition rates are 0% for ZSUS234, BTR70 and T72). In the next architecture, we added a hidden layer to the perceptron.

- 2nd architecture

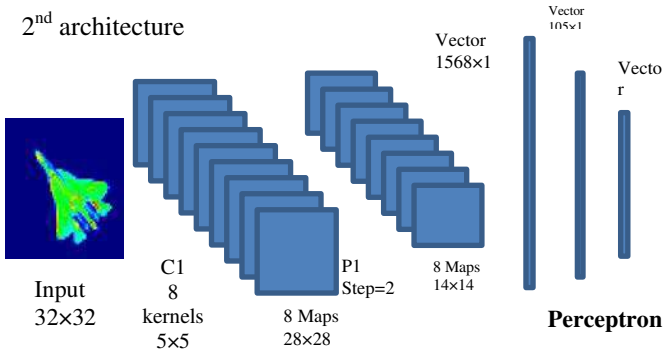

Fig. 6. CNN with one convolutionnel layer and with hidden layer.

In this configuration a hidden layer was added according to rule [17]. For the ISAR dataset, the input layer is composed of 1568 neurons and the output layer consists of 7 neurons so we inserted a hidden layer of 105 neurons. We obtained a recognition rate of 95.31% with a learning time of 27 seconds for all ISAR images of learning dataset. For the MSTAR database, the perceptron is configured with a hidden layer of 125 neurons, we obtained a recognition rate of 70.74%. The learning time for this learning database is 85 seconds.

We can see from these two configurations that by adding a hidden layer the recognition rate increases. It should also be noted that the learning time becomes more important by increasing the number of hidden layers. With this configuration, the adding of a hidden layer improves the system performances. We observe that this CNN can recognize two classes (ZSUS234, BTR70) but the third remains always unknown (T72). The next architecture, we append a second convolution layer to the previous architecture.

- 3rd architecture

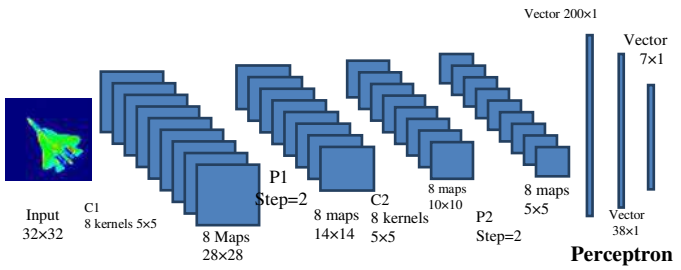

Fig. 7. CNN with two convolutionnel layers and one hidden.

Using this architecture, we can assess the impact of convolution and sub-sampling, by adding the second convolution layer. The 8 images of the last sub-sampling step are of size 5×5. After the concatenation of these images, the feature vector is of size 200.For the connected layers, the input layer is 200 neurons, the output layer is 7 neurons (ISAR data base) and the hidden layer according to [17] is of 38 neurons. For the MSTAR datasets which have 10 classes, so we configured the hidden layer with 45 neurons. We obtain a recognition rate of 97.19% with a learning time of 37 seconds for the ISAR images and recognition rate of 75.98% with 105

seconds on learning SAR images dataset. We have found that adding a convolution layer the recognition rate as well as the learning time increases. In this proposed architecture, all classes are recognized with enhancing of the classification rate but make the learning time more long.

The table II summarizes the results obtained by the different CNNs on the two databases. We noticed that adding a hidden layer to the PMC or adding a convolution layer results improve the recognition rate. On the other hand it serves to increase the learning time.

For the improvement of the recognition architecture by reducing the learning time, we opted for the last configuration which gives us a good classification. We then use a convolutional auto-encoder for the generation of filters.

TABLE II. RECOGNITION ACCURACIES FOR CNN ARCHITECTURE

		CNN			
		One conv. layer		Two conv.layer	
Perceptron		Without hidden layer	One hidden layer	Without hidden layer	One hidden layer
ISAR Dataset	Recog. rate	92.27%	95.31%	96.48%	97.19%
	Learn. *t*	21 s	27 s	30 s	37 s
MSTAR Dataset	Recog. rate	62.27%	70.74%	72.82%	75.98%
	Learn. *t*	55 s	85 s	90 s	105 s

D. CNN using CAE

As previously described, Convolutional Auto-Encoder (CAE) addresses the task of defining filters in a different perspective. Instead of using CNN convolutional random filters, we used the CAE to learn the optimal kernels that minimize the reconstruction error, after we use these filters for the two layers of our CNN. Since we have 2 layers of convolutions, we first start by implementing a 2-layer CAE, each layer gives us the 8 filters intended to be used in the CNN.

For the ISAR database, we obtained 96.72% with a learning time of 40 seconds. Howevern for MSTAR database we obtained a rate accuracy of 82.15% with 110 seconds. For learning step including CAE times learning. Note that the accuracies are roughly equal to that found without CAE with increased learning time. In order to reduce the learning time without loss system's performances, we introduce an AE in order to obtain the optimal weights that will be used for the hidden layer of the perceptron. Then, we opted to reduce the number of filters given by CAE.

E. CNN With CAE and AE.

In this step and in ordrer to optimize the previous archticture (with CAE solution), we integrate an auto-encoder to optimize the weights of the perceptron.

- 4th architecture

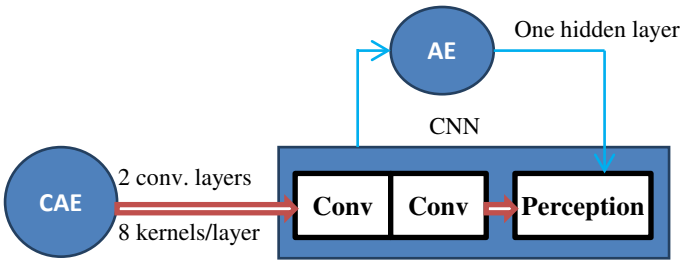

Fig. 8. CNN with CAE and AE architectures.

Using the weights given by the auto-encoder, we obtain 97.65% as a classification rate for a learning time of 47 seconds with the ISAR database, and a rate of 83.27% with a learning time is 117 seconds for the MSTAR database.

Note here that the recognition rate is improved while the learning time has increased. In order to reduce learning time, we sought to reduce the number of filters generated by the CAE. The first simulations are carried out with 8 filters in the first convolution layer and 8 filters in the second (8, 8), we have reduced the number to (2, 2) to bring out the influence of this step on the classification rate and learning time. With two filters in the two convolutions layers, a recognition rate of 98.12% with a learning time of 25 seconds, is obtained, which implies an improvement in the results in terms of rate and calculation time. We found the same improvement for the MSTAR database, a classification rate of 90.09% is achieved with a learning time of 78 seconds. The obtained results using CAE and AE are presented un Table 3.

TABLE III. RECOGNITION ACCURACIES FOR CNN WITH CAE AND AE ARCHITECTURES.

		CNN archi; using CAE (2 convolutions) and AE (one hidden layer)	
Number of kernels (conv1, vonv2)		Conv. Layers (CAE)	Conv. Layers (CAE)
		(8,8)	(2,2)
ISAR Dataset	Recog. rate	97.65%	98.12%
	Learn *t*	47s	25s
MSTAR Dataset	Recog. rate	83.27%	90.09%
	Learn *t*	117s	78s

CONCLUSION

In this work, we studied and applied different classification methods: K-NN, RN, and CNN, which are evaluated in terms of recognition rates and learning time. We evaluated also the influence of the addition of the layers of convolutions and the hidden layers on the performances of the network.

The results presented in the last section shows the effectiveness of deep learning methods for the classification of radar images. By varying the number of filters with the use of various deep learning techniques (CAE and CNN), we achieved a better classification rate with optimal number of filters and one hidden layer.

REFERENCES

[1] A. Moreira, P. Prats-Iraola, M. Yonis, G. Krieger, I. Hajnsek and P. Papathanassiou, "A tutorial on synthetic aperture radar," IEEE Geoscience remote sensing, vol. I, no. 1, pp. 6 - 43, 2013.

[2] Y. Yang, Y. Qui et C. Lu, "Automatic target classification, experiments on the MSTAR SAR images," *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, Towson, MD, USA, 2005, pp. 2-7. 2005.

[3] D. Scherer, A. Muller et S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," 20th Internationnal conference on artificial neural networks, Thessaloniki, Greese, septembre 2010.

[4] J. Geng, H. Wang, F. Fan et X. Ma, "Deep supervised and contractive neural network for SAR image classification," IEEE Transactions on geosience and remote sensing, vol. 55, no. 14, pp. 2442 - 2459, 2017.

[5] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," In Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1237-1242, June 2011.

[6] D. Weimer, B. Scholz-Reiter and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," Elsevier , vol. 65, pp. 417-420, 2016.

[7] J. Nagi, F. Ducatelle, G. DiCaro, D. Cireşan, U. Meier, A. Giusti and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," IEEE International Conference, In Signal and Image Processing Applications, pp. 342-347, November 2011.

[8] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, no.14, pp. 193-202, 1980.

[9] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent et S. Bengio, "Why does unsupervised pre-training help deep learning?," Journal of Machine Learning Research, vol. 11, pp. 625-660, 2010.

[10] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang et A. Madabhushi, "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," IEEE transactions on medical imaging, vol. 35, no. 11, pp. 119-130, 2016.

[11] Y. Zhang, E. Zhang et W. Chen, "Deep neural network for halftone image classification based on sparse auto-encoder," Elsevier, Engineering Applications of Artificial Intelligence, vol. 50, pp. 245-255, 2016.

[12] A. Ng, "Sparse autoencoder," CS294A Lecture notes, 2011.

[13] J. Geng, J. Fan, H. Wang, X. Ma, B. Li et F. Chen, "High-resolution SAR image classification via deep convolutional autoencoders," IEEE Geoscience and Remote Sensing Letters, vol. 12, no.111, pp. 2351-2355, 2015.

[14] J. Masci, U. Meier, D. Cireşan and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," Springer,Artificial Neural Networks and Machine Learning–ICANN, pp. 52-59, 2011.

[15] S. Chen, H. Wang, F. Xu et Y. Q. Jin, "Target classification using the deep convolutional networks for SAR images," IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no.18, pp. 4806-4817, 2016.

[16] V. K. Vatsavayi et H. K. Kondaveeti, "Efficient ISAR image classification using MECSM representation," Journal of King Saud University-Computer and Information Sciences, 2016.

[17] V. Venugopal et W. Baets, "Neural networks and Statistical Techniques in Marketing research: A Conceptual Comparison," Marketing Intelligence & Planning, Marketing intelligence and planning, Vol. 12 Issue. 7, pp.30-38 1994.

[18] A. Toumi and A. Khenchaf, "Target recognition using IFFT and MUSIC ISAR images," in Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on. IEEE, 2016, pp. 596–600.

[19] A. El Housseini, A. Toumi and A. Khenchaf, "Deep Learning for target recognition from SAR images," Seminar on Detection Systems Architectures and Technologies (DAT), Algiers, , pp. 1-5, 2017.

[20] A. Toumi, A. El Housseini, and A. Khenchaf, "Aircrafts recognition using convolutional neurons network," International Conference on Radar Systems. Radar'17.UK. 2017.