Performance study of the smart networks for remote sensing image textures identification

[1] MARRAKCHI CHARFI OLFA, [2] MBAINAIBEYE JEROME
[1] Department of Physic and instrumentation, National High Institute of Applied Sciences and Technology, Centre Urbain Nord, Box 676, 2080. Tunis, Tunisia. [2] National Polytechnic High Institute of Mongo, Chad.
olfa.charfi@gmail.com mbai_jerome@yahoo.com.

Abstract
In this paper the performance evaluation of smart networks to identify highly heterogeneous textures remote sensing images was investigated. These networks are Feed Forward Neural Networks (FFNN), Quantum Neural Network (QNN) and Support Vector Machine (SVM). This evaluation is performed through an optimization training time and number of parameters of smart networks in the constraint to achieve optimal identification rate of the textures. The study also concerns the influence of the nature of heterogeneous textures on the choice of smart networks parameters to obtain elementary unit of textures. The objective is to study the impact of the textural information on the network design and considering that the samples of textures have a textural complexity due to the textural correlation and the overlapping rates of species in these textures. Textures bases used in this study are taken from different remote sensing images sources: an airborne radar image and an ASTER satellite whose resolutions are totally different. We have studied the influence of the spatial resolution on the textures identification and network performance relative to each of the two types of images.

Key-Words: Smart networks, textures, classification, identification, remote sensing images.

1. Introduction

Several studies have shown that smart classifiers are fairly robust tools for the identification of remote sensing image textures [1][2]. However, the development of such networks must take in account the minimization of computation time and the convergence to an optimal solution. All this is related to the complexity of the data in terms of their dimensions and correlations.

Neural networks called Feed Forward Neural Networks (FFNN), are widely used as the classifiers, through their approximation and generalization capabilities [3].

Quantum neural networks (QNN) are also called FFNN fuzzy-based networks [4][5]. The development of neuro-fuzzy systems, through the merger of neuronal modeling with the concepts of fuzzy logic, was realized to treat the incapacity of FFNN networks to properly estimate the belonging class of two data from two regions with a high correlation due the sharp decision to partition the observation space [4]. The QNN networks are the FFNN networks able to classify the uncertain data. The main difference between these two networks is the shape of activation functions related to hidden layers, which are nonlinear.

Separator Large Marge Networks or Support Vector Machine (SVM) are powerful tools recognized for solving the training problems [6]. With their speed and their power, the SVM networks have been exploited in various applications area such as image classification [7], texture classification based on the wavelet coefficients [8]. There are two main methods of SVM training of more than two classes [9]: the first method is called “one against all” (1AA) and the second method is called “one against one” (1A1). However, the 1AA method is best suited for image classification, but it is slower than the 1A1 method.

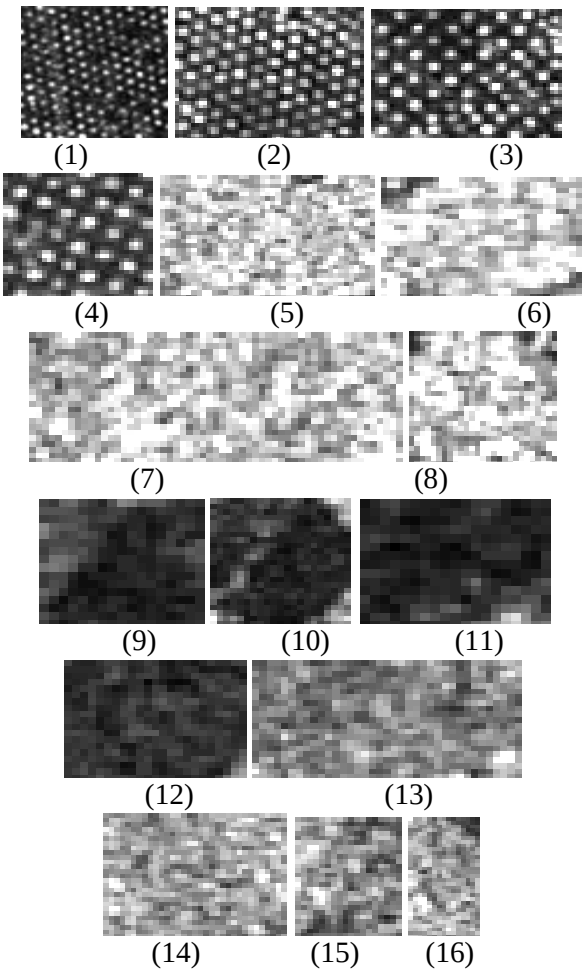
For this study we have considered two remote sensing images with different acquisition modes. An image acquired by the airborne radar called synthetic aperture radar (SAR) and another acquired by the sensor Terra of ASTER satellite. The image areas are in different places of the Tunisian territory (semi-arid and forested area). Our interest is the influence of textural parameters on the neural network implementation, able to identify such textures knowing that the units constituting such image textures with their details, are conditioned by the spatial resolution (most fine is the resolution, more are the details).

In the first time, we perform an analysis of image textures by taking in account the nature of their complexity. In the second time, we highlight the compromise between the development of an optimal architecture for each type of network and textural parameters. In the third time, we evaluate the performance of the networks in the textures identification.

2. Databases

2.1 Texture analysis

Textures bases used are extracted from remote sensing images: an image acquired by a synthetic aperture radar (SAR) airborne C-band and horizontal polarization and an image acquired by the multi-spectral sensor of Terra ASTER satellite : Red (R), Green (V) and Near Infrared (NIR). A thematic study on the image areas has allowed us to identify the different existing textures in these images. Due to the higher resolution of the SAR image (4.6 meters of spatial resolution), this image shows more details than the Aster image for which the spatial resolution is 15 meters (see figure 1 and table 1).



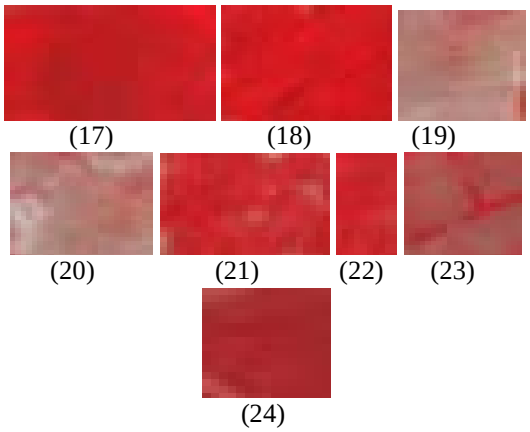


Fig.1 Samples of textures of remote sensing images: from (1) to (16) are those of SAR image and from (17) to (24) are those of ASTER image.

Table 1 Characteristics of texture samples of the two images.

A: Olivier on covered soil by vegetation, B: Build on covered soil by vegetation, **C:** Naked soil with presence of vegetation
D: Vegetation, E: Oak cork with a recovery =5 % - 10**%,** F: Oak cork with a R recovery = 50%-75%, G: Oak cork with a R recovery R > 75%, H: Pine of Alps with a recovery R=100 %

Number	Label	Textures	Size in pixels
1, 2, 3, 4	O.V	**A**	66 x 63, 37 x 47 39 x 49, 31 x 31
5, 6, 7, 8	B.V	**B**	29 x 41, 23 x 31 23 x 57, 27 x 27
9, 10, 11, 12	SN.V	**C**	16 x 17, 23 x 21 15 x 19, 19 x 23
13, 14, 15, 16	V	**D**	21 x 37, 27 x 33 26 x 18, 32 x 15
17, 18	C.L5-10	**E**	25 x 38, 26 x 33
19, 20	C.L50-75	**F**	22 x 20, 25 x 26
21, 22	C.L+75	**G**	20 x 25, 22 x 11
23, 24	P.A100	**H**	25 x 22, 16 x 15

Selected textural windows will be used later, for training purposes of the different smart networks. The textures of the images are irregular and containing common textural elements. The high existing textural correlation can be observed visually from the histogram distributions of texture samples from each of the two images (figures 2 and 3). The dynamic of gray levels is wide in the case of textures of SAR image; this allows more information of these textures. The existence of common textural elements causes the overlapped distributions for the most textures. For example, the textures C.L5-10, C.L50-75 and C.L + 75 of ASTER present the textural element " oak cork " common to these textures but have a variable overlapping percentage from one texture to another. For the textures O.V and SN.V of the SAR

image, they have the common textural element which is the "naked soil ".
We note therefore that the textures of the ASTER image present, for certain textures, a spectral inter-band correlation R and RIP. Indeed, the spectral response of the texture P.A100 in the V band has a distribution similar to that given in the NIR band; so we can deduce that there is inter-band redundancy information.
In this study, the redundancy information is one of the parameters which can influence the design of networks as the data are used in their raw state (digital pixel value) to perform training.

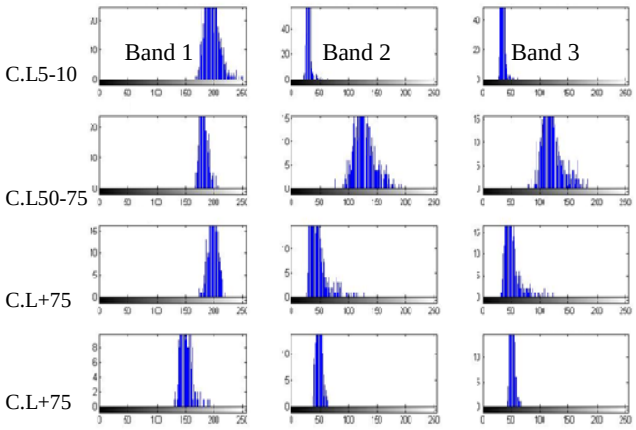


Fig.2 Histograms of the textures for ASTER image

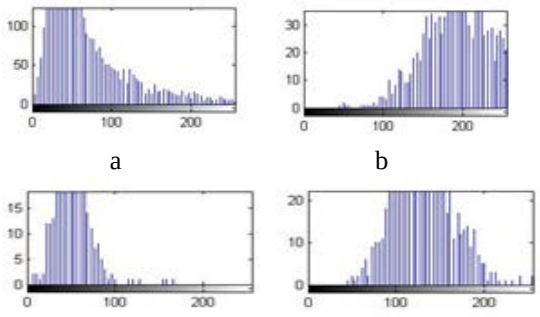


Fig.3 Histograms of the textures for SAR image: a) O.V, b) B.V, c) SN.V, d) V.

2.2. Design of data bases for training and test

The windows of textures of (figure 1 and table 1) extracted from remote sensing images were used as texture samples to achieve training and testing of FFNN, SVM, and QNN networks. The examples are designed from a scan, using a window with size of K_b x K_b where K_b belongs to the following list {3, 5, 7, 9, 11}, realized on these textural windows. For the training of the networks, the examples are randomly selected from all the examples where the number of examples is the same for each of these textures. Indeed, after a random mixture of these examples, we have considered a number equal to 2/3 of the total number of examples used for the training phase and the 1/3 was used for the test phase of the different networks (see figure 4).

Table 2 Number of samples of remote sensing image textures depending on the size of the scanning window

K_b	3	5	7	9	11
SAR	3124	2444	1844	1332	908
ASTER	1532	1184	880	620	400

3. Smart Network Design

3.1 Optimal network design procedure

The first step in the development of a network is to initialize a number of network parameters. Some parameters are adjusted iteratively according to two criteria: obtain the convergence of the Mean Square Error (MSE) given by equation 4 after training (MSE_L) toward a minimum MSE value (EQM_{Lmin}) and ensure that the MSE determined in the test phase (EQM_T) is greater than EQM_L (Fig.4). There may be several architectures responding to the imposed conditions. Choosing the optimal network depends, therefore, on the results of texture identification rate of the image after their identification by different networks. The texture identification rate procedure is defined by equation 1.
Depending on the size of the input vector, we have obtained several possibilities of the networks whose may be used for image texture classification. Indeed, successful networks are those have well responded to the convergence criteria of MSE. To avoid the possibility to have several network architectures, we have imposed the rapidity of convergence by a minimum convergence time Tc during the training phase. So, only are considered the networks for which the convergence time is inferior or equal to Tc.

$$I_d(T) = \sum_{i=1}^{c} \left(p(I_s = i \,/ T_s = T) / p(T_s = T / I_s = i) \right) \times 100 \quad (1)$$

Where c is the number of classes, Is is the pixel label s and $p(s = i / Ts = T)$ is the conditional probability that the pixel s is classified in class i knowing that it is the texture T.

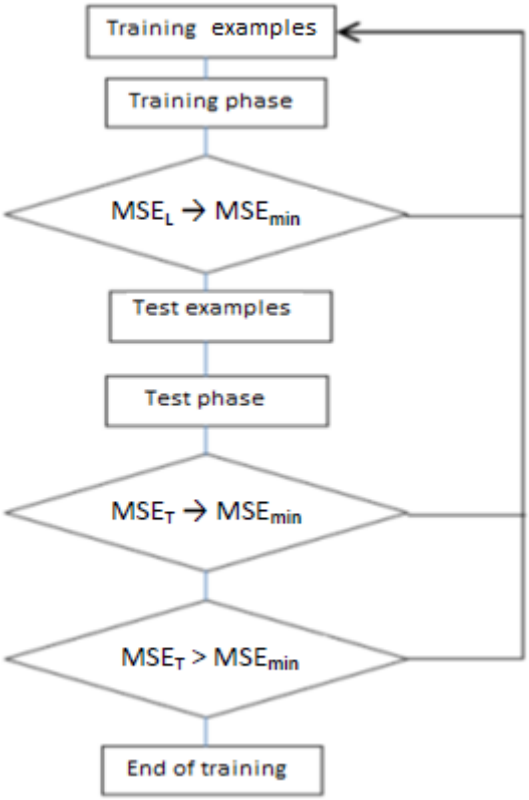


Fig.4 Flowchart of the training and test phases.

3.2. Neural network classification (FFNN)

The architecture of the neural network FFNN that we have used is the multilayer perceptron (figure 5) [10]. The adaptation of the synaptic weights of the network, W_{ijk} is done according to the MSE back-propagation algorithm defined by equation 2 using a descending gradient method defined in equation 3. The activation functions used are the same for all network layers. A suitable function is the sigmoid function *f* which values belong to the interval [-1,1] (see equation 5). The first stage of the network development is to initialize a certain number of parameters: the number of input nodes *N* which is conditioned by the size of the scanning window K_b where K_b x K_b = *N*; the number of neurons of the output layer is equal to the number of classes *Ns*; the number of neurons of the first hidden layer *n*; the number of the second hidden layer *m* and the training step µ. The synaptic weights W_{ijk} and the parameter µ are randomly initialized and are adjusted iteratively and sequentially (after passage of an example or sample) according to the criteria imposed on MSE_L and MSE_T (figure 4). In the case where the MSE_L does not converge to MSE_{min}, after a given number of iterations *iter*, the number of neurons per layer and the value of the training step will be adjusted.

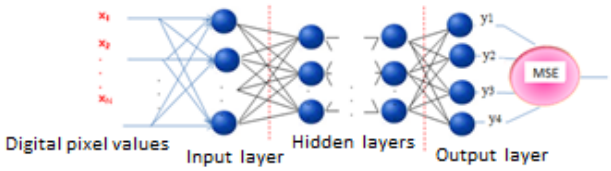


Fig.5 Architecture of a FFNN network

The adaptation algorithm of the synaptic weights by the method of back-propagation of descending gradient is defined by equations 2 and 3:

$$W_{ijk}(p+1) = W_{ijk}(p) + \Delta W_{ijk}(p) \quad (2)$$

Where $W_{ijk}(p+1)$ is the synaptic connection weight between neuron *i* of the layer *k* and the neuron *j* of the layer (*k-1*) at iteration (*p* + *1*) and $\Delta W_{ijk}(p)$ is the variation of the synaptic weight between iteration *p* and (*p* + *1*) defined by equation 3

$$\Delta W_{ijk}(p) = -\frac{1}{2}\mu \frac{\partial E(p)}{\partial W_{ijk}(p)} \quad (3)$$

The MSE at iteration *p*, between the desired output d_s (table 2) and the output of the network y_s^p is defined by the equation 4:

$$E(p) = \sum_{s=1}^{N_s} \left(d_s - y_s^p\right)^2 \quad (4)$$

$$y_s^p = f\left(Y_s^p\right) \quad (5)$$

Where f and Y_s^p are defined respectively by the equations 6 and 7

$$f(Y) = \frac{1-e^{-\beta_0 Y}}{1+e^{-\beta_0 Y}} \quad (6)$$

Where β_0 is the slop of the sigmoid function

$$Y_i^p = \sum_{j=1}^{N_1} W_{ijk}(p) x_j^p \quad (7)$$

3.3. Classification by quantum neural networks (QNN)

The implementation of QNN network requires two steps [15]. Initially, a simple FFNN network is designed, consisting of a single input layer, containing N nodes, which is equal to the size of the scanning window K_b, a hidden layer containing N_h neurons and an output layer containing N_s which is equal to the number of classes (figure 6). By varying the value of the training step µ of the synaptic weights (W_{ij}, V_{ij}) according to descending gradient algorithm (equation 2) and the number of neurons N_h, we have obtained the convergence of the FFNN network to a minimum EQM_L (equation 4) but this convergence is not necessarily optimal. In the second step, we have considered that only the activation functions of neurons in the hidden layer of the same network FFNN are in multilevel configuration (figure 7). The number N_s of activation functions level is considered as the same for all the neurons in this layer. The position of the jump levels Θ^r of the multi-level activation functions are also adjusted according to

the descending gradient algorithm but by the minimization of the objective function G (equation 10). In this case, Θ^r are adjusted with a training step α_θ . Once the parameter α_θ is optimized, the number of thresholds N_s is adjusted to allow the convergence of MSE_L to the lowest possible value.

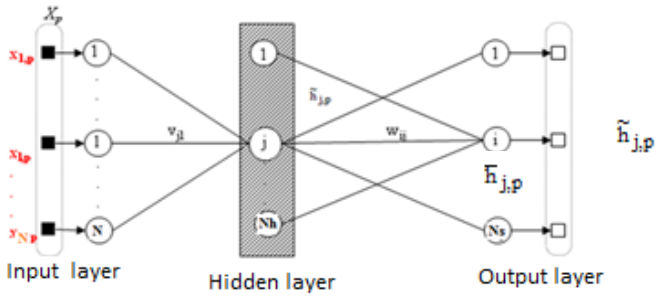


Fig.6 Architecture of a QNN network.

$$\hat{y}_{i,p} = g_0\left(\sum_{j=0}^{N_h} W_{ij}\,\tilde{h}_{j,p}\right) \quad (8)$$

With

$g_0(x) = \frac{1}{1+e^{(-\beta_0 x)}}$

$\tilde{h}_{j,p} = g\left(\bar{h}_{j,p}\right)$

$\bar{h}_{j,p} = \sum_{l=0}^{N} V_{jl}\, x_{l,p}$

$g(x) = \frac{1}{n_s}\sum_{i=1}^{n_s} g_0\left(\beta_h(x-\theta^r)\right)$

The minimization of the MSE defined by equation 9

$$E(p) = MSE = \frac{1}{2}\sum_{i=1}^{N_s}\left(y_{i,p} - \hat{y}_{i,p}\right)^2 \quad (9)$$

The minimization of the objective function G defined by equation 10

$$G = \frac{1}{2}\sum_{j=1}^{N_h}\sum_{m=1}^{N_s}\sum_{p\in C_m}\left(< \tilde{h}_{j,C_m} > - \tilde{h}_{j,p}\right)^2 \quad (10)$$

With $< \tilde{h}_{j,C_m} >$ is the average value of these outputs associated with samples of a given class C_m and β_h the slope of the multi-levels sigmoid functions.

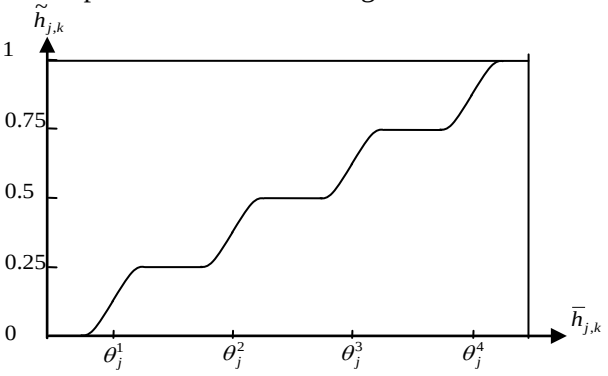


Fig.7 Example of four-level activation function (ns = 4) applied to the j-th neuron of the hidden layer from the k-th observation vector.

3.4. Classification by wide margin separators networks (SVM)

Wide margin separators are mainly used to treat the non-linear discrimination problem [12][6] such as SAR and ASTER image textures. We have used the predefined functions of the software library Matlab7.10 (R2009a) [13] to realize the training and the test phases of the networks (see figure 4). The predefined function used for the training accepts six parameters in the input and deliver four output parameters (see figure 9 a). The predefined function for the test accepts six parameters whose have been adjusted for the determination of the hyper-plane in the training phase, the support vectors $\hat{y}_{ip}$ and the test examples, and delivers only one output, namely the real output for each test sample (see figure 9 b).

The SVM networks [11] architecture has four SVM (Architecture "1A1"), relative to the number of textures per image (figure 9). After the training phase, each SVM*i* network recognizes the samples associated to the T*i* texture of all samples used in the training of QNN and FFNN networks. However, for one SVM*i* network among the four networks, corresponds one output y_i which converges to +1 and the other SVM*j* networks with j ≠ i converge towards -1 (table 3). Note that the four networks are used in the training phase at the same time. To obtain the converge of these networks to the desired outputs for all samples in the training database, some parameters of global SVM network such as the standard deviation (σ), the variable spring (ξ), the width of the margin (C) and the number of support vectors (NSV) should be adjusted.

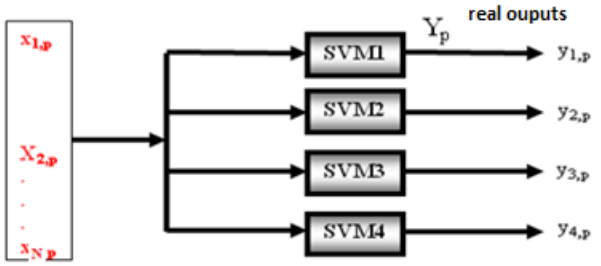
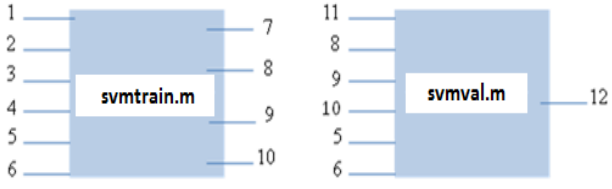


Fig.8 Architecture of a SVM network.



a b

Fig.9 Parameters of predefined functions in the software library "SVM and Kernel Methods Matlab Toolbox": a) training phase with svmval.m, b) test phase with function svmtrain.m

The different inputs are:

1: Designed samples for training of the networks
2: Desired outputs for each training sample
3: The constant C which allows controlling the compromise between the classification errors number and the marge width.
4: The variable which describes the classification error of the examples
5: The kernel used to compute the scalar product
6: The adjustment kernel parameter
7: The real outputs for each training sample
8: The support vectors
9: Weights W
10: Slant W_0
11: Designed samples for test
12: Real outputs for each test sample.

The parameter from 3 to 6 is to be adjusted.

3.5. Images classification procedure

Remote sensing images are sampled with a scan window for which the size is K_b x K_b defining the number of input nodes *N* of the network designed after training phase. During the classification, each texture sample is presented to the network input (see table 2). The network output index (among *Ns* networks), providing the minimum value of the MSE between the real output of the network and the output of the different codes previously attributed to the textures (table 3), is considered as the label (or class) associated with the current sample. The resulting image is an image of labels; it represents the image classified by the network.

Tab.3 Codes of desired outputs of the network for each type of texture
a) FFNN and SVM, b) QNN.

(a)

Classes	1	2	3	4
	d_1	d_2	d_3	d_4
	1	-1	-1	-1
	-1	1	-1	-1
	-1	-1	1	-1
	-1	-1	-1	1

(b)

Classes	1	2	3	4
	d_1	d_2	d_3	d_4
	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1

4. Results and interpretation

4.1. Classification results of FFNN networks

To the textures of both ASTER and SAR images is assigned several neural architectures possible that have satisfied both two criteria: rapidity of the convergence of the adaptation algorithm of the synaptic weights to the *EQMmin*, and maximal texture identification. For these neural configuration cases, the simulations with the training examples and testing examples (see figure 4), MSE_L and MSE_T are quite low. Therefore, the image texture identification rates are all greater than 75% [14]. For each size of scanning window, a neural network was designed to give the best results in texture classification (see tables 4 and 5). For these networks cases, the best identification rate is more than 90% for the textures in the SAR image and was obtained using a FFNN network composed of a single layer of 100 neurons (see tables 5 and 6). Also, the best identification rate is more than 98% for texture ASTER image, obtained using a FFNN network composed of two layers: the first layer contains 120 neurons and the second layer contains 45 neurons (see tables 4 and 6).

Table 4 Optimal parameters of FFNN for ASTER image.

K_b	n	m	iter	Tc(mn)	μ	MSEmin	MSE_L	MSE_T
3	5	15	119000	35	0,06	0,001	0,013	0,026
5	200	60	43230	30	0,05	0,0001	0,0081	0,0143
7	120	45	248124	14	0,05	0,001	0,037	0,042
9	120	45	33500	30	0,03	0,0001	0,0144	0,0149
11	120	45	137970	30	0,05	0,00001	0,005	0,0057

Table 5 Optimal parameters of FFNN for SAR image.

K_b	n	m	iter	Tc(mn)	μ	MSEmin	MSE_L	MSE_T
3	30	0	5.10^5	277	1	0,001	0.304	0.7
5	20	0	10^5	30	1,5	0,001	0,048	0,342
7	100	0	214411	135	0,05	0,001	0,027	0,158
9	30	0	79633	25	0,05	0,001	0,034	0,202
11	100	0	30949	21	0,05	0,001	0,038	0,176

Table 6 Average rate of textures identification of Aster and SAR images (in %) obtained by FFNN network

K_b	3	5	7	9	11
ASTER	98,33	98,43	98,4	98,68	97,6
SAR	80,55	87,45	90,93	87,83	84,7

From the results, the basic unit textures, or texton, of ASTER image varies with the nature of the texture (see table 7 and figure 10). Indeed, for textures C.L5-10 and C.L50-75, a window of size $K_b = 3$ allows a best representation of these textures. In this case, the network is optimized in terms of neurons number (20 neurons) but not optimized in terms of training time (T = 35 min). For texture C.L50-75, the window size $K_b = 9$ gives good identification rate, but the network requires a higher number of neurons (165 neurons). For texture C.L + 75, both the window sizes $K_b = 7$ and $K_b = 9$, give

the identification rates more than 99% for FFNN network with 15 neurons and a training time varying between 14 min and 30 min. Finally, the texture P.A100, is better represented by the window size Kb = 5 or more; in this case, the number of neurons can vary from 165 to 260 for a training time varying between 14 and 30 min. The analysis results confirmed that the size of the window increases with the overlapping rate (R) of the plant species. We conclude that the FFNN network requires more neurons.

Table 7 Performance of FFNN network for ASTER image

	C.L5-10	C.L50-75	C.L+75	P.A100
K_b	3	3, 5, 7, 9, 11	7, 9	5, 7, 9, 11
Identification rate (%)	97	> 99	> 99	> 99
n+m	20	20, 260, 165, 165, 165	165,16	260,165, 165, 165
Tc(min)	35	35, 30, 14	14, 30	30, 14, 30, 30

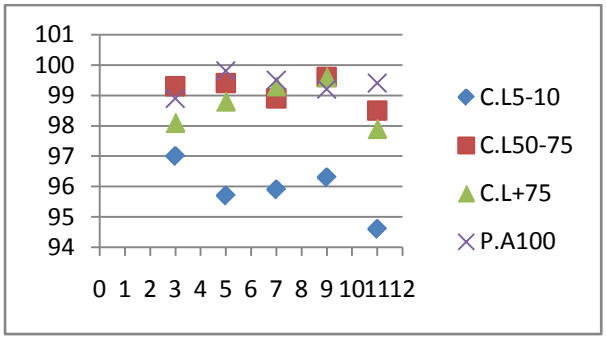


Fig.10 Textures identification rate of ASTER image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for FFNN networks.

The criteria used in the design of FFNN networks for the identification of textures of the image ASTER were also applied to networks designed to identify the textures of the SAR image. The classification results show that the textures in the SAR image require neuronal architecture composed of a smaller number of neurons than that of the image ASTER (table 5). The texture identification rates of SAR image are greater than 90 % but are inferior to those of ASTER; this is due to the texture mixture which exists in SAR (heterogeneous texture) and the higher resolution (more details) of the image SAR than the image ASTER (see figure 11). The basic unit of the textures of the SAR image is variable (see table 8). Thus, for the textures O.V and B.V, a window of size K_b = 7 can describe these textures. The texture SN.V can be represented by a window of size K_b = 5, with a FFNN network for which the number of neurons is lower 20 neurons than other networks. Finally, the texture V requires a window of size K_b = 11, much wide than that of other textures.

From these results, textures are represented by the variable size windows depending of the textural nature: more the form is randomly distributed; more is the number of input nodes required by the network and consequently a larger window size.

Table 8 Performance of FFNN network for SAR image

	O.V	B.V	SN.V	V
K_b	7, 9, 11	7	5	11
Identification rate (%)	> 90	~ 90	~ 90	> 90
n+m	100, 30, 100	100	20	100
Tc(mn)	13, 25, 21	135	30	21

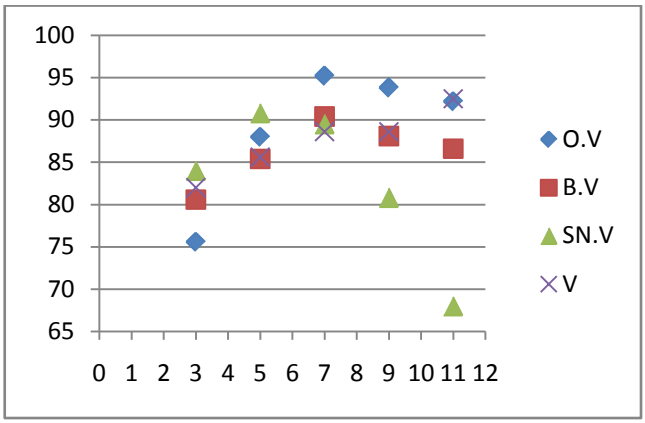


Fig.11 Textures identification rate of SAR image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for FFNN networks.

In conclusion, the results of the texture identification by FFNN networks show that the identification rate values depend on the mode of image acquisition. Also, the spatial resolution, the characteristics and the texture overlapping have a direct impact on FFNN network architecture. Simulations of FFNN networks were performed by Matlab 7.4 using an Intel CPU Q8200 2.33GHz processor Core2Quad.

4.2. Classification results of QNN networks

The design of QNN network is based mainly on the optimization criterion of MSE_L. For this purpose, five QNN networks per image, according to $K_b \in \{3, 5, 7, 9, 11\}$, were selected (see tables 9 and 10). The decision at this step is the texture identification rate results of the two images (see figures 12 and 13).

Table 9 Optimal parameters of QNN network for ASTER image, EQMmin = 0.01, $\boldsymbol{\alpha_\theta = 0.0001}$

K_b	N_h	n_s	i	Tc	μ	MSE_L
3	17	5	2000	18h46mn31s	0,99	0,11
5	19	10	2000	19h49mn20s	0,99	0,09
7	20	6	2000	14h42mn21s	0,4	0,06
9	20	7	2000	05h05mn44s	0,8	0.02
11	18	7	2000	2h42mn42s	0,4	0,02

Table 10 Optimal parameters of QNN network for SAR image, MSEmin = 0.01, $\boldsymbol{\alpha_\theta = 0.0001}$

K_b	N_h	n_s	i	Tc	μ	MSE_L
3	12	7	2000	16h29min21s	0,9	0,08
5	19	8	2000	11h39min04s	0,4	0,03
7	12	8	2000	6h02min14s	0,3	0,02
9	12	8	2000	2h41min34s	0,6	0,004
11	20	6	2000	1h44min28s	0,1	0,05

The classification results show that the best identification rate is more than 95% for the textures in the SAR image; these results were obtained by using a QNN network with 20 neurons in the hidden layer (tables 9 and 10). Also, the best identification rate is more than 90% for ASTER textures image; this result was obtained using a QNN network with 18 neurons in the hidden layer (see tables 11 and 12). The identification texture depends on the size K_b of the scanning window and the nature of the texture (see tables 12 and 13). The textures C.L5-10 and C.L50-75 of ASTER image are better identified for large values of K_b. In this case the identification rate reaches the values more than 70% for K_b =11. The texture C.L + 75 is very well identified for K_b = 11 because the identification rate is nearly 100%. The texture P.A100 is 100% identified regardless of the size of the scanning window.

The textures of the SAR image are all identified to more than 70% for the sizes of scanning windows greater to K_b = 5. Thus, for the texture O.V, the best identification rates are obtained for K_b = 5 or K_b =11 while for the texture B.V, the maximum rate is obtained when K_b = 11. Two cases are possible for the texture SN.V because we have obtained more than 99% of good classification when K_b = 5 or K_b = 11. The identification rate of texture V is more than 95% for K_b = 9 or K_b = 11. For this image, we have obtained the identification rate which is nearly 100%.

Table 11 Average identification rate for textures of ASTER and SAR images (in %) using QNN network.

K_b	3	5	7	9	11
ASTER	27.93	25.07	54.77	78.16	90.86
ROS	57.87	86.20	88.01	85.74	95.09

All textural forms of ASTER image can be represented by a scanning window of size K_b = 11. A unique choice of QNN network for all textures of ASTER image composed of 18 neurons in the hidden layer and 7 neurons in the output layer allows the faster training and allows classifying this image with best results (see table 12).

Table 12 Performance of QNN network for ASTER image

	C.L5-10	C.L50-75	C.L+75	P.A100
K_b	11	11	11	3, 5, 7, 9, 11
Identifi-cation rate in (%)	~91	~72	> 99	100
N_h	18	18	18	17, 19, 20, 20, 18
Tc	2h42min42s	2h42min42s	2h42min42s	18h46min31s 19h49min20s 14h42min21s 05h05min44 2h42min42s

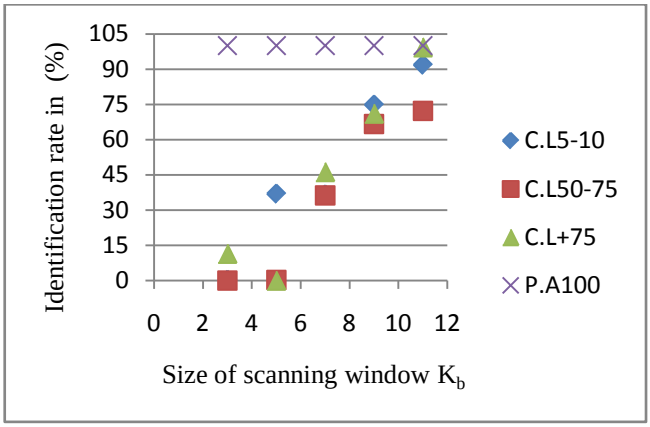


Fig.12 Textures identification rate of ASTER image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for QNN networks.

All textures of the SAR image are also well represented by a scanning window of size K_b =11; however the textural element of each texture can be identified by smaller sizes. We have observed that this may cause a degradation of QNN network performance in terms of training time with a compromise on the number of neurons in the hidden layer (see table 13).

Table 13 Performances of QNN network for SAR image

	O.V	B.V	SN.V	V
K_b	7,11	11	3, 5, 7, 11	9, 11
Identification rate in (%)	~85	~95	> 95	> 95
N_h	12, 20	20	12, 19, 12, 20	12, 20
Tc	6h02min14s, 1h44min28s	1h44min28s	16h29min21s 11h39min04s 6h02min14s 1h44min28s	2h41min34s 1h44min28s

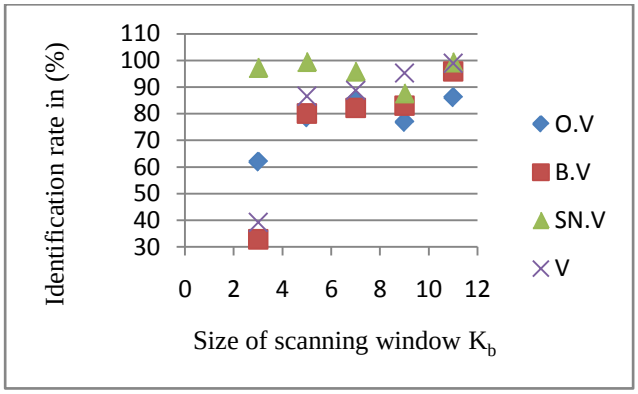


Fig.13 Textures identification rate of SAR image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for QNN networks.

All the programming of QNN and SVM networks was performed under a processor based platform Centrino ™ with a frequency of 1.7GHz.

4.3. Classification results of SVM networks

Several SVM networks can be used for textures classification. Successful networks are the ones whose respond to the convergence criteria of the MSE_L, the speed of convergence during training phase (see tables 14 and 15) and for which the evaluation of texture classification has reached the maximum identification rate. Note that MSE_L has reached 10^{-5}.

The textures of the ASTER image have an average classification rate of 100% regardless the size K_b x K_b of the scanning windows. However for the SAR textures image, the average identification rate is not high. The best average textures identification rate is obtained by the SVM network for K_b = 11 (see table 16).

Table 14 Optimal parameters of SVM network for SAR image. Inf: infinity

K_b	NSV1	NSV2	NSV3	NSV4	C	σ	Tc(s)	ξ
3	606	569	560	585	500	0,5	480	10^{-05}
5	548	628	218	501	Inf	0.5	202	10^{-03}
7	141	124	101	129	Inf	2	17	10^{-03}
9	138	107	80	101	Inf	2	11	10^{-07}
11	142	76	53	70	Inf	2	5	10^{-07}

Table 15 Optimal parameters of SVM network for ASTER image. C = Inf., σ = 2.

K_b	NSV1	NSV2	NSV3	NSV4	Tc(s)	ξ
3	36	6	60	22	9.60	10^{-05}
5	23	6	44	20	5.26	10^{-05}
7	23	6	44	17	6.90	10^{-07}
9	24	13	37	20	4.95	10^{-08}
11	25	15	38	22	2.24	10^{-08}

The textures of the SAR image classification rates has slightly the lower values than those of ASTER image (see figures 14 and 15) although the number of examples of texture used for training phase is roughly equal to twice of those of the ASTER image (see figure 3).

Table 16 Identification average rate of textures images
ASTER and SAR (in %) for SVM network

K_b	3	5	7	9	11
ASTER	99,53	99,93	99,43	100	100
SAR	80.91	92,50	96.05	97,66	98,44

The K_b size of scan windows is not a mandatory parameter for the identification of the textures of the ASTER and SAR images. However, more the scanning windows size K_b increases, less the SVM networks require support vectors and therefore training is faster (see tables 16 and 17). Note that the SVM networks designed for the classification of SAR image require a much larger number of support vectors than those designed for the classification of ASTER image (see tables 14 and 15).

Table 17 Performance of SVM network for ASTER image

	C.L5-10	C.L50-75	C.L+75	P.A100
K_b	3, 5, 7, 9, 11	3, 5, 7, 9,11	3, 5, 7, 9, 11	3, 5, 7, 9, 11
Identifi-cation rate in (%)	> 99, 99, > 99, 100, 100	100	> 99, 100, 100,100,100	100
NSV1+NSV2 + NSV3+NSV4	124; 93; 90; 94; 100	124; 93; 90; 94; 100	124; 93; 90; 94; 100	124; 93; 90; 94; 100
Tc(s)	9.60; 5.26; 6.90; 4.95; 2.24	9.60; 5.26 6.90; 4.95; 2.24	9.60; 5.26; 6.90; 4.95; 2.24	9.60; 5.26; 6.90; 4.95; 2.24

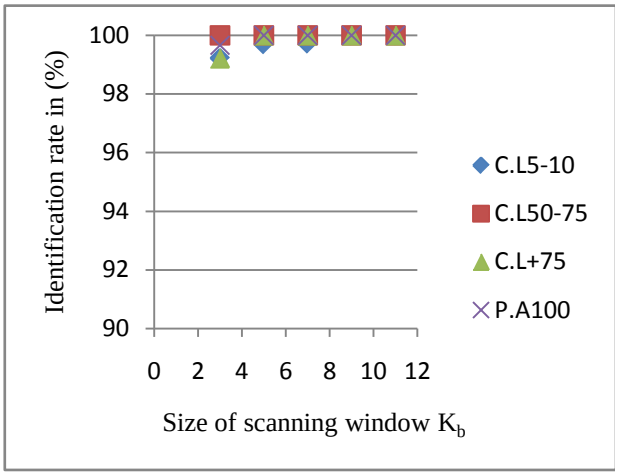


Fig.14 Textures identification rate of ASTER image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for SVM networks.

Table 18 Performance of SVM networks for SAR image

	O.V	B.V	SN.V	V
K_b	5, 7, 9, 11	5, 7, 9, 11	3, 5, 7, 9, 11	5, 7, 9, 11
Identifi-cation rate in (%)	> 91, > 94, > 96, > 96	> 90, > 94, > 96, >98	> 93, > 97, > 99, > 99	> 90, > 95, > 97, > 98
NSV1+NSV2 + NSV3+NSV4	1895; 495; 426; 341	1895; 495; 426; 341	2320; 1895; 495; 426 ; 341	1895; 495; 426; 341
Tc(s)	202; 17; 11; 5	202; 17; 11; 5	480; 202; 17; 11; 5	202; 17; 11; 5

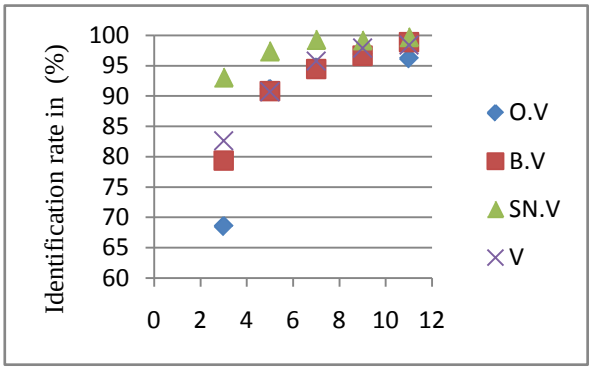


Fig. 15 Textures identification rate of SAR image in function of the different scanning window sizes K_b = 3, 5, 7, 9, 11 for SVM networks.

5. Confrontation of the classification results

The best results of image classification are those obtained by using the SVM networks (see table 18). These networks also have the advantage of being the faster during the training phase and are also more robust than FFNN and QNN networks. However, they require a large number of NSV and the adjustment of several parameters at once.

The QNN networks achieve texture identification rate of the SAR image slightly higher than those obtained with FFNN networks and also have the advantage of being faster and use fewer neurons that FFNN networks. However, for the ASTER image, the results of the identification rate of textures by the FFNN networks are better than those obtained by the QNN networks. The FFNN networks have the advantage of being much faster during the training phase but require more neurons (see tables 4 and 9).

Table 19 Textures identification rate for the remote sensing images obtained by the different classification methods. (a) Image ASTER, (b) Image SAR.

(a)

K_b	FFNN	QNN	SVM
3	**98,33**	27.93	**99,52**
5	**98,43**	25.07	**99,92**
7	**98,40**	54.77	**99,43**
9	**98,68**	78.16	**100**
11	**97,60**	**90.86**	**100**

(b)

K_b	FFNN	QNN	SVM
3	80,55	57.87	80.91
5	87,45	86.20	**92,50**
7	**90,93**	88.01	**96.05**
9	87,83	85.74	**97,66**
11	84,70	**95.09**	**98,44**

For the textures of SAR image, the networks whose allow obtaining the best average identification rates are themselves the faster for the training. In these cases, the size of the scanning window differs from one network to another (see table 20). Therefore, we cannot define a dimension of the representative texton of the texture by evaluating the average identification rates results. The results differ for the ASTER image textures and especially in the case of

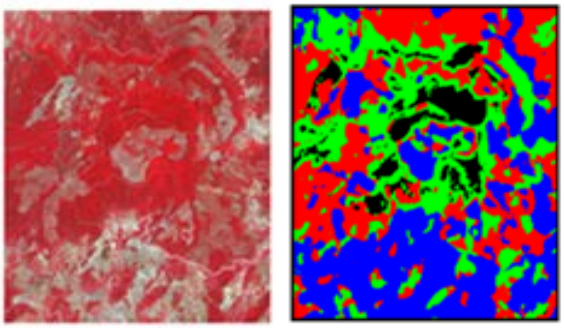
the FFNN networks. Indeed, they identify all textures with the average rates more than 90% but the faster is the one for K_b = 9 (see tables 19a and 19b). The results of classified images on table 19 are represented by figures 16 and 17.

Table 20 Performance of networks and optimal texture identification rates for remote sensing images obtained by the different classification methods.
(a) ASTER image, (b) SAR image.
(a)

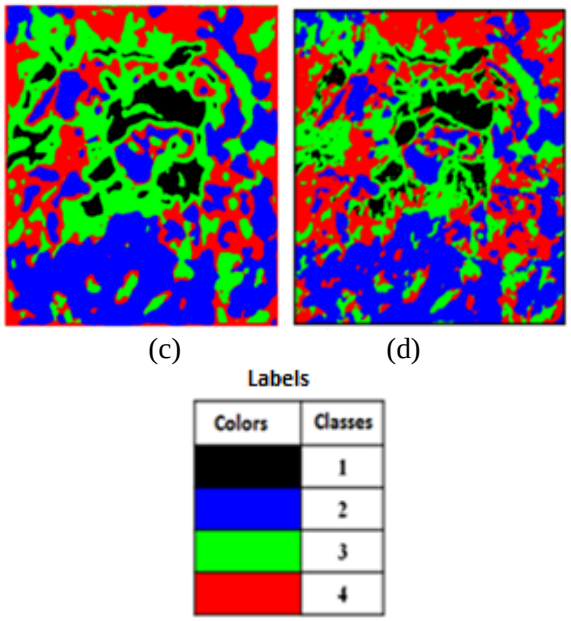
		FFNN	QNN	SVM
K_b		9	11	9 or 11
MSE_L		0,014	0,019	10^{-05}
Tc		30mn	2h42mn42s	11s, 5s
Identification rate in (%)	C.L5-10	96,3	91,8	100
	C.L50-75	99,6	72,2	100
	C.L+75	99,6	99,4	100
	P.A100	99,2	100	100

(b)

		FFNN	QNN	SVM
K_b		7	11	11
MSE_L		0,027	0,046	10^{-05}
Tc		2h15mn	1h44mn28s	2,24s
Identification rate in (%)	O.V	95,2	86,2	96,2
	B.V	90,4	95,9	98,9
	SN.V	89,5	99,3	99,7
	V	88,6	99	98,4

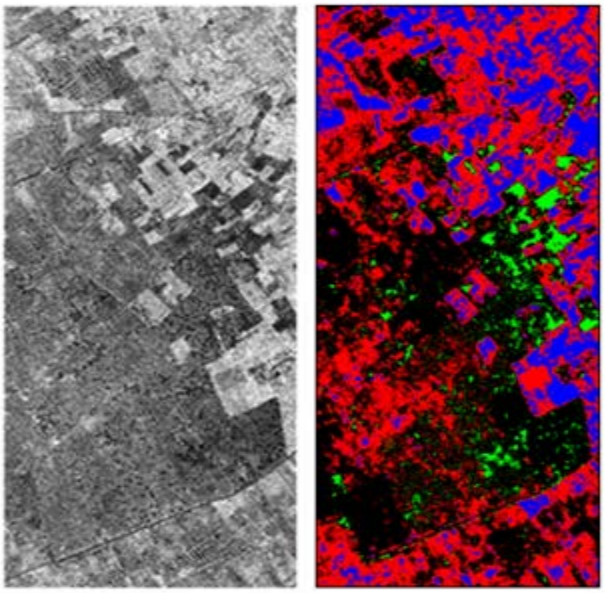


(a) (b)

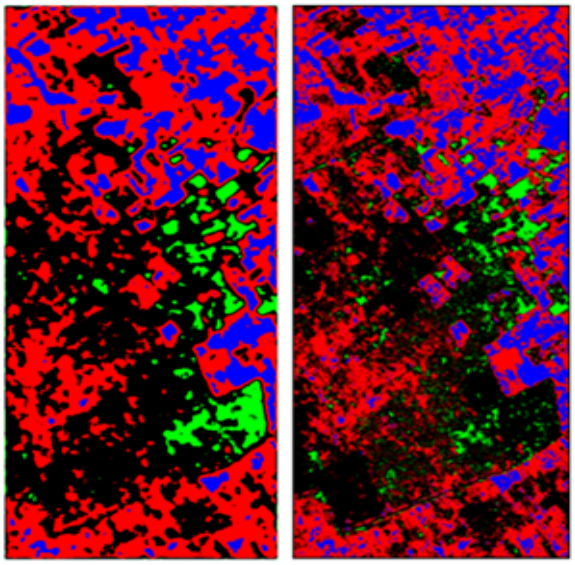


(c) (d)

Fig.16 Better classification results from the ASTER image. (a) Original image,
(b) FFNN, (c) QNN, (d) SVM.



(a) (b)



(c) (d)

Fig.17 Better classification results of the SAR image. (a) Original image (b) FFNN, (c) QNN, (d) SVM.

6. Conclusion and perspectives

In this paper, we have shown the influence of some textural parameters on the design of optimal architectures of the FFNN, QNN, and SVM networks, able for the textures identification of the remote sensing images with different spatial resolutions and acquisition modes.

The parameters of the networks have been adjusted according to the criterion of minimum mean squared error. The successful networks are those that achieve the maximum texture identification rates and this for the faster networks in training time.

The results of the texture identification show that it is possible to assign to a given texture a classification method and a window size of texture analysis, which is accurate for its identification with the classification rates superior to 96%. A multitude of parameters must beings optimized to achieve satisfactory classification results.

In the perspectives of this work, we will automatically design the SVM networks; the goal is to facilitate the training phase and in the same time, reduce the user's action. We think that it is possible to optimize the number of vector supports which may be obtained by realizing the parameterization of the SVM inputs than the direct utilization of the digital pixel values of the textures. This parameterization may be realized by using the orthogonal transforms applied to textures. Also, to reduce the duration of the simulations, we will realize the implementation of the developed programs using the compiled languages such as the C or C++ than Matlab used in this work.

References

[1] Szenasi S, Vamossy Z, Kozlovszky M, "Preparing initial population of genetic algorithm for region growing parameter optimization", 4th International Symposium on Logistics and Industrial Informatics: LINDI 2012, Smolenice, 2012, pp.47-54.

[2] Z. Kerekes, Z. Toth, S. Senasi, Z. Vamossy, Sz. Sergyan, "Colon Cancer Diagnosis on Digital Tissues Images", Proceeding of IEEE 9th International Conference on Computational Cybernetics. Tihany, 2013, pp.159-163.

[3] Hornik K., Stinchcombe M., and White H.: "Multilayer feed forward Networks are universal approximators", Neural Networks, Vol.2, pp.359–366, 1989.

[4] Hammerstrom D. "A digital VLSI architecture for real-world applications", in An Introduction to Neural and Electronic Networks, S.F. Zornetzer, J.L.Davis, C.Lau, and T.McKenna, Eds. New York: Academic, 1995.

[5] Cybenko G. "Approximations by superposition of sigmoidal function", Math. Contr., Signals, Syst., Vol.2, pp: 303–314, 1989.

[6] David V., Sanchez A.: "Advanced support vector machines and kernel methods", Neurocomputing, N° 55, 2003, pp. 5-20.

[7] Chapelle O., Haffner P. and Vapnik V.N.: "Support Vector Machines for Histogram-Based Image Classification", IEEE Transactions on neuronal networks, Vol.10, N°.5, September 1999, pp.1055-1064.

[8] Li S. and *al.* "Texture classification using the support vector machines", Pattern recognition, N°.36, 2003, pp. 2883-2893.

[10] Gidudu A., Hulley G. and Marwala T., "Image classification using SVMs: One-against-One vs One-against-All", IOP publishing, November 2007, 6 pages.

[10] Looney C.G. "Pattern recognition using neural networks, theory and algorithms for engineers and scientists", Edition Oxford university press, USA, 1997, 480 pages.

[11] Krishnapuram B., Sichina J.et Carin L. "Physics-based detection of targets in SAR imagery using Support Vector Machines", IEEE Sensors journal, Vol.3, N°.2, Avril 2003, pp: 174-185.

[12] Cortes C., Vapnik V. "Support-Vector Networks", Machine Learning, 20, 1995, pp. 273-297.

[13] http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/SVM-KM.zip
[14] Marrakchi O., Agina A. et El Ghali A., "Evaluating SVM and BPNN classifiers for remote sensing data", International Review on Computers and Software (I.RE.CO.S.), Vol. 4, N°. 5, pp. 600, Sept. 2009.
[15] Purushothaman G. et Karayiannis N. B.: "Quantum Neural Networks (QNN's) :Inherently Fuzzy Feedforward Neural Networks" IEEE transactions on neural networks,Vol.8, N°.3, pp: 679-693, 1997.