Journal of Engineering and Applied Sciences 13 (11): 4020-4024, 2018
ISSN: 1816-949X

Recognition Textures of the Tumors of the Medical Pictures by Neural Networks

Youssef Douzi, Nabil Kannouf, Tarik Hajji, Touria Boukhana,
Mohamed Benabdellah and Abdelmalek Azizi
Laboratory of ACSA, Faculty of Sciences, University Mohammed First, Oujda, Morocco

Abstract: Texture plays a very important role in identifying and extracting the thematic information contained in the image. Texture analysis is a vast field whose objective is to identify the nature of a texture, either via. classification algorithms or via. synthetic algorithms aimed at the creation of a texture, visually similar to the original texture. As specialists are looking for radio-tracers to use in order to do a more advanced study on diseases that infect the skin and organs in general, we have come back to thinking about using ultrasound as ultrasound may well replace radiography in some cases like breast cancer screening. Our goal is to introduce methods to classify different diseases which infect the skin and organs leaving traces by adaptive texture analysis of ultrasound images, i.e., to make a recognition of different types of tumors on medical images and to describe a new approach to automatic texture recognition in digital images using Artificial Neural Networks (ANN).

Key words: Textures, recognition, tumors, medical images, learning, artificial neural networks

INTRODUCTION

A rguably, textures represent a region in a digital image that has uniform characteristics. These characteristics are such a basic pattern that can be repeated or frequency characteristics. Several methods have been invented to make textures of the analysis in a digital image.

Structural methods (Fazia, 2013): They take into account the structural and contextual information of a form and are particularly well suited to macroscopic textures. The analysis steps are first identifying the components and the definition of investment rules. The two most important structures are graph structures and syntactic structures (Jan and Hsueh, 1998; Philip, 1989).

Statistical methods (Fazia, 2013): The texture is considered to be the realization of a stationary stochastic process. Following the modality of images to study, the most discriminating signature texture is to seek either in methods that directly exploit the statistical properties of the texture (co-occurrence matrix (Haralick, 1979), run length arrays matrix (Galloway, 1975), neighborhood matrix, autocorrelation function, Markov Model, model auto regressive (Chellappa and Kashyap, 1985) models from mathematical morphology (Serra, 1982)) or in methods that use the statistical properties from a transformed plane in which we rewrite the texture image (spectral density, local extrema methods (Mitchelle *et al.*, 1977), processing methods fourier, methods of transformation of Karhunen-Loeve, Walsh Hadamard). The methods based on forms studies (Fazia, 2013) are found at the intersection of pattern recognition, characterization of defaults and the macrotextural analysis. The textural regions of the image include specific forms and can be characterized by parameters so-called forms (Peet and Sahota, 1985; Besl and Jain, 1986).

From the point of view of image synthesis, fractal methods (Fazia, 2013) are apart because they allow to synthesize very realistic images. In texture analysis, the fractal dimension which is a measure of the degree of irregularity of an object describes a certain property of the texture. The fractal model is essentially based on the estimate by spatial methods, the fractal dimension of the gray levels representing the surface of the image (Brown *et al.*, 1992; Pentland, 1984; Keller *et al.*, 1989; Sarkar and Chaudhuri, 1992).

In the filtering methods, researchers are interested in the texture analysis by multi-channel filtering. Most of the proposed techniques use benches of selective filters in orientation, frequency and scale. Literature is rich concerning the analysis of textures by benches filters of Gabor (Haley and Manjunath, 1995) and wavelet (Unser, 1995). The ANN represent problem solving units able to hang a blur of information in parallel and come out with one or more results representing the postulated solution. The basic unit of ANN is a nonlinear combinatorial function called Artificial Neuron (AN).

Corresponding Author: Youssef Douzi, Laboratory of ACSA, Faculty of Sciences, University Mohammed First, Oujda, Morocco

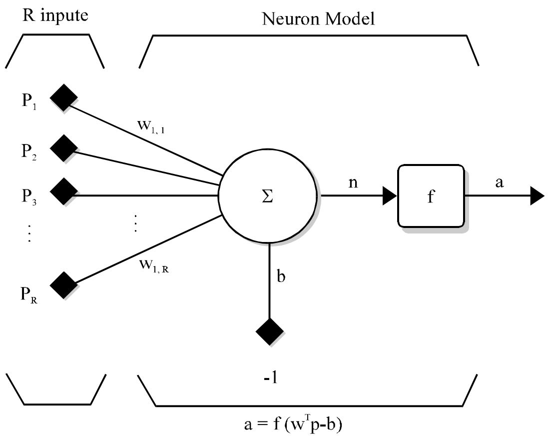

Fig. 1: Artificial neuron

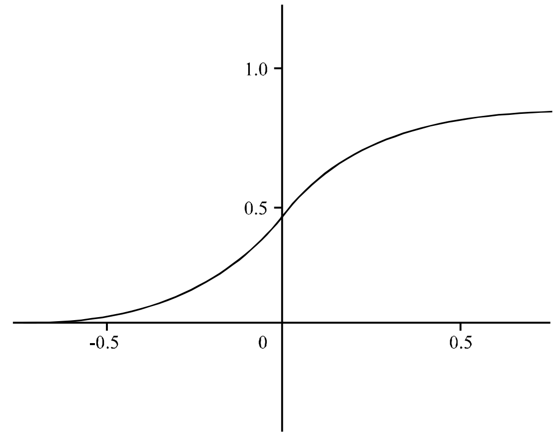

Fig. 2: Sigmoid function

AN represents a computer simulation of a biological neuron of the human brain. Each AN is characterized by a vector of information which is presented at the input and a non-linear mathematical operator able to calculate a relative output on this vector. Figure 1 shows an AN:

X_i represent the elements of the input vector and we have for each entry an associated value. $W_{i,j}$ represent the synapses (the weights) of the neuron j, they are real numbers between 0 and 1. In the creation of a AN these weights are initialized randomly.

The function of combinations is a summation function between active synapses associated to the same neuron. The activation function is a nonlinear operator able to return a real value or rounds in the range (0-1). We often use the sigmoid function:

$$f(x) = 1/(1+\exp(-x)) \quad (1)$$

Figure 2 shows the graph of this function. A ANN is composed by a collection of inter-connected artificial neurons between them to form a neural system able to learn the knowledge and understand the understanding mechanisms. Each ANN is characterized by his own architecture, it is meant by this architecture the number of neurons of the input layer, the number of hidden layers, the number of neurons in each hidden layer and the number of neurons in the layer of exit. A layer of neurons in an ANN represents a group of artificial neurons that have the same level of importance as in Fig. 3.

The ANN of Fig. 3 is characterized by an input layer of 8 artificial neurons, so, a vector of size 8 must be presented at the input of this network as the network entry information. It is also characterized by two hidden layers, the first is formed by 11 artificial neurons and the second is formed by only 4. The last layer (output layer) is formed by a single AN. And as illustrated in Fig. 4, the neurons of every hidden layer communicate only with the neurons of the two adjacent layers.

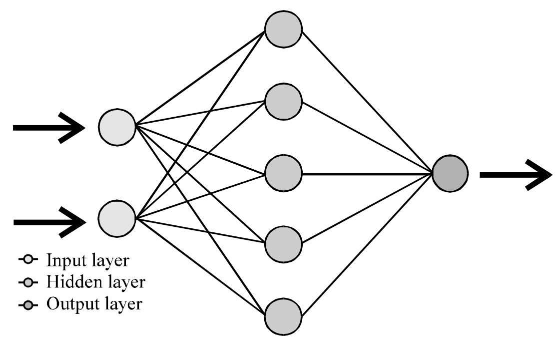

Fig. 3: Artificial neural network

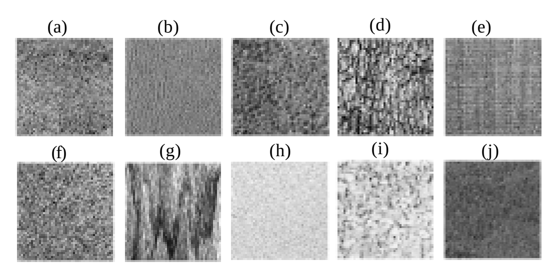

Fig. 4: a-j) Images of the Brodazt database

The principle of operation of ANN is similar to operation of the human brain. First, it is essential to pass on the learning phase to record the knowledge in the memory of ANN. The memorization of knowledge is done by the principle of repetition and correction on a collection of data that forms the learning base. We have a lot of algorithms able to teach an ANN as back propagation.

The back propagation is a method of calculating the weights for a ANN that has a supervised learning that consist in minimizing the quadratic output error. It is to correct the errors depending on the importance of the elements that justly participated in the realization of these errors, the synaptic weights that help to generate a significant error will be changed more significantly than the weights that generated a marginal error.

The weights in the neural network are first initialized with random values. Then we consider a data set that will be used for learning. Each sample has its target values which are those that the neural network must eventually predict when we present the same sample to it. After the learning phase is closed, we proceed to test our ANN on a carefully selected data sample named the test base this base is also used to calculate the rate of convergence of the network of artificial neurons. And finally, the ANN can be used to solve the new cases through the knowledge stored in the synapses of the ANN with the same manner as the human brain.

Our method: We propose the use of a neural network with supervised learning and taught. It will recognize textures in an image. The back propagation gradient is the learning algorithm. We mention some advantages of our method.

The first advantage of this method is that texture's recognition is independent of image characteristics (co-occurrence matrix, length arrays matrix, neighborhood matrix, autocorrelation function, Markov Model, autoregressive model, derived models from mathematical morphology). The second advantage is that the time of execution is shorter compared to another method which takes more time, over than being slow. Thirdly, the recognition of textures in a digital image in a short time (complexity) will allow us to increase the recognition rate compared to other approaches texture.

MATERIALS AND METHODS

We use an artificial neural network, a learning database (Brodazt) and the gradient backpropagation algorithm. In this research, we select some textures to recognize from the Brodazt database.

Neural network architecture used: Based on the research by Tarik *et al*. (2014), we used a multi layer neural network characterized by an input layer composed of a square matrix of size 8*8 artificial neurons which a single hidden formed by size of 4*4 matrix and an output layer composed of 10 artificial neural to reflect the number of texture to recognize who is 10.

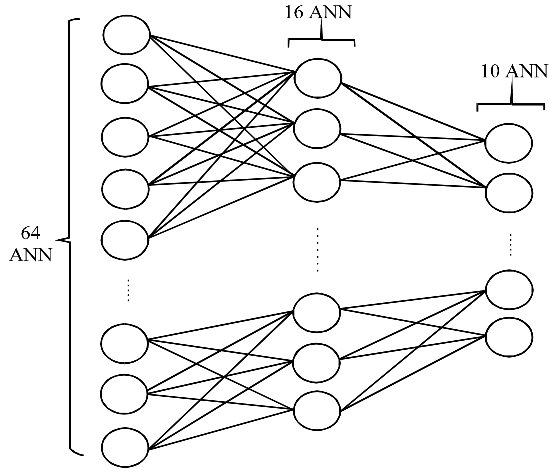

Fig. 5: Our neural network

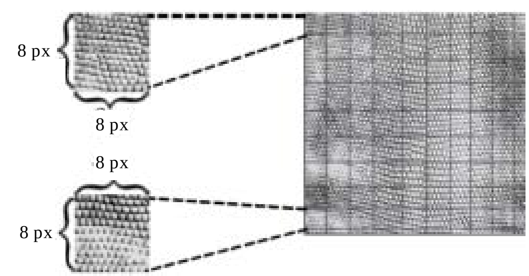

Fig. 6: Example of an image

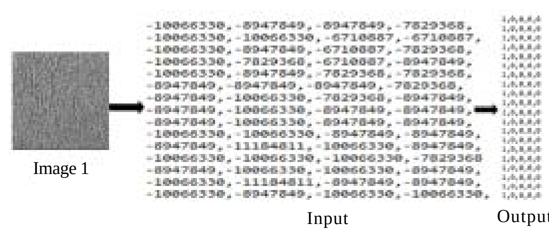

Fig. 7: Example of a texture and his class

The learning database: We took the Brodatz album then, we divided each image that makes up the album to blocks of size 8*8. It must be noted that each image represents a texture class.

Technically, we have created two files: An input and an output. The first one is organized to record in each row the values of pixels of a block of the image, the second one corresponds to the class of an input block (Fig. 5-7).

Learning algorithm: Learning algorithm that, we used was inspired by the back propagation of the gradient and it is as follows. According to the size n of the input layer of the ANN concerned (the number of artificial neurons that compose the input layer), we subdivide each image in the learning database into blocks of size n. For each block Xi which we put at the entrance of ANN, we propagate the forward signal in the hidden layers and the output layer using Eq. 2:

$$X_j^n = Y_j = g^{(n)}(h_j^{(n)}) = g^{(n)}\left(\sum_{k=1} W_{jk}^{(n)} X_k^{(n-1)}\right) \qquad (2)$$

At the output layer ANN, we generate a block T_j corresponding to the output generated by the network for the block X_j. We calculate the error between the output generated by T_j network and the desired output X_j by using Eq. 3:

$$e_i^{sortie} = g'(h_j^{sortie})\,[T_i\text{-}X_i] \qquad (3)$$

Where:
G () = Activation function
H () = Aggregate function
$W_j k$ = Synaptic weights between neuron $X_k^{(n-1)}$ and the $X_i^{(n)}$

We disperse the error backward:

$$e_j^{(n-1)} = g'^{(n-1)}\ (h_j^{(n-1)}) \sum_i W_{ij}\ e_i^{(n)} \qquad (4)$$

With:

$$e_j^{(n)} \sum_i \frac{\partial y_i}{\partial h_j^{(n)}} \qquad (5)$$

We update the weights with the following relationship:

$$\Delta W_{ij}^{(n)} = e_i^{(n)}\ x_j^{(n-1)} \qquad (6)$$

We redo this process with the next block.

Implementation: We used the Java programming language to implement our neural system. We created 5 Java classes (Fig. 8).

- Class 1 (backpropagation) for implementation of the learning algorithm
- Class 2 (JavaTraitmentImages) to implement the various operations that can be performed on an image such as extraction of the blocks, loading, recording
- Class 3 (layer), the creation of a layer of neurons in the neural network
- Class 4 (Node) for the implementation of the AN class
- Class 5 (ANN), instantiating a network of artificial neurons

Fig. 8: Classes Java

RESULTS AND DISCUSSION

Test database: This database is composed of the learning database in addition to two other databases that make the test database:

- The images of the first test database is similar to the images of the tested database
- The images of the test database are one is similar to the learning database and the other is taken randomly

Figure 9 shows the evolution of the quadratic error in terms of the number of iterations. While the number of iterations is high then we get a good result (Table 1).

- BA: learning database
- T1: database similar to the existing DB
- T2: database taken randomly

We can see in line 2, we have 10 blocks of learning database (BA) from image 1 previ-ously seen in learning. These blocks are all well ranked in image 1, however, we have 9 blocks well ranked in image 1 and one misclassified in picture 4 (the last 10 blocks belong to the database T1). We get to know 8 blocks instead of 10 of the database T2, 8 well ranked and two misclassified. We got for the BA 100% of recognition but for the basic T1 and T2, we couldn’t reach that number because of the following problems:

- Sampling: the criteria, we have taken into consideration to select the blogs that participate in learning (we took just a small part of the picture randomly)
- Convergence rate: the accepted quadratic error
- Random initialization of W_i (Weights)
- Weights saturation problems of ANN because of the large amount of information treated
- Local minimum problem

Table 1: Experimental results

Texture	1	2	3	4	5	6	7	8	9	10
BA	**10**	0	0	0	0	0	0	0	0	0
T1	**10**	0	0	0	0	0	0	0	0	0
T2	**9**	0	0	0	0	0	0	1	0	0
BA	0	**10**	0	0	0	0	0	0	0	0
T1	0	**9**	1	0	0	0	0	0	0	0
T2	0	**8**	1	0	0	0	0	0	1	0
BA	0	0	**10**	0	0	0	0	0	0	0
T1	0	0	**10**	0	0	0	0	0	0	0
T2	0	0	**10**	0	0	0	0	0	0	0
BA	0	0	0	**10**	0	0	0	0	0	0
T1	0	0	0	**10**	0	0	0	0	0	0
T2	0	0	1	**9**	0	0	0	0	0	0
BA	0	0	0	0	**10**	0	0	0	0	0
T1	0	0	0	0	**9**	1	0	0	0	0
T2	0	0	0	0	**9**	0	1	0	0	0
BA	0	0	0	0	0	**10**	0	0	0	0
T1	0	0	0	0	0	**9**	0	0	1	0
T2	0	0	0	0	0	**8**	0	0	1	1
BA	0	0	0	0	0	0	**10**	0	0	0
T1	0	0	0	0	0	0	**10**	0	0	0
T2	0	0	1	0	0	0	**9**	0	0	0
BA	0	0	0	0	0	0	0	**10**	0	0
T1	0	0	0	0	0	0	0	**10**	0	0
T2	0	0	0	0	0	0	0	**10**	0	0
BA	0	0	0	0	0	0	0	0	**10**	0
T1	0	0	0	0	0	1	0	0	**9**	0
T2	0	0	0	0	0	1	0	0	**9**	0
BA	0	0	0	0	0	0	0	0	0	**10**
T1	0	0	0	0	0	0	0	0	0	**10**
T2	0	0	0	0	0	0	0	0	0	**10**

*Bold values in table shows the BA, T1 and T2 significant values

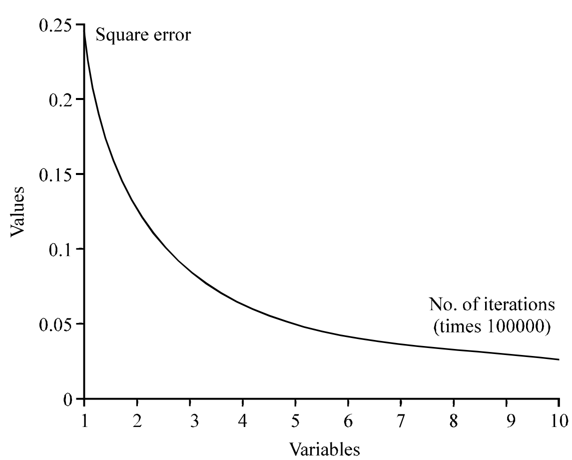

Fig. 9: Convergence graph

Soon, we will try to increase the recognition rate by testing with Helber Accelerator and solve the sampling problem as we will study the problem of W_i initialization.

CONCLUSION

As we have seen in the results, we reached a recognition order of 100% for the BA, 96% for T1 and 91% T2 which reflect the power of our approach. On the other hand, this approach research well if, we specify exactly textures as in medical imaging and if we were interested in a specific type. As perspective of this research, we will develop a new compression approach texture analysis.

REFERENCES

Besl, P.J. and R.C. Jain, 1986. Invariant surface characteristics for 3D object recognition in range images. Comput. Vision Graphics Image Process., 33: 33-80.

Brown, G., G. Michon and J. Peyriere, 1992. On the multifractal analysis of measures. J. Stat. Phys., 66: 775-790.

Chellappa, R. and R.L. Kashyap, 1985. Texture synthesis using 2-D noncausal autoregressive models. IEEE. Trans. Acoust. Speech Signal Process., 33: 194-203.

Fazia, A.M.R.O.U.N., 2013. [Extracting the textured component of an image]. Ph.D Thesis, Mouloud Mammeri University of Tizi-Ouzou, Tizi Ouzou, Algeria.

Galloway, M.M., 1975. Textural analysis using gray level run lengths. Comput. Graphics Images Process., 4: 172-179.

Haley, G.M. and B.S. Manjunath, 1995. Rotation-invariant texture classification using modified Gabor filters. Proceedings International Conference on Image Processing, Vol. 1, October 23-26, 1995, IEEE, New York, USA., ISBN:0-8186-7310-9, pp: 262-265.

Haralick, R.M., 1979. Statistical and structural approaches to texture. Proc. IEEE., 67: 786-804.

Jan, S.R. and Y.C. Hsueh, 1998. Window size determination for gran-ulemetricale structurel texture classication. Pattern Recognit. Lett., 19: 439-446.

Keller, J.M., S. Chen and R.M. Crownover, 1989. Texture description and segmentation through fractal geometry. Comput. Vision Graphics Image Process., 45: 150-166.

Mitchell, O.R., C.R. Myers and W. Boyne, 1977. A max-min measure for image texture analysis. IEEE. Trans. Comput., 26: 408-414.

Peet, F.G. and T.S. Sahota, 1985. Surface curvature as a measure of image texture. IEEE. Trans. Pattern Anal. Machine Intell., 7: 734-738.

Pentland, A.P., 1984. Fractal-based description of natural scenes.

Sarkar, N. and B.B. Chaudhuri, 1992. An efficient approach to estimate fractal dimension of textural images. Pattern Recognit., 25: 1035-1041.

Serra, J., 1982. Image Analysis and Mathematical Morphology. 1st Edn. Academic Press, London.

Tarik, H., M. Kodad and E.J. Miloud, 2014. Digital movements images restoring by artificial neural networks. Comput. Sci. Eng., 4: 36-42.

Unser, M., 1995. Texture classification and segmentation using wavelet frames. IEEE Trans. Image Process, 4: 1549-1560.