A Fingerprint Recognition Framework Using Artificial Neural Network

Ridouane OULHIQ, Saad IBNTAHIR, Marouane SEBGUI *, Zouhair GUENNOUN *

*LEC Laboratory *, EMI, Mohammed V University*
Rabat Morocco

(ridouaneoulhiq, saadibntahir)@student.emi.ac.ma, (sebgui, zouhair)@emi.ac.ma

Abstract— **Fingerprinting is one of the most used biometrics for people identification, it relays on image processing and classification algorithms. In this work we propose and test a framework that enables fingerprint detection using a set of image pre-processing algorithm. Concerning the features extraction, we propose the use of the number of bifurcations in image localities, and we propose the use of Artificial Neural Network (ANN) for the classification. The performance of our framework is evaluated for three different activation functions and show that we can reach an accuracy of 81%.**

Keywords—***fingerprint recognition; neural networks; minutiae extraction; image pre-processing; logistic sigmoid.***

I. INTRODUCTION

Biometrics is the science area that measures characteristics of life. Taking into account the results of biometrics, systems that are able to identify people are designed using either their physical characteristics (Fingerprints [1], Retina [2]) or behavioral ones (Voice [3], Handwriting [4]). The physical characteristics have the advantage of being unique and constant for everybody even for twins, while the behavioral are changeable throughout life time.

Among all biometric traits, fingerprints have one of the highest levels of reliability [5] [6] and have been extensively used by forensic experts in criminal investigations as in [7].

In our work we developed a framework that enables fingerprint identification using the Artificial Neural Networks (ANNs). ANNs can be configured and trained to adapt to the variations of fingerprints images of one's person; especially when fingerprint images are affected with noise and missing parts. This paper presents a comparison between different activation functions based on their performances in the two phases of training and validation. For each activation function, the optimum number of units in the hidden layer, which gives the best results in the recognition process, has been determined. Then minutiae extraction is a critical step that allow extracting true details from a bad quality input image. After this step of extracting minutiae, the image is divided into 64 zones and then modeled with a vector having the number of bifurcations in each zone, which will provide a basic fault tolerance to image rotation and translation. The vector is then considered as the input of the neural network.

The rest of this paper is organized as follows: Section II presents an overview of the related works. In section III we describe the proposed algorithms for image pre-processing and neural networks. Section IV presents our simulation and the performance results. We conclude our work in section V.

II. BACKGROUND AND RELATED WORK

A. Background

The study about the properties of fingerprints started at 16th century. In 20th century, fingerprint recognition was formally accepted as valid personal identification [8]. Since, fingerprints have become one of the most used methods for human recognition.

In the biometric process of finger scanning, a ridge is the curved line in the finger image. Some ridges are continuous curves, while others terminate at specific points called ridge endings. In other cases, two ridges come together at a point called a bifurcation (Fig. 1). Ridge endings and bifurcations are known as minutiae.

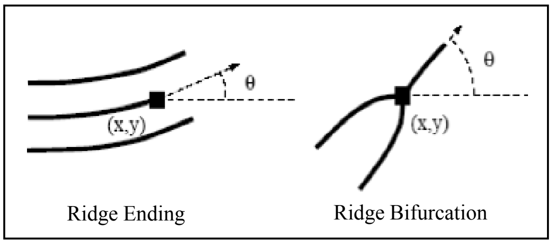


Fig.1: Minutiae points

There exist many different algorithms of fingerprint recognition. Some methods are based on minutiae matching, while others look for similarities in the bigger structure of the fingerprint. The performances of each system can be evaluated using the accuracy of the algorithm, print matching speed, robustness to noise and poor image quality.

According to [9], fingerprints matching systems are mainly divided into 3 parts: Correlation matching system, Minutiae matching system and Pattern matching system. It turns out that Minutiae system is the most popular and it is based on detecting certain unique characteristics on the individual's fingerprint called minutiae points. Several researchers try to devise better fingerprint matching algorithms than the previous ones. Many of them try to create new ways to match. Other researchers search to combine the existing matching algorithms to come up with a better one. And others like [10]

978-1-4799-7560-0/15/$31 ©2015 IEEE

modify the existing algorithms and matching methods to increase their performances.

B. Related work

The research work on fingerprints recognition based on neural networks has begun in the early 1990's, and since, it has become one of the most commonly used classifier for fingerprints classification systems. Researchers have developed many different neural based classification approaches.

The K-L transform is a neural approach that was developed by NIST (National Institute of Standards and Technology) for the FBI in 1990. In this approach, a massively parallel fingerprint classification system is described which uses image-based ridge-valley features, K-L transform, and neural networks to perform pattern level classification, Which is a problem of accurately classifying the fingerprints into one of five classes from an input fingerprint image. The five classes are Arch, Left Loop, Right Loop, Tented Arch and Whorl. The database used in this work is the NIST Special Database 4. The fingerprint's directional image is registered with respect to the center of the fingerprint image. The dimensionality of the orientation field is reduced using the K-L transform. Then, a probabilistic neural network (PNN) [11] is used in [12] to classify the feature vector.

Basing on the same database (NIST Special Database 4), authors in [13] used a classification scheme based on fingerprint feature extraction. A fuzzy-neural network classifier is used to implement the classification of input feature codes according to the well-known Henry system.

Another approach consists on using Wavelet. Authors in [14] used a feed-forward neural network with a single hidden layer, to classify features vectors consisting of 64 wavelet coefficients. The wavelet-based method used was sensitive to rotation and translation. However, this weakness was minimized by the correlation-based segmentation of the images and the former is less incident when patterns of thumbs are used.

As a different way for fingerprints classification, an unsupervised learning consisting on Self Organization feature Maps (SOM) is used in [15]. It is based on Kohonen algorithm and is trained to produce a low-dimensional and discretized representation of the input space of the training data. The basic SOM method is modified using a certainty parameter to deal with fingerprints having distorted regions. The features being used are directional images of fingerprints (ridge direction images).

III. Proposed Detection Framework

In our work we propose a framework, as represented in Fig.2 that consists in three main processing blocs, namely the pre-processing which is based on five steps, the features extraction which consists on minutia extraction and profile building and finally the classification or identification which uses an ANN.

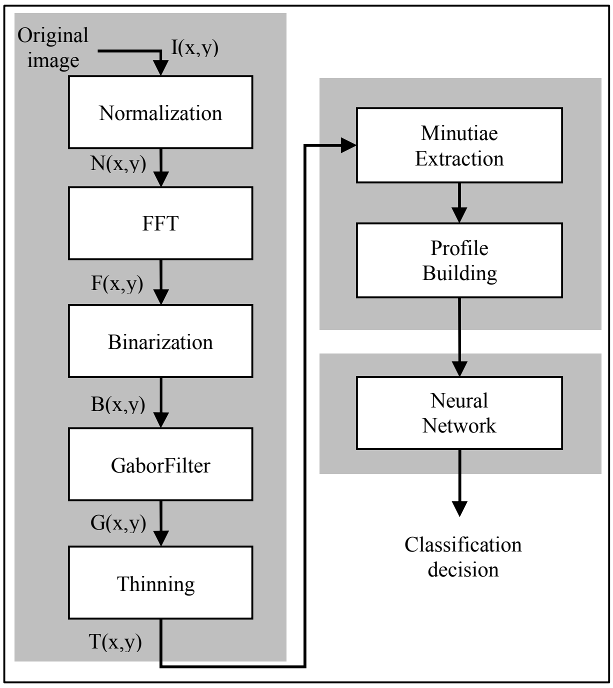


Fig.2: Fingerprint recognition system

A. Image Pre-Processing

Noise and poor image quality are the most causes of the fingerprint identification error. To improve detection results, the fingerprint pre-processing is needed. It allows transforming the original image Fig.3 (a) to a thinned image Fig.3 (f) which is used for minutiae extraction. It is a five-step process, namely normalization, enhancement, binarization, filtering and thinning.

1) *Normalization* [16]: The main purpose of normalization is to standardize the grey level values in fingerprint image so that it becomes within a desired range of values. The normalized image (Fig.3 (b)) is calculated using the equation :

$$N(x,y)=\begin{cases} M_0+\sqrt{\dfrac{VAR_0\times(I(x,y)-M)^2}{VAR}}, if \quad I(x,y)>M \\ M_0-\sqrt{\dfrac{VAR_0\times(I(x,y)-M)^2}{VAR}}, otherwise \end{cases} \quad (1)$$

N(x,y) is the normalized image I(x,y) is the original image. M_0 and VAR_0 are the desired mean and variance values respectively. While M and VAR are the real mean and variance of the input image.

2) *Fingerprints enhancement by Fourier Transform* [17]: The input image is first divided in blocks of m by n pixels, then the Fourier transform is applied to each block according to equation (2):

$$f(u,v)=\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}F(x,y)e^{-j2\pi\times(\frac{u.x}{m}+\frac{vy}{n})} \quad (2)$$

For x = 0,1,2,...,m-1 and y = 0,1,2,...,n-1.

The FFT enhancement takes into account the dominant frequency of the block. To obtain the enhanced block we multiply the FFT of the block by its magnitude a set of times according to the equation:

$$g(u,v) = F^{-1}\left[F(x,y)\times\left|F(x,y)^{k}\right|\right] \quad (3)$$

The k in formula (3) is an experimentally determined constant, which we set to k=0.45. The higher "k" is the more the algorithm improves the appearance of the ridges, filling up small holes in ridges. However, having too high value of "k" can result in false joining of separate ridges. Thus a termination might become a bifurcation.

Here F^{-1} *(F (x,y))* is computed using equation (4)

for $u = 0,1,2...\text{m-1}$ and $v = 0,1,2...\text{n-1}$.

$$F(x,y) = \frac{1}{mn}\sum_{u=0}^{m-1}\sum_{v=0}^{n-1} f(u,v)e^{j2\pi\times(\frac{u.x}{m}+\frac{vy}{n})} \quad (4)$$

The advantage of the FFT is to connect some falsely broken points on ridges and to decrease some of background noise like false connections between ridges.

3) *Binarization* [18]*:* The level of gray in the fingerprint image is more related to the image quality and does not have considerable effect on person identification. In the next step, the 256-level of gray is transformed into a black and white with no loss of information.

For this purpose, a global intensity threshold should be defined for the gray image, then a comparison between the local gray-value of each pixel with the threshold is made. The pixel value turns to 0 (Black) if it's less than the threshold and to 1 (white) if it is greater than the threshold. By the end of this process, all pixel values within the image are either zero or one, and the image has been converted to binary format. Practically a value of 1 is corresponding to the information (ridges) and a value of 0 to the background (0 for black, 1 for white).

$$B(\mathrm{x},\mathrm{y}) = \begin{cases} 1 & ,if \quad F(x,y) < T \\ 0 & ,otherwise \end{cases}$$

Where T is the threshold gray value of $F(x,y)$.

4) *Gabor Filter* [19]*:* To remove noise and preserve true ridge and valley structures, the Gabor filter is used. This later is a Gaussian function (with variances s_x and s_y along x and y-axes respectively) modulated by a complex sinusoid. It is used to remove noise and preserve true ridge and valley structures.

The Gabor filter is described by the following equation:

$$g(x,y,f,\theta) = k.e^{\frac{-1}{2}\times d\times M(x,y,f,\theta)} \quad (5)$$

Where:

$$\bullet\, k = \frac{1}{2\times\pi\times s_x\times s_y}$$

$$\bullet\, d = (\frac{x}{s_x})^2 + (\frac{y}{s_y})^2 \quad (6)$$

$$\bullet\, M(x,y,f,\theta) = \cos(2\times\pi\times f\times(x\times\cos(\theta)+y\times\sin(\theta)))$$

Where s_x and s_y are the variances along x and y-axes respectively, f is the frequency of the sinusoidal function and θ is the orientation of Gabor filter.

5) *Thinning* [20]: The objective in this step, is to eliminate the redundant pixels in ridges till it becomes one pixel wide. In each scan of the full fingerprint image, the algorithm marks down redundant pixels in each small image window (3x3).

The algorithm used is as follows:

1. Set the center point as P and the eight neighborhood points clockwise around P as P1, P2... P8

If the following four conditions are satisfied at the same time, point P is deleted.

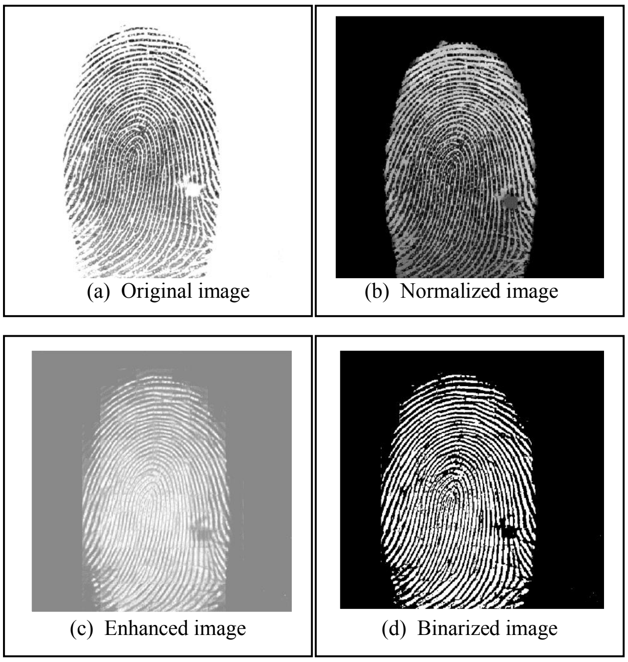
(a) 2⩽N(P)⩽6 (b) S(P)=l (c) P1*3*P5=0 (d) P3*P5*7=0

Where N(P) is the number of the non-zero points around P, and S(P) is the frequency of changes from 1 to 0 according to the sequence of P1, P2... P8, P1. All the points that satisfy the conditions above are marked and deleted when all boundary points are checked.

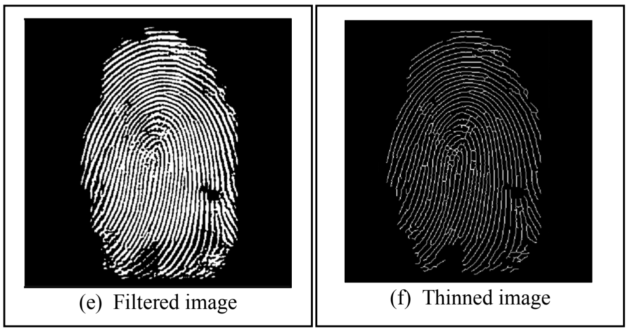
2. As for the first one, except that conditions (c) and (d) have changed as:

(c)P1•P3•P7=0 (d) P1•P5•P7=0

Several iterations should be performed until there is no point meeting these conditions. The ridge's skeleton is made up of the left points.



(a) Original image
(b) Normalized image
(c) Enhanced image
(d) Binarized image



(e) Filtered image (f) Thinned image

Fig.3: Results of the pre-processing steps

B. Features Extraction

1) *Minutiae Extraction*: For the minutiae extraction, we use the cross number method [20] which generally includes the endpoint or bifurcation. The bifurcations and ridges ending are extracted from the thinned image by scanning the local neighborhood of each ridge pixel in the image using a 3×3 window. The value of the cross number Cn is calculated using equation (7):

$$Cn = \frac{1}{2}\sum_{1}^{8}\left|P_{i-1} - P_i\right| \qquad (7)$$

If Cn=1, point P is an endpoint.
If Cn=3, point P is a bifurcation.

Because the ridges ending are sensitive to noise, we choose to use the bifurcations for identification.

1) *Profile Building:* To build the profile corresponding to the image, the fingerprint image is divided to a grid of 64 zones (Fig.4) and then the number of the bifurcations in each zone is calculated, which gives the system a basic fault tolerance to image rotation and translation. Each neuron of the input layer corresponds to the number of bifurcations (the normalized value) in the equivalent zone.

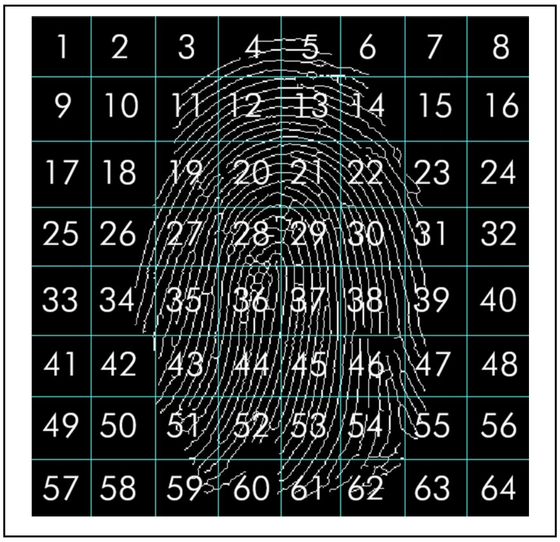


Fig.4: The thinned image divided to 64 zones

C. Classification/Identification

Artificial Neural Networks (ANNs) offer many advantages like adaptive learning, fault tolerance and generalization.

The objective of this step is to train an ANN to perform classification of different fingerprint images as belonging to their owners. When the learning is achieved the algorithm will enable identification.

The feedforward backpropagational neural network is one of the most popular neural networks (Fig.5), because it can be applied to many different tasks. The term feedforward describes how this neural network works. In a feedforward neural network, neurons are only connected forward. Each neuron of a layer is connected to all the other neurons of the next layer, but there are no connections back. The second term backpropagation describes how this type of neural networks is trained. An iterative weight-adjusting scheme is used to propagate backward the error term by modifying the weights for all the connections in the neural network structure. In Fig.5 we define:

$(U_1 ... U_I)$: vector of input, $U_{I+1} = 1$ is the bias unit.
W: Matrix of weights of the neural network.
$(S_1 … S_K)$: vector of output for classification
I: number of units in the input layer (without the bias)
J: number of units in the hidden layer (without the bias)
K: number of units in the output layer

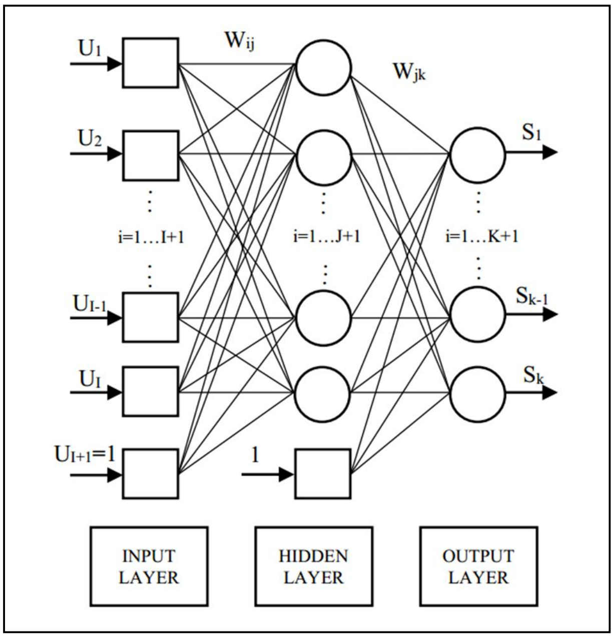


Fig.5: Schematic representation of a neural network

The number of hidden layers was chosen to be 1 and the range J between 2 and 120, so that we can evaluate the performance of the neural network according to the number of units in the hidden layer.

Each neuron of the neural network has an activation function which specifies the output of a neuron to a given input. In this paper, we tried to test different activation functions.

IV. SIMULATION

A. Simulation results

Coming to simulation, a database of 384 fingerprints of 504×480 pixels was used; 336 fingerprints for training and 48 fingerprints for validation.

The database is constructed of 8 persons, 6 fingers for each person and 8 scans for each finger.

The output layer has been set to 8 neurons (8 persons) of binary values (0 or 1) so that only one output is activated (i.e. its value equal to 1) for each input vector.

For our experiments, we use the values:

M	VAR	m	n	S_x	S_y	f	θ
127	2000	32	32	4.24	4.74	0.1001	45°

For analysis sake, a comparison between different activation functions has been done by calculating the performance ratio for different numbers of hidden layers, in order to find the optimum number that gives the best performance for each activation function:

$$Performance = \frac{\text{NPEI}}{\text{NAI}} \tag{8}$$

Where *NPEI* is the number of persons effectively identified and *NAI* is the number of all individuals.

The activation functions tested in the simulation are:

- Hyperbolic sigmoid Activation Function :

$$y = \frac{1}{2} \times \frac{1-e^{-2x}}{1+e^{-2x}} + \frac{1}{2} \tag{9}$$

The Hyperbolic function produces the 1 and 0 output values for minus and plus infinity, respectively.

- Elliot Activation Function :

$$y = \frac{x/2}{1+|x|} + \frac{1}{2} \tag{10}$$

The Elliot activation function [21] is higher-speed approximation of the Hyperbolic Tangent function. For saturation limits the output range is 0 to 1.

- Logistic Sigmoid Activation Function :

$$y = \frac{1}{1+e^{-x}} \tag{11}$$

The sigmoid function produces positive numbers between 0 and 1. It is one of the most used activation functions.

Simulation is done using OCTAVE [22], through the command line interface. Our computation results are done as follow:

1. We set the activation function
2. We define the variation interval for J the number of neurons in the hidden layer. In our case we use [2,120]
3. The training and the test are performed for each J. The performances are computed using formula in equation (8)

1) Training:

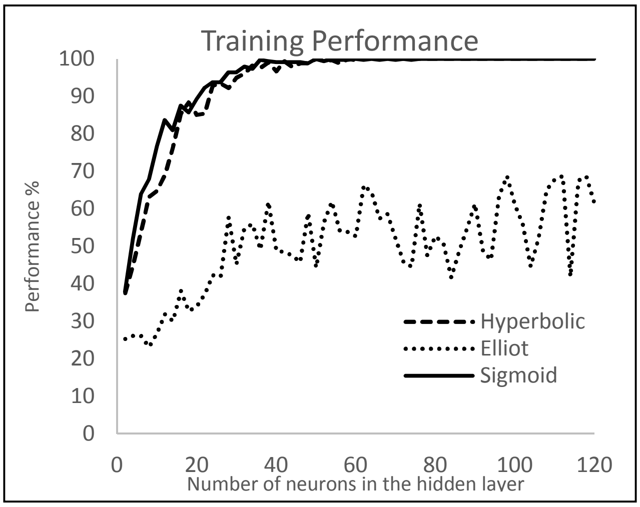


Fig.6 System performance at the training phase

Fig.6 allows us to compare the minimal needed number of neurons in the hidden layer that ensure a perfect learning (100% performances). The lowest value, which is approximately 40 neurons, is obtained for the logistic sigmoid function. In other words, the logistic sigmoid function offers the best learning ratio (performance/number of neurons) followed by the hyperbolic sigmoid function, which leads the close performances, while the Elliot function tend to perform a random and unstable learning.

2) Test:

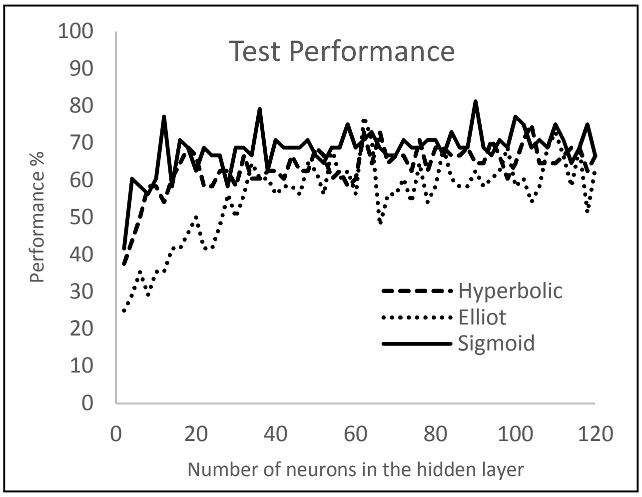


Fig.7 System performance at the testing phase

In the test phase, the logistic sigmoid function presents once again the best performance comparing to other functions, it got approximately the same appearance as the hyperbolic sigmoid function, but with higher performance, while the Elliot presents the lowest test performance. Concerning the Sigmoid function, if we consider a number of neurons in the hidden layer higher than 40 (which is approximately the minimum number needed for a perfect training) the performance curves vary between 64.58% and 81.25% with an average of 70.02%.

B. Comparison

Even though our results are underperformed, the system proposed in this paper, is not just for a classification of fingerprints into the five classes (Arch, Left Loop, Right Loop, Tented Arch and Whorl), but it aims to identify a person using six fingers with a basic fault tolerance to image rotation and translation and efficiency against noise and missing parts in the fingerprints.

Work	**Database**	**Technique**	**Classifier**	**Best perf.**
[12]	NIST Special DB 4	KL transform	ANN	88%
[13]	NIST Special DB 4	Fuzzy logic	ANN	98.5 %
[14]	Not specified	Wavelet	ANN	94.3 %
[15]	1000 fingerprints (500 pairs)	SOM	Kohonen	90.2 %
This work	8 scans 6 fingers 8 persons	Image preprocessing	ANN	81.3 %

V. Conclusions

The preprocessing block enhances the fingerprint's images so that the minutiae (bifurcations) can be extracted. Using the positions of bifurcations in the image, we build, for each image, a profile that consists of a vector, this later is the input for the ANN. For each of the activation functions chosen (Logistic Sigmoid, Elliot and Hyperbolic Sigmoid), we performed the forward-propagation and the back-propagation for a set of number in the hidden layer [2,120]. Comparing the performances, in both training and test, we came to the conclusion that the Logistic sigmoid presents a quick learning ratio, it also presents the highest performance for a number of units in the hidden layer higher than 40 neurons. This performance reaches 81.25%.

In future works we plan to evaluate the effect of different profile building algorithms such as polar coordinates instead of Cartesian ones.

References

[1] Jain, L.C. et al. (Eds.). 1999. Intelligent Biometric Techniques in Fingerprint and Face Recognition. Boca Raton, FL: CRC Press.

[2] Retina and Iris Scans. Encyclopedia of Espionage, Intelligence, and Security. Copyright © 2004 by The Gale Group, Inc.

[3] Person Authentication by Voice: A Need for Caution. Jean-Francois Bonastre, Frédéric Bimbot, Louis-Jean Boe, Joseph P. Campbell, Douglas A. Reynolds, Ivan Magrin-Chagnolleau (September 2003). 8th European Conference on Speech Communication and Technology. Geneva, Switzerland: isca-speech.org.

[4] Chapran, J. (2006). "Biometric Writer Identification: Feature Analysis and Classification". International Journal of Pattern Recognition & Artificial Intelligence: 483–503.

[5] J. Berry, "The history and development of fingerprinting," in Advances in Fingerprint Technology, H.C. Lee and R.E. Gaensslen, Eds., pp. 1–38. CRC Press, Florida, 1994.

[6] E. Newham, "The biometric report," SJB Services, 1995.

[7] Federal Bureau of Investigation, the Science of Fingerprints: Classification and Uses, Washington, D.C., 1984, U.S. Government Printing Office.

[8] History of Fingerprint, http://www.crimtrac.gov.au/our_services/ History of Fingerprinting.html.

[9] Neeraj Bhargava, Ritu Bhargava, Manish Mathuria, Minaxi Cotia, "Fingerprint Matching using Ridge-End and BifurcationPoints" , International Conference in Recent Trends in Information Technology and Computer Science (ICRTITCS - 2012)Proceedings published in International Journal of Computer Applications® (IJCA) (0975– 8887)

[10] Hemlata Patel, Mr. Vishal Sharma Computer Science & Engg. Department, Jawaharlal Institute of Technology Borawan, Khargone (M.P.) India." Fingerprint Recognition by Minutiae Matching Method for Evaluating Accuracy", International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue5-May 2013

[11] https://en.wikipedia.org/wiki/Probabilistic_neural_network

[12] Wilson CL, Candela GT, Grother PJ, Watson CI, Wilkinson RA,"Massively parallel neural network fingerprint classification system". National Institute of Standards and Technology; NISTIR 4880, 1992

[13] Mohamed S, Nyongesa H,"Automatic fingerprint classification system using fuzzy neural techniques". In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Washington, DC, April, 2002

[14] Neto HV, Borges DL," Fingerprint classification with neural networks". In: Proceedings of the 4th Brazilian Symposium on Neural Networks, Goiania, Brazil, 3-5 December, 1997

[15] Halici U, Ongun G,"Fingerprint classification through self-organizing feature maps modified to treat uncertainties". Proc IEEE 84(10):1497)1512, 1996

[16] Ching-Tang Hsieh* and Chia-Shing Hu, A Fingerprint Identification System Based on Fuzzy Encoder and Neural Network. Tamkang Journal of Science and Engineering, Vol. 11, No. 4, pp. 347355(2008).

[17] Fingerprint Recognition using MATLAB Graduation project - Zain S. Barham Supervised by: Dr. Allam Mousa.

[18] Fingerprint Matching using Ridge-End and Bifurcation Points - Neeraj Bhargava, Ph.D. - Ritu Bhargava, Ph.D. - Manish Mathuria - Minaxi Cotia.

[19] Ahmad Poursaberi, Faulty of Engineering, Electrical & Computer Department, Tehran University, Iran, June 2004

[20] Jing WANG, Jingcui LI, Liulin CAO, An Improved Fast Thinning Algorithm for Fingerprint Image and Its Application. Journal of Computational Information Systems 7: 7 (2011) 2285-2292.

[21] P.SIBI, S.ALLWYN JONES, P.SIDDARTH, Student, SASTRA University, Kumbakonam, India, "Analysis of different activation functions using back propagation neural networks", Journal of Theoretical and Applied Information Technology 31st January 2013. Vol. 47 No.3.*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[22] GNU Octave, http://en.wikipedia.org/wiki/GNU_Octave.