Journal of Intelligent Learning Systems and Applications, 2016, 8, 39-49
Published Online May 2016 in SciRes. http://www.scirp.org/journal/jilsa
http://dx.doi.org/10.4236/jilsa.2016.82004

Fingerprint Recognition with Artificial Neural Networks: Application to E-Learning

Stephane Kouamo, Claude Tangha

Department of Computer Science, University of Yaounde I, Yaounde, Cameroon
Email: skouamo@gmail.com, ctangha@gmail.com

Received 7 February 2016; accepted 27 May 2016; published 30 May 2016

Abstract

Fingerprint recognition is a mature biometric technique for identification or authentication application. In this work, we describe a method based on the use of neural network to authenticate people who want to accede to an automated fingerprint system for E-learning. The idea is to apply back propagation algorithm on a multilayer perceptron during the training stage. One of the advantages of this technique is the use of a hidden layer which allows the network to make comparison by calculating probabilities on template which are invariant to translation and rotation. Results come both from the NIST special database 4 and a local database, and show that a proposed method gives good results in some cases.

Keywords

Neural Networks, Pattern Recognition, Fingerprint, Back-Propagation, E-Learning

1. Introduction

Usual identification methods based on what we have (*magnetic card*, *identity card*, *chip*, etc.) or on what we know (*password*, *encoding*, etc.) cause big problems of reliability [1] [2]. The recent terrorist acts and threats to many countries, coupled with the growing evolution of cybercrime in a world where all tasks tend to become automatic identification/authentication of flawless natural persons have become a problem crucial for security reasons (border controls, access to public or virtual sites, transport, ...). All these problems have encouraged the further development of biometrics as an identification and authentication tool. The use of fingerprints is one of the oldest and most successful methods of identification/authentication biometrics [3] [4]. Concerning a fingerprint, F. Galton was the first to prove the existence of the papillary drawings from birth to death [5]; this particular arrangement of the papillary lines forms points called *minutiae*, which are the cause of uniqueness and

How to cite this paper: Kouamo, S. and Tangha, C. (2016) Fingerprint Recognition with Artificial Neural Networks: Application to E-Learning. *Journal of Intelligent Learning Systems and Applications*, **8**, 39-49.
http://dx.doi.org/10.4236/jilsa.2016.82004

immutability of the fingerprint.

The digital fingerprint forms a specific class of pattern with a particular singularity and statistics characteristics known. Then, fingerprint recognition seems to be more constraining than the other problem of pattern recognition (like handwritten character recognition) where neural network has been well applied [6]-[8].

The main question we therefore consider here is: how to automatically authenticate flawless people who wish to access a computer system (or electronic) with the lowest possible error rate? The approach consists in the use of neural networks to develop such a system. The built model will then be applied to E-learning with a relatively stable database. In fact, neural network has shown their ability to deal with many domains. Their peculiarity is the ability to adapt to the data to be processed that they offer, and the ability to perform calculations in parallel, allowing them to work in various fields of application.

In this paper, we present a model of authentication by fingerprints, based on the use of a neural network having a multilayer perceptron structure and extraction algorithm and classification minutiae at two levels.

The first level consists of the fingerprint classification at the input layer considering its general topography according to the classification of Henry [9], and is divided into two stages: the first stage permits the classification of the input fingerprint according to the prototypes of the first class of constructed data by simple calculation of probabilities, exactly as done by Baldi [4]; the second stage serves as a control of the first classification and reduces the risk of incorrect answers. This second control can reduce the rate of false rejection, even when the number of detected minutiae is low.

The second level which is actually devoted to correspondence or authentication is to compare in the network training phase, the feature vector of the input fingerprint (which is invariant to translation and rotation) to all those belonging to the class winner after the two previous classifications, by a simple similarity measure (the Euclidean norm for example).

The rest of the document is organised into three sections. The first section presents algorithms of neural network for fingerprint recognition. The second section describes the proposed method based on multilayer perceptron with two hidden layers. The third section presents the application of studied methods and obtained results; we finish by a conclusion.

2. Fingerprint Recognition and Neural Network

From a technical point of view, there are two different problems in the field of fingerprint analysis, namely: classification and correspondence. The classification consist in putting together some fingerprints that have templates that are similar in predefined subclasses. Today, there are five classes of fingerprints defined by Henry [9]. Correspondence implies authentication and identification, is based on other more subtle elements (minutiae, streaks' orientations) and requires a much more detailed analysis of the fingerprint image. The whole picture is no longer taken into consideration here, but a set of elements which as a whole constitutes the imprint template or the feature vector: This is the minutiae and their guidelines when making the correspondence. This technique involves the use of pre-processing (binarization, thinning, filtering, etc.) that will enable the refinement of the image to extract the most useful features [10] [11]. As part of this work we will present algorithms that deal exclusively with the appearance of correspondence, the classification being standardised.

2.1. Comparison Based on Correlation (Minimum Distance)

This method aims simply to compare pixels array of two fingerprint images and to calculate the correlation of that pixels [12] [13]. For two fingerprint images to be compared, this method just compares the minimum distance between the mean of data pixels of each pattern. The technique based on the correlation of pixels is not very effective, because we can get many different images for one fingerprint.

Remarks

- If images of the same fingerprint are different, the values of their pixels are automatically different;
- In addition two images taken at different resolutions (from two different scanners) may not have the same pixel values.

For this method, we can give a threshold (for example if the difference between the values of the two pixels is less than 3, it is considered that these pixels are equal); but this technique is clearly not rigorous.

2.2. Comparison Based on Minutiae (Image Mapping)

Comparison based on minutiae (Image mapping) it's the most widely used method because it's based on individual features in each person, the *minutiae* [14] [15]. Minutiae are extracted from two prints and are represented as a set of points in a two dimensional plans depending on the model coordinates (type of minutiae, coordinates, angle). The comparison is to find a proper alignment of the two fingerprint minutiae (*F*1 and *F*2) which produces a maximum of pairs of similar minutiae.

By taking two fingerprints *F*1 and *F*2 to be compared as vectors of minutiae, wherein each *m* is represented by a vector (x_i, y_j, θ), where x_i and y_j are coordinates of the location of the minutiae in the fingerprint and θ the direction of the minutiae. We consider that the distance *sd* between two minutiae of two different fingerprint must be less than a level r_0 and the difference *dd* of their direction should be less or equal to an angular tolerance θ_0:

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \le r_0 \tag{1}$$

and

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|) \le \theta_0. \tag{2}$$

Remarks

- Computation number increase with vector number already classified;
- Classification result depends on a critical level (two fingerprints are identical when a number *t* of identical minutiae is reached, this number can vary from one system to another depending on the degree of security chosen by the designer).

2.3. Probabilistic Neural Networks

The algorithm described by Baldi and Chauvin [16] that exclusively deals with the aspect of correspondence consists of two main steps, namely: a preprocessing step and a decision step. The preprocessing stage basically aligns the two images and extracts, from each one of them, a central region. The two central regions are fed to the decision stage, which is the proper neural network part of the algorithm and subject to training from examples. Whereas the preprocessing stage is fairly standard, the decision stage is innovative and based on a neural network that implements a probabilistic Bayesian approach to the estimate of the probability p of a match. The network is formed by the descent of the gradient using a training set of n pairs of images from several different fingers. Adding some additional fingers to make using phase strengthens the robustness of the neural network created and tested beforehand. Given two fingerprint images *A* and *B*, the proposition that they match (or do not match) will be denoted by $M(A, B)$ [or $\bar{M}(A, B)$]. The purpose then is to design a neural network algorithm that when presented with a pair (*A*, *B*) of fingerprint images outputs a number $p = p(M) = p[M(A, B)]$ between 0 and 1 corresponding to a degree of confidence or probability that the two fingerprints match.

2.3.1. Advantages

- Use of a central region of the image with alignment and data compression;
- Calculation of probabilities to make the correspondence between two patterns provided at the network entry after several successive filtering;
- The algorithm is adapted for authentication of a person who wishes to access a system in which its fingerprint is previously known;
- The algorithm is suitable for systems using a small database.

2.3.2. Limitations

- Use complex formulas for the calculation of probabilities;
- Usable for a limited number of fingerprint's pictures including the learning phase.

2.4. Locally Connected Neural Network

For this algorithm, pattern recognition systems typically consist of two stages: construction of feature vectors and classification of pattern [17].

1) In the first step, a fingerprint is passed in the system entry and is reduced to a feature vector (which is invariant to translation and rotation), this throws operations of transformation (calculating an activation for each neurone of the network and a global energy). After having extracted feature vectors, prototype vectors are formed by subclass of data.

2) In the second step, a classification or direct comparison which led to the recognition or not of the pattern is performed between the input feature vector and the prototypes of the database. This method based on the locally connected neural network is more appropriate for verification applications, security and identification. Mathematically, the network activation is iteratively computed by [18]:

$$\alpha_{ij}(n+1) = \frac{I_{ij} + \sum_{kl\in N_r(ij)} \left[w_{ij;kl}\,\alpha_{kl}(n) \right]}{\sum_{ij} \left(I_{ij} + \sum_{kl\in N_r(ij)} \left[w_{ij;kl}\,\alpha_{kl}(n) \right] \right)} \cdot E \tag{3}$$

where:

- $kl \in N_r(ij)$ are the coordinates kl of a point that falls within a radius r of the neighbourhood of neurone *ij*;
- $\alpha_{kl}(n)$ is the current activation level of neurone *kl* in $N_r(ij)$;
- $w_{ij;kl}$ is the weight of the synaptic connection between neurone *ij* and neurone *kl*;
- I_{ij} is the input pattern pixel value at location *ij*;
- E is the global network energy constant;
- n is the iteration number.

The peculiarity of this technique is the use of feature vectors that are invariant to the translation and rotation of the fingerprint image. This technique is based on three classification algorithms which are: average classes, nearest neighbour and k-nearest neighbours.

2.4.1. Advantages

- Use of prototypes (feature vectors of fingerprint by subclass of data) invariants to translation/rotation to make correspondence;
- This algorithm is suitable for identification of person in a great set of data;
- Use of nearest neighbours to perform the classification.

2.4.2. Limitations

- The learning time is high;
- Relatively high rate of false rejections;
- Using thresholds and activation calculations to perform the correspondence.

3. A Two Hidden Layers Perceptron for Fingerprint Recognition

By leveraging the benefits of the previously mentioned methods, we have developed a new technique based on the use of a multilayer perceptron. The proposed structure of the neural network (see **Figure 1**) will be that of a locally connected neural network, which uses the calculation of probabilities to classify the input image subblocks considering the henry's classification. We also use a threshold (to determine whether there is matching or not between two patterns) and the retro-propagation algorithm to train the network.

The neural network used here is a multilayer perceptron with an input layer, a hidden layer which itself contains three layers (two classification layers and a correspondence layer) and finally an output layer (see **Figure 1**). The principle is the same like a usual recognition method based on the use of neural network and is based on two main steps: *training* and *recognition stage* [19]. Before applying the training stage, we classify the entire sample of data into four main classes (*arch*, *whorl*, *left* and *right loop*) according to the *Henry's classification* and we determine two prototypes for each class of data.

After this strict classification of the database and the determination of a $K1$ *prototypes* for the 1st and 2nd *classification layer*, the training stage can be proceeded as follow: the three layers of our network are strongly connected among its. *Input layer* is connected to the 1st *classification layer*, the 1st *classification layer* is connected to the 2nd *classification layer*, the 2nd *classification layer* and *correspondence layer* are connected to-

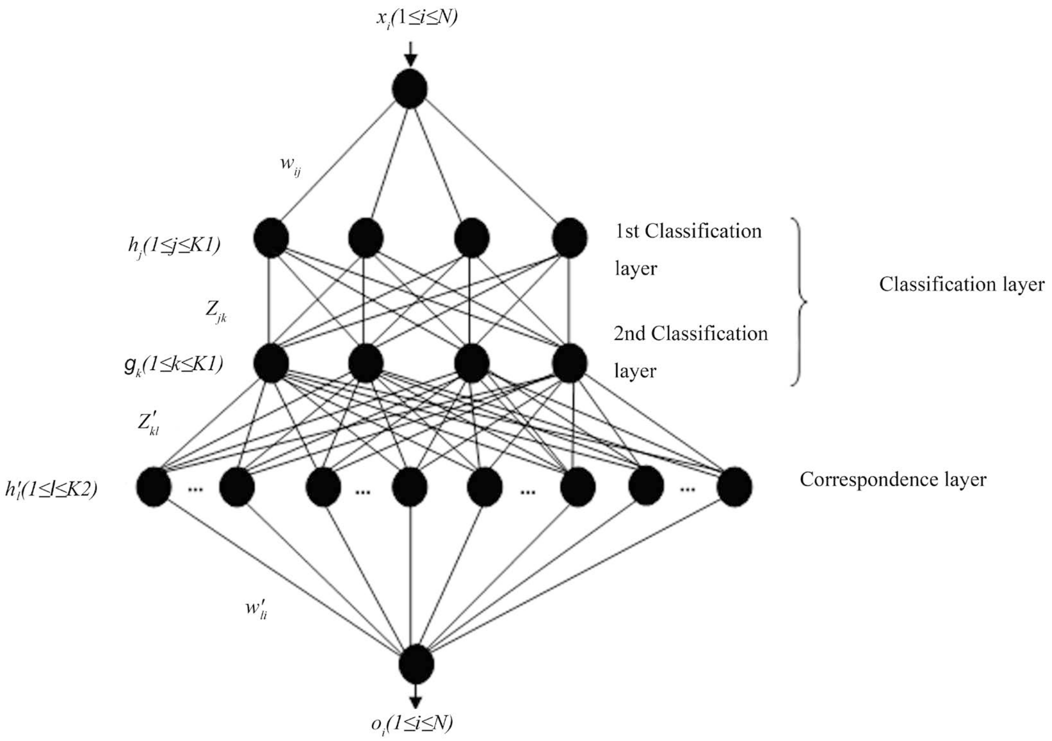

Figure 1. Structure of the proposed neural network. Where: x_i represents the input image vector while o_i is the expected output; h_j are prototypes of each class of data (according to Henry classification); w_{ij} the connection weight between the input layer and the 1st classification layer; Z_{jk} the connection weight between the 1st and the 2nd classification layer; g_k are prototypes of the second classification layer; z'_{kl} the connection weight between the 2nd classification and the output layer; h'_l are the vectors of the entire database for the correspondence layer; w'_{li} the connection weight between the correspondence layer and the output layer.

gether and then, the *correspondence layer* is connected to the *output layer* by connection weights. The N input fingerprint images to be recognised (after pretreatment operations), is applied at the input layer of our neural network. Connection weights of the 1st and 2nd *classification layer* are respectively initialised by $K1$ (here $K1 = 4$ because of four main classes) prototypes of every data class already known according to the fingerprint class made. Then, connection weights of the correspondence layer are initialised by all the $K2$ expected fingerprint image of the database for the training stage for each class of data determined above. So $K2$ is the number of pattern of a data class, for example is the winner neurone after the two classification belong to the class of whorl then $K2$ will be equal to the number of pattern of the whorl class.

The input vector is compared to all prototype of every data class of the 1st and 2nd classification layers, and it's assigned to the class of winner neurone (the second classification serve as control of the first and help to reduce error rate). After this, the input vector is compared to all data of the class of the winner neurone at the correspondence layer, and it's assigned the winner neurone too. If the winner neurone is not the prototype of the right class, committed errors are back propagated. The output layer gives us the activated winner neurone.

Mathematically, proposed algorithm can be applied like this:

1) Choose the initial prototypes two by two for each data class, according to the classification of Henry.
2) Initialise neurone of the first classification layer by the first prototype of each data class.
3) Initialise neurone of the second classification layer by the second prototype of each data class.
4) Initialise every correspondance layer neurone weight with the database pattern class by class.
5) Apply an input vector.
6) Apply the similarity measure on the first classification layer by probability calculation.

$$h_j = H\left(P\left(M\left(x_i, w_{ij}\right)\right)\right) \quad 1 \le i \le N, 1 \le j \le K1,$$

P() represent a set of featured vectors obtained after pretreatment operations;
H() is a combination of featured vectors and their minutiae orientation.
7) Select the winner neurone.
8) Apply the similarity measure on the second classification layer.

$$g_k = \sqrt{\sum_{j=1}^{K1}\left(h_j - z_{jk}\right)^2} \quad 1 \le k \le K1,$$

9) Select the winner neurone.
10) Apply the similarity measure on the correspondance layer (if the winner is in the right class).

$$h'_l = \sqrt{\sum_{k=1}^{K1}\left(g_k - z'_{kl}\right)^2} \quad 1 \le l \le K2,$$

11) Select the winner neurone.
12) Calculate the error between the expected and the obtained output.

$$o_i = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(h'_l - w'_{li}\right)^2} \quad 1 \le l \le K2,$$

13) Check the thresholds.
14) Modify connection weight of winners neurone:

$$w_{ij}\left(n+1\right) = \frac{x_i + \sum_{ij}\left[w_{ij} \cdot \alpha_{kl}\left(n\right)\right]}{\sum_{jl}\left(x_i + \sum_{li}\left[w'_{li} \cdot \alpha_{kl}\left(n\right)\right]\right)} \cdot E$$

15) Update prototypes of winners neurone:

$$m^l_{n+1} = \frac{m^l_n + x^l_{n+1}}{n+1}$$

16) Return to (5) till finishing input vector.

4. Applications and Results

Application fields of biometrics are many and diverse (criminal police, airport control, election system, medical file, ...). We apply our work in particular to an online learning system (E-learning) whose purpose is to enable secure access, including learners located at long distances from the place of learning [20]. E-learning is a learning mode that takes advantage of the use of technology of information and communication at all levels of training activity. It particularly refers to a training system whose main purposes can be defined as: independent learning, distance learning, individualised learning paths and development of pedagogical relationships online.

To compute different training stage and tests, we use: a laptop Intel? CoreTM i52450M CPU@ 2.50 GHz 2.50 GHz with 4Go RAM; a desktop Intel Pentium CPU G645@ 2.90 Hz 2.90 GHz with 2Go de RAM; the OS Ubuntu 12.10 and GNU Octave 3.6.1 and the gcc 3.4 compiler with OpenMPI library.

The model thus constructed is performed using two databases:

- A database (called BDAL) consisting of 500 fingerprint images sized 248×338 with 8-bit grayscale, from 100 different people (all being E-learners especially trained and cooperatives which permits to avoid having latent images and very noisy) with 5 pictures of fingerprints per person (it's the same fingerprint is taken into account);
- International NIST database 4 which is a specific database for fingerprint recognition; it consists of 2000 pairs of fingerprint images in the format 512×512 with 8-bit grayscale [21].

Performances indicators are:

- The Recognition Rate (or Acceptation Rate) $\text{AR} = \frac{M}{N}$.
- The Error Rate constituted of: False Rejection Rate (FRR) and False Acceptation Rate (FAR)

$$\text{FRR} = \frac{R}{N},\ \text{FAR} = \frac{A}{N}$$

- The training and recognition time.

Table 1 and Table 2 below show training results, while Table 3 and Table 4 show obtained testing results with both NIST and BDAL database.

Table 1. Training results with images sub-blocs sized of 8 × 8.

	Training stage with 80% of NIST's data					
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training time	No	66 min	48 min	54 min	62 min	74 min
Recognition time	45 s	5 s	4 s	5 s	9 s	11 s
AR (en %)	91.8	94.6	49.8	88.3	92.7	94.7
FRR (en %)	7.7	4.8	17.3	11.3	6.7	4.8
FAR (en %)	0.5	0.6	32.9	0.4	0.6	0.5
	Training stage with 80% of BDAL's data					
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training time	No	22 min	16 min	18 min	21 min	25 min
Recognition time	29 s	3 s	2 s	3 s	7 s	9 s
AR (en %)	91.4	94.2	49.9	87.9	92.3	94.3
FRR (en %)	8.0	5.1	17.6	11.6	7.0	5.1
FAR (en %)	0.6	0.7	33.0	0.5	0.7	0.6

Table 2. Training results with images sub-blocs sized of 16 × 16.

	Training stage with 80% of NIST's data					
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training time	No	46 min	28 min	46 min	48 min	58 min
Recognition time	41 s	5 s	4 s	5 s	9 s	11 s
AR (en %)	91.5	94.3	49.5	88.0	92.4	94.4
FRR (en %)	8.0	5.1	17.6	11.6	7.0	5.1
FAR (en %)	0.5	0.6	32.9	0.4	0.6	0.5
	Training stage with 80% of BDAL's data					
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training time	No	15 min	9 min	15 min	16 min	19 min
Recognition time	27 s	3 s	2 s	3 s	7 s	9 s
AR (en %)	91.1	93.9	49.1	87.6	92.0	94.0
FRR (en %)	8.3	5.4	17.9	11.9	7.3	5.4
FAR (en %)	0.6	0.7	33.0	0.5	0.7	0.6

Table 3. Testing results with images sub-blocs sized of 8 × 8.

	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training stage with 80% of NIST's data						
Recognition time	26 s	4 s	3 s	4 s	6 s	7 s
AR (en %)	92.4	95.2	51.8	89.3	93.3	95.3
FRR (en %)	7.1	4.2	15.3	10.3	6.1	4.2
FAR (en %)	0.5	0.6	32.9	0.4	0.6	0.5
Training stage with 80% of BDAL's data						
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Recognition time	19 s	2 s	1 s	2 s	4 s	6 s
AR (en %)	92.0	94.7	51.5	89.0	92.9	94.9
FRR (en %)	7.4	4.6	15.5	10.5	6.4	4.5
FAR (en %)	0.6	0.7	33.0	0.5	0.7	0.6

Table 4. Testing results with images sub-blocs sized of 16 × 16.

	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Training stage with 80% of NIST's data						
Recognition time	21 s	4 s	3 s	4 s	6 s	7 s
AR (en %)	92.1	94.8	51.1	88.9	92.9	94.9
FRR (en %)	7.4	4.6	16.3	11.7	7.3	4.6
FAR (en %)	0.5	0.6	32.6	0.4	0.6	0.5
Training stage with 80% of BDAL's data						
	Classic method	Probabilist method	Mean classes LCNN	1-*PPV* LCNN	*K-PPV* LCNN (*K* = 8)	Proposed method
Recognition time	15 s	2 s	1 s	2 s	4 s	6 s
AR (en %)	91.8	94.6	50.8	88.6	92.6	94.8
FRR (en %)	7.6	4.7	16.2	10.9	6.7	4.6
FAR (en %)	0.6	0.7	33.0	0.5	0.7	0.6

5. Discussion

The images of the databases (BDAL and NIST) used to perform the above tests are proportionally distributed among the sub-classes defined by Henry (arch, arch attempted, whirl and loops).

By using vector blocks size of 16×16 compared to vector blocks size of 8×8, a significant gain of 6.40 minutes and 1.5 seconds on average can be achieved respectively in the learning time and the recognition time. In return, there has been an average fall of 0.3% and a slight increase of 0.3% respectively on the acceptance rate (AR) and the false rejection rate (FRR). In fact, when using blocks of size 8×8 compared to those of size 16×16, the quantity of information to handle is reduced and therefore easier to compute with regard to the detection of streaks, singular points, valleys, … However, the number of sub-blocks to be processed increases and promotes the long processing time.

The proposed method, based on the use of a perceptron with three hidden layers offers best results in terms of recognition rate and false rejection rate as shown by Table 3 and Table 4. This can be explained by the fact that the proposed network uses three hidden layers to perform the classification, using calculation of the probabilities on features vectors (which are invariant to translation and rotation). In fact, the use of the second classification layer has been introduced to control wether the winner neurone after the first classification is really a pattern of the right class, and this contribute to reduce the proportion of error rate (especially false acceptance rate). Furthermore, the use of prototype (with process update) instead of the entire database to make classification contribute to the robustness of our proposed method.

One of the strong results of our experience shows that the more the decision threshold is strict, the more the probability that the system provides a satisfactory result is high (as shown in **Figure 2**). But in return, the complexity of the algorithm is increased, resulting in increased learning and recognition time. Also the choice of a reasonable decision threshold is recommended to avoid the perverse effect of a surcharge on the recognition rate due to threshold considered too strict.

The proposed algorithm, performs matching in less than 7 s on average. On both databases used, as the ϵ decision threshold (from 0.1 to 10^{-7} with 20 minutiae), the overall system has a false acceptance rate (FAR)

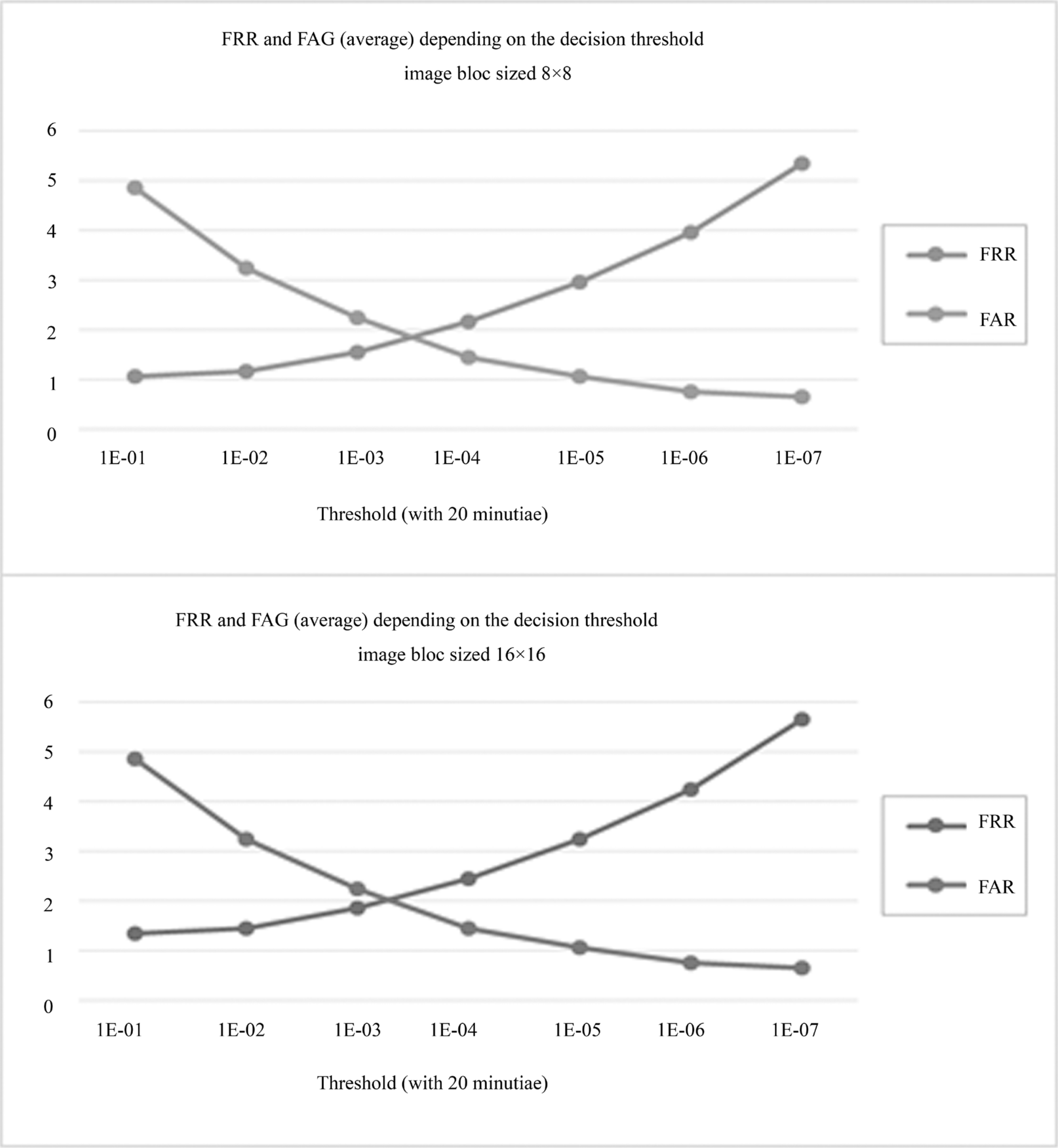

Figure 2. FRR and FAR depending on the decision threshold.

ranging from 0.6% to 4.8%, a false rejection rate (FRR) ranging from 1.1% to 5.3% as shown in **Figure 2** above.

6. Conclusions

The aim of this paper was to provide a fingerprint recognition algorithm based on artificial neural networks for authentication. The results obtained with different techniques studied show that using blocks of processed image enables a recognition with low false acceptance rate in the range of $\pm 0.5\%$ on average. For against, when the recognition system uses the mean classes of entire images, these false acceptance rates are very high and reflect the subjectivity of this method.

Many fingerprints' recognition systems for authentication have already been established in the literature, but the method we propose is quite innovative and takes into account both the calculation of probabilities and the comparison of templates (invariants to translation/rotation) to perform classification, an unexplored way till now (to our knowledge). A special effort has been used to take into account other singularities of fingerprint lines for the case of images from the same individual. Tests conducted with this technique have led us to some encouraging results, although upgradeable. Indeed we limited ourselves to perform tests with a maximum set at 10^{-7} squared error between the expected output and that obtained; because beyond this value (which is already very high), the results are somewhat biased. One small technical issue that we propose is that it does not handle very well the case of latent images due to the objective that we set at the start which was to minimise error rate (in particular false rejection rates).

As it stands, the performance of different approaches is considered satisfactory for images of good qualities. However further study on the choice and number of times to be applied pretreatment operations on the fingerprint images would have more powerful algorithms better suited to the case of latent images or visible half. Future work will focus on the establishment of an authentication system that uses more fingerprints, other biometrics such as face shape, the dynamics of the plot of the signature and/or dynamics keystrokes.

The combination of these techniques will highlight a fundamental element in the recognition before and during use of the online training system, namely the signing of the user. Indeed, the use of signing rather than the template is unique that it allows recording in addition to the fingerprint features, a set of other elements specific to individual behaviour of each user and it supports authentication before and during system access.

References

[1] Jain, A.K., Feng, J.J. and Nandakumar, K. (2010) Fingerprint Matching. *IEEE Computer Society*, **43**, 36-44. http://dx.doi.org/10.1109/MC.2010.38

[2] Maltoni, D., Maio, D., Jain, A.K. and Prabhakar, S. (2003) Handbook of Fingerprint Recognition. Springer, New York.

[3] Chatterjee, A., Mandal, S., Atiqur Rahaman, G.M. and Mohammad Arif, A.S. (2010) Fingerprint Identification and Verification System by Minutiae Extraction Using Artificial Neural Network.

[4] Sathiaraj, V. (2012) A Study on the Neural Network Model for Finger Print Recognition. *International Journal of Computational Engineering Research*, **2**, 70.

[5] Galton F. (1892) Fingerprint. McMillan and Co., London.

[6] Kouamo, S. and Tangha, C. (2012) Handwritten Character Recognition with Artificial Neural Network. *Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing*, **151**, 535-543. http://dx.doi.org/10.1007/978-3-642-28765-7_64

[7] Sakshica, Gupta, K., Vidyapith, B., Campus, J. and Jaipur (2015) Handwritten Digit Recognition Using Various Neural Network Approaches. *International Journal of Advanced Research in Computer and Communication Engineering*, **4**, 4.

[8] Jagtap, V.N. and Mishra, S.K. (2014) Fast Efficient Artificial Neural Network for Handwritten Digit Recognition. *International Journal of Computer Science and Information Technologies*, **5**, 2302-2306.

[9] International Biometric Group (2011) The Henry Classification. www.biometricgroup.com

[10] Yu, L., Laaraiedh, M., Avrillon, S. and Uguen, B. (2011) Fingerprint Localisation Based on Neural Networks and Ultra-Wide Band Signals. *IEEE International Symposium on Signal Processing and Information Technology* (*ISSPIT*), Bilbao, 14-17 December 2011, 184-189.

[11] Xia, X. and O'Gorman, L. (2003) Innovations in Fingerprint Capture Devices. *Pattern Recognition*, **36**, 361-369. http://dx.doi.org/10.1016/S0031-3203(02)00036-5

[12] Li, R., Li, C.T. and Guan, Y. (2015) A Compact Representation of Sensor Fingerprint for Camera Identification and Fingerprint Matching. *IEEE International Conference on Acousticsi Speech and Signal Processing* (*ICASSP*), Brisbane, 19-24 April 2015, 1777-1781. http://dx.doi.org/10.1109/icassp.2015.7178276

[13] Wiilis A.J. and Myers, L. (2001) A Cost-Effective Fingerprint Recognition System for Use with Low-Quality Prints and Damaged Fingertips. *Pattern Recognition*, **34**, 255-270. http://dx.doi.org/10.1016/S0031-3203(00)00003-0

[14] Ojha, A.K. (2015) ATM Security Using Fingerprint Recognition. *International Journal of Advanced Research in Computer Science and Software Engineering*, **5**, 170-175.

[15] Cappelli, R., Lumini, A., Maio, D. and Maltoni, D. (2002) Synthetic Fingerprint-Database Generation. *Proceeding of the* 16*th International Conference on Pattern Recognition*, **3**, 744-747. http://dx.doi.org/10.1109/ICPR.2002.1048096

[16] Baldi, P. and Chauvin, Y. (1993) Neural Networks for Fingerprint Recognition. *Neural Computation*, **5**, 402-418. http://dx.doi.org/10.1162/neco.1993.5.3.402

[17] Hamsa, A.A. (2012) Fingerprint Identification System Using Neural Networks. *Nahrain University*, *College of Engineering Journal* (*NUCEJ*), **15**, 234-244.

[18] Thomas, T.J. (2000) Locally-Connected Neural Network for Fingerprint Recognition. *Proceedings of the ISATED International Conference*, *Intelligent Systems and Control*, Honolulu.

[19] Kouamo, S. and Tangha, C. (2013) Images Compression with Artificial Neural Network. *Advances in Intelligent and Systems and Computing*, **189**, 515-524. http://dx.doi.org/10.1007/978-3-642-33018-6_53

[20] Batchakui, B., Tangha, C. and Kameni, J.S. (2012) xCCM: An Alternative Method of Appropriate Contents Creation on an e-Learning Platform. *IEEE Global Engineering Education Conference* (*EDUCON*), Morocco, 17-20 April 2012, 1-7. http://dx.doi.org/10.1109/educon.2012.6201053

[21] Watson, C.I. and Wilson, C.L. (1992) NIST Special Database 4 Figerprint Database. National Institute of Standards and Technology, Advanced Systems Division, Image Recognition Group.