



## Brief paper

## Fingerprint matching using multi-dimensional ANN

Rajesh Kumar<sup>\*,1</sup>, B.R. Deva Vikram

Department of Electrical Engineering, Malaviya National Institute of Technology, 302017 Jaipur, India

## ARTICLE INFO

## Article history:

Received 17 May 2008

Received in revised form

25 December 2008

Accepted 16 November 2009

Available online 31 December 2009

## Keywords:

AFIS

FAR

FRR

Minutiae

Matching

MDANN

Training

Weights

## ABSTRACT

Fingerprint matching is one of the most widely used biometric technique for personal identification. This identification is achieved in this work by using the concept that every fingerprint has a unique pattern of distribution of the minutiae points. In this paper, a new method of recognition of this pattern of distribution of the minutiae points of an enhanced image is considered by using a multi-dimensional artificial neural network (MDANN). The proposed technique has the distinct advantage of using the entire resized minutiae image as an input at once. It is capable of excellent pattern recognition properties as the distribution of the minutiae points are used directly for recognition. The proposed approach shows significant promise and potential for improvement, compared with the other conventional matching techniques with regard to time and efficiency of results.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

Biometrics is a technology that uniquely identifies a person based on his physiological or behavioral characteristics. Some of the biometrics which have excellent potential for identification are the fingerprint, iris, retinal scan (Jain et al., 1997). Fingerprint matching is one of the oldest and trusted biometric for identification (Lee and Gaensslen, 1991). The manual classifications of the fingerprints were time consuming and prone to errors, hence it was not possible to promote it on a commercial scale. The first scientific paper on fingerprint was given by English morphologist N. Grew in 1684, in which he had reported his study on the ridge, furrow and pore structures in the fingerprint. In 1823, Purkinje developed the first fingerprint classification scheme in which the fingerprints were classified into nine categories according to the ridge configuration (Lee and Gaensslen, 1991). Sir F. Galton after an extensive study of fingerprints introduced the minutiae features for the fingerprint classification in the year 1888. The minutiae points are the local discontinuities in the ridge flow pattern. The FBI invested a lot of effort since the early 1960s in the development of the AFIS (Federal Bureau of Investigation, 1984). Ever since the Automatic Fingerprint Identification System (AFIS) was developed in 1991, rapid

progress has been made in improving the recognition rates. This development has greatly improved the scope of using fingerprint for identification purposes and in reducing the cost of hiring and training human experts for fingerprint matching. The current works in this field focus on mainly reducing the computation time for feature extraction and matching and improving the accuracy of the results. Today, with the crime rates increasing everyday there is an urgent need for a system which is safe and fast.

The general fingerprint recognition system comprises of two broad steps namely: (i) feature extraction and (ii) classification of the image using the features extracted. Some of the important feature extraction methods include: (i) filter based approach (ii) structure based approach (iii) statistical based approach which also comprises the method of extracting the minutiae points (Ern and Sulong, 2001). The methods of matching the fingerprint using the features extracted is carried out by using one of the many approaches namely: (i) syntactics, (ii) singularities based, (iii) structural, or (iv) neural network based (Ern and Sulong, 2001). Some of the recent work in the field of fingerprint matching includes the chain coded string matching technique (Paul et al., 2007), use of the combination of fast Fourier transform (FFT) and Gabor filters for the enhancement of the captured image (Aguilar et al., 2007) which have the potential of improving the speed and the performance of feature extraction and matching.

One of important methodology used in the fingerprint recognition system is the use of minutiae points for matching the fingerprints. Some of the popular minutiae matching techniques are: (i) using the Delaunay triangulation method (Bebis et al., 1999), (ii) using local structural similarity (Ratha

<sup>\*</sup> Corresponding author. Tel. +91 141 2713372; fax +91 141 2529092.

E-mail addresses: rk.rlab@gmail.com (R. Kumar), devavikram@gmail.com (B.R. Deva Vikram).

<sup>1</sup> Home Address: C-7, Warden Quarters, Staff Colony, Malaviya National Institute of Technology, 302017 Jaipur, India.

et al., 2000), (iii) using neural networks or a combination of neural and fuzzy logic (Arantus et al., 2002; Hong and Hua, 2003). In this paper, we propose to use the MDANN in which the 2-D resized minutiae image is taken as an input directly and this method involves the training of the entire pattern at once as against the other earlier works of training, in which the whole image is either divided into a number of small images and then trained one by one or use the 1-D feature vector for training. This method is proposed as it uses the comprehensive data of the fingerprint and hence the chances of failure to recognize the authentic fingerprints are very remote.

The rest of the paper is organized as follows: Section 2 describes the image enhancement techniques along with the binarization and thinning techniques. Section 3 explains the minutiae extraction and its authentication method. Section 4 introduces the resizing algorithm briefly. Section 5 introduces the concept of MDANN and its algorithm. The results are presented in Section 5 and the concluding remarks are presented in Section 6.

## 2. Image enhancement

The quality of the fingerprint ridge structure is very important as they possess the necessary information for the extraction of minutiae points. Ideally the ridges and valleys should alternate with a clear demarcation and flow in a locally constant direction. Due to a number of factors, the obtained fingerprints may not have well defined ridge/valley structures and might contain a lot of disturbance in the image (Jain et al., 1997). So the fingerprint image is first enhanced before further processing and comprises of the following steps: (i) segmentation, (ii) normalization, (iii) orientation estimation, (iv) ridge frequency estimation, (v) Gabor filtering (vi) binarization, (vii) thinning (Jain et al., 1999).

### 2.1. Segmentation

Segmentation is the process of distinguishing the background region from the foreground regions. It is observed that the background regions have a low gray-scale variance value as compared to the foreground region. So a method based on variance thresholding is used for segmentation (Mehtre, 1993).

$$V(k) = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (I(i, j) - M(k))^2 \quad (1)$$

where  $V(k)$  is the variance and  $M(k)$  is the mean gray-level value of the block  $k$ .  $I(i, j)$  is the pixel value at the pixel  $(i, j)$ .

### 2.2. Normalization

Normalization is done in order to standardize the dynamic variation in the gray-level values.

$$N(i, j) = M_o + \sqrt{\frac{V_o(I(i, j) - M)^2}{V}} \quad \text{if } I(i, j) > M \quad (2)$$

$$M_o - \sqrt{\frac{V_o(I(i, j) - M)^2}{V}} \quad \text{otherwise} \quad (3)$$

where ' $N(i, j)$ ' is the normalized gray-level value and  $I(i, j)$  is the gray-level value at pixel  $(i, j)$ . ' $M$ ' and ' $V$ ' are the estimated mean and variance of  $I(i, j)$  respectively and ' $M_o$ ' and ' $V_o$ ' are the desired mean and variance values respectively.

### 2.3. Orientation estimation

The orientation field of a fingerprint defines the local orientation of the ridges and is necessary for the Gabor filtering stage to effectively enhance the image (Jain et al., 1997). The local orientation  $\theta$  for the block centered at the pixel  $(i, j)$  is calculated by using the following equations.

$$O_x(i, j) = \sum_{u=i-W/2}^{i+W/2} \sum_{v=j-W/2}^{j+W/2} 2\partial_x(u, v)\partial_y(u, v) \quad (4)$$

$$O_y(i, j) = \sum_{u=i-W/2}^{i+W/2} \sum_{v=j-W/2}^{j+W/2} \partial_x^2(u, v)\partial_y^2(u, v) \quad (5)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \frac{O_x(i, j)}{O_y(i, j)} \quad (6)$$

The orientation image is converted into a continuous vector field. ' $\phi_x$ ', ' $\phi_y$ ' being the x and y components of the vector field respectively. The Gaussian smoothing is then performed and the equations for the process are as under,

$$\phi_x(i, j) = \cos(2\theta(i, j)) \quad (7)$$

$$\phi_y(i, j) = \sin(2\theta(i, j)) \quad (8)$$

$$\phi'_x(i, j) = \sum_{u=-q/2}^{q/2} \sum_{v=-q/2}^{q/2} G(u, v)\phi_x(i-uw, j-vw) \quad (9)$$

$$\phi'_y(i, j) = \sum_{u=-q/2}^{q/2} \sum_{v=-q/2}^{q/2} G(u, v)\phi_y(i-uw, j-vw) \quad (10)$$

where ' $G$ ' is a Gaussian low pass filter of size  $q \times q$  and the final smoothed orientation field ' $O$ ' at pixel  $(i, j)$  is given by

$$O(i, j) = \frac{1}{2} \tan^{-1} \frac{\phi'_x(i, j)}{\phi'_y(i, j)} \quad (11)$$

### 2.4. Ridge frequency estimation

Along with the orientation field, the ridge frequency is also required for the construction of Gabor filter. This represents the local frequency of the ridge in the fingerprint (Jain et al., 1997).

### 2.5. Gabor filtering

The orientation and the ridge frequency information is used to design the even-symmetric Gabor filter (Daugman, 1985). An even-symmetric Gabor filter is defined in the spatial domain as

$$G(x, y; \theta, f) = \exp\left\{-\frac{1}{2} \left[ \frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2} \right]\right\} \cos(2\pi f x_\theta), \quad (12)$$

$$x_\theta = x \cos \theta + y \sin \theta \quad (13)$$

$$y_\theta = -x \sin \theta + y \cos \theta \quad (14)$$

where  $\theta$  and  $f$  refer to the orientation and the frequency of the Gabor filter, while  $\sigma_x$  and  $\sigma_y$  refer to the standard deviation of the Gaussian envelope. The application of the Gabor filter to the image is performed as follows.

$$E(i, j) = \sum_{u=-M_x/2}^{M_x/2} \sum_{v=-M_y/2}^{M_y/2} G(u, v; O(i, j), F(i, j))N(i-u, j-v) \quad (15)$$

where  $E, O, F, N$  are the enhanced, orientation, ridge frequency and normalized fingerprint images respectively.  $M_x, M_y$  being the width and height of the Gabor filter mask. The range of frequency

to which the filter responds is determined by the parameters  $\sigma_x$  and  $\sigma_y$  and this is chosen to be a function of the ridge frequency.

$$\sigma_x = k_x F(i, j) \quad (16)$$

$$\sigma_y = k_y F(i, j) \quad (17)$$

In order to enable the filter to accommodate the majority of the Gabor waveform information, the filter size is allowed to vary according to the bandwidth of the Gabor waveform.

$$M_x = 6\sigma_x \quad (18)$$

$$M_y = 6\sigma_y \quad (19)$$

where  $k_x$  and  $k_y$  are constants for  $\sigma_x$  and  $\sigma_y$  respectively.

## 2.6. Binarization

Binarization is performed to convert the gray-scale image into a binary image having either zero or one as the pixel intensity value using a certain threshold (Aguilar et al., 2007).

## 2.7. Thinning

This is the final stage of image enhancement. This is a morphological process which corrodes the image successively until a skeletal image of the fingerprint which is one pixel wide is obtained (Aguilar et al., 2007). This image is in turn used for the minutiae extraction process.

## 3. Minutiae extraction and its authentication

The enhanced image is used to extract the minutiae points of the fingerprint. There are a number of features that could be used for the identification of fingerprints, but mostly the minutiae points are restricted to two types namely the ridge ending and the ridge bifurcation. Ridge endings are the point where the ridge curve terminates and the ridge bifurcations are the points where the ridge curve splits from a single path to two on a Y-junction.

### 3.1. Minutiae extraction

The minutia points are extracted from the enhanced fingerprint image by using the Crossing Number (CN) concept (Amengual et al., 1997; Mehtre, 1993; Kasaei and Boashash, 1997). The CN value for a ridge pixel  $P$  is given by the equation.

$$CN = \frac{1}{2} \sum_{i=1}^8 |P(i) - P(i+1)|, \quad P(9) = P(1) \quad (20)$$

The value of the CN number is used to differentiate between the types of minutiae. The ridge ending will have a CN value of one while the ridge bifurcation will have a value of three.

### 3.2. Authentication of the minutiae points

The minutiae points extracted by using the CN concept might contain some false minutiae points. The spurious minutiae points may be due to either a loop in the ridge structure which could be classified as a bifurcation point or a few isolated points may be assumed to be ridge endings (Tico and Kuosmanen, 2000).

## 4. Resizing of the minutiae image

The minutiae image obtained from the previous stage clearly depicts the pattern of the fingerprint minutiae points. This pattern

is to be trained on a MDANN. Since, the normal image size of 512\*512 cannot be used to train the network as it will be highly time consuming and moreover it is not possible to train an image of this size on a system with a 512 MB RAM. So it is mandatory to reduce the size of the image without losing the necessary information from the image. This is performed by dividing the minutiae image into 225 (15\*15) matrices and then add up the pixel intensity values of every individual matrix. Now we will be left with a 15\*15 matrix, in which the pixel intensity value is given a maximum of one and the rest are given a value of zero. This smaller matrix retains the minutiae pattern and hence can be used to train the network.

## 5. Recognition by MDANN

In the neural networks, learning is easy because it is dependent only on the locally available information, but since the information is mixed together at the storage elements, unambiguous retrieval of stored information is not simple. Recognition involves knowing whether or not a new input has been presented before and stored in memory. The gradient descent is generalized of to a non-linear, multi-layer feed forward network called backpropagation. This is a supervised learning algorithm that learns by first computing an error signal and then propagates the error backward through the network by assuming the network weights are the same in both the backward and forward directions. Backpropagation is a very popular and widely used network learning algorithm (Anderson, 1995).

In this paper, we explain our work on training the minutiae images on the ANN by using the backpropagation algorithm and then matching the fingerprint for recognition from the database. The MDANN used here consists of 20 neurons and was designed to take the matrix inputs. Fig. 1 shows the network structure of MDANN but the connections between the input and the hidden layer are not shown completely for the purpose of clarity and the complete structure is obtained by extending the connections to the entire image.

The inherent advantage of using a multidimensional ANN in this system is that the entire matrix containing the minutiae points can be used to train the network which means that the relative orientation of the minutiae points are automatically used in the recognition.

### 5.1. Algorithms for the MDANN

Here,  $S_1$  and  $S_2$  are structures containing  $n$  layers and  $k$  elements in each field and comprise of the internal weights  $W_{dl}, W_{dr}, W_{dl1}, W_{dr1}$ , are matrices denoting the entire strengths of

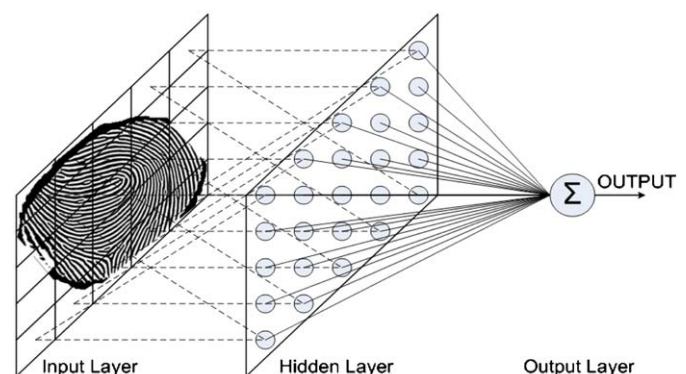


Fig. 1. Structure of MDANN.

the image to neuron connection and the components  $A_w, A_{dw}, A_b, A_d, A_{d1}$  are single row matrices denoting the weights of the connection of the neurons with the image pixel points, their differential values, the bias values for the neurons, their differential values.

*Step 1: calculation of the derivatives:*

Now 'i' refers to the number of patterns we are trying to train on the network. 'j' refers to the number of regresses which is given by one in excess of the number of neurons and 'k' refers to the dimension of the input matrix. 'n' refers to the number of neurons.

$$\Delta = \text{actual output} - \text{required output} \quad (21)$$

*For the weights:*

$$A_d(j) = A_d(j) + \Delta, \quad \text{if } j = 1; \quad (22)$$

$$A_d(j) = A_d(j) + \Delta \left( \frac{2}{1 + \exp(-2(S_{j1} \times I(i) \times S_{j2}))} - 1 \right), \quad \text{if } j > 1 \quad (23)$$

where the symbols ' $S_{j1}$ ' and ' $S_{j2}$ ' refer to the  $j$ th layer elements of the structures ' $S_1$ ' and ' $S_2$ ' respectively and  $I(i)$  refers to the  $i$ th matrix input to the network. The values are calculated for all values of  $j$ . Now ' $as$ ' refers to the activation signal value and is given by the equation:

$$as = \frac{-2}{1 + \exp(-2(S_{j1} \times I(i) \times S_{j2}))} \quad (24)$$

*For the internal weights:*

$$W_{d1}(j, k) = W_{d1}(j, k) + (A_w(j) \times \Delta \times as \times (S_{j2} \times I(i, k, :))) \quad (25)$$

$$W_{dr1}(j, k) = W_{dr1}(j, k) + (A_w(j) \times \Delta \times as \times (S_{j1} \times I(i, :, k))) \quad (26)$$

Where  $W_{d1}(j, k)$ ,  $W_{dr1}(j, k)$  refer to elements in the  $j$ th row and  $k$ th column of the matrices  $W_{d1}$  and  $W_{dr1}$  respectively.

$I(i, k, :)$  refers to the  $k$ th row elements of the  $i$ th input in the structure.  $I(i, :, k)$  refers to the  $k$ th column elements of the  $i$ th input in the structure. These values are calculated for all the coordinates  $(j, k)$ .

*For the bias values:*

$$A_{d1}(j) = A_{d1}(j) + (\Delta \times as \times A_w(j)) \quad (27)$$

*Step 2: updating the weights and bias values and setting their derivatives back to zero:*

*For all values of j*

*For the weights*

$$A_w(j) = A_w(j) + (\alpha \times A_{dw}(j)) - (\eta \times A_d(j)) \quad (28)$$

$$A_{dw}(j) = (\alpha \times A_{dw}(j)) - (\eta \times A_d(j)) \quad (29)$$

$$A_d(j) = 0 \quad (30)$$

*For the bias values*

$$A_b(j) = A_b(j) + (\alpha \times A_{db}(j)) - (\eta \times A_{d1}(j)) \quad (31)$$

$$A_{db}(j) = A_{db}(j) - (\eta \times A_{d1}(j)) \quad (32)$$

$$A_{d1}(j) = 0 \quad (33)$$

where  $\eta$  and  $\alpha$  refer to the initial learning rate and the momentum factor respectively.

*Step 3: updating the internal weights and setting the derivatives back to zero:*

*For all the coordinates (j,k)*

$$S_{jk1} = S_{jk1} + (\alpha \times W_{d1}(j, k)) - (\eta \times W_{d1}(j, k)) \quad (34)$$

$$W_{d1}(j, k) = (\alpha \times W_{d1}(j, k)) - (\eta \times W_{d1}(j, k)) \quad (35)$$

$$W_{dr1}(j, k) = 0 \quad (36)$$

$$S_{jk2} = S_{jk2} + (\alpha \times W_{dr1}(j, k)) - (\eta \times W_{dr1}(j, k)) \quad (37)$$

$$W_{dr}(j, k) = (\alpha \times W_{dr}(j, k)) - (\eta \times W_{dr}(j, k)) \quad (38)$$

$$W_{dr1}(j, k) = 0 \quad (39)$$

where  $S_{jk1}$  and  $S_{jk2}$  refer to the  $k$ th element in the  $j$ th layer of the structure  $S_1$  and  $S_2$  respectively

*Step 4: calculating the output values after the training*

$$\text{output} = A_w(j), \quad \text{for } j = 1; \quad (40)$$

$$\text{output} = A_w(1) + \sum_{j=2}^n A_w(j) \times \left( \frac{2}{1 + \exp((S_{j1} \times I(i) \times S_{j2}) + A_b(j))} - 1 \right) \quad (41)$$

During the training stage of the network with the fingerprint images, each fingerprint image is assigned a distinct arbitrary value. Now the different images of the same fingerprint are trained on the network with the same arbitrary value as the target value. The iteration goes on until the value calculated in Step 4 is the target value that was assigned to the image. This value is unique for every distinct fingerprint image and when a random fingerprint is tested on the network, this value concludes as to whether the image is already present in the database or not and if present the exact image to which it corresponds to.

The fingerprint images were collected from the institute student data bank using paper and ink and then scanned to give an image that can be used for processing and matching. Each fingerprint was collected three times and was collected from a total of 3500 individuals. Out of these images, fingerprints of 500 individuals were reserved just for the test purposes and were not used for the training of the MDANN. The other fingerprint images (two each) from the 3000 individuals were used for training the MDANN. These images were firstly subjected to image enhancement process wherein the discontinuities and the disturbances in the image were removed and subsequently the resized image was obtained from the minutiae image. The resized image was then trained on the MDANN as mentioned earlier.

Now the trained MDANN is subjected to a test to evaluate if the network is capable of recognizing a fingerprint image that was already trained on it and also to evaluate if it can reject a fingerprint image that was not trained on it. So the images that are used for the test purpose are: (a) the 500 images that were isolated initially and reserved only for the test purposes and (b) out of the three images each collected from the 3000 individuals, one image each from these 3000 individuals which had not been used during the training of the network.

Now the test images are subjected to the same process of image enhancement, minutiae extraction and resizing and then applied to the MDANN. The value obtained from Eq. 40 for the test image is compared with the values that were assigned earlier during the training of the 3000 unique fingerprint images. If the values are found to match, then the image is concluded to be present in the database and the value of Eq. 40 will exactly indicate as to which image it is in the database. If the values are not matched, then the image is said to be not present in the database.

## 6. Result

The actual image was first enhanced in order to enable the proper extraction of the minutiae points. The segmented image which separates the background from the foreground area is shown in Fig. 2(a). The normalized image where the dynamic intensities are standardized so that the values are in a specific range is shown in Fig. 2(b).





Fig. 2. (a) Segmented image, (b) normalized image.

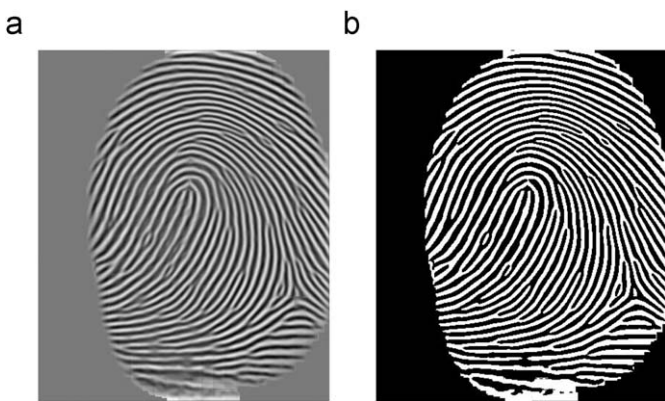


Fig. 3. (a) Enhanced image, (b) binarized image.

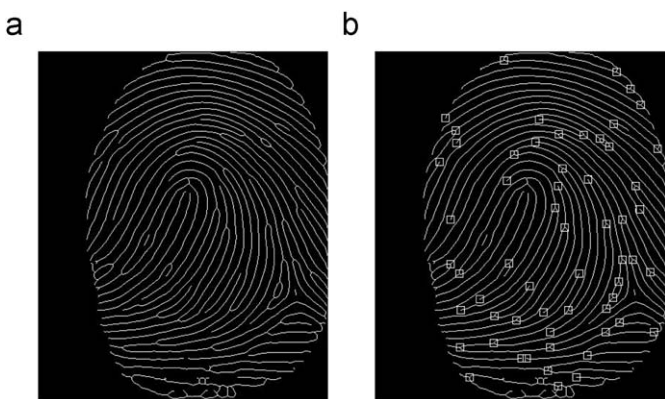


Fig. 4. (a) Thinned image, (b) thinned image showing the minutiae points.

The enhanced image is shown in Fig. 3(a) and this is better in terms of the continuity of the ridge structure and Fig. 3(b) shows the binarized image which has a better clarity and continuity of the ridge. This binarized image has only two level of intensity values namely zero and one and clearly distinguishes between the ridges and the valleys.

The thinned image obtained from the binarized enhanced image is shown in Fig. 4(a) and is used for calculating the CN values and then for extracting the minutiae points. The minutiae points obtained after the authentication of these points obtained from the CN method is shown in Fig. 4(b).

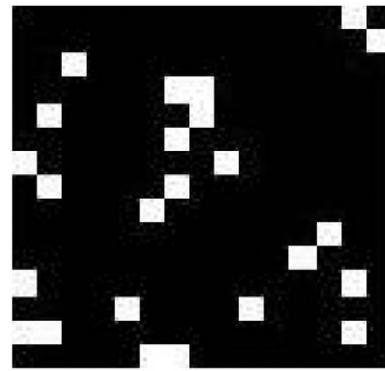


Fig. 5. Resized minutiae image.

The resized image is shown in Fig. 5 and when this image is trained on the multi-dimensional neural network by using the gradient descent training procedure gave the following performance curve, where performance is defined as square of the difference between the target value and the actual value obtained during simulation. The performance curve has been plotted against the number of iterations. It is observed that the performance is steadily improving and leading to a value around zero.

The rate of learning curve has been shown in Fig. 6, and this refers to the rate at which the network is learning. This rate is altered continuously in the algorithm every time the network is subjected to iteration (see Fig. 7).

The algorithm was tested on a database of 3000 fingerprints and was found to give better recognition result as compared to other algorithms as shown in Table 1. It was observed that with an increase in the number of unique fingerprints in the database to values above 3500, the recognition rate started declining considerably.

It was also observed that, with a bigger resized minutiae image instead of 15\*15, this limit of 3500 was increased to support a larger database but with a decline in the speed of the recognition.

The simulations were carried out on MATLAB 7.02 version on a 512 MB RAM, 1.7 GHz processor. The trained network was able to give the exact value every time an image which was one of those which had been trained earlier was given as input. The results were given fast and were observed to have good rate of recognition for a database of 3000 fingerprint inputs.

## 7. Conclusion

The fingerprint matching technique that uses the minutiae points to identify fingerprints have been presented in this paper. Preprocessing techniques were first applied to enhance the image and then obtain a thinned binary fingerprint ridge structure that can be used for minutiae extraction. The extracted minutiae images of the fingerprints from the database are then trained on a multi-dimensional artificial neural network which takes in the two-dimensional minutiae image matrix as an input and a target output. The trained network can then be used for the purpose of matching the fingerprints from the database. This method relies on portraying the pattern in two-dimensions and hence retains the details about the pattern of the minutiae in an image in the best possible way. This method is found to give results that are comparable in terms of efficiency of recognition with the other pattern recognition techniques and hence the results are encouraging to work on improving the MDANN for better matching capabilities. This neural based technique was also

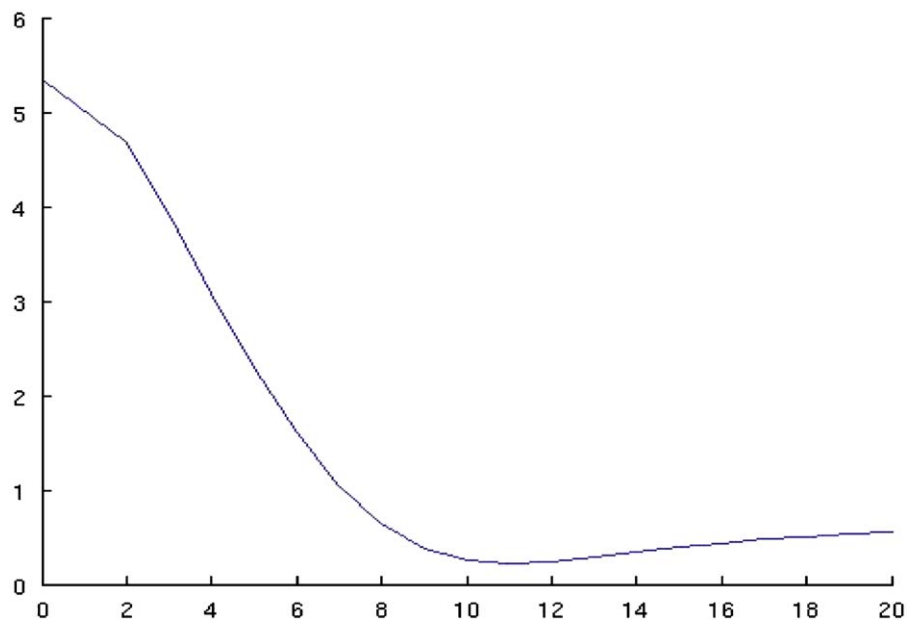


Fig. 6. Performance vs. number of iterations.

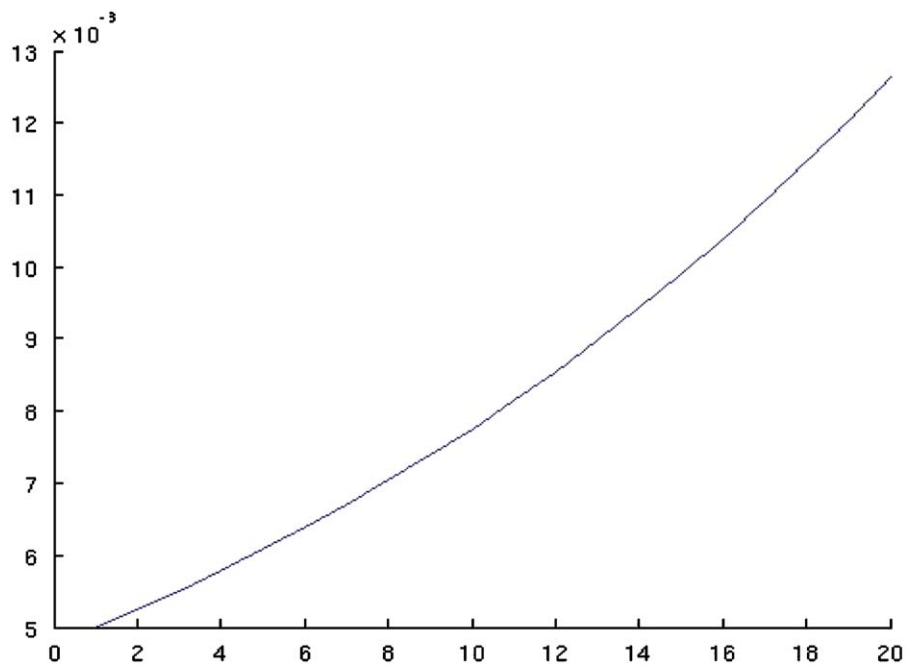


Fig. 7. Rate of learning (eta) vs. the number of iterations.

Table 1

Performance comparison the result obtained with other results presented in literature.

Algorithm	Recognition rate (%)	False acceptance rate (FAR) (%)	False rejection rate (FRR) (%)
Zhang's (Wang and Zhang, 2000) algorithm	91.3	3.1	4.4
G. Aguilar's (Aguilar et al., 2007) algorithm	94.1	2.3	3.6
Proposed algorithm	97.37	1.13	1.5

found to support fast fingerprint classification rates. Moreover further work in this field could be in optimizing the trade-off between the speed of recognition and the size of the database that can be supported by the network.

## References

- Aguilar, G., Sanchez, G., Toscano, K., Salinas, M., Nakano, M., Perez, H., 2007. Fingerprint Recognition. Second International Conference on Internet Monitoring and Protection, 32.

- Amengual, J.C., Juan, A., Prez, J.C., Prat, F., Sez, S., Vilar, J.M., 1997. Real time minutiae extraction in fingerprint images. *Proceedings of the 6th International Conference on image processing and its application*, 871–875.
- Anderson, James A., 1995. In: *An Introduction to Neural Networks*. Prentice Hall Publications.
- Arantus, M., Ide, A.N., Saito, J.H., 2002. A system for fingerprint minutiae classification and recognition. *IEEE Proceedings of the Ninth International Conference on Neural Information Processing* 5, 2474–2478.
- Bebis, G., Deaconu, T., Georgiopoulos, M., 1999. Fingerprint Identification using Delaunay Triangulation". *IEEE Transactions on Information Intelligence and Systems*, 452–459.
- Daugman, J.G., 1985. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters. *Journal of the Optical Society of America (A)*, 1160–1169.
- Ern, L.C., Sulong, G., 2001. Finger print classification approach—an overview. *IEEE Sixth International Symposium on Signal Processing and its Application* 1, 347–350.
- Federal Bureau of Investigation, 1984. In: *The Science of Fingerprints: Classification and Uses*. GPO, Washington, DC.
- Hong, H., Hua, L.J., 2003. Fingerprint matching using ANFIS. *IEEE International Conference on Systems, Man and Cybernetics* 1, 217–222.
- Jain, Anil K., Hong, Lin, Pankanti, Sharath, Boll, Rudd, 1997. An Identity Authentication System Using Fingerprints. *Proceedings of the IEEE* 85 (9), 1365–1388.
- Jain, A.K., Prabhakar, S., Hong, L., 1999. A multi-channel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 335–348.
- Kasaei, M.D.S., Boashash, B., 1997. Fingerprint feature extraction using block-direction on reconstructed images. *IEEE Region TEN Conference Digital Signal Processing Applications, TECON*, 303–306.
- Lee, H.C., Gaensslen, R.E. (Eds.), 1991. *Advances in Fingerprint Technology*. Elsevier, New York.
- Mehre, B.M., 1993. Fingerprint Image Analysis for automatic identification. *Machine Vision and its Applications* 6, 124–139.
- Paul, R., Nasif, M.S., Farhad, S.M., 2007. Fingerprint recognition by chain coded string matching technique. *International Conference on Information and Communication Technology*, 64–67.
- Ratha, N.K., Pandit, V.D., Bolle, R.M., Vaish, V., 2000. Robust Fingerprint authentication using local structural similarity. *Fifth IEEE Workshop on Applications of Computer Vision*, 29–34.
- Tico, M., Kuosmanen, P., 2000. An algorithm for fingerprint image post-processing. *Proceedings of the Thirty Fourth Asilomar Conference on Signals, Systems and Computers* 2, 1735–1739.
- Wang, Y., Zhang, W., 2000. Core Based Structure matching Algorithm of Fingerprint Verification. *Proceedings of the IEEE International Conference on Pattern Recognition*, 70–74.