A Study on Artificial Neural Network for Fingerprint Recognition

Thanh S.Nguyen, Phuong V.T.B
Faculty of Electronic and Telecommunication
Danang University of Science and Technology
Danang, Vietnam
Email: nsthanh102@gmail.com
vophuong197@gmail.com

Tuan V.Pham
Faculty of Electronic and Telecommunication
Center of Excellence
Danang University of Science and Technology
Danang, Vietnam
Email: pvtuan@dut.udn.vn

***Abstract*— Fingerprint recognition has been popularly used in the biometric systems. A lot of methods were developed to enhance performance and improve the reliability of the recognition system. In this paper, an effective method based on the trained artificial neural network model using multi-layer perceptron with back-propagation algorithm was researched and presented to recognize the fingerprint. In addition, the quality of ridge line is enhanced with the combination of Gabor filter and Fast Fourier Transform algorithm. This system was evaluated by using three database sets FVC2000, FVC2002, FVC2004. The performance of proposed method is compared with matching method. The recognition results indicated that the approach of using neural network had high recognition result and reliability. The best performance of this system on the same condition database sets is over 90% with both FAR and FRR are 4%.**

Keywords—fingerprint recognition; Gabor filter; FFT algorithm; Artificial Neural Network; Back-propagation algorithm.

I. INTRODUCTION

Fingerprint recognition system is one of the biometric systems widely used for recognition because of its uniqueness and stability and high performance. To recognize the fingerprint, firstly taking the information of minutiae before matching with the database stored in the system. There are a lot of proposed recognition approaches. Based on matching technique, there are matching based on Artificial Neural Network, matching using parallel processors or another architectures [1]. Based on the feature, there are three main methods, namely the pattern-based matching [2], the correlation-based matching [3], the feature-based matching [4]. In which, the feature-based matching is a most popular method. This approach was done by Sachine Harne [5], R.Dharmendra Kumar and his partners [6].

The minutiae extraction plays an important role, making the database for the recognition process. Therefore, the input image must be through preprocessing step to enhance the quality of image. Some methods had been proposed such as Wiener filter, Histogram Equalization, Isotropic and Anisotropic filter, Gabor filter and FFT algorithm. Among them, the combination of Gabor filter and FFT algorithm are commonly used in the previous studies and produce the image with the best quality [7][8].

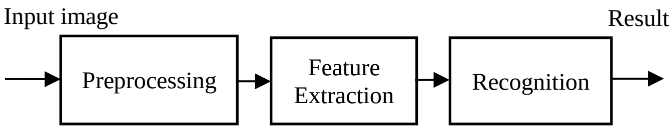

Fig. 1. The block diagram of the fingerprint recognition system

The image enhancement using FFT algorithm help to increase the quality of ridge lines, to fill the holes, to connect the interrupted ridges [5]. Meanwhile, Gabor filter help to exclude noise and to improve the structure of ridges and valleys [8].

In this paper, the Artificial Neural Network was chosen as the recognition method. Feature vector was created on the basic of minutiae points (bifurcation point and ending point) extracted from the fingerprint image, then it was put into the neural network for training. The end of the training process create a neural network model with the weights of the network. The multi-layer perceptron with back-propagation algorithm was used to train the neural network [12]. The system was built and evaluated on three different conditions with the database taken from three database sets FVC2000, FVC2002, FVC2004 [9][10][11]. The performance of proposed method is compared with matching method. The block diagram of the fingerprint recognition system is shown in Fig.1.

The paper is organized as follows: In section II, describes the algorithm used in the fingerprint recognition system includes preprocessing process, feature extraction and neural network. In section III, describes the matching method. The experimental results and evaluation are carried out in section IV. Finally, conclusion and future research of the paper are in section V.

II. FINGERPRINT RECOGNITION ALGORITHM

A. *Preprocessing*

The input image usually has low quality, so it needs to be through the preprocessing to improve the quality for feature extraction process. In this paper, the preprocessing process includes normalization, segmentation, orientation estimation, frequency estimation, Gabor filter and filtering in the frequency domain using Fast Fourier Transform.

1) *Normalization*

The main purpose of normalization is to reduce the variations of the gray level values between ridges and valleys [5]. The gray level value of the pixels is adjusted through normalization process to achive mean gray level value (mean) and gray level variance (variance) as desired.

Let *I(i,j)* and *N(i,j)* denote the gray level value at pixel (i,j) of the input image and normalized image, respectively. The formula to calculate the value of normalized image is defined as follows [7]:

$$N(i,j) = M_0 + \frac{std_0}{std}\left(I(i,j) - M\right) \tag{1}$$

M_0 and std_0 are the desired mean gray level value and gray level variance. M and std which denote the mean gray level value and gray level variance of image I with $M \times N$ size.

2) Segmentation

Segmentation is to take neccessary regions for recognition procedure including clear ridges and valleys. Other regions are background that stores no information with recognition.

The proposed method uses a T threshold based on gray level variance to segment. Let *I(i,j)* is gray level value of image at pixel (i,j). The algorithm is summarized as follows [8]:

- Deviding the image into successive, non-overlapped blocks, each block's size is $W \times W$.
- Computing the gray level variance *v(i,j)* at centered pixel (i,j) for each block.

$$v(i,j) = \frac{1}{W^2}\sum_{i=0}^{W-1}\sum_{j=0}^{W-1}\left(I(i,j) - M(i,j)\right)^2 \tag{2}$$

- Taking blocks with the gray level variance larger than T threshold.

3) Orientation Estimation

In this paper, orientation of the ridge line has been used for Gabor filter [8]. If the orientation is wrong, the quality of image is reduced. The least mean square is popularly method to estimate the orientation field. This method bases on gradient value to find the best orientation.

Given a normalized image, the main steps of the algorithm are as follows [8]:

- Estimating for each successive, non-overlapped block of image with each block's size is $W \times W$ (proposed W is 16).
- Calculating the gradient value in x direction - $\partial x(i,j)$ and y direction - $\partial y(i,j)$ for each pixel of each block. The used gradient operator is Sobel:

$$V_x(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}}\sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2\partial_x(u,v)\partial_y(u,v) \tag{3}$$

$$V_y(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}}\sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \left(\partial_x(u,v)^2 - \partial_y(u,v)^2\right) \tag{4}$$

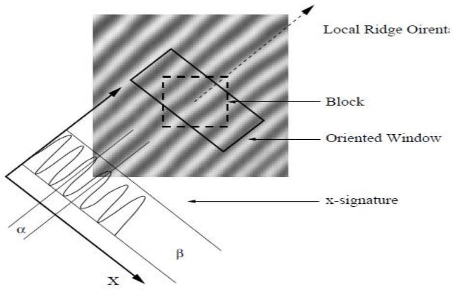

Fig. 2. Orientation window and x-signature

$$\theta(i,j) = \frac{1}{2}\arctan\left(\frac{V_x(i,j)}{V_y(i,j)}\right) \tag{5}$$

With $\theta(i,j)$ is the least mean square estimation value of the local ridge orientation at the centered pixel (i,j) of each block.

- Since the local ridge orientation noise varies slowly in a local neighborhood where no singular points appear, the Gaussian filter is used to smooth the orientation fields and modify the incorrect local ridge orientation [8].

4) Frequency Estimation

Ridge frequency also affects to Gabor filter. In this system, Counting-based method is chosen to measure the ridges distance [8]. The ridges distance is the mean number of dots between two peaks of ridges. This method is very suitable for the case which there is no core or delta point.

Let N is the normalized image, θ is ridge orientation field. According to this method, the frequency estimation algorithm includes the following steps [8]:

- Breaking image into blocks with $W \times W$ size.
- For each centered block at pixel (i,j), calculate the orientation window with $l \times w$ size.
- For each centered block at pixel (i,j), calculate the x-signature of ridges and valleys in the orientation window, in which:

$$X[k] = \frac{1}{w}\sum_{d=0}^{w-1} I(u,v), \quad k = 0,1,\ldots,l-1 \tag{6}$$

5) Gabor filter

Gabor filter have the properties of supporting both frequency-selection and orientation-selection [7]. Hence, it can efficiently remove undesired noise and preserve true ridge and valley structures. [1]. The even-symmetric Gabor filter has the general form [13]:

$$G(x,y,\theta,f) = \exp\left\{-\frac{1}{2}\left(\frac{x_\theta^{\ 2}}{\delta_x^{\ 2}} + \frac{y_\theta^{\ 2}}{\delta_y^{\ 2}}\right)\right\}\cos\left(2\pi f x_\theta\right) \tag{7}$$

With θ is the orientation of the Gabor filter, $[x_\theta, y_\theta]$ are the coordinates of [x,y] after a clock-wise rotation of the descartes axes by an angle of $(90° - \theta)$, f is the frequency of sinusoidal plane wave, δ_x và δ_y are the Gaussian spatial constants along with the x-axis and y-axis, respectively.

The image filtered by Gabor filter is given by following formula:

$$E(i,j)=\begin{cases}0 & if\ \ mask(i,j)=0\\ \sum_{u=-\frac{W}{2}}^{\frac{W}{2}}\sum_{v=-\frac{W}{2}}^{\frac{W}{2}} G(u,v,O(i,j),f)N(i-u,j-v) & if\ \ mask(i,j)=1\end{cases}$$

6) Frequency Domain Filtering

This approach is proposed by Watson, Candela and Grother (1994). Instead of calculating the orientation and the frequency of ridge line, the quality of image is enhanced in frequency domain by using the Fast Fourier Transform algorithm. It is based on root filter, the Fast Fourier transform of the block is multiplied by the amplitude spectrum a set of times *k* [1][6].

The algorithm includes main steps as follows:

- Breaking the image N into blocks with $W\times W$ size (*32×32* pixels) and implement 2D Fast Fourier transform by formula:

$$F(u,v)=\sum_{x=0}^{W-1}\sum_{y=0}^{W-1} f(x,y)\times\exp\left\{-j2\pi\times\left(\frac{ux+vy}{W}\right)\right\} \tag{8}$$

- To enhance a specific block by its dominant frequencies, we multiply the FFT of the block with amplitude a set of times *k* of that block. The block after enhancing:

$$I_{enh}(x,y)=FFT^{-1}\left\{F(u,v)\,|\,F(u,v)|^{k}\right\} \tag{9}$$

B. Feature Extraction

1) Binarization

The combination of Gabor filter and FFT is used to enhance the quality of image and binarization for minutiae extraction [6].

With two enhanced images, algebraic sum is made. Then binarization method is used. The black pixels have a value of 0, if the intensity value lower than 255. The white pixels have a value of 1, if the intensity value larger than 255.

$$I(x,y)=\begin{cases}1 & if\ I(x,y)\ge 255\\ 0 & if\ \ I(x,y)<255\end{cases} \tag{10}$$

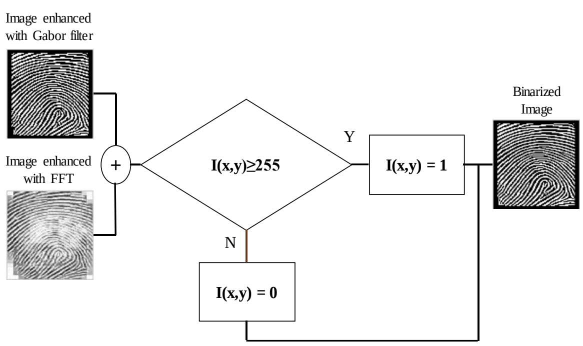

Fig. 3. Binarization Algorithm

2) Thinning

With binarized-based image, the thinning algorithm makes the ridge thickness 1 pixel width to extract features easier. The conditions are proposed by L.Lams [14]:

Condition C1:

$$X_H(p)=1 \text{ with } X_H(p)=\sum_{1}^{4} b_i$$

$$b_i=\begin{cases}1 & if\ x_{2i-1}=0\, and\, (x_{2i}=1\, or\ x_{2i+1}=1)\\ 0 & otherwise\end{cases}$$

Condition C2:

$$2\le min\{n_1(p),n_2(p)\}\le 3 \text{ with}$$

$$n_1(p)=\sum_{i=1}^{4} x_{2k-1}\vee x_{2k},\ n_2(p)=\sum_{i=1}^{4} x_{2k}\vee x_{2k+1}$$

Condition C3:

$$(x_2\vee x_3\vee \overline{x_8})\wedge x_1=0$$

Condition C3’:

$$(x_6\vee x_7\vee \overline{x_4})\wedge x_5=0$$

Thinning procedure is done as following algorithm:

Deviding the image into two separate subfields:

- In the first subfield, pixel p is removed if all three conditions C1, C2 and C3 are satisfied.
- In the second subfield, pixel p is removed if all three conditions C1, C2 and C3’ are satisfied.

3) Minutiae Extraction

The minutiae are extracted from thinned image, including ending points and bifurcation points. In this paper, the proposed algorithm to mark minutiae is Crossing Number [1]. Assuming, p is a point on the thinned ridge line and $p_0, p_0, p_1, \ldots, p_7$ are 8-neighborhood of p, the crossing number *cn(p)* of pixel p is defined as half the sum of the differences between pairs of adjacent pixels in the 8-neighborhood:

$$cn(p)=\frac{1}{2}\sum_{i=1}^{8}\left|val(p_{i\,\mathrm{mod}8})-val(p_{i-1})\right| \tag{11}$$

With $val(\mathrm{p})\in\{0,1\}$ is the pixel value.

- cn(p) = 1, p is a ridge ending point.
- cn(p) = 2, p is an intermediate ridge point.
- cn(p) = 3, p is a bifurcation point.
- cn(p) > 3, defines a more complex minutiae (e.g., crossover).

However, after the minutiae extraction process, there are always many false minutiae. So it must be through some algorithms to exclude false minutiae before saving to database or recognition process.

C. Training Artificial Neural Network for recognition

Artificial Neural Network is a system including many simple processing elements (also called neurons) operating in parallel and connected by weighted links to excite or inhibit neurons. There are a lot of different neural network architectures such as feed-back network, feed-forward network and self-organizing network [12][15].

1) Training Neural Network

a) Learning methods

The basic feature of the neural network is capable of learning, capable of reproducing the learned image and data. In the learning state, information is transmitted in both directions several times to learn the weights. Three main learning methods are supervised learning, unsupervised learning and reinforcement learning [15].

b) Training algorithm

In this paper, multi-layer perceptron with back-propagation algorithm is used to train the neural network [12].

The training procedure is the process of training templates $X_s = \{x_1, x_2, \ldots, x_n\}$ to find desired values $T_s = \{t_1, t_2, \ldots, t_n\}$.

- Perceptron procedure:

The output value at j[th] neuron of any layer:

$$out_j = \frac{1}{1+e^{-net_j}} \text{ with } net_j = \sum_{i=0}^{m} w_{ji} x_{ji} \qquad (12)$$

With w_{ji}, x_{ji} corresponding to link weight and input value from i[th] input to j[th] neuron, m is number of elements of previous layer.

- Back-propagation procedure:

At each output neuron k, calculate error value,

$$\delta_k = o_k\left(1-o_k\right)\left(t_k - o_k\right) \qquad (13)$$

With t_k is k^{th} desired output value.

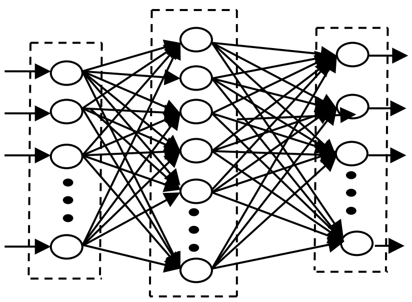

Fig. 4. The architecture of Artificial Neural Network using multi-layer perceptron

For each neuron in hidden layer,

$$\delta_h = o_h\left(1-o_h\right) \sum_{k \in outputs} w_{jk}\delta_k \qquad (14)$$

With *outputs* are a set of neurons of output layer, w_{jk} is link weight from k neurons of output layer to j[th] neuron of hidden layer.

The update process for the weights:

$$w_{jk} = w_{jk} + \Delta w_{jk} \qquad (15)$$

If let η is learning weight, $\Delta w_{jk} = \eta\delta_h o_k$. After update these weights, then the template in the set X_s is put into the network, this process is done until error value $E_d < \varepsilon$, with ε is a desired error value:

$$E_d = \frac{1}{2}\sum_k \left(t_k - o_k\right)^2 \qquad (16)$$

With t_k is desired output value of neuron k for training template d, o_k is real output value of neuron k.

2) Neural Network configuration

To optimize the network, we initialized the original neural network with following parameters: inputs are feature vectors containing minutiae extracted in feature extraction phase, 100 neurons in hidden layer and 50 neurons in output layer, using log-sig function in neurons of both layers. The learning rate of the neural network is set to $\eta = 0.01$ and the mean square error value is MSE = 0.001. In addition, the weights and biases of all units are randomly initialized before training the neural network.

III. MATCHING METHOD

This method is based on the template and input fingerprint. Let T and I be the representation of the template and input fingerprint, respectively. The presentation here is a feature vector (of variable length) whose elements are the fingerprint minutiae. Each minutiae may be described by a number of attributes, including its location in the fingerprint image, orientation, a weight based on the quality of the fingerprint image in the neighborhood of the minutiae, and so on. Matching algorithm includes 2 process: alignment and matching.

A. Alignment

Fingerprint pattern matching is understood as setting a fingerprint template in need of comparing with samples that will need to match in a position so that there is consistency between them by a threshold.

Alignment algorithm is as follows:

- Choose a pair of minutiae of input and template fingerprint.
- Rotate templates woth respect to inputs.
- Calculate satisfied extent S_p of two samples template and input until getting a given threshold T_r

$$S_p = \frac{\sum_{i=0}^{m} x_i X_i}{\sqrt{\sum_{i=0}^{m} x_i^2 X_i^2}} \qquad (17)$$

B. Matching

After we got two sets of transformed minutiae points from alignment stage, we use the elastic match algorithm to count the matched minutiae pairs. Rotating operator RA is defined:

$$RA = \begin{pmatrix} COS\theta & sin\theta & 0 \\ cos\theta & -sin\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (18)$$

Next, we count minutiae pairs allowed wrong in term of position and angle in the limitation of ε.

IV. Experimental results and evaluation

A. Database

The database used for fingerprint recognition includes training sets and testing sets based on three database sets FVC2000, FVC2002, FVC2004. To evaluate the system, the database was built including three datasets in three different conditions based on DB2A of FVC2000 [9], DB2A of FVC2002 [10] and DB2A of FVC2004 [11] as Table I. The detail description of training and testing datasets are in Table II.

B. Evaluation criterias

In this study, the confusion matrix are used to evaluate the efficiency of the system: *Recall* [%], Equal Error Rate (*EER*) [%] and Accuracy (*ACC*) [%]. [1]

- *Recall*: this criteria is used when a part of database used for training and the rest of database used for testing. The testing sets don't have the occurrence of strangers.

$$Recall = \frac{\text{Total number of recognized fingerprint samples}}{\text{Total number of fingerprint samples}} \quad (19)$$

- *EER*: with the occurrence of strangers, the system using a threshold to determine whether that person is accepted or not. Table III is the classification matrix for two type of objects, with P represent the competent person and N represent the incompetent person. From the Table III, the number of times of incompetent person accepted by the system is FN. Similarly, we can find the definition of TN, TP and FP. False Acceptance Rate (FAR) is defined as the ratio of the number of impostor images considered as authentic by the system to the total number of impostor images. False Rejection Rate (FRR) is defined as the ratio of the number of authentic images not considered qualified by the system to the total number of authentic images. FAR and FRR is calculated as following formula:

$$FAR = \frac{FN}{TN+FN}, \quad FRR = \frac{FP}{TP+FP} \quad (20)$$

With different thresholds, FAR and FRR have the corresponding values. When FAR and FRR are equal, we call it Equal Error Rate (EER). The reliability of the system is generally measured in terms of EER.

- *ACC*: *ACC* is the rate that the system properly recognize objects. It is calculated by the formula:

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad (21)$$

TABLE I. CHARACTERISTIC OF FVC DATABASES

Database	**Sensors**	**Image size**	**Number of images**	**Resolution**
FVC2000 DB2A (type 1)	Low-cost Capacitive Sensor	256x364	100x8	500 dpi
FVC2002 DB2A (type 2)	Optical Sensor FX2000	296x560	100x8	569 dpi
FVC2004 DB2A (type 3)	Capacitive Sensor URU 4000	328x364	100x8	500 dpi

TABLE II. THE STRUCTURE OF THE TRAINING AND TESING DATABASE

Datasets	**Training image**	**Testing image**
Set 1	50 persons x 3 images = 150 images of type 1	50 images of type 1; 25 images of type 2
Set 2	50 persons x 3 images = 150 images of type 2	50 images of type 2; 25 iamges of type 3
Set 3	15 persons x 3 images = 45 images of type 1; 45 images of type 2; 20 persons x 3 images = 60 images of type 3	40 images of type 1; 40 images of type 2; 70 images of type 3

TABLE III. THE CLASSIFICATION MATRIX FOR TWO TYPES OF OBJECTS

		State	
		Accept	**Reject**
Type of objects	Possitive (P)	True (T)	False (F)
	Negative (N)	False (F)	True (T)

TABLE IV. RECOGNITION RESULTS

Datasets	**Methods**	
	Matching (%)	**Neural network (%)**
Set 1	88	92
Set 2	92	94
Set 3	86	90

TABLE V. RELIABILITY RESULTS

Datasets	**EER**		**ACC (%)**	
	Matching	**Neural network**	**Matching**	**Neural network**
Set 1	0.2	0.05	85.33	90.67
Set 2	0.12	0.04	88	92
Set 3	0.27	0.12	82.67	88

C. Analysis results

1) System performance

The efficiency of the considered systems is correct recognition for the objects in the database (no occurrence of stranger). Table IV presents proper recognition rate when the system was tested on three datasets with different condition using matching method and neural network.

From Table IV, it indicates the performance of neural network better than matching method in the same dataset. With the most simple dataset (set 2), matching method achieves high performance. However, it is still worse than the neural network. When the input data varies, recognition performance using neural network is also more stable than the matching method.

2) System reliability

The efficiency of the system is considered with the occurrence of stranger not in the database. EER and ACC are used to evaluate the reliability of the systems.

The system using neural network works quite well with the different datasets. EER is low, accuracy higher and more stable than matching method with increasing of the complexity of databases. EER of the system using neural network varies from 0.04 to 0.12 with the difference of 0.08. That difference is much smaller than EER of matching method (0.15).

V. CONCLUSION

With three datasets sampled in different condition as well as evaluation methods indicated in this paper, performance of the system using neural network is over 90% with reliability better than matching methods in almost of cases. However, to get the recognition result with higher performance needs to optimize the structure and neural network parameters which requires longer training and testing time.

However, in this paper, the recognition results were done in particular conditions. Applying the system into reality will have currently challenges or problems such as environment parameters, image quality, then performance and reliability of the system significantly decrease because of occurrence of many fake points in thinning procedure, brightness change effecting segmentation process. To overcome these limitations, the feature extraction algorithm on the gray image as well as the segmentation algorithm based on ridge direction is used. In addition, consideration use another models such as K-nearest Neighbor (KNN) or Support Vector Machine (SVM) to recognize in the future. Building a good database also contribute to increase the recognition performance.

ACKNOWLEDGMENT

We would like to give a special thanks Duy Huu Le for his instructions that has supported us in the process to complete this research.

REFERENCES

[1] David Maltoni, Dario Maio, Anil K.Jain, Salid Prabhakar, "*Handbook of Fingerprint Recognition*", Second Edition, Springer, 2009.

[2] Dorasamy, Kribashnee, *"Fingerprint Classification Using a simplified Rule-set Based On Directional patterns And Singularity feartures"*, IEEE 2015 International Conference on Biometrics, pp. 400-407, May 2015.

[3] Akhtar, Zahid, *"Correlation based fingerprint liveness detection"*, IEEE, pp. 305-310, May 2015.

[4] Ozkan, Savas, *"Fingerprint recognition with geometric relation of minutiae points"*, IEEE, pp. 875-878, May 2015.

[5] Sachin Harne, Prof. K. J. Satao, *"Minutiae Fingerprint Recognition Using Hausdorff Distance"*, Sachin Harne et al UNIASCIT, Vol. 1, pp. 16-22, 2011.

[6] R. Dharmendra Kumar, kaliyaperumal Karthikeyan, T.Ramakrishn, "*Fingerprint Image Enhancement Using FTT for Minutia Matching with Binarization*", International Journal of Engineering Research & Technology (IJERT) Vol. 1, Issue 8, October 2012.

[7] J. S. Chen, Y. S. Moon, K. F. Fong, "*Efficient Fingerprint Image Enhancement for Mobile Embedded Systems*", Biometric Authentication Lecture Notes in Computer Science Vol. 3087, pp. 146 – 157, Sept 2008.

[8] IG. Babatunde, AO. Charles, AB. Kayode, O. Olatubosun, *"Fingerprint Image Enhancement: Segmentation to Thinning"*, International Journal of Advanced Computer Science and Applications, Vol. 3, pp. 8-14, 2012.

[9] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K.Jain, *"FVC2000: Fingerprint Verification Competition"*, IEEE PAMI, Vol. 24, No. 23, pp. 402-412, March 2002.

[10] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K. Jain, *"FVC2002: Second Fingerprint Verification Competition"*, in Proceedings 16th International Conference on Pattern Recognition, Québec City, Vol. 3, pp. 811-814, 2002.

[11] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K. Jain, *"FVC2004: Third Fingerprint Verification Competition",* in Proceedings 1st International Conference on Biometric Authentication, LNCS 3072, pp. 1-7, 2004.

[12] Atanu Chatterjee, Shuvankar Mandal, G. M. Atiqur Rahaman, and Abu Shamim Mohammad Arif, *"Fingerprint Identification and Verification System by Minutiae Extraction Using Artificial Neural Network"*, JCIT, Vol. 01, Issue 01, 2010.

[13] Gualberto Aguilar, Gabriel Sánchez, Karina Toscano, Mariko Nakano-Miyatake, Héctor PérezMeana, *"Automatic Fingerprint Recognition System Using Fast Fourier Transform and Gabor Filters",* Científica Vol. 12 Núm. 1 pp. 9-16, 2008.

[14] L.Lam, S. Lee, C.Y.Suen, *"Thinning Methodologies - A Comprehensive Survey"*, IEEE Transactions On Partern Analysis and Machine Intelligence, Vol. 14, No. 9, Sept. 1992.

[15] Kalpna Kashyap, Meenakshi Yadav, *"Fingerprint Matching Using Neural Network Training"*, International Journal Of Engineering And Computer Science, Volume 2, Issue 6 June, pp. 2041-2044, 2013.