Fingerprints detection using neural networks suitable to physical changes of fingerprint

Detección de huellas dactilares usando redes neuronales adecuadas a cambios físicos de la huella

M. R. Díaz[1*], M. A. Dueñas[1], K. D. Acevedo[1]

Received: 15 July 2017

Accepted: 15 September 2017

Summary

This working paper shows the results of a finished research, using image processing techniques to improve the fingerprint obtained from a database, where the image is normalized and segmented to get only the interesting section of the image with the fingerprint; then Gabor`s filter is applied and it corrects defects in ridges and valleys, allowing continuity, that way if the fingerprint has a physical defect, the filter is able to correct it, as long as the segment orientation is correct. Once improved the fingerprint, it is binarized and thinned for minutiae extraction. The false minutiae are filtered and eliminated in order to ensure the operation of the algorithm. Finally, it is necessary a training with the minutiaes of all fingerprints in the database, in order to this determines individually which user belongs the fingerprint entered. As a result, the system has a reliability of 81% of the process, being crucial the pre-processing part in order to guarantee the correct extraction of the characteristics of fingerprints.

Keywords*: Gabor filter, image processing, ridges, minutiae, physical defect.*

Resumen

En este trabajo se muestra los resultados generados de una investigación terminada, usando técnicas de procesamiento de imágenes para mejorar la huella obtenida de una base de datos donde se normaliza y segmenta la imagen para obtener solo la región de interés en la imagen con la huella; posteriormente se aplica el filtro Gabor, que corrige los defectos en las crestas y los valles, permitiendo mantener la continuidad en los mismos, y de esta manera, si la huella

[1] Unidades Tecnológicas de Santander

[*] Correspondence author MsC Telematics – mabelrociod1@yahoo.com

Díaz M.R., Dueñas M.A., Acevedo K.D. 2017. Fingerprints detection using neural networks suitable to physical changes of fingerprint. Revista CINTEX. Vol. 22 nº2, pp. 35-50

tienen un defecto físico, el filtro es capaz de corregirlo, siempre y cuando la orientación del segmento sea correcta. Una vez mejorada la huella se puede binarizar y adelgazar para la extracción de minucias. Las minucias son filtradas para eliminar todas las falsas minucias y asegurar el funcionamiento del algoritmo. Por último se hace un entrenamiento con las minucias de todas las huellas en la base de datos, para que esta determine individualmente a que usuario pertenece la huella ingresada. Los resultados arrojan una confiabilidad de un 81% del proceso, siendo vital la parte de pre-procesamiento para garantizar la correcta extracción de las características de las huellas.

Palabras Claves: Filtro Gabor, procesamiento de imágenes, crestas, minucias, defecto físico.

1. Introduction

Fingerprint is a unique and different pattern of a human finger. One of the characteristics of the fingerprints is their individuality, even the same individual has different fingerprints in each of their fingers. This feature allows the use of fingerprints as one of the most reliable recognition methods used in various applications.

Multiple approaches to the treatment of fingerprints using neuronal networks have been proposed in the literature, being one of the pioneers in the Jain classification [1], which utilizes two stages in the use of neural networks. Initially used for the detection of false minutiae, and then for the classification of the fingerprints in five groups depending on the orientation of the fingerprint. This classification is given by *NIST4*, which is a database of the FBI with 4000 fingerprints, classified in those five groups. A similar approach has been used to perform the classification, using neuronal networks for the detection of patterns in the fingerprints and thus classify them [2]-[3].

The use of neural networks for the classification of fingerprints is the most used approach, some authors have used this approach in classification depending on gender, that is, depending on the fingerprint it can be said if it is a man or a woman [4]-[5]. On the other hand, the *feedforward architecture* has been implemented for the training of the neuronal networks, whose objective is to thin the fingerprint, ensuring the connectivity of valleys and ridges [6]. In order to have a correct extraction of the characteristics and that in the thinning process information is not lost, it is necessary to have a pre-processing of the robust footprint for which the *Gabor filter* is used [7]-[8]-[9], with which the greater accuracy in the improvement of the image with the fingerprint is obtained. In addition, the use of neuronal networks for people recognition has several approaches. Werner extracts fifteen minutiae of the area of interest which is located in the center of the fingerprint and with this, trains the neuronal networks, using the *architecture of the multilayer perceptron* [10]-[11]. Likewise, algorithms that use fingerprint minutia have been developed, but in this case it only takes the terminations for authentication, obtaining more than 90% of accuracy when a fingerprint matches the database [12].

Therefore, in this paper a biometric digital system is developed based on fingerprints that aims to provide greater ease and safety when accessing a site, based on neural networks so that if the fingerprint is physically modified, there is no impediment for the user to be identified. The techniques to improve the quality of the fingerprint image, extraction of minutiae and training of neural networks are first addressed, and then in the results section, the approaches achieved with the algorithms proposed in this research are shown.

2. Materials And Methods

a. Fingerprint detection improvement

i. Standardization

The mean and the variance of the images obtained by the sensor depend on the particular characteristics of it and the pressure level exerted in by the user on the sensor; the normalization stage aims to set the mean and the variance of the image to a fixed level [7]. Being I(*i,j*) the gray level value in the pixel (*i,j*) , *M* and *VAR* denote the estimated mean and the variance I respectively, and G(*i,j*) denote the standardized gray level value to the pixel (i,j). Hong [7] defines the standardized image as it is shown in (1) .

$$G(i,j) = \begin{cases} M_0 + \sqrt{\dfrac{VAR_0(I(i,j)-M)^2}{VAR}} & if\ I(i,j) > M \\ M_0 - \sqrt{\dfrac{VAR_0(I(i,j)-M)^2}{VAR}} & In\ other\ case \end{cases} \tag{1}$$

Where M_0 and VAR_0 are the mean and the desired variance values, respectively. Standardization is an operation in pixels. That does not change the clarity of the ridge and valley structures. The main objective of normalization is to reduce the variations in the gray level values along the ridges and valleys, facilitating the following processing stages.

ii. Segment

It consists on separating the area of the image containing the fingerprint (where the valleys and ridges are present) which is really the important thing regarding the background, which lacks of information for the calculation of the minutiae [7]. In a fingerprint image the background regions generally represent a gray scale, while the foreground regions have a very high variance, therefore, it is used a method based on the variance threshold. The image is divided into blocks and variance of grays is calculated for each block in the image, if the variance is less than the global one, it is assigned to be a background region otherwise, it is assigned to be a part of the foreground. The variance of the gray level for a block of size W x W pixels is defined according to (2).

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

Where is the variance for the block is the value of the gray level in the pixel and is the average value of the gray level of the block .

iii. Orientation

The orientation field of a fingerprint image defines the orientation of each of the ridges contained in the fingerprint [8]. The angles of the orientation field represent the direction in which the ridges of the fingerprint flow. The orientation estimation is a fundamental step in the process of improving the fingerprint, which is obtained by dividing the image by blocks of size pixels, calculating the orientation of the central pixel and applying this orientation to the entire block.

Before calculating the orientation of the central pixel, the gradient in x most be calculated, and as shown in (3) and (4), applying the y masks, for this the convolution of each pixel is done with a mask of 3x3.

$$
\begin{aligned}
Sx(i,j) &= \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\
Sy(i,j) &= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}
\end{aligned}
\tag{3}
$$

$$Vx(i,j) = \sum_{u=i+\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2\,\partial x(u,v)\partial y(u,v) \tag{4}$$

$$Vy(i,j) = \sum_{u=i+\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \partial x^2\,(u,v) - \partial y^2(u,v) \tag{5}$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\frac{Vy(i,j)}{Vx(i,j)} \tag{6}$$

Where $\theta(i,j)$ is the orientation of the block centered on the pixel (i,j).

The orientation calculated according to (5) is a quite acceptable result, however, the elimination of errors can be done by performing a Gaussian filter on the obtained result. To do this, two vectors $\Phi x(i,j)$ and $\Phi y(i,j)$, are calculated according to (7) and (8).

$$\Phi x(i,j) = \cos\,(2\theta(i,j)) \tag{7}$$

$$\Phi y(i,j) = \sin\,(2\theta(i,j)) \tag{8}$$

Where Φx and Φy are the components X and Y of the vector field, respectively. After calculating the vector field, the Gaussian smoothing is performed with (9) and (10).

$$\Phi' x(i,j) = \sum_{u=-\frac{w\Phi}{2}}^{\frac{w\Phi}{2}} \sum_{v=-\frac{w\Phi}{2}}^{\frac{w\Phi}{2}} G(u,v)\Phi x(i-uw, j-vw) \tag{9}$$

$$\Phi' y(i,j) = \sum_{u=-\frac{w\Phi}{2}}^{\frac{w\Phi}{2}} \sum_{v=-\frac{w\Phi}{2}}^{\frac{w\Phi}{2}} G(u,v)\Phi y(i-uw, j-vw) \tag{10}$$

Where G is the low pass Gaussian filter of $w\Phi$ x $w\Phi$ size. Finaly, the orientation value is calculated following (11).

$$O(i,j) = \frac{1}{2}\tan^{-1}\frac{\Phi' x(i,j)}{\Phi' y(i,j)} \tag{11}$$

iv. Gabor filter

The Gabor filter allows to keep and improve line structures with certain orientation and a given frequency in the domain of the figure [8]. The Gabor filter is the real part of the Gabor function, which is composed by a cosine wave modulated by a Gaussian. The general formula of the symmetric Gabor filter comes by (12).

$$G(x,y;\theta,f) = exp\left\{-\frac{1}{2}\left[\frac{x^2{}_\theta}{\sigma^2{}_x} + \frac{y^2{}_\theta}{\sigma^2{}_y}\right]\right\}\cos(2\pi f x\theta) \tag{12}$$

$$x_\theta = x\ cos\theta + y\ sin\theta \tag{13}$$

$$y_\theta = -x\ sin\theta + y\ cos\theta \tag{14}$$

Where θ is the Gabor filter orientation, f is the frequency that in this case 0,15 is used, σ_x and σ_y are the standard deviations of the Gaussian envelope through the X and Y axes, respectively. The parameters selection from σ_x and σ_y are empirically fixed as it is shown in the Hong's article [7], where those have a fixed value equal to 4,0. To filter the image, a convolution of each pixel is performed (i,j) in the image, the corresponding orientation value $O(i,j)$ and the frequency value.

The G Gabor filter application to obtain the E image is shown in (15).

$$E(i,j) = \sum_{u=-\frac{Wx}{2}}^{\frac{Wx}{2}} \sum_{v=-\frac{Wy}{2}}^{\frac{Wy}{2}} G(u,v;O(i,j),F)N(i-u, j-v) \tag{15}$$

Where O is the image orientation, F is the frequency, N is the improved image and Wx and Wx are the width and height of the Gabor's.

v. Binary

The image binary process consists in leaving all the values in 0 and 1, for this, the pixels which had a positive value were left, to which assigned value was 1, and, the ones who had a zero-negative value respectively, as its shown in (16).

$$G(i,j) = \begin{cases} 1 & if\ I(i,j) > 1 \\ 0 & Demás \end{cases} \tag{16}$$

vi. Thinning

The thinning is a morphologic operation performed before the minutiae extraction, this operation continuously erode far from the forefront until the pixels are a pixel of width.

The Guo & Hall algorithm [13] works through the performance of a wipe on the multiple image pixels, thus, the matching pixels with the criteria set of connected areas are eliminated; [14] the operation is performed using the two sub-iterations in each of them, the algorithm correlates the binary image with a neighbor pixels mask, then, based on the pixel elimination set, it is checked if the central pixel must be eliminated in that sub-iterations.

b. Minutiae extraction

The most common used minutiae extraction method is the *Crossing Number*, this method implies the usage of the skeleton image in which the crested flow pattern is connected eight times [8]. The minutiae are extracted through the neighbor pixels exploration of each crest pixel in the image using a 3 x 3 block. Next, the CN value is calculated, which is defined as the sum differences half between adjacent pair of pixels.

The *Crossing Number* (*CN*) method is used to perform the minutiae extraction; this method extracts the endings and forks of the skeleton image crest, looking into the neighbor pixels of each pixel using a 3 x 3 window. The CN for a P crest pixel is given by (17).

$$CN = \frac{1}{2}\sum_{i=1}^{8} |P_i - P_{i+1}|,\ \ P_9 = P_1 \tag{17}$$

Where P_i is the pixel value of the P near pixels. For a P pixel, its eight neighbor pixels are scanned counterclockwise of the following way:

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

After calculating the CN for a crest pixel, the pixel could be classified according the property of its CN value with a CN of one corresponding to a crest ending and a CN of three corresponding to a bifurcation.

i. False minutiaes elimination

After finishing the minutiae extraction stage, the result has been analyzed and proven that several found minutiae are wrong, the minutiae that will be eliminated are the following:

- When the bifurcation is part of a line which length its smaller than N pixels and when an ending is near the N pixels bifurcation.
- If there are two close bifurcations that that form a loop and this difference is less than N pixels, these minutiae are considered false.

c. Neural networks training

The usage of the *feed-forward back propagation (newff)* is proposed to solve the individuals classification and identification problem as Yang [15]. The selected neural network architecture, is based in three hidden layers with 24, 23 and 25, neurons each layer respectively.

To perform the training, a database of 15 fingerprints per individual were taken, of which 10 fingerprints to train and 5 fingerprints to check such training were used. Trainings for a network per individual in the database were performed, because when one single neural network was done for all the users, the network never converged on a solution. The block diagram that represents the system performance is shown in Fig1.

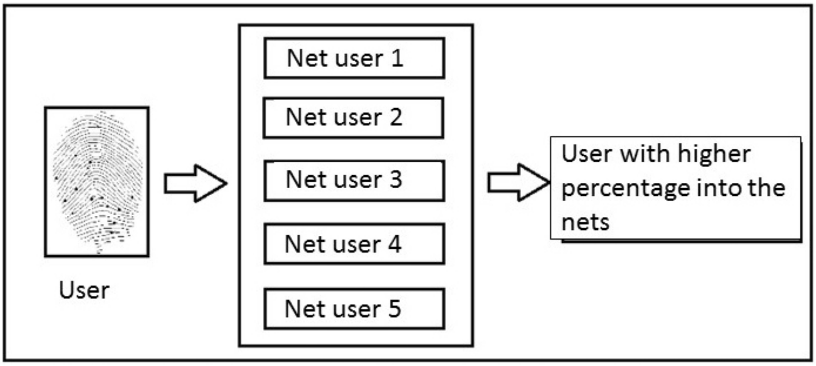

FIGURE 1. CLASIFICATION SYSTEM ARCHITECTURE

SOURCE: AUTHORS

The used neural network architecture was composed by 5 layers with three hidden layers, plus the entrance and exit layer, the three layers have 24, 23 and 25 neurons each respectively.

3. Results

For the performance and efficiency analysis of the proposed development, an algorithm which has been implemented in Python programming language is proposed, using several libraries for the image treatment such as OpenCV, PIL, and SCIPY, which allow the image operation, matrices usage and complex mathematical calculations with ease. Likewise, such algorithm has been tested in a Laptop and in an embedded system, which works with different operative systems and characteristics. On one hand, the Laptop with Core i5 at 2.5 GHz, 8GB of RAM and a Windows 7 operative system is used, while the embedded system is a raspberry which has a 4× ARM Cortex-A53 at 1.2GHz processor, 1GB of RAM LPDDR2 (900 MHz) and a Raspbian JESSIE operative system.

The proposed algorithm pseudocode is shown in Table I.

TABLE 1. PROPOSED ALGORITHM

Definitions	Elements
Include libraries	PIL, numpy, matplotlib, cv2, scipy, math, neural networks
Image reading	
Pre-processing	
	Standardize
	Segment
	Find Orientation
	Gabor Filter
	Binary
	Thinning
Characteristics extraction	
	Minutiae extraction
	Eliminate false minutiaes
Neural networks	
Result	

SOURCE: AUTHORS

a. Pre-processing of the fingerprint

Fig. 2 illustrates the result of the binary application and a fingerprint picture thinning without any stage of pre-processing. In contrast with Fig. 3, the binary picture in Fig. 2 is not well connected and contains significant amounts of noise and damage elements. Therefore, when thinning is applied to this binary picture, the results shows that the minutiae precise extraction of this picture would not be possible due to the big number of produced noise. So, it can be proved than the use of a series of stages of picture improvement before the thinning is effective to facilitate the minutiae reliable extraction.

On Fig. 5 each one of the pre-processing stages can be evidenced, by watching on Fig. 5g the finger print after the Gabor filter is applied, then on 5c binary and finally on 5d the picture thinned, ready for the minutiae's extraction. Unlike Fig. 2 and Fig. 3 where the process of binary and thinning with and without pre-processing stage are shown.

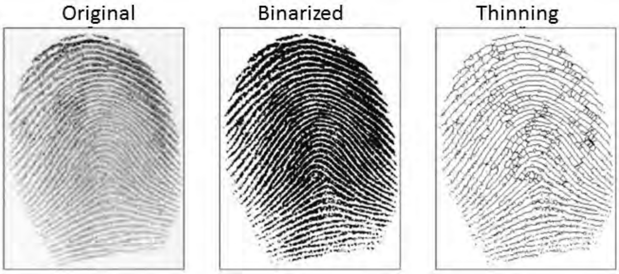

FIGURE 2. RESULTS OF A IMAGE WITHOUT PRE-PROCESSING

SOURCE: AUTHORS

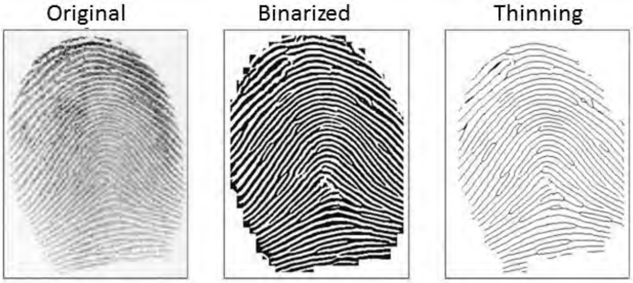

FIGURE 3. RESULTS OF A IMAGE WITH PRE-PROCESSING

SOURCE: AUTHORS

b. *Minutiae extraction*

The minutiae extraction has been made using the *crossing number (CN)* algorithm, then a false minutiae elimination method has been proposed, leaving only the bifurcations that are alone in a ten pixels window, that way, the possible noises in the minutiae extraction are eliminated. Fig. 4 shows the minutiae extraction and the false minutiae elimination, leaving only the bifurcations, allowing, to have the less among of parameters, so in the classification stage the neuronal network algorithm can converge easily.

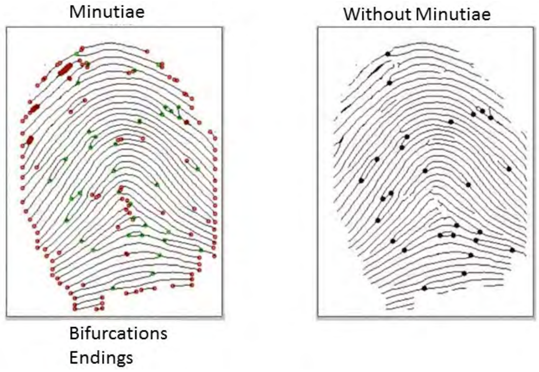

Figure 4. False minutiae extraction and elimination

Source: authors

c. Confusion matrices in the detection algorithm

For the neuronal networks training, each one of the fingerprint data base minutiae were used as an entry parameter. A data base with 6 users were used, with 15 fingerprints per user, of which initially five users were trained as shown in Table II, that is, a neuronal network per user, specifying which fingerprint belonged to the user and which didn't. After the training was concluded for the five users, it was trained with the sixth user to observe what reliability it had to add a new user without having to train all the networks again as shown in Table III.

Table ii. General confusion matrix of the five users

	U1	U2	U3	U4	U5
U1	13	2	6	1	1
U2	4	12	0	0	0
U3	2	2	12	2	0
U4	0	2	2	12	2
U5	1	4	0	1	14

Sourse: authors

Table III. General confusion matrix of all users plus an additional user

	U1	U2	U3	U4	U5	U6
U1	13	2	6	1	7	1
U2	4	12	0	0	1	0
U3	2	2	12	2	4	0
U4	0	2	2	12	0	2
U5	3	0	4	3	14	2
U6	1	4	0	1	4	14

Source: Authors

Additionally, a database with cut fingerprints was used to prove the validity of the Gabor filter and the general algorithm, in order to evidence that the proposed system responds to physical changes in the fingerprint. On Fig. 5 the pre-processing system performance is shown, by eliminating the cut thanks to one of the filter's input parameters is the general orientation of the fingerprint, whereby, filtering eliminates the imperfection.

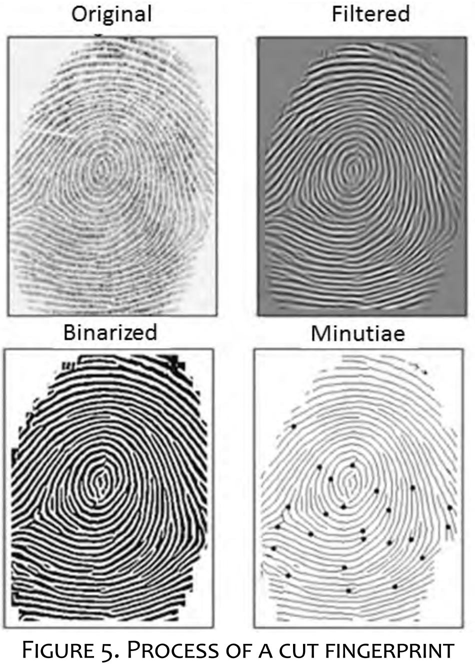

Figure 5. Process of a cut fingerprint

Source: Authors

The confusion matrix made for this data base is shown in Table IV.

TABLA IV. CONFUSION MATRIX OF CUT FINGERPRINTS

	U1	U2	U3	U4	U5	U6
U1	2	1	0	0	2	1
U2	1	0	0	0	0	0
U3	0	0	1	0	1	0
U4	0	0	1	2	0	0
U5	0	0	1	1	1	0
U6	0	1	0	0	0	1

SOURCE: AUTHORS

4. Conclusions

In this paper, an algorithm that allows detecting fingerprints using neural networks was developed, based on various strategies to the pre-processing of the image with the fingerprint, such as segment, Gabor filter and false minutiae elimination. This set of strategies allows the algorithm to present robustness. When segmenting the image only the parts of the image that contains the fingerprints are passed, which allows the next process to operate faster because they don't have to do a full image analysis.

Gabor filter allows to rebuild the fingerprint even though the noise the image might have, as long as the directions of the valleys and the crests are defined and show continuance, which allows the filter rebuild the way basing on the orientation and separation between the valley and the crest.

In the development of this project, the calculation of the distance between the valley and the crest was omitted, known as the segment's frequency in which the filter wants to be applied, due to the frequency measurement was always affected by the excess noise, and it caused the filter not to be applied properly, therefore it was decided to take a fixed frequency for the Gabor filter's operation, generating good results, as long as the segment orientation can be recognized and the filter can be applied in the right direction, correcting the mistakes in the continuity, on the contrary, if the right orientation isn't extracted, the filter will finish damaging the filtered segment.

Besides, in order to reduce the among of input data in the neuronal networks, all the false minutiae were eliminated and only the bifurcations were took to make the training, which allowed to the different neuronal networks to learn correctly which fingerprint belonged to which persons; one of the processes were treating to do a general neuronal network for all the fingerprints, which wasn't possible because during the multiples trainings the networks never converged to a solution, that is, they didn't train property, due to that, it was decided to make a neuronal network per person, which allowed that each network learned to identify the information obtain from each fingerprint and that way will determinate to which person belonged each one.

On the other hand, the execution time of the algorithm oscillate a lot when being executed in a computer and in an embedded system, because in the computer the algorithm spends an average of 14 seconds and in the Raspberry approximately 133 seconds per execution. This is possible to prove thanks to that the code was developed in PYTHON and the majority of the functions pass transparently.

5. References

[1] A. K. Jain, S. Prabhakar, and L. Hong, "A Multichannel approach to fingerprint classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 348–359, 1999.

[2] C. L. Wilson, G. T. Candela, and C. I. Watson, "Neural Network Fingerprint Classication," *J. Artical Neural Networks*, vol. 1, no. 2, 1993.

[3] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Second Edi. Lodon, 2009.

[4] S. Gupta and A. P. Rao, "Fingerprint Based Gender Classification Using Discrete Wavelet Transform & Artificial Neural Network," vol. 3, no. 4, pp. 1289–1296, 2014.

[5] B. Vanathi, K. Shanmugam, and V. Rymand Uthairaj, "A secure M-commerce architecture for service provider to improvize quantity and quality of the products using fingerprint authentication and gender classification," *Asian J. Inf. Technol.*, vol. 15, no. 2, pp. 232–242, 2016.

[6] P. G. Goyal and R. Luthra, "Neural network Based approach for Image skeletonization on Gurumukhi Characters," vol. 3, no. 2, pp. 1325–1329, 2015.

[7] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, 1998.

[8] R. Thai, “Fingerprint Image Enhancement and Minutiae Extraction,” p. 63, 2003.

[9] A. L. Velasco, “Tratamiento dıgital de huellas dactilares,” p. 38.

[10] G. Á. Werner and L. H. P. D, “Tuning an artificial neural network to increase the efficiency of a fingerprint matching algorithm,” pp. 105–109, 2016.

[11] Karam ekhlas, J. Hanady, and Ali ekbal, “Fingerprint Recognition Using Discrete Wavelet Transform And Neural Network For Estimation Rotation Region,” *J. Eng. Dev.*, vol. 18, no. 3, pp. 55–66, 2014.

[12] A. Agarwal, A. K. Sharma, and S. Khandelwal, “Fingerprint Recognition System by Termination Points Using Cascade-Forward Backpropagation Neural Network,” pp. 203–211, 2016.

[13] Z. Guo and R. W. Hall, “Parallel thinning with two-subiteration algorithms,” *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[14] L. Lam, S.-W. Lee, and C. Y. Suen, “Thinning methodologies-a comprehensive survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, 1992.

[15] G. Z. Yang, “A Robust Handwriting Recognition System in MATLAB using Artificial Neural Network,” University of Malaya.