International Journal of Innovative Computing, Information and Control
Volume **15**, Number **6**, December **2019**

 ISSN 1349-4198
pp. **2183–2196**

HUMAN IDENTITY VERIFICATION VIA AUTOMATED ANALYSIS OF FINGERPRINT SYSTEM FEATURES

Amal Ahmad[1], Shereen Ismail[2] and Mohammad Abdul Jawad[3]

[1]Department of Electrical Engineering
[3]Department of Basic Sciences
Al-Zaytoonah Private University of Jordan
Amman 11733, Jordan
{ amal.q; m.abduljawad }@zuj.edu.jo

[2]Department of Computer Science and Engineering
American University of Ras Al Khaimah
PO Box: 10021, American University of Ras Al Khaimah Road, Ras Al Khaimah, UAE
shereen.subhi@aurak.ac.ae

Received March 2019; revised July 2019

Abstract. *Automatic biometric recognition systems are considered recently a milestone of many potential commercial and civilian applications. Fingerprint, is one of the biometric traits, which has proved its reliability as a distinctive feature for an individual. In this paper, a fingerprint recognition system has been proposed that extracts unique features to distinguish one fingerprint from others in order to develop a piece of security system for identity verification. Although there are several algorithms for fingerprint authentication, there is still a need to close the gap of accurateness in minutiae matching and pattern matching method. In this paper, we will study the detection of minutia coordinates and ridge orientation to determine uniqueness and do matching, focusing on three main phases: Fingerprint Image Preprocessing, Segmentation, and Feature Extraction. Through MATLAB simulation, the proposed algorithm performance is verified and the results show that recognition rate accuracy reaches up to 96.11%.*
Keywords: Fingerprint recognition, Identity verification, Biometrics

1. **Introduction.** Biometrics technology is continuously developing to improve accuracy, robustness and security. There are a variety of biometric characteristics that can be collected from humans including hand geometry, iris, fingerprint, facial recognition, and voice recognition. Fingerprint is the ridges and furrows pattern on the tip of the human finger, and they have been used for identification because of their immutability and individuality. Immutability refers to the permanent and unchanging character of the pattern on each finger. Individuality refers to the uniqueness of ridge details across individuals [1]. The fingerprint is composed of groups of ridges and furrows. In many cases, poor quality in terms of ridge clarity and background noise requires several stages of preprocessing before applying the recognition system. However, to distinguish one fingerprint from another we do not depend on the ridges and furrows themselves but on some abnormal features called minutia. The minutia lay on ridges and is grouped into many types.

A biometric system can be operated in two modes: verification and identification. The fingerprint matching is either for the 1-to-1 verification or 1-to-many identification [2]. In the verification mode, a biometric system either accepts or rejects a user's claimed identity while a biometric system operating in the identification mode establishes the identity of the user without a claimed identity. Fingerprint identification is much more

DOI: 10.24507/ijicic.15.06.2183

complicated than fingerprint verification because of the huge number of comparisons to all users in the database to be performed in identification stage [3]. In this study, we have focused on a biometric system operating in a verification mode. Verification involves comparison with only those templates corresponding to the claimed identity. Nowadays, many emerging applications require reliable fingerprint verification to confirm the identity of an individual. Traditionally, passwords and ID cards have been used to restrict access to secure systems but these methods can easily be breached and are unreliable. Biometric cannot be borrowed, stolen, or forgotten, and forging one is practically impossible.

The lines that make up a person's fingerprint are a unique sequence of ridges and furrows [4]. These ridges and furrows (valleys) have unique features to them. Sometimes they end, sometimes they split, and sometimes they cross. In this paper, we concerned about the first two cases. Actually, these unique characteristics represent what is called minutia where a human's fingerprint derives its uniqueness. Another defining characteristic of a fingerprint is called the core. The core represents the north most points of the inner most ridge line. Fingerprints can be classified into four major categories as seen in Figure 1 based on their respective core; they are Left Loop (LL), Right Loop (RL), Whorl, and Tented Arch [5].

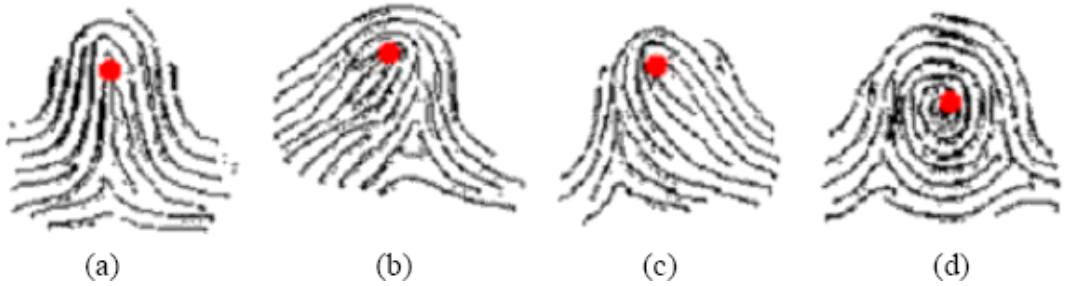


FIGURE 1. Core points on different fingerprint patterns: (a) tented arch, (b) right loop, (c) left loop, and (d) whorl

The core is an important part of the fingerprint recognition process; however, the backbone of fingerprint recognition is found in recognizing minutia and pattern matching method, such as Fast Fourier Transform as one of its image process to produce better image quality to be used for minutiae identifications that have been employed in [6], and the construction of both ridge and minutia dictionaries in a two-step multiscale patch based sparse representation to enhance the ridge using ridge dictionaries and enhance the minutia with both dictionaries proposed in [7]. A novel hybrid shape and orientation descriptor is designed to address the detection problems. This hybrid descriptor can effectively filter out spurious or unnatural minutiae pairings while simultaneously using the additional ridge orientation in improving match score calculation proposed in [8]. In [9], authors used the ridges and valleys on fingerprints extraction combined with the orientation field for alignment to improve accuracy.

In [6], the authors proposed a fingerprint recognition system through several stages: pre-processing stage for image enhancement, binarization, thinning fingerprint image; then the feature extraction stage from the thinning image ridge ending extracting; and the last stage is the matching stage to match two minutiae points by using minutiae matcher stage in which similarity and distance measure are used. In [8], the proposed algorithm has the following stages: the pre-processing, feature extraction step using point pattern matching; then the generated template will be matched with the stored template; and interfacing of LCD and GSM was implemented through Arduino UNO controller. LCD is used to display results.

In this paper, we proposed an algorithm that combines several stages to achieve the required accuracy through studying the detection of minutia coordinates and ridge orientation to determine uniqueness and do matching. Firstly, in the fingerprint image preprocessing phase, we use Histogram Equalization and Fourier Transform. Afterwards, Gabor filter is used to enhance images. Then the enhanced image is binarized using locally adaptive threshold method. In the next phase, image segmentation will be applied that consists of block direction estimation and segmentation by direction intensity and Region of Interest (ROI) extraction using some morphological methods and standard deviation. After that, ridge thinning and filtering using morphological operations are applied in order to use their resultant image to extract minutia. The minutia extraction algorithm is based on detecting the image pixels values to determine whether it is a termination or a bifurcation.

The rest of this paper is organized as follows. In the following section, detailed description of the proposed fingerprint verification system has been discussed. Section 3 presents simulation scenarios and obtained results. Finally, this paper is concluded in Section 4.

2. **Proposed Fingerprint Verification Algorithm.** In this section, the developed fingerprint verification system and algorithms used to implement the preprocessing, segmentation, feature extraction and selection have been discussed. It also presents detailed description of the verification technique. Figure 2 shows the block diagram of the proposed fingerprint verification system. It consists mainly of five phases: Preprocessing, Segmentation, Feature Extraction, Feature Matching and Analysis as it will be explained in detail shortly.

FIGURE 2. Fingerprint verification system block diagram

2.1. **Preprocessing.**

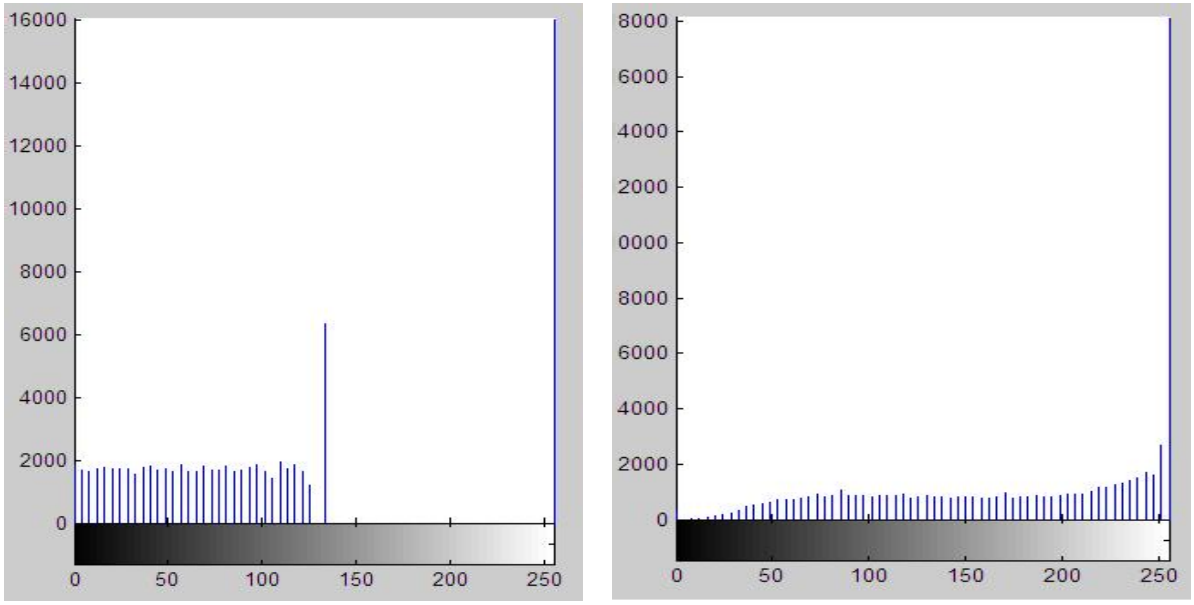
2.1.1. *Histogram equalization.* Histogram equalization enhances the contrast of images by expanding the pixel value in an intensity image; so that the histogram of the gray component of the indexed image is approximately equally distributed, to increase information perception. The histogram visualization effect is enhanced after histogram equalization. See Figure 3. Figure 4 shows the effect of histogram equalization.

2.1.2. *Fourier transform.* Fast Fourier Transform (FFT) enhancement method was fully derived from [6]. On (32×32) Pixels Blocks of the fingerprint image, we applied the Fourier transform. The enhanced image after FFT usually connects some falsely broken points on ridges and removes some spurious connections between ridges. After applying Fourier transform, FFT image is also needed to be processed with histogram equalization to improve the image enhancement as follows:

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

$$\text{For } u = 0, 1, 2, \ldots, 31 \text{ and } v = 0, 1, 2, \ldots, 31$$

where the image is divided into small processing blocks $(32 \times 32$ pixels). Multiplying the FFT of the block by its magnitude is applied in order to enhance a specific block by its dominant frequencies. The magnitude of the original $FFT = abs(F(u,v)) = |F(u,v)|$.



(a) (b)

FIGURE 3. (a) The original histogram of a fingerprint image, (b) histogram after the histogram equalization

(a) (b)

FIGURE 4. Histogram enhancement: (a) original fingerprint image, (b) enhanced image

Get the enhanced block according to:

$$g(x,y) = F^{-1}\left\{F(u,v) * |F(u,v)|^k\right\} \tag{2}$$

where $F^{-1}(F(u,v))$ is done by:

$$f(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \times \exp\left\{j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N}\right)\right\} \tag{3}$$

For $x = 0, 1, 2, \ldots, 31$ and $y = 0, 1, 2, \ldots, 31$

The k in Formula (2) is an experimentally determined constant, $k = 0.45$. Having a lower k will not make that influence on ridges appearance, while having a higher k improves their appearance, filling up small holes in ridges, having too high a k can result in false joining of ridges. Thus a termination might become a bifurcation. Figure 5 presents the image after FFT enhancement. Through applying image binarization, we

(a) (b)

FIGURE 5. (a) Original fingerprint image, (b) image after FFT enhancement

found that image became better for the viewer. In such a case and although FFT has a side effect on the image we can simply ignore for most images since it is not so harmful eventually, Figure 5(b) shows the fingerprint image after FFT enhancement.

2.1.3. *Gabor filter.* The performance of minutia extraction algorithms and other fingerprint recognition techniques depend heavily on the quality of the fingerprint images input. Some researchers proposed an effective method based on Gabor filters [7]. An initial enhancement routine would be to perform a contrast enhancement on the image. Gabor Filtering Routines are taken into consideration as a very successful way to enhance low quality fingerprints in order to more effectively extract minutia features from it. Gabor filtering creates a kind of context sensitive filter for each pixel and does a better job of maintaining overall ridge structure than other methods [7]. The basic idea is to use ridge orientation and frequency to exploit certain properties around a pixel. The basic formula for this filter is:

$$G(x, y : \theta, f) = \exp\left\{-1/2\left[\left(x_\theta^2/\sigma_x^2\right) + \left(y_\theta^2/\sigma_y^2\right)\right]\right\} * \cos(2 * pi * f * x_\theta) \tag{4}$$

In this formula, θ is the angle from the Local Orientation Image discretized to some set of banked filters, and (x_θ, y_θ) are the coordinates of the pixel (x, y). The f parameter is the ridge frequency which involves calculating the number of ridges in a local window in the fingerprint image, other part of this equation are the constants σ_x and σ_y. Specifying these constants involves a tradeoff. The higher these constants the more robust the filters are at filtering out noise, but the tradeoff is that false ridges and furrows can be introduced. Conversely the smaller these numbers the less robust to noise the filters are, but they will not introduce any false ridges and valleys. A good value for these constants could be in the range 3 to 5, but could also be higher or lower depending on context. Figure 6 shows the image before and after applying the filter.

2.1.4. *Fingerprint image binarization.* Binary images contain only 0's and 1's. Pixels with the value 0 are displayed as black; pixels with the value 1 are displayed as white. To binarize fingerprint image, a locally adaptive thresholding method is performed. Binarization is to transform the 8-bit gray fingerprint to a 1-bit image with 0-value for furrows and 1-value for ridges. The mechanism of this method is based on transforming a pixel value to 1 if the value is larger than the mean intensity of (16×16) blocks and transforming it to 0 if it is smaller. Figure 7 shows the fingerprint image before and after applying adaptive binarization.

(a) (b)

FIGURE 6. (a) The image before the filter, (b) the image after the filter

(a) (b)

FIGURE 7. (a) Original fingerprint image, (b) image after binarization

2.2. **Segmentation.** In general, only a Region of Interest (ROI) is useful to be recognized for each fingerprint image. The areas without effective ridges and furrows will be first discarded since it only holds background information. Then, the bound of the remaining effective area is sketched out since the minutia in the bound region is confusing with that spurious minutia that is generated when the ridges are out of the sensor. To extract the ROI, a two-step method is applied. The first step is block direction estimation and direction variety check [9], while the second has two optional parts: the first one is based on standard deviation and the second is based on morphological methods.

2.2.1. *Block direction estimation.* Afterwards, we applied the Gaussian filter. To estimate each block direction we perform the following algorithm on a window size $(W \times W)$ (W is 16 pixels by default).

A. Calculate the gradient values along x-direction (g_x) and y-direction (g_y) for each pixel of the block. Two Sobel filters are used to fulfill the task and two Gaussian filters.
B. For each block, use the following formula to get the Least Square approximation of the block direction named $tg2\beta$.

$$tg2\beta = 2\sum\sum(g_x * g_y) \Big/ \sum\sum\left(g_x^2 - g_y^2\right) \tag{5}$$

The formula is easy to understand by regarding gradient values along x-direction and y-direction as cosine value and sine value. So the tangent value of the block direction named $tg2\theta$ is estimated nearly the same as the way illustrated by the following formula:

$$tg2\theta = 2\sin\theta\cos\theta \,/\left(\cos\theta^2 - \sin\theta^2\right) \tag{6}$$

After we finished estimation of each block direction, those blocks without considerable information on ridges and furrows are discarded based on the following formula:

$$E = \left\{2\sum\sum(g_x * g_y) + \sum\sum\left(g_x^2 - g_y^2\right)\right\} \Big/ W * W * \sum\sum\left(g_x^2 + g_y^2\right) \tag{7}$$

Then the enhanced image is binarized using locally adaptive threshold method, for each block, if its certainty level E is below a threshold, then the block is regarded as a background block. The direction map is shown in Figure 8. Table 2 in Section 3 will show the match score (%) for three persons' fingerprint with different threshold values.

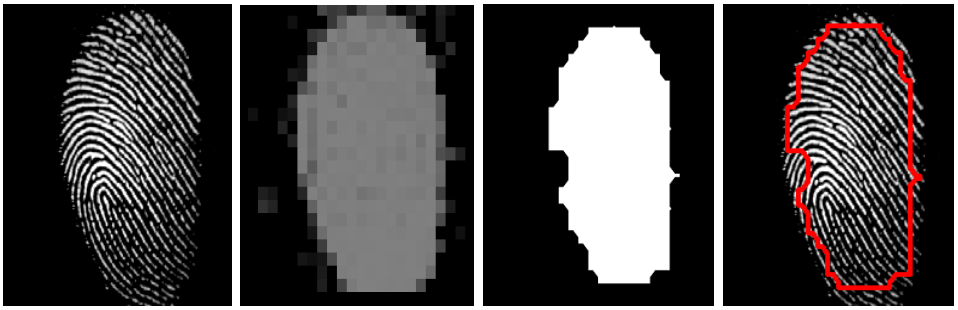
(a) (b)

FIGURE 8. (a) Original image, (b) direction map image

2.2.2. *ROI extraction using standard deviation method.* At this step the standard deviation is calculated for each block in the fingerprint image, block size $W \times W$ where W equals 16. Then a Comparative value is calculated by taking 85% of the standard deviation value of the whole image as follows:

$$\text{Comparative value} = 0.85 \times \text{standard deviation of the whole image} \tag{8}$$

Next, for each block if the local standard deviation is less than the Comparative value the block is treated as background. Figure 9 shows the effect of performing standard deviation to extract ROI area.



(a) (b) (c) (d)

FIGURE 9. (a) Original image, (b) image after standard deviation, (c) ROI after erosion, (d) ROI area

2.2.3. *ROI extraction using morphological method.* Three morphological operations called 'CLOSE', 'FILL', and 'EROSION' are adopted. The 'CLOSE' operation can shrink images and eliminate small cavities Figure 10(a). 'FILL' operation fills the small holes. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image so it will fill small region holes to reduce error in calculating the ROI

Figure 10(b). The 'EROSION' method is used to smooth the outer bound of the fingerprint image after closing and filling operations Figure 10(c). Another method is used and is called 'BWAREAOPEN' Figure 10(d) which is a MATLAB function used to remove small objects and works as follows:

A. Determine the connected components.
B. Compute the area of each component.
C. Remove small objects.

(a) (b) (c) (d)

FIGURE 10. (a) ROI after CLOSE operation, (b) ROI after FILL operation, (c) ROI after EROSION, (d) ROI after BWAREAOPEN operation

2.3. **Minutia extraction.**

2.3.1. *Fingerprint ridge thinning.* Thinning is an operation that is used to remove selected foreground pixels from binary image somewhat like erosion or opening, we used it so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a connected ring halfway between each hole and the outer boundary. In this study, ridge thinning is achieved through applying built in morphological thinning function in MATLAB. To eliminate the redundant pixels of ridges till the ridges are just one pixel wide. Thinning comes after binarization which is implicitly enforced since only pixels with maximum gray intensity value are remained in the ROI area. The thinned ridge map is then filtered by other three morphological operations called 'HBREAK', 'CLEAN' and 'SPUR' to remove some H breaks isolated points and spikes respectively. Figure 11 shows the output image after thinning.

2.3.2. *Minutia marking.* Minutia marking is getting easier after the fingerprints ridge thinning. In general our approach in minutia marking is based on using (3×3) window and checks whether the central pixel is one, and has exactly three neighbors with pixel values one, then the central pixel is a ridge branch, see Figure 11(a). However, when the central pixel is one and it has only one neighbor with pixel value one, then the central pixel is a ridge ending, see Figure 11(b).

2.3.3. *False minutia removal.* The preprocessing phase does not totally repair the fingerprint image. For example, false ridge breaks due to insufficient amount of ink and ridge cross-connections due to the fact that over inking is not totally eliminated also background noise effect. Actually all the earlier phases themselves occasionally introduce some artifacts which later lead to spurious minutia. This false minutia will significantly affect the accuracy of matching if they are simply regarded as genuine minutia. So some

(a) (b) (c) (d)

FIGURE 11. (a) Image after binarization, (b) image after thinning, (c) thinned image after removing H breaks, (d) thinned image after removing spikes

mechanisms of removing false minutia are essential to keep the fingerprint verification system effective.

Our procedures in removing false minutia are the following.

A. If two terminations are within a distance D and their directions are coincident with a small angle variation, and they suffice the condition that no any other termination is located between the two terminations, then the two terminations are regarded as false minutia derived from a broken ridge and are removed (Case m4, m5, m6).
B. If the distance between two bifurcations is less than D and they are in the same ridge, remove the two bifurcations (Case m2, m3).
C. If the distance between one bifurcation and one termination is less than D and the two minutiae are in the same ridge (m1 case), remove both of them. D is the average inter-ridge width representing the average distance between two parallel neighboring ridges.
D. If two terminations are located in a short ridge with length less than D, remove the two terminations (m7).
E. As a checking procedure we used slices and strips to scan the image and detect if two minutiae lay on the same ridge and are within the slice height or the strip width both are regarded as false minutiae and removed.

Our proposed procedures in removing false minutia have two advantages. One is that the ridge ID is used to distinguish minutia and the seven types of false minutia are strictly defined comparing with those loosely defined by other methods. The second advantage is that the order of removal procedures is well considered to reduce the computation complexity. Figure 12 shows the final result after minutia extraction and false minutia removal.

2.4. **Minutia match.** Given two sets of minutia of two fingerprint images, the minutia match algorithm determines whether the two minutia sets are from the same finger or not. An alignment-based match algorithm is used in our study and we made some adjustments on the concepts used in it. It includes two consecutive phases: one is alignment phase and the second is match phase.

A. Alignment phase. Given two fingerprint images to be matched, choose any one minutia from each image, calculate the similarity of the two ridges associated with the two referenced minutia points. If the similarity is larger than a threshold, transform each

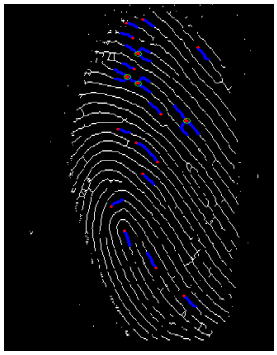


FIGURE 12. Fingerprint image after minutia extraction, red dots represent terminations, green circles represent bifurcations, and blue lines represent the ridge path map.

set of minutia to a new coordination system whose origin is at the referenced point and whose x-axis is concurrent with the direction of the referenced point.

B. Match phase. After we get two sets of transformed minutia points, we use the elastic match algorithm to count the matched minutia pairs by assuming two minutiae having nearly the same position and direction.

2.4.1. *Alignment phase.*

A. The ridge associated with each minutia is represented as a series of x-coordinates $(x_1, x_2, \ldots, x_n)$ of the points on the ridge. A point is sampled per ridge length L starting from the minutia point, where the L is the average ridge length. And n is set to 10 unless the total ridge length is less than $10 * L$.

So the similarity of correlating the two ridges is derived from:

$$S = \sum_{i=0}^{m} x_i X_i \Bigg/ \left[\sum_{i=0}^{m} x_i^2 X_i^2 \right]^{0.5} \tag{9}$$

where $(x_i \sim x_n)$ and $(X_i \sim X_N)$ are the set of minutia for each fingerprint image respectively. And m is minimal one of the n and N value. If the similarity score is larger than 0.8, then go to Step B; otherwise, continue to match the next pair of ridges.

B. For each fingerprint, translate and rotate all other minutia with respect to the reference minutia according to the following formula:

$$\begin{pmatrix} x_i\_\text{rotate} \\ y_i\_\text{rotate} \\ \theta_i\_\text{rotate} \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_i - x \\ y_i - y \\ \theta_i - \theta \end{bmatrix} \tag{10}$$

The rotation angle is calculated from all the sparsely sampled ridge points. The method uses the rotation angle calculated earlier by densely tracing a short ridge start from the minutia with length D. Since we have already got the minutia direction at the minutia extraction phase, obviously the method reduces the redundant calculation but still holds the accuracy.

2.4.2. *Match phase.* The matching algorithm for the aligned minutia patterns needs to be flexible since the strict match that requires that all parameters (x, y, θ) are the same for two identical minutiae is impossible due to the slight deformations and inexact quantization of minutia. Our approach to elastically match minutia is achieved by placing a bounding box around each template minutia. If the minutia to be matched is within

the rectangle box and the direction difference between them is very small, then the two minutiae are regarded as a matched minutia pair. Each minutia in the template image either has no matched minutia or has only one corresponding minutia.

The final match ratio for two fingerprints is the number of total matched pair over the number of minutia of the template fingerprint. The percentage match is (100*ratio) and ranges from 0 to 100. If the percentage match is larger than a pre-specified acceptance threshold, the two fingerprints are from the same finger.

3. **Results.** This section is concerned with the performance evaluation of fingerprint verification system, using confusion matrix on multi data bases. And it makes it possible to better understand how current fingerprint recognition system works and to define useful research directions for the future.

3.1. **Performance.** A confusion matrix contains information about actual and predicted classifications done by a classification system. A confusion matrix is used for checking the accuracy of a classification. Performance of such systems is commonly evaluated using the data in the matrix. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

Confusion matrix contains actual information for matching images with databases. To simplify the idea of this matrix, Table 1 shows the confusion matrix for a two class classifier:

TABLE 1. Confusion matrix

		Predicted	
		1	2
Actual	1	**a**	**b**
	2	**c**	**d**

The confusion matrix idea will be used to measure the performance of fingerprint verification algorithm in the following section. The idea that every person in the database has 8 fingerprint images, so the code modified to do recognition to all images in data set. Define a matrix of size (10×10). Every 8 loops of the program the counter moves to next column in the confusion matrix, which means a new fingerprint image. When the code stops running the matrix will be filled, the diagonal of it will be the number of times that the code finds the match correctly. Simply the performance will be:

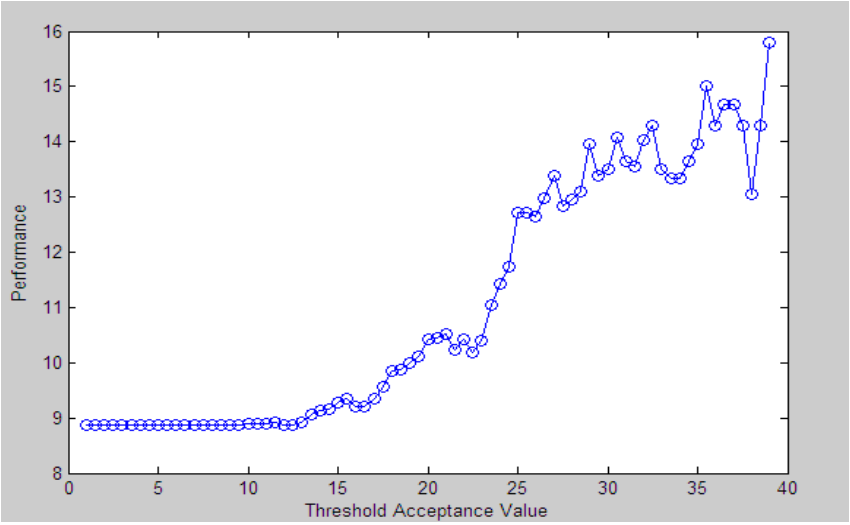
$$\text{Performance} = \frac{\text{sum of diagonal values}}{\text{total number of images}} \times 100\% \tag{11}$$

Calculating the performance of our fingerprint verification system is done by getting the Similarity Matrix which is then used to evaluate the confusion matrix.

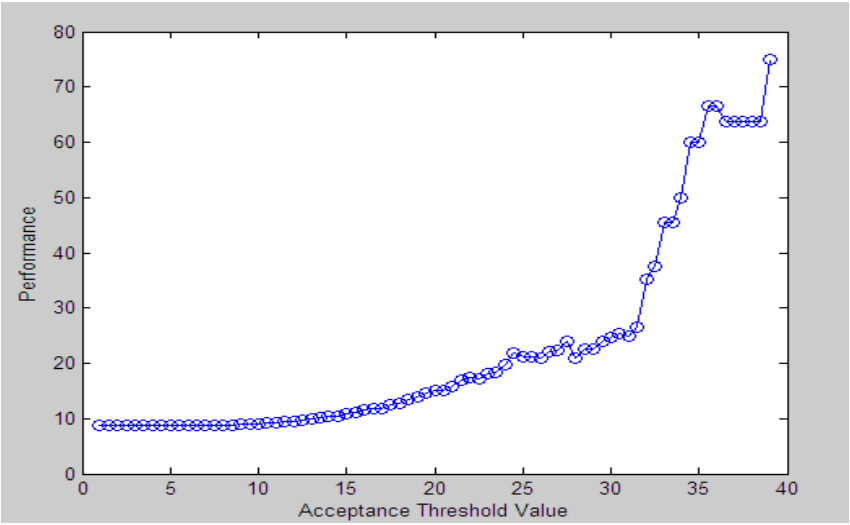
Similarity Matrix carries the matching percentages of the input images in the system database. Suppose that the database that our system works on contains 10 persons' images and 8 images for each; Similarity Matrix size is (80×80) and the confusion matrix size is (10×10).

3.2. **Performance analysis.** We studied the performance of the fingerprint verification system under varying conditions. We used three different databases; each one had fingerprint images for 10 persons with 8 images per person.

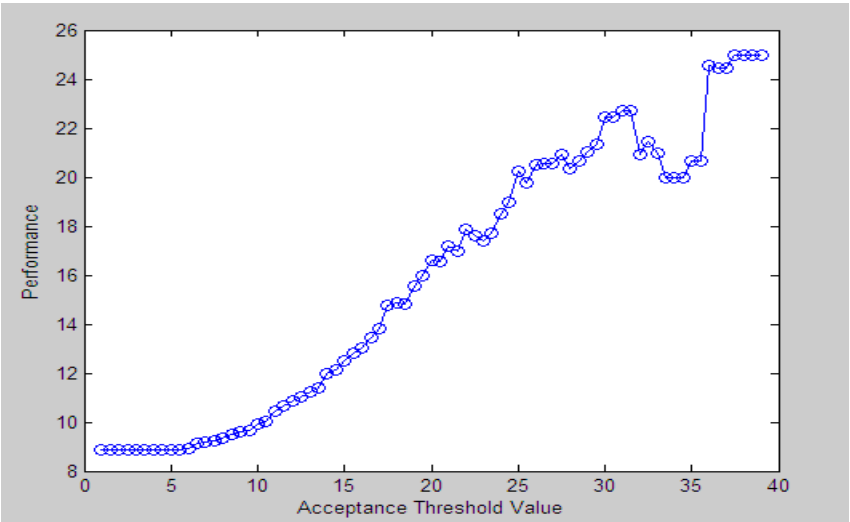
Figure 13 shows the performance results that were conducted after using DB1-FVC2002 [10], DB2-FVC2002 [11] and DB3-FVC2004 [12] databases, respectively. On the other



(a) The results using database DB1-FVC2002 [10]



(b) The results using database DB2-FVC2002 [11]



(c) The results using database DB3-FVC2004 [12]

FIGURE 13. The performance results of the system

TABLE 2. Matching score for different threshold values

Threshold value	**Matching score (%)**		
	Person (1)	**Person (2)**	**Person (3)**
1.0	32.46	31.22	32.15
0.75	41.27	39.46	38.22
0.5	55.79	50.73	48.4
0.25	78.26	75.68	74.8

TABLE 3. Recognition rate for different persons

Fingerprint sample using database DB2-FVC2002	**Person 1**	**Person 2**	**Person 3**	**Person 4**	**Person 5**	**Average**
Recognition rate (%)	97.73	95.02	94.78	96.67	96.35	96.11

hand, we used different acceptable match threshold values to consider matching, Table 2 shows the match score (%) for three persons' fingerprint with different threshold values, and Table 3 shows the recognition rate for different persons (using database DB2-FVC2002) and consider threshold value to be 0.25 which results the highest matching score as shown in Table 2.

We found that the best results of our system were after applying it to DB2-2002 because of the quality of the acquired images. The other databases had low quality images with much noise in the background that could not be repaired by the enhancement procedures.

Our proposed enhancement algorithm minutia estimation algorithm is compared with [6]. The algorithms are tested using fingerprint images from different databases like FVC2000 and FVC2002 databases. Table 4 shows results of comparison between the two algorithms for the same databases.

TABLE 4. Results validity compared with the proposed algorithm in [6]

Fingerprint database	**Recognition accuracy**	
	Proposed algorithm	**Algorithm proposed by Ali et al. [6]**
DB1-FVC2000	82.43%	80.03%
DB1-FVC2002	96.11%	98.55%

Several methods to build a minutia extractor and a minutia matcher have been employed. The combination of multiple methods came up from a wide investigation into the available techniques like segmentation using morphological operations and standard deviation, also, choosing suitable filters in enhancement, segmentation and direction estimation phases respectively. The accuracy obtained from the authentication system proposed in [8] was on average 89.1% compared with the average accuracy obtained from our proposed system which reaches up to 89.6%.

4. **Conclusion.** In this study, a fingerprint verification algorithm has been proposed to verify human fingerprints in security systems. Moreover, enhancing bad quality fingerprint images was another goal in order to be able to do feature extraction and matching. Some techniques have been applied to removing false detected features. The proposed algorithm performance is investigated through conducting many MATLAB simulations to build the consecutive phases of our fingerprint recognition system and analyzing the output performance. The recognition rate can reach up to 96.11%. This high accuracy

result proves that the collection of image enhancement techniques can further improve the recognition rate and matching score by using minutiae extraction.

4.1. **Recommendations for further research.**

A. Develop a match algorithm other than the Elastic match algorithm that may improve the system performance.
B. Develop an algorithm to calculate the ridge frequency as a fourth parameter in addition to the x, y coordinates and the orientation of the fingerprint.
C. Work on the enhancement phase, Gabor filter, to have more efficient output by calculating the ridge frequency prior to running this filter.
D. Expand the study to do identification instead of verification.

4.2. **Data availability.** The data generated during the study is available upon request from authors.

REFERENCES

[1] K. Cao and A. K. Jain, Automated latent fingerprint recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.41, no.4, pp.788-800, 2018.
[2] S. Guennouni, A. Mansouri and A. Ahaitouf, Biometric systems and their applications, in *Eye Tracking and New Trends*, IntechOpen, 2019.
[3] M. S. Mahmood, Fingerprint identity watermark to authenticate digital camera images, *Advances in Natural and Applied Sciences*, vol.11, no.9, pp.117-126, 2017.
[4] C. Champod, C. J. Lennard, P. Margot and M. Stoilovic, *Fingerprints and Other Ridge Skin Impressions*, CRC Press, 2017.
[5] D. Peralta, I. Triguero, S. García, Y. Saeys, J. M. Benitez and F. Herrera, Robust classification of different fingerprint copies with deep neural networks for database penetration rate reduction, *arXiv Preprint*, arXiv:1703.07270, 2017.
[6] M. M. Ali, V. H. Mahale, P. Yannawar and A. T. Gaikwad, Fingerprint recognition for person identification and verification based on minutiae matching, *IEEE the 6th International Conference on Advanced Computing*, pp.332-339, 2016.
[7] M. Xu, J. Feng, J. Lu and J. Zhou, Latent fingerprint enhancement using Gabor and minutia dictionaries, *IEEE International Conference on Image Processing (ICIP)*, pp.3540-3544, 2017.
[8] S. Sindhu and B. Arunadevi, Fingerprint authentication based on adaptive greedy registration of minutiae pairs, *The 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp.1360-1364, 2018.
[9] Y. Xu, G. Lu, Y. Lu, F. Liu and D. Zhang, Fingerprint pore comparison using local features and spatial relations, *IEEE Trans. Circuits and Systems for Video Technology*, 2018.
[10] *FVC 2000*, http://bias.csr.unibo.it/fvc2000/db1.asp.
[11] *FVC 2000*, http://bias.csr.unibo.it/fvc2000/db2.asp.
[12] *FVC 2000*, http://bias.csr.unibo.it/fvc2000/db3.asp.