

Design and Implementation of Fingerprint Identification System Based on KNN Neural Network

Israa Ghazi Dakhil¹, Ali Abdulhafidh Ibrahim²

¹College of Information and Communication Engineering, Al-Nahrain University, Baghdad, Iraq

²Al-Nahrain University, Baghdad, Iraq

Email: israaghazi94@yahoo.com, dr_ali9@yahoo.com

How to cite this paper: Dakhil, I.G. and Ibrahim, A.A. (2018) Design and Implementation of Fingerprint Identification System Based on KNN Neural Network. *Journal of Computer and Communications*, 6, 1-18.

<https://doi.org/10.4236/jcc.2018.63001>

Received: January 30, 2018

Accepted: March 5, 2018

Published: March 8, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Fingerprint identification and recognition are considered popular technique in many security and law enforcement applications. The aim of this paper is to present a proposed authentication system based on fingerprint as biometric type, which is capable of recognizing persons with high level of confidence and minimum error rate. The designed system is implemented using Matlab 2015b and tested on a set of fingerprint images gathered from 90 different persons with 8 samples for each using Futronic's FS80 USB2.0 Fingerprint Scanner and the `frScanApiEx.exe` program. An efficient image enhancement algorithm is used to improve the clarity (contrast) of the ridge structures in a fingerprint. After that core point and candidate core points are extracted for each Fingerprint image and feature vector have been extracted for each point using `filterbank_based` algorithm. Also, for the matching the KNN neural network was used. In addition, the matching results were calculated and compared to other papers using some performance evaluation factors. A threshold has been proposed and used to provide the rejection for the fingerprint images that does not belong to the database and the experimental results show that the KNN technique have a recognition rate equal to 93.9683% in a threshold equal to 70%.

Keywords

Fingerprint, Core Point and Candidate Core Points, Filterbank_Based Algorithm, Weightless Neural Network, KNN

1. Introduction

Establishing the identity of a person is a critical task in any identity management

system. Surrogate representations of identity such as passwords and ID cards are not sufficient for reliable identity determination because they can be easily misplaced, shared, or stolen. Biometric recognition is the science of establishing the identity of a person using his/her anatomical and behavioral traits. Commonly used biometric traits include fingerprint, face, iris, hand geometry, voice, palm print, and handwritten signatures [1]. The fingerprint recognition system is one of the widely used biometric authentication systems, the biometric authentication system based on two modes: Enrolment and Recognition. In the enrolment mode, the biometric data are acquired from the sensor and stored in a database along with the person's identity for the recognition. In the recognition mode, the biometric data are re-acquired from the sensor and compared to the stored data to determine the user identity. Finally, the decision module makes the identity decision [2]. Fingerprints are graphical patterns of ridges and valleys on the surface of fingertips and every person has a unique fingerprint from any other person. In general, researchers have categorized Fingerprint Recognition system techniques into several groups, due to the Enrollment of Fingerprint images, Preprocessing including the Enhancement step, Feature Extraction and Matching. During the Enrollment, the fingerprint is scanned using the fingerprint scanner. The preprocessing module includes preparing the fingerprint to the feature extraction module and enhances the fingerprint to be combatable with the system performance. Feature extraction module processes the scanned biometric data to extract the feature sets. The matcher module accepts two biometric feature sets from template and query, resp. as inputs, and outputs a match score indicating the similarity between the two sets.

2. Related Work

In the literature, various approaches have been proposed by researchers to provide the best recognition rate. For example Jain, in 2001 [3], has developed a novel filter based representation technique for fingerprint recognition. The technique exploits both the local and global characteristics in a fingerprint image to make a verification. Each fingerprint image is filtered in a number of directions and a fixed-length feature vector is extracted in the central region of the fingerprint. The matching stage computes the distance between the template feature vector (finger code) and the input finger code. D. Batra, G. Singhal and S. Chaudhury in 2004 [4], use a Gabor filter based Feature extraction scheme to generate a 384 dimensional feature vector for each fingerprint image. The classification of these patterns is done through a novel two stage classifier in which K Nearest Neighbour (KNN) acts as the first step and finds out the two most frequently represented classes amongst the K nearest patterns, followed by the pertinent SVM classifier choosing the most apt class of the two. 6 SVMs have to be trained for a four class problem, (6 C2), that is, all one-against-one SVMs. Using this novel scheme and working on the FVC 2000 database (257 final images) and achieved a maximum accuracy of 98.81% with a rejection percentage of 1.95%. S.

Chikkerur, C. Wu and Govindaraju in 2004 [5], Enhancement algorithm based on Fourier analysis that is simultaneously able to extract local ridge orientation, ridge frequency and ridge quality measures is proposed. Contextual filtering in the Fourier domain based on these features is explored. Jihong, Baorui, and De-qin, in 2007 [6], presented a hardware implementation of ANN based on modification of Boolean k-nearest neighbor (BKNN) classifier proposed by Gazula and Kabuka. BKNN is a kind of supervised classifier using Boolean Neural Network, which has binary inputs and outputs, integer weights, fast learning and classification, and guaranteed convergence. The emphasis of this design is that it is implemented on Field Programming Gate Array (FPGA) chip. Dhia, and Thikra, in 2014 [7], proposed fingerprint identification algorithm is introduced, it has been used Genetic Algorithm (GA) as a feature selection tool for fingerprint identification. The proposed system contains four main steps: preprocessing, features extraction, feature selection and classification. The preprocessing sub stages consist of some image processing techniques as: Enhancement and Segmentation. A feature has been extracted from ridges that find around core point then DCT has been used to extract features and has been got a few coefficients. Important features have been selected using genetic algorithm filter. Finally, classification step has been done by using k-Nearest Neighbors (k-NN), where the database contains samples for 28 persons, 7 sample for each person. The recognition rate reached to 98%.

3. Proposed System

The proposed system flowchart provides a definition of the system modules and sub modules. **Figure 1** illustrates the system flowchart. The system consists of four modules:

3.1. The Enrollment Module

The task of this module is to enroll the fingerprint of the user in to the system database using Futronic's FS80 USB2.0 Fingerprint Scanner and the ftrScanApiEx.exe program. In this work 90 fingerprint image have been collected with 8 samples for each person; so as a total 720 fingerprint image have been captured. The FP images have been capture from different age persons like teenagers, college students, middle age and old persons (see **Figure 2**), also in multiple rotations, Image resolution is 480×320 pixel, 500 DPI. One picture for 90 person have been added to a database.

3.2. The Pre-Processing Module

The task of this module is to prepare the FP image for the feature extraction module and enhance and increase the FP quality to get rid from noise if there was any so it can be compatible with the system performance. This module includes the enhancement using the Fourier domain analysis filtering and segmentation.

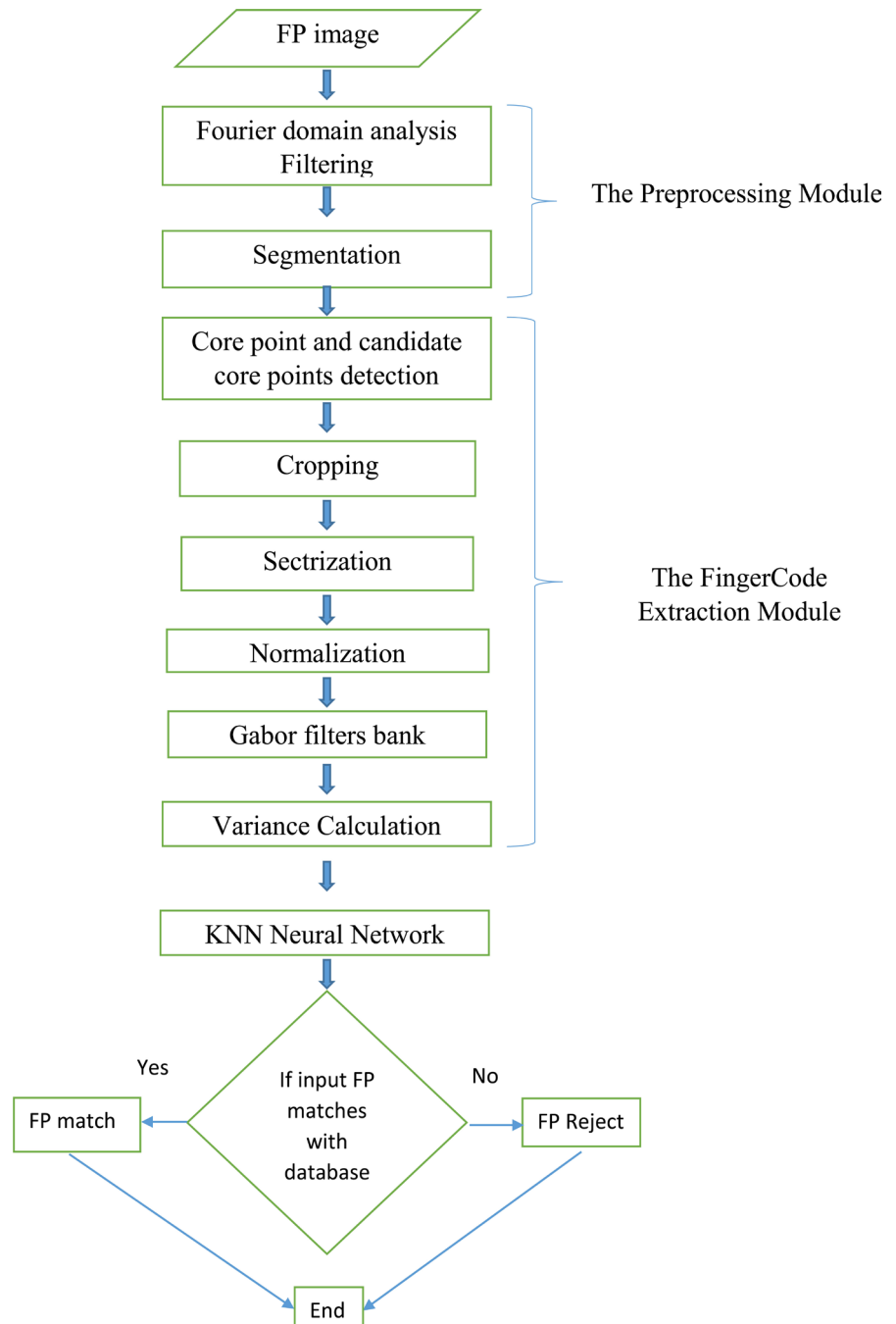


Figure 1. The proposed FP recognition system flowchart.

1) Fourier domain analysis filtering: we enhance the FP image as in [5] through a number of steps: Fourier Domain Analysis, Directional Field Estimation, Ridge Frequency Estimation, Energy Map and Enhancement.

In the Fourier domain analysis A local region of the FP image can be model as a surface wave according to Equation (1):

$$i(x, y) = A \cos\{2\pi f(x \cos \varnothing + y \sin \varnothing)\} \quad (1)$$

This orient wave can be characterize completely by its orientation \varnothing and



Figure 2. FP images from different age persons: (a) Teenager; (b) college student; (c) middle age and (d) old person.

frequency f . The Fourier spectrum and its inverse is obtained by using Equations (2) and (3).

$$F(u, v) = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} f(x, y) e^{-j2\pi(ux+vy)/MN} \quad (2)$$

$$f(x, y) = \sum_{u=0}^{x=M-1} \sum_{v=0}^{y=N-1} F(u, v) e^{-j2\pi(ux+vy)/MN} \quad (3)$$

The Directional Field Estimation Is the process used to find the orientation (angles) of fingerprint ridges. Let the Fourier coefficients be represented in the polar co-ordinates as $F(r, \varnothing)$. We define a probability density function as $F(r, \varnothing)$ and the marginal density functions as $f(\varnothing)$, $f(r)$.

$$f(r, \varnothing) = \frac{|F(r, \varnothing)|^2}{\iint |F(r, \varnothing)|^2 d\varnothing dr} \quad (4)$$

$$f(r) = \int f(r, \varnothing) d\varnothing \quad (5)$$

$$f(\varnothing) = \int f(r, \varnothing) dr \quad (6)$$

The orientation \varnothing is assume as a random variable that has the probability density function $f(\varnothing)$. The expected value of the orientation may then be obtained by equation (7):

$$E\{\varnothing\} = \int \varnothing \cdot f(\varnothing) d\varnothing \quad (7)$$

For the Ride Frequency Estimation, the image frequency represents the local frequency of the ridges in a fingerprint [8]. The average ridge frequency is estimated in a manner similar to the ridge orientation. We can assume the ridge frequency to be a random variable with the probability density function defined by $f(r)$ as in Equation (5). The expected value of the ridge frequency is given by Equation (8). The frequency map that obtained was smoothed by applying a 3×3 Gaussian mask.

$$E\{r\} = \int r \cdot f(r) dr \quad (8)$$

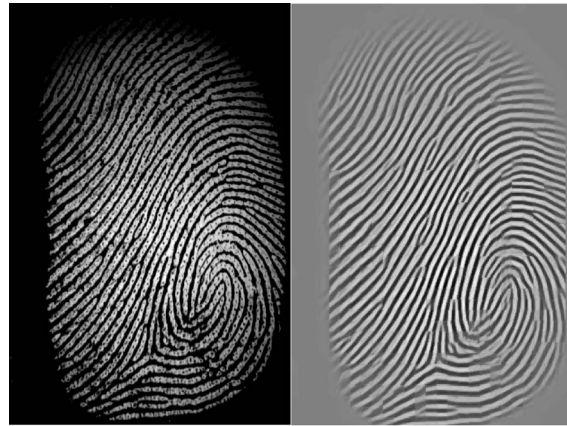
Either for the Energy Map, is the presence of ridges contributes to the energy content in the Fourier spectrum. The energy content of a block may be obtained through Equation (9). We define an energy image $E(x,y)$ where each value indicates the energy content of the corresponding block. The fingerprint region may be differentiated from the background by thresholding the energy image so the FP may be easily to segmented based on the energy map. We take the logarithm values of the energy to obtain a linear scale. The same technique is used to visually represent a frequency spectrum [9].

$$E = \sum_u \sum_v |F(u,v)|^2 \quad (9)$$

Finally in the Enhancement, the image is divided in to 12×12 overlapping blocks with a 6 pixel overlap between adjacent blocks. The block is multiplied by a raised cosine window in order to eliminate any artifacts due to rectangular windows. Each block is filtered in the frequency domain by multiplying it with orientation and frequency selective filter whose parameters are based on the estimated local ridge frequency and orientation. Block-wise approaches have problems around the singularities where direction of the ridges cannot be approximated by a single value. The bandwidth of the directional filter has to be increased around these regions. This is achieved in [10] by making the filter bandwidth a piece-wise linear function of the distance from the singularities. The directional histogram has been obtained in the estimation of the orientation image. It is reasonable to assume that the probability distribution $f(\varnothing)$ is unimodal and centered around $E\{\varnothing\}$. The bandwidth is defined as the angular extent centered on the mean that contains half the energy content of the block. This is obtained from the directional histogram by finding all values of angle \varnothing such that $p\{|\varnothing - E\{\varnothing\}| < \varnothing Bw\} = 0.5$. **Figure 3** show the enhanced FP image.

The filter H is separable in angle and frequency and is obtain by multiplying separate frequency and angular band pass filter of order n. the filters are defined in [10] and are given in Equations (10)-(12).

$$H(r, \varnothing) = H(r) \cdot H(\varnothing) \quad (10)$$



(a) (b)

Figure 3. (a) FP image; (b) the enhanced FP image.



(a) (b)

Figure 4. (a) FP image; (b) the Segmented FP image.

$$Hr(r) = \sqrt{\frac{(rrBW)^{2n}}{(rrBW)^{2n} + (r^2 - rBW^2)^{2n}}} \quad (11)$$

$$H(\varnothing) = \begin{cases} \cos \frac{\pi(\varnothing - \varnothing_c)}{2\varnothing BW} & \text{if } |\varnothing - \varnothing_c| \leq \varnothing BW \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

2) Segmentation: In this operation, the image is segmented and the background is separated from the fingerprint image. This can be performed using a simple block-wise variance approach, since the background is usually characterized by a small variance. The image is first binary closed (Matlab command `imclose`), then eroded (Matlab command `imerode`), in order to avoid holes in the fingerprint image and also to avoid an undesired effect at the boundary (between the fingerprint and the background). **Figure 4** shows the segmented FP image.

3.3. Feature Extraction

For each point from the core and candidate core points, the filterbank-based al-

gorithm will be implemented [3]. **Figure 5** illustrate the filterbank_based algorithm structure.

a) Core Point and Candidate Core Points Detection: The core point is detected through a number of steps:

- 1) Estimate the orientation from the enhanced FP image as it described.
- 2) The orientation field is used to obtain a logical matrix where pixel (i, j) is set to 1 if the angle of the orientation is $\leq \text{PI}/2$ (PI 3.1415926535897...).

3) After this computation, the complex filtering output [11] [12] of the enhanced fingerprint image must be calculated; Complex filter, for the detection of patterns with radial symmetries is modelled by $\exp\{im\varphi\}$. A polynomial approximation of the complex filter in Gaussian windows yields $(x + iy)g(x, y)$ where g is a Gaussian defined as $g(x, y) = \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\}$ [13]. A Gaussian is used as window because the Gaussian is the only function which is orientation isotropic (in polar coordinates, it is a function of radius only) and separable [11].

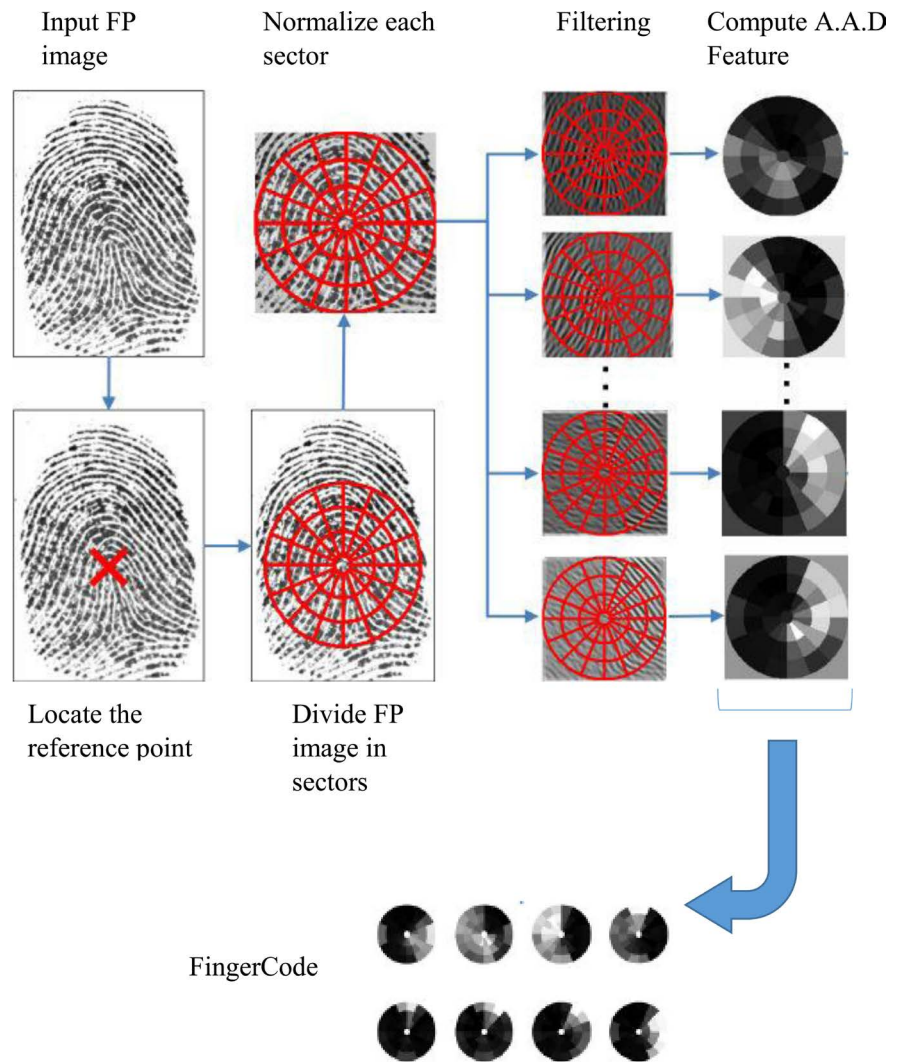


Figure 5. Scheme of Filterbank_based algorithm [3].

In this thesis, symmetry complex filter is used to detect the core point:

$$h(x, y) = (x + iy) g(x, y) = r \exp\{i\varphi\} g(x, y) \quad (13)$$

Figure 6 show the complex filter h .

We identify the candidate core points by its special symmetry properties. Therefore, in order to detect the candidate core points, complex filter designed for detect rotational symmetry extraction is applied. After calculating the complex filtering output of the enhanced fingerprint image, the maximum value of complex filtering output where the pixels of the logical image are set to one were found.

Figure 7 shows the enhanced FP image with core point localization.

4) Repeat steps 2 - 3 for a wide set of angles (... , $\text{PI}/2 - 3 * \text{alfa}$, $\text{PI}/2 - 2 * \text{alfa}$, $\text{PI}/2 - 1 * \text{alfa}$, $\text{PI}/2$, $\text{PI}/2 + 1 * \text{alfa}$, $\text{PI}/2 + 2 * \text{alfa}$, $\text{PI}/2 + 3 * \text{alfa}$, ... where alfa is an arbitrary angle). Each time a point is determined (Note: each of them will be a candidate for fingerprint matching).

5) Subdivide all the points found in step 4 into subsets of points, which are quite near each other. It will be N subsets. For each subset there will be a certain number of candidates and only subset with a number of candidates ≥ 3 will be

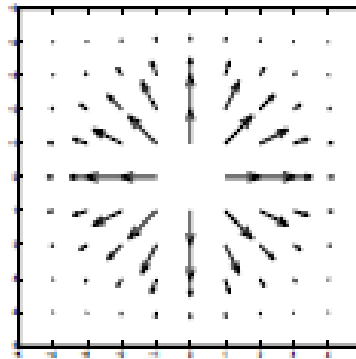


Figure 6. The complex filter h [13].

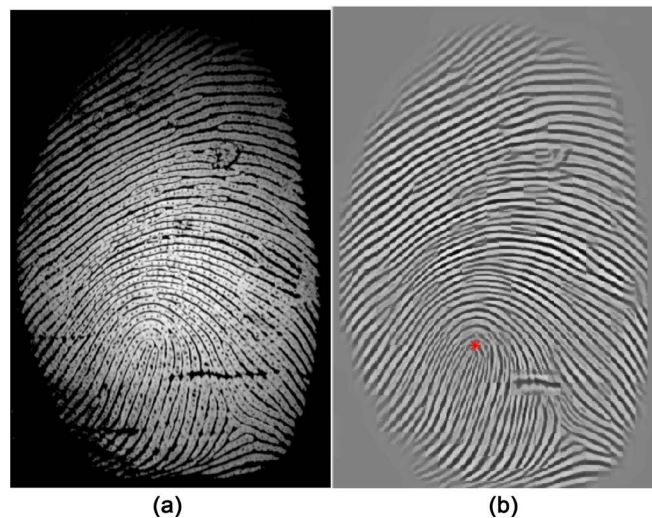


Figure 7. (a) FP image; (b) the enhanced FP image with core point localization.

considered. For each of this subset, consider the subset with the greatest x-averaged coordinate. In this subset, the core point with the greatest x-coordinate is considered. This is a good approximation in standard fingerprint image. The number of core point and candidate core points that have been extracted from this are different from FP image to another, **Table 1** shows the number of core point and candidate core points for some FP images.

b) Cropping

A square of a certain size region around the calculated point is extract in this step. This square area contains the part that will be the input of Gabor filter-bank. Input image is pad with a proper border of zeros in order to avoid any size error.

c) Sectorization

Cropped fingerprint image is sectorized into 4 concentric bands. The bands are centered on the pseudo-center point. All 4 bands have a radius of 20 pixels, and a center hole radius of 12 pixels. Each band is divided into 16 sectors, ignoring center band as it has very small area. Each sector thus formed will capture information corresponding to each Gabor filter. See **Figure 7**.

d) Normalization

Each sector is individually normalized to a constant mean and variance to eliminate variations in the fingerprint pattern. Normalization of each sector is done as in [14]:

$$N(i, j) = \begin{cases} M^\circ + \frac{\sqrt{V^\circ(I(i, j) - M)^2}}{V} & \text{if } I(i, j) > M, \\ M^\circ - \frac{\sqrt{V^\circ(I(i, j) - M)^2}}{V} & \text{else} \end{cases} \quad (14)$$

where $I(i, j)$ grey level at pixel (i, j) , $N(i, j)$ is normalized grey level at pixel (i, j) , M and V are estimated mean and variance of (i, j) , respectively. M° and V° are desired mean and variance, respectively.

Table 1. The number of core point and candidate core points for some FP images.

FP Number	Number of core point and candidate core points
1	10
2	5
3	8
4	12
5	12
6	10
7	12
8	9
9	16
10	11

e) Gabor Filters Bank

The Gabor filter capture both local orientation and frequency information from a fingerprint image. This filter is suited for extracting the texture information from images because by tuning a Gabor filter to specific frequency and direction, the local frequency and the orientation information could be obtained.

The definition of GF in spatial domain is given as follows [15]:

$$G(x, y; f, \theta) = e^{-\frac{1}{2} \left[\frac{x^{\circ 2}}{\sigma_x^2} + \frac{y^{\circ 2}}{\sigma_y^2} \right]} \cos(2\pi f x^{\circ}) \quad (15)$$

$$x^{\circ} = x \cos \theta + y \sin \theta$$

$$y^{\circ} = y \cos \theta - x \sin \theta$$

where θ is the orientation of the GF, f is the frequency of the cosine wave, σ_x and σ_y are the slandered deviations of the Gaussian envelope along the x and y axes, respectively, and x° and y° define the x and y axis of the filter coordinate frame, respectively.

The Normalized image is passing through eight Gabor filters. Each filter produces a 33×33 filter image for 8 angles ($0, \pi/8, \pi/4, 3\pi/8, \pi/2, 5\pi/8, 3\pi/4$ and $7\pi/8$) which is convolved with the fingerprint image. The angles are $\theta = \{0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5\}$.

4 concentric bands are considered around the detected reference point for feature extraction. Each band is 20 pixel wide and segmented in to 16 sectors. Thus, we have a total of $16 * 4 = 64$ sectors. Each sector image will be input in to the eight Gabor filters bank. So, as a total 512 ($64 * 8$) filtered images for each core point in FP image will be extract.

f) Variance Calculation

The Variance of the all pixel values in each sector is calculated after obtaining the 512 filtered images that gives the concentration of fingerprint ridges going in each direction in that part of the fingerprint. $Vi\theta$ is the average absolute deviation from the mean and it is calculate using the equation [5]:

$$Vi\theta = \sum_{Ki} (Fi\theta(x, y) - Pi\theta)^2 \quad (16)$$

$Fi\theta$ are the pixel values in the i th sector after a Gabor filter with angle θ has been applied. $Pi\theta$ is the mean of the pixel values. Ki is the number of pixels in the i th sector. A higher variance in a sector means that the ridges in that image were going in the same direction as is the Gabor filter. A low variance indicates that the ridges were not, so the filtering smoothed them out. The resulting 512 variance values (8×64) is the feature vector of the fingerprint scan. Therefore, each feature vector will have 512 value. **Table 2** shows 16 value out from 512 value of feature vector.

3.4. Fingerprints Matching Using KNN Neural Network

K-nearest neighbor classification is the simplest technique in machine learning, if you have a labeled data set $\{x_i\}$, and you want to classify some new item y , find the k elements in your data set that are closest to y , and then somehow average

Table 2. 16 value of feature vector from FP image.

317.755472110611	85.0608871268424
347.876625885873	144.168111602665
208.831596658908	198.447152449966
119.702662365649	267.354621591008
67.6820626488398	285.626540499418
58.8557991705111	270.602200430766
32.0457471125907	118.844510888489
15.3282659340089	266.967014183922

their labels to get the label of y [16].

The k-NN predict is computed using the features assembled in the matrices in a two-step process. In the first step, we have been calculating the distance between the features in the new dataset (test set) and the features in the previous dataset (training sets). In the second step, choosing k-NNs and have k smallest distance from distance set [17].

To find the k-NN based on the Euclidean distance, this mathematical equation is used [17]:

$$d(x, y) = \sqrt{\sum_{j=1}^N W_j^2 (x_j - y_j)^2} \quad (17)$$

where d is the number of forecast instances in the test set. We can calculate the distance between two scenarios using some distance function $d(x, y)$, where x , y are the matrix scenarios composed of N features $x = \{x_1, x_2, \dots, x_N\}$, $y = \{y_1, y_2, \dots, y_N\}$, N is the length of data, and the distance between the current performance and previous condition, W_j is the weight value of the dependent variable members of k-NN (kernel function) and j is the order of the k-NN based on their distance from the current performance condition and which the nearest with used the lowest order ($j = 1, \dots, K$) [17].

The training and testing of the FP images using the KNN neural networks will be explained in details:

a) Training Phase

In the training phase, core point and candidate core points is extracted from the FP image. For each point a feature vector is extracted, so the FP image in the training phase will have a number of feature vectors depending on the number of the points that the FP image have. Therefore, each FP image will have different number of feature vectors.

One image is trained for each person and this process will be repeat for all the 90 persons stored in the database and all the features data set will be stored outside the KNN neural network.

b) Testing Phase

The same process happen for the training FP image will be done in the testing phase for the testing FP image. Core point and candidate core points is extract from the test FP image. For each point a feature vector is extracted, so the FP

image in the testing phase will have a number of feature vectors depending on the number of points that the FP image have. Therefore, each FP image will have different number of feature vector.

The second step is the calculation of the distance (which will be the Euclidean distance) between each vector from the input vectors N and the whole number of vector for all the FP images in the database and find the minimum distance and store the person who it belong to. Moreover, repeat this step for all the input vectors N and finally it will be N suggested persons; the most repeating person will be the final identification matching result.

There are 8 images for each person, and one image has been used for training and 7 images for testing, and repeats this for all the 90 person. **Figure 8** will illustrates the flowchart of the data set for the train and test images. **Figure 8** will illustrates the flowchart of the data set for the train and test images.

3.5. Threshold Selection

The threshold selection process has been proposed. After suggest a person for each input vector and repeat this for all N input vectors. We calculate percentage or a score and called it matching score by taking the most repeating person divided by the all number of suggested persons (which is the same number of the FP input image vectors N) multiplying by 100%:

$$\text{Score} = \frac{\text{number of most repeating person}}{\text{all number of suggested person}} \times 100\%$$

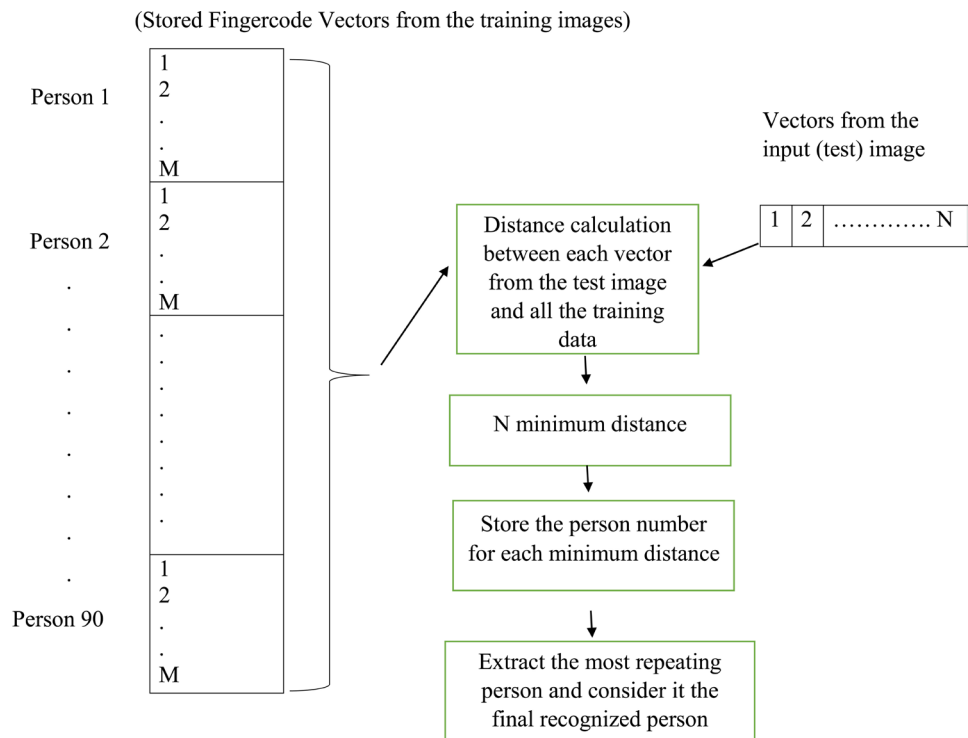


Figure 8. The flowchart of the data set for the train and test images. M and N means there are different number of feature vectors for each FP image.

If the test image or the unknown input image have Score larger than a specified threshold then the image will be accept and if else, the image will be reject.

For example, you can choose the threshold such high, that really no impostor scores will exceed this limit. As a result, no patterns are falsely accepted by the system. On the other hand, the client patterns with scores lower than the highest impostor scores are falsely rejected. In opposition to this, you can choose the threshold such low, that no client patterns are falsely rejected. Then, on the other hand, some impostor patterns are falsely accepted. If you choose the threshold somewhere between those two points, both false rejections and false acceptances occur [18].

4. Results and Discussion

The recognition rate (RR) has been extract for a range of thresholds values:

$$RR = \frac{\text{number of Right Accept FP images}}{\text{all number of FP images}} \times 100$$

Figure 9 shows the recognition rate line for the KNN in different range of threshold values. Depending on the choice of the score of the threshold, between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR) [19]. If a score of the threshold that is too high is applied to the classification scores, some of the client patterns are falsely rejected. Depending on the value of the threshold, between none and all of the client patterns will be falsely rejected. The

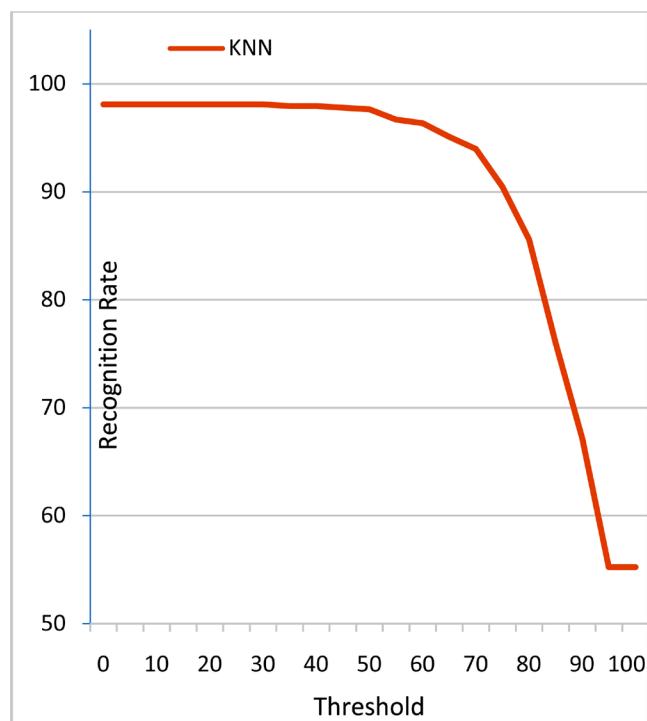


Figure 9. The Recognition Rate in the KNN for a range of threshold values.

fraction of the number of rejected client patterns divided by the total number of client patterns is called False Rejection Rate (FRR) [19]. So whenever the FAR and FRR are low it is better.

$$\text{FAR} = \frac{\text{number of False Accept FP images}}{\text{all number of FP images}} \times 100$$

$$\text{FRR} = \frac{\text{number of False Reject FP images}}{\text{all number of FP images}} \times 100$$

Also **Figure 10** show the FAR and FRR line for a range of threshold and for a total number of FP images equal to 720 and also **Table 3** shows the RR, FAR and FRR in a range of thresholds and show how while the threshold increased, the FAR will be decreased but the FRR will be increased. So, we find out that the 70% threshold is the best threshold can be consider because it have a good recognition rate for the FP images which is 93.9683% in the KNN matching technique and the FAR equal to 1.2698% and the FRR equal to 4.7619%. In comparing our work to [7], also they using k-Nearest Neighbors (k-NN) and the database contains 28 person, 7 sample for each person and the recognition rate reached to 98%, also [4] use KNN and FVC 2000 database (257 final images) and achieved a maximum accuracy of 98.81% and we use 90 person and 8 sample for each and used one for training and 7 for testing so as a total 720 test image have been used and the recognition rate was equal to 93.9683%. Our recognition rate is lower than [7] and [4] but we use large database than they do, and whenever the database is large, the error rate will be increased.

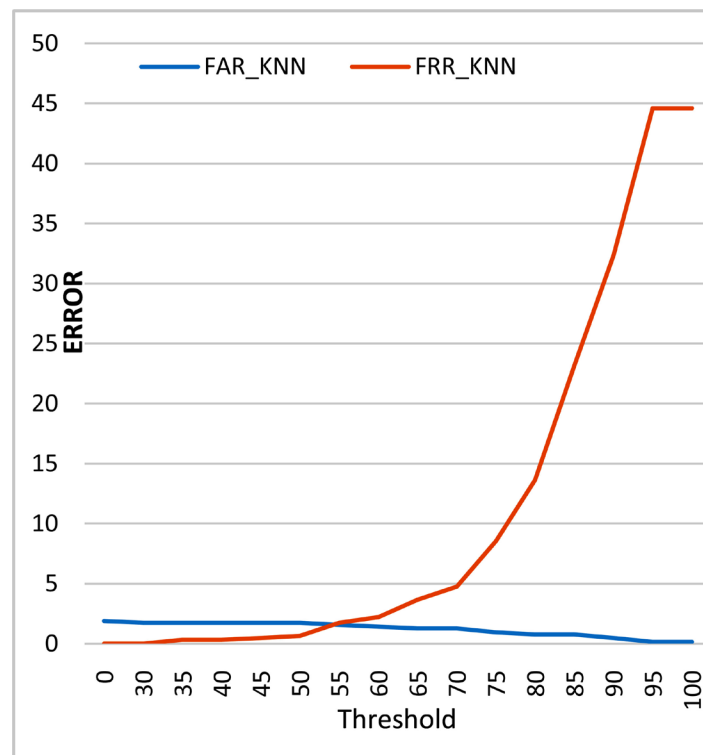


Figure 10. The FAR and FRR in the KNN for a range of threshold.

Table 3. The RR, FAR, and FRR for a range of thresholds.

Threshold %	Recognition Rate %	FAR %	FRR %
0	98.0952	1.9047	0
5	98.0952	1.9047	0
10	98.0952	1.9047	0
15	98.0952	1.9047	0
20	98.0952	1.9047	0
25	98.0952	1.9047	0
30	98.0952	1.7460	0
35	97.9365	1.7460	0.3175
40	97.9365	1.7460	0.3175
45	97.7778	1.7460	0.4762
50	97.6190	1.7460	0.6350
55	96.6667	1.5873	1.7460
60	96.3492	1.4285	2.2223
65	95.0794	1.2698	3.6508
70	93.9683	1.2698	4.7619
75	90.4762	0.9523	8.5715
80	85.5556	0.7936	13.6508
85	76.0317	0.7936	23.1747
90	67.1429	0.4761	32.3810
95	55.2381	0.1587	44.6032
100	55.2381	0.1587	44.6032

5. Conclusions

This paper presents a design and implementation of Fingerprint recognition system using Filterbank_based algorithm for a number of core point and candidate core points in the feature extraction step and using KNN neural matching techniques in the matching step and threshold selection technique has been propose. During the implementation of the case studies, a number of conclusions have been drawn based on the practical results obtained from the implemented systems and the followings are the most important ones:

- 1) Taking 8 images for 90 persons from our reality with different ages and rotating the fingerprint image as possible as we can means that the final results are more real and applicable.
- 2) Include image enhancement in the fingerprint identification system improves the quality of input fingerprint image, reduces extraction of false features vectors and minimizes matching errors.
- 3) Core point and candidate core points extraction algorithm is a good algorithm and appropriate as a base for the feature extraction algorithm.

4) Feature extraction algorithm based on Filterbank_based algorithm produces a good feature vector in comparing among fingerprints that different from one person to another.

5) KNN Neural networks provide an appropriate matching result and the 70% threshold value of the threshold technique provide appropriate and good results for FP images (90 persons and 8 samples for each) that belong to database which is 93.9683% recognition rate, 1.2698% FAR and 4.7619% FRR.

References

- [1] Sangeetha, S., Radha, N., Sangeetha, S. and Radha, N. (2013) A New Framework for IRIS and Fingerprint Recognition Using SVM Classification and Extreme Learning Machine Based on Score Level Fusion. *Proceedings of 7th International Conference on Intelligent Systems and Control (ISCO 2013)*, Coimbatore, 4-5 January 2013, 183-188. <https://doi.org/10.1109/ISCO.2013.6481145>
- [2] Ali, M.M.H., Mahale, V.H., Yannawar, P. and Gaikwad, A.T. (2016) Overview of Fingerprint Recognition System. *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)—2016*, Chennai, 3-5 March 2016, 1334-1338. <https://doi.org/10.1109/ICEEOT.2016.7754900>
- [3] Jain, A.K., Prabhakar, S., Hong, L. and Pankanti, S. (2000) Filter Bank Based Fingerprint Matching. *IEEE Transactions on Image Processing*, **9**, No.5. <https://doi.org/10.1109/83.841531>
- [4] Batra, D., Singhal, G. and Chaudhury, S. (2004) Gabor Filter Based Fingerprint Classification Using Support Vector Machines. *Proceedings of the IEEE INDICON, First India Annual Conference*, Kharagpur, 20-22 December 2004, 256-261. <https://doi.org/10.1109/INDICO.2004.1497751>
- [5] Chikkerur, C.W. and Govindaraju, V. (2004) A Systematic Approach for Feature Extraction in Fingerprint Images. *ICBA 2004*, Hong Kong, 15-17 July 2004. https://doi.org/10.1007/978-3-540-25948-0_48
- [6] Liu, J., Li, B. and Liang, D. (2007) Design and Implementation of FPGA-Based Modified BKNN Classifier. *IJCSNS International Journal of Computer Science and Network Security*, **7**, No3.
- [7] Alzubaydi, D.A. and Abed, T.M. (2014) Adaptive Genetic Algorithm and KNN for Fingerprint Identification. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, **1**, No. 7.
- [8] Aveenmohee (2007) Development of Fingerprint Recognition System. Thesis, University of Al-Nahrain, Al-Nahrain.
- [9] Gonzalez, R.C. and Woods, R.E. (2002) *Digital Image Processing*. Prentice Hall, Upper Saddle River.
- [10] Sherlock, B.G., Monro, D.M. and Millard, K. (1994) Fingerprint Enhancement By Directional Fourier Filtering. *IEE Proceedings—Vision, Image and Signal Processing*, **141**, 87-94. <https://doi.org/10.1049/ip-vis:19949924>
- [11] Nilsson, K. and Bigun, J. (2002) Complex Filters Applied to Fingerprint Images Detecting Prominent Symmetry Points Used for Alignment. In: Tistarelli, M., Bigun, J. and Jain, A.K., Eds., *Biometric Authentication*, Springer, Berlin. https://doi.org/10.1007/3-540-47917-1_5
- [12] Nilsson, K. and Bigun, J. (2003) Localization of Corresponding Point in Fingerprint by Complex Filtering. *Pattern Recognition Letters*, **24**, 2135-2144.

[https://doi.org/10.1016/S0167-8655\(03\)00083-7](https://doi.org/10.1016/S0167-8655(03)00083-7)

- [13] Nilsson, K. and Bigun, J. (2003) Localization of Corresponding Points in Fingerprints by Complex Filtering. School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University, Halmstad.
- [14] Hong, L., Wan, Y. and Jain, A. (1998) Fingerprint Image Enhancement: Algorithm and Performance Evolution. *IEEE Transactions on Pattern Analysis and Matching Intelligence*, **20**, No. 8.
- [15] Thai, R. (2003) Fingerprint Image Enhancement and Minutiae Extraction. PhD. Thesis, University of Western Australia, Perth.
- [16] <http://andrew.gibiansky.com/blog/machine-learning/k-nearest-neighbors-simplest-machine-learning/>
- [17] Chen, C.-R. and Unit Three Kartini (2017) k-Nearest Neighbor Neural Network Models for Very Short-Term Global Solar Irradiance Forecasting Based on Meteorological Data. Energies, Taiwan.
- [18] SYRTs Technology Corp (2004) Technical Document about FAR, FRR and EER. SYRTs Technology Corp, ver. 1.0, 2004.
- [19] Jain, A.K., Feng, J.J. and Karthik, N. (2010) Fingerprint Matching. *IEEE Computer Society*, **43**, 36-44. <https://doi.org/10.1109/MC.2010.38>