

## Artificial Neural Network-Based Fingerprint Classification and Recognition



Zainab S. Naser<sup>1</sup>, Hiyam N. Khalid<sup>1</sup>, Asraa Safaa Ahmed<sup>2</sup>, Mustafa Sabah Taha<sup>3\*</sup>, Mohammed Mahdi Hashim<sup>4</sup>

<sup>1</sup> Department of Computer Engineering, Imam Ja'afar Al-Sadiq University, Baghdad 10047, Iraq

<sup>2</sup> Department of Computer sciences, College of Science, Diyala University, Diyala 32001, Iraq

<sup>3</sup> Missan Oil Training Institute, Ministry of Oil, Baghdad 10064, Iraq

<sup>4</sup> Interior Design Colleges, Uruk University, Baghdad 10069, Iraq

Corresponding Author Email: [mustafa@moti.oil.gov.iq](mailto:mustafa@moti.oil.gov.iq)

<https://doi.org/10.18280/ria.370116>

### ABSTRACT

**Received:** 4 September 2022

**Accepted:** 28 December 2022

#### **Keywords:**

*fingerprint, biometric technique, image data, image enhancement, image enhancement, deep learning*

The most commonly used biometric technique for identifying people is fingerprint-based biometrics. It is divided into two parts: verification (if this individual is genuinely himself) and identification (identifying a person from a pool of persons). Due to the enormous number of comparisons required, the Automatic Fingerprint Identification System (AFIS), which typically conducts two stages: feature extraction and matching, had difficulties with a large database of fingerprint photos for the real-time application. So, more classification stages for complete fingerprint data can make it faster for the AFIS to identify a person. In this paper, we presented a classification method for identifying detailed fingerprint information by utilizing a deep learning approach to support the operations for classifying, identifying, and recognising the fingerprint. The proposed method was designed to differentiate certain fingerprint information, such as left-right hand classification, sweat-pore classification, scratch classification, and finger classification. We privately created our fingerprint image dataset due to high personalization and security concerns (25 fingerprint images in the dataset with seven features for each image through the scanning technique). Finally, the research results for the proposed study were accurate and outperformed previous results.

## 1. INTRODUCTION

Customers typically use various identifying evidence amounts, keys, or smart cards to protect their hardware, software, or systems that contain sensitive data. However, these defense mechanisms could be misunderstood, stolen, or disregarded. Consequently, biometric systems rather than the conventional security measures have been deployed. Biometrics, which is the automatic identification of an individual based on unique physiological or behavioral characteristics, are inherently more reliable and eligible than conventional methods like PIN and passwords in identifying a trustworthy person from a fraudulent impostor [1-5].

The Automated Fingerprint Identification System (AFIS) is a very popular security system to identify or recognize the right person because the fingerprint is unique and constant [6, 7]. It is one of the most reliable biometric technologies among the several major biometric technologies that are either currently ready-made or under investigation. The AFIS is a biometric identification method that collects, examines, and stores fingerprint data using digital imaging technology. The Federal Bureau of Investigation (FBI) of the United States frequently used the AFIS in criminal investigations [8].

The fingerprint is a mark for everyone, and it is located at the ends of all fingers, as confirmed by the Czech anatomy scientist 'Purkinje' in 1823 [9], when he discovered the fact of fingerprints and found that fine lines located in the headers of the fingers (bony) differ from person to another person. Later, in 1858, William Herschel (England) adopted the fingerprint

to identify human beings [10]; usually, the left thumb fingerprint is enough to prove identity. It consists of heights and depressions in the skin-forming lines that affect them after touching the surfaces [11].

The fingerprint as a biometric system is classified into two systems. Initially, verification is needed to determine if this person is truly himself. Secondly, identification is to identify a person from a pool of people [12]. The fingerprint is a visual representation of local parallel edges and valleys that characterize the surface of the fingertips [13].

Since there are many variations in the finger shapes of young persons under the age of 18 [14], the discrimination of human fingerprints is a challenging problem. Because of this, many techniques may be used to accurately identify fingerprints [15-17]. However, there are numerous methods for feature extraction and numerous learning processes that go into the recognition of human fingerprints. The technology for fingerprint recognition currently encounters various challenges. Particularly after converting the image to two-dimensional (2D) photos, certain details may be overlooked and destroyed, and the fingerprint that is recovered might not be accurate and full; for instance, it might be missing some areas or contain scars that are twisted or stained. These will have a negative impact on the effectiveness of feature extraction and fingerprint recognition [18]. In order to remove the obstacles that lower fingerprint classification accuracy and matching accuracy, new algorithms are required.

In order to extract features from fingerprint images and match them with a dataset after training the Neural Network

NN for more security with a high level of accuracy, this paper's main objective is to manually create a dataset of 25 fingerprint images with seven features for each image through the scanning technique.

The remainder of this paper is structured as follows. The proposed methodology for the wavelet-based fingerprint recognition framework is discussed in Section 2. The testing of the suggested system was described in Section 3. The simulation results and discussion were reported in Section 4. Finally, Section 4 presents the planned study's final observations.

## 2. PROPOSED METHOD

The four required steps are described step-by-step in the proposed system's four stages. The first step involves gathering fingerprints (building the database); the second step involves picture preprocessing (enhancing, digitalization), which turns the image into a formula that the computer can understand. The third stage entails training NNs using fingerprint image data. The fourth stage entails classification and matching procedures for evaluating fingerprint images and their compatibility with the database in order to diagnose tested fingerprints and determine ownership. The general phases of the proposed fingerprint system are explained in Figure 1.

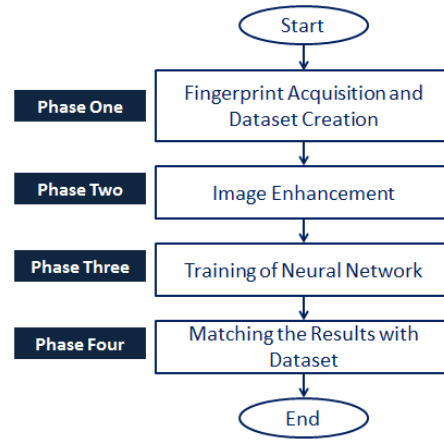


Figure 1. The general block diagram of fingerprint design

### 2.1 Image acquisition and database creation

In this work, the input, the fingerprints, is taken manually and in traditional ways by pressing the donor's finger that contains ink on a white paper several times for the same finger. Ink is applied to the donor's finger for every input. In this scenario, the left thumbprint is considered. The image of the fingerprint is put into the computer using scanners that come with most homes.

Table 1. The collection of fingerprint samples

Sequence	Training image	Training image	Training image	Training image	Testing image
Donor no.1					
Donor no.2					
Donor no.3					
Donor no.4					
Donor no.5					

Due to high security and personalization cases, we privately constructed our own fingerprint image database. Moreover, during the database collection, any movement leads to the fingerprint's distortion, such as not pressing correctly. This unwanted movement will produce a non-desired shape and distorted image; even dust and soil in the air can cause

inaccurate fingerprint images. For that reason, accuracy is required. For more knowledge, a computer cannot extract information from a low-quality image, and processing the distortion fingerprint image becomes very hard. Therefore, for the reasons that have been mentioned, the fingerprint is collected in the following manner: An A4 size sheet is divided

into several fields of 44 cm, and the fingerprint donor places his left thumb finger in the middle of the area several times in different positions (actually 5 times), as it has been used in reference [16]. This process gives five fingerprint images, and four of them are used in the training of NNs, while the remaining one was used for the testing process. As shown in Table 1, after collecting samples from a volunteer, fingerprint samples are entered into a computer by a scanner, and robust and special algorithms are used to cut and convert them into single images. The images are labeled to ensure that each fingerprint image belongs to the same person and are later entered into the database. Using scanners to take the image directly from the person and store it in the database is the best way to make sure that the fingerprint belongs to the right person.

The proposed method is divided into two approaches: the first one is to extract the characteristics and train the NN, while the second one is to test the first approach. The general approach of the proposed system is depicted in Figure 2. The flowchart is simply divided into several steps, and each step represents an important stage of image processing.

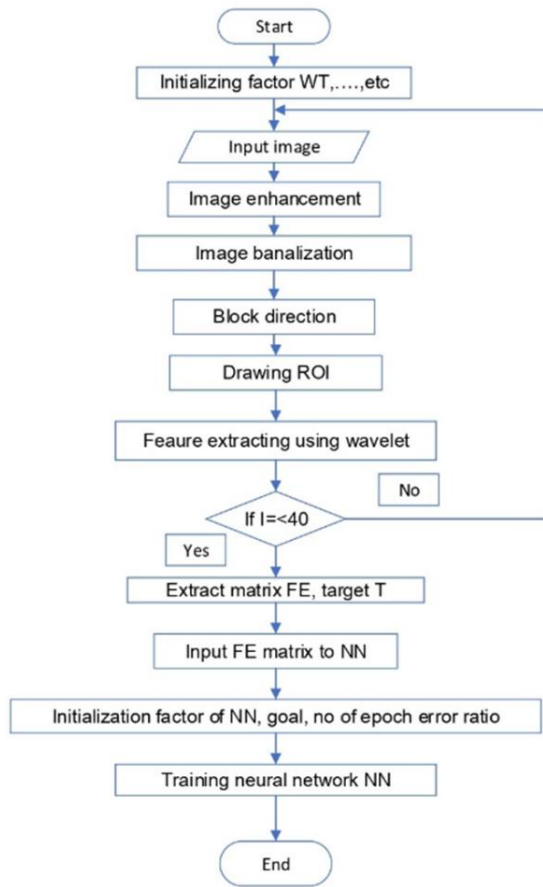


Figure 2. The flowchart of the proposed system

### 2.1.1 Input image

The process of inserting images requires the creation of a counter identical to the number of images used to ensure that all images have been correctly entered into the system database. The images are processed separately. It is necessary to reach the maximum number of entered images; if that is achieved, the NN will initiate the training process. Figure 3 illustrates the flowchart of image counter initialization.

### 2.1.2 Image enhancement

After inserting the images into the system, the process of image enhancement begins. Each image has its own characteristics, and these characteristics are used as input in the subsequent processes; thus, the images must be improved. There are several ways to enhance the images; in this paper, the Fast Fourier Transform (FFT) method has been used to enhance the fingerprint images.

In the FFT method, the image is divided into small processing blocks (32 by 32 pixels) before performing the Fourier transform as follows:

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

where,  $u=0, 1, 2, \dots, 31$  and  $v=0, 1, 2, \dots, 31$ .

When enhancing a particular block by its dominant frequencies, we repeatedly multiply the block's FFT by its magnitude, where the magnitude of the original FFT is equal to  $\text{abs}(F(u,v)) = |F(u,v)|$ , the results in an enhanced block will be looks like the Eq. (2).

$$g(x,y) = F^{-1} \{ F(u,v) \times |F(u,v)|^k \} \quad (2)$$

where,  $F^{-1}(F(u,v))$  is done by the Eq. (5), and  $(k)$  in formula is an experimentally determined constant, which we choose  $k=0.45$  to calculate. While having a higher "k" improves the appearance of the ridges, filling up small holes in ridges, having too high a "k" can result in false joining of ridges.

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

where,  $x=0, 1, 2, \dots, 31$  and  $y=0, 1, 2, \dots, 31$ .

In the present study, the above equations have been utilized to enhance the images. Figure 4 shows the image after FFT enhancement. While Figure 5 illustrates the image enhancement flowchart.

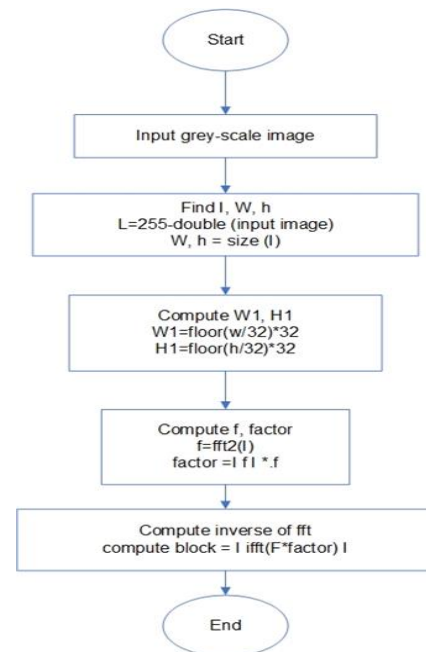
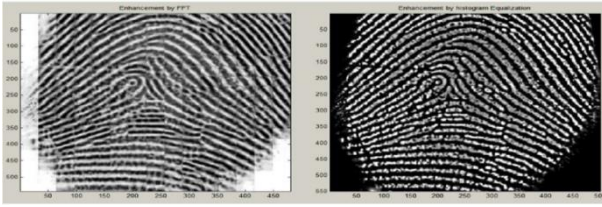
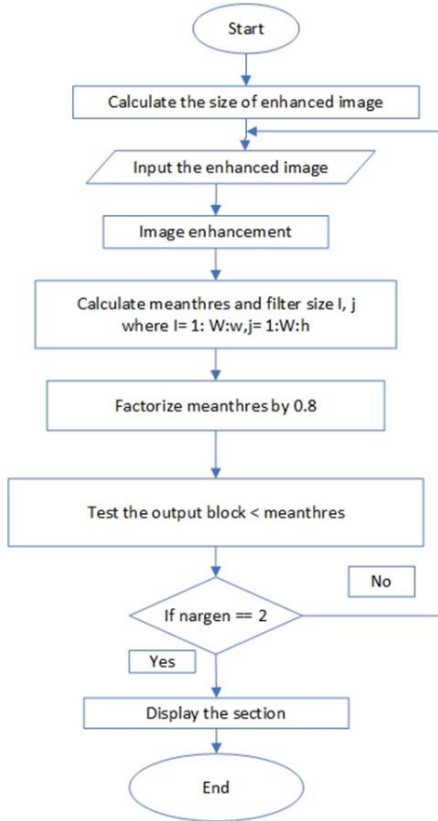


Figure 3. Image counter initialization flowchart



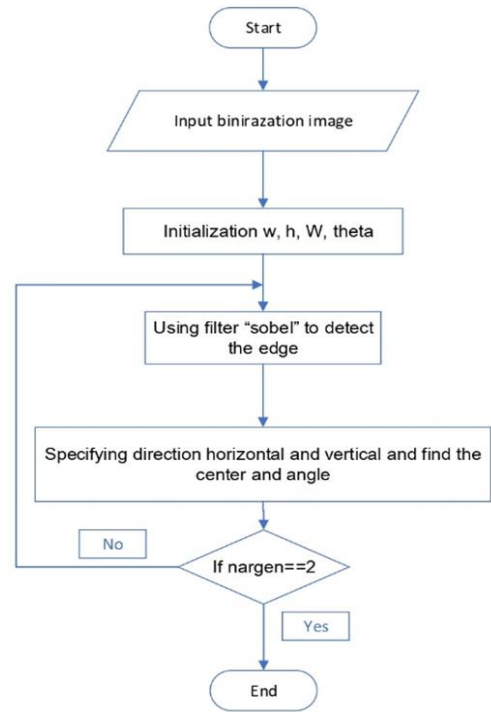
**Figure 4.** The enhancement process of fingerprint using FFT: Original image (right), Enhanced image (left)



**Figure 5.** Image enhancement flowchart

### 2.1.3 Image binarization

There are various different types of images. Some of these formats include binary, gray scale, and color (two-color). However, the image must be of a type that complies with the algorithm when working on an image matching system. Some filters can only be used with gray scale images, some only with color images, while some only need a two-color image. Because the proposed system technique deals with the function (wavelet), which requires the picture to be binary, two-color images is handled in this paper. Each image's characteristics are extracted. The flowchart displays the procedure for transforming the input image into a two-color image (Figure 6). In this case, nargen returns the quantity of function input parameters supplied in the call to the active function. Only the function body of a function uses this syntax. The image (C) provides the data as an array and makes use of the entire color spectrum in the color map. Each C element identifies the color for 1 picture pixel. An m-by-n grid of pixels, where m is the number of columns and n is the number of rows in C, makes up the final image. The centers of the relevant pixels are determined by the row and column indices of the elements. Knowing the pace of the resulting matrix, which is here referred to as the most important factor, when the filter begins to move on the image (mean threes).



**Figure 6.** Image binarization flowchart (The input image is in gray scale)

### 2.1.4 Block direction estimation

The process of determining trends is very important because it gives the angle of deviation of the lines that form the fingerprint, which is the characteristic that can be distinguished by the fingerprint and the other. The following steps explain the procedures of the block direction estimation.

- a) For each block of the fingerprint image with a size of WW, estimate the block direction (W is 16 pixels by default). What the algorithm performs is:
  - i. Determines the gradient values along the block's x and y axes ( $g_x$  and  $g_y$ , respectively). To complete the work, two Sobel filters are employed.
  - ii. For each block, the following formula is used to approximate the block direction using the least square method.

$$\tan 2\beta = 2 \sum \sum (g_x * g_y) / \sum \sum (g_x^2 - g_y^2) \quad (4)$$

Understanding the formula is simple if you think of the gradient values along the x- and y-axes as cosine and sine values. As a result, the block direction's tangent value is approximated almost exactly as shown by the formula below.

$$\tan 2\theta = 2 \sin \theta \cos \theta / (\cos \theta - \sin^2 \theta) \quad (5)$$

- b) Once all block directions have been evaluated, the blocks with no significant information (ridges) are removed using the formula below:

$$E = \{2 \sum \sum (g_x * g_y) + \sum \sum (g_x^2 - g_y^2)\} / W * W * \sum \sum (g_x^2 + g_y^2) \quad (6)$$

where, all the border of summation from (0) to (32).

If there is only one fingerprint in each image, the block is considered a background block if its certainty level (E) is lower than a threshold for each block. The Sobel operator highlights high spatial frequency regions that correspond to

edges while measuring a 2-D spatial gradient in an image. It is typically employed to determine the roughly absolute gradient magnitude at each location in an input gray scale image. The operator is dependent on two 3x3 convolution kernels, as shown in Figure 7, at least in theory. One kernel is just the other kernel turned 90 degrees. The Roberts Cross operator and this are practically identical.

-1	0	+1
-2	0	+2
-1	0	+1

G<sub>x</sub>

+1	+2	+1
0	0	0
-1	-2	-1

G<sub>y</sub>

**Figure 7.** Utilizing Sobel convolution kernels

These kernels, one for each of the two perpendicular orientations, are made to react as strongly as possible to edges that run vertically and horizontally in relation to the pixel grid. The input picture and the kernels can be used individually to create distinct measurements of the gradient component in each orientation (call these G<sub>x</sub> and G<sub>y</sub>). These can then be combined to determine the gradient's direction and absolute magnitude at each site. The gradient's magnitude can be calculated by:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (7)$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y| \quad (8)$$

The edge's direction with respect to the pixel grid, which creates the spatial gradient, is determined by:

$$\theta = \arctan(G_y/G_x) \quad (9)$$

In this instance, orientation 0 is understood to indicate that the direction of the image's maximum contrast, from black to white, travels from left to right. Subsequent angles are then measured counterclockwise from this. The two components of the gradient are readily computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 8. Frequently, this absolute magnitude is the only output the user sees.

Using this kernel, the approximate magnitude is given:

$$by: |G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)| \quad (10)$$

P1	P2	P3
P4	P5	P6
P7	P8	P9

**Figure 8.** Pseudo-convolution kernels used to quickly compute approximate gradient magnitude

### 2.1.5 Drawing the ROI

For each fingerprint image, it is typically only advantageous to recognize a Region of Interest (ROI). First, the image area without useful ridges is eliminated because it only contains background data and probably noise. As the minutiae in the bound zone are confused with those false minutiae produced when the ridges are outside of the sensor, the bound of the remaining effective area is then sketched out. Two steps are taken to obtain the ROI. Block direction estimate and direction variety checking make up the first stage, while various morphological approaches are used in the second.

### 2.1.6 Fingerprint thinning

This step is preceded by the process of extracting properties, which is a very important process because it naturally gives the best shape to the image and narrows the scope in which the properties of the image will be extracted. It works on the binary image only.

When used with the 'thin' option, (bwmorph) [13] uses the following algorithm: Firstly, in sub iteration, delete pixel p if and only if G1, G2, and G3 are all satisfied. Secondly, in sub iteration, delete pixel p if and only if the conditions G1, G2, and are all satisfied.

#### a) Condition G1:

$$X_H(P) = 1$$

where,

$$X_H(P) = \sum_{i=1}^4 b_i \quad (11)$$

where,

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

x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>8</sub> are the values of the eight neighbors of p, starting with the east neighbor and numbered in counterclockwise order.

#### b) Condition G2:

$$2 \leq \min\{n_1(P), n_2(P)\} \leq 3$$

where,

$$n_1(p) = \sum_{k=1}^4 x_{2k-1} \vee x_{2k} \quad (12)$$

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

#### c) Condition G3:

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

#### d) Condition G3':

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

The proposed study uses a fast thinning method, it iterates once during the two sub-iterations. The iterations are performed until the image stops altering. In this study, users enter an infinite number of repetitions (n=Inf) in a way that keeps the computation process unheavy.

### 2.1.7 Feature extraction

There are various ways to extract properties from images and the most famous ones are as shown below:

- i. Gabor filter work with grayscale image.
- ii. Wavelet work with binary image.

In this paper, the second method (wavelet) is used [19], a filter that works with two-color images to extract the most important characteristics of the image and give it in the form of a matrix of one line and seven columns, where each image is turned into seven values and expressed energy. A wavelet is a two-dimensional wavelet analysis function.

$$C = [ A(N) | H(N) | V(N) | D(N) | \dots H(N-1) | V(N-1) | D(N-1) | \dots | H(1) | V(1) | D(1) ]$$

where, A, H, V, D, are row vectors such that:

V=vertical detail coefficients

D=diagonal detail coefficients

Each vector is the vector column-wise storage of a matrix.

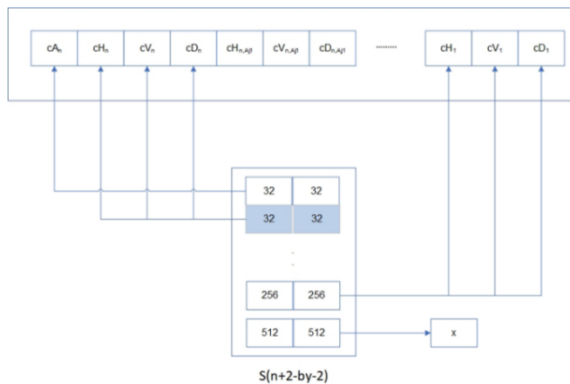
Matrix S is such that

S(1,:)=size of approximation coefficients(N).

S(i,:)=size of detail coefficients(N-i+2) for i=2, ...N+1 and

S(N+2,:)=size(X).

Figure 9 explains the feature extraction process using wavelet transform algorithm.



**Figure 9.** Feature extraction process using wavelet transform algorithm

### 2.1.8 Initializing target and training of NN

After obtaining the characteristics from the last filter, we will have a matrix of characteristics corresponding to a matrix of goals that are identical, giving us an equal number of people whose fingerprint has been inserted into the training. Once the target matrix has been made, the NN is given initial values to start the training. Initial values are given as follows:

1) Input, 7 neurons; hidden layer, 50 neurons; output, 10 neurons.

2) The number of training times is 1000 times and after doing trial and error operation, the NN is trained to be stabilized on the following data.

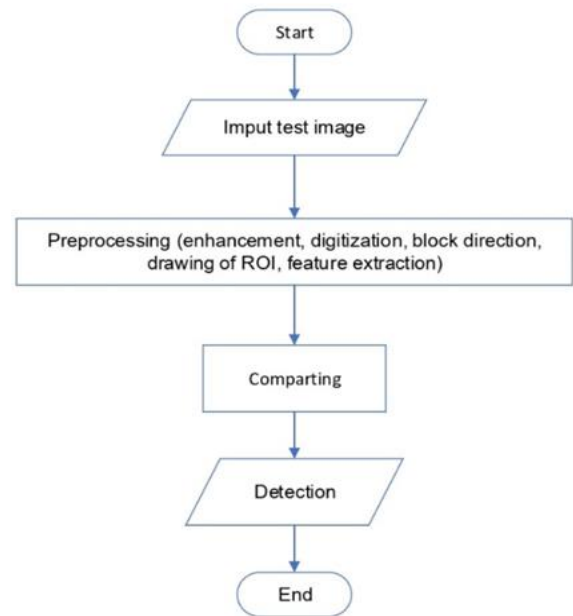
3) Input, 7 neurons; hidden, 450 neurons; output, 10 neurons with 14332 epochs.

These results will be discussed in detail in the next section.

## 3. FINGERPRINT IMAGE TEST

After training the proposed system, the NN is ready for work and ready for testing. As we pointed out that the number of fingerprints collected was 25 fingerprints. The network was trained on 15 fingerprints and 10 left for testing. When the

samples are tested, the match ratio was 100%. Therefore, there are another 10 fingerprint images gotten from a strange donor that were tested with the testing program. The other four fingerprint images were distorted by 10% to 20% using computer programs for image processing. Figure 10 represents the flowchart of the testing operation. The testing process does not require the training of the network again because it has already been trained on the training samples only. The test images are entered into the test program without training the neural network. Once all the processing steps mentioned above have been completed on the image to be tested, the program will compare it with the image obtained with the matrix that is extracted after the NN training. The value from the testing process is compared to the highest value from the NN training in the matrix.



**Figure 10.** Test operation flowchart

## 4. RESULTS AND DISCUSSION

After preparing all the images and the parameters, the NN will start the learning process. It is required to satisfy the best value of its internal weight; for this reason, the NN needs to run several times and repeat the training on the same image to achieve the best value. The network includes many variables, but the important ones are the number of hidden layers and the number of epochs using trial and error. The training process is repeated several times depending on variations in the number of the hidden layers and the number of epochs, as shown in Table 2. The network achieved the best value and achieved good results.

**Table 2.** The results at a fixed number of epochs (10000)

No.	Input neuron	Hidden neuron	Output neuron	Training
1	7	50	10	0.9849
2	7	100	10	0.96874
3	7	200	10	0.98763
4	7	225	10	0.9923
5	7	250	10	0.9849

Figure 11 shows the result of first time of training with 50 hidden neurons. The mean square error is high approximately

near  $2.8 \times 10^{-3}$ , and the target is approach to 0.9849. There result is so far from specified mean square error about ( $1 \times 10^{-5}$ ) and target equal to 1.

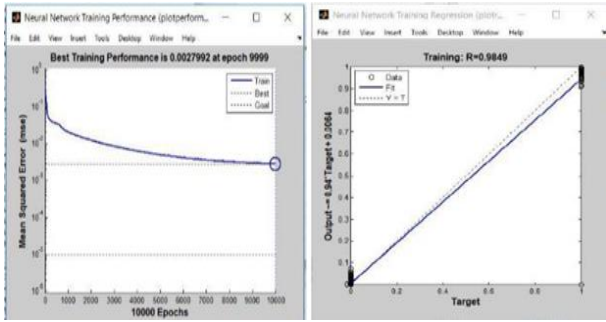


Figure 11. Mean square error and training goal of row No. 1

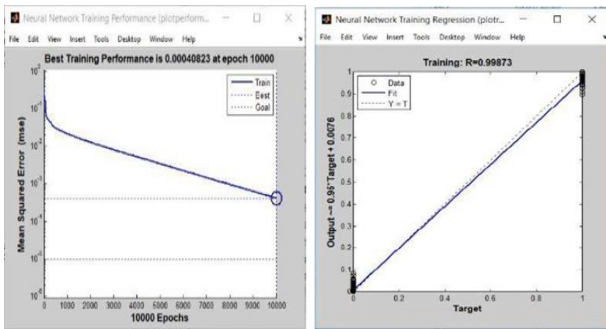


Figure 12. Mean square error and training goal of row No. 2

Figure 12 shows the result of the second time of training with 100 hidden neurons. The mean square error was about  $5.8 \times 10^{-3}$ , and the target was almost 0.96874. These results are far from the specified mean square error of about  $1 \times 10^{-5}$  and target value of 1.

Figure 13 shows the result of the third time of training with 200 hidden neurons. The mean square error showed a decrease to about  $4.1 \times 10^{-4}$  and the target was around 0.99873. These results are also far from the specified mean square error of about  $1 \times 10^{-5}$  and target value of 1.

Figure 14 shows the result of the fourth time of training with 225 hidden neurons. The mean square error decreased to approximately  $2.5 \times 10^{-4}$ , and the target was 0.99923. Although these results are far from the specified mean square error of about  $1 \times 10^{-5}$  and target value of 1, they are still the best results so far achieved.

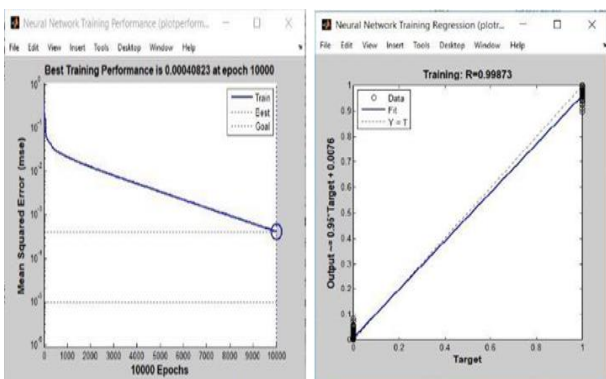


Figure 13. Mean square error and training goal of row No. 3

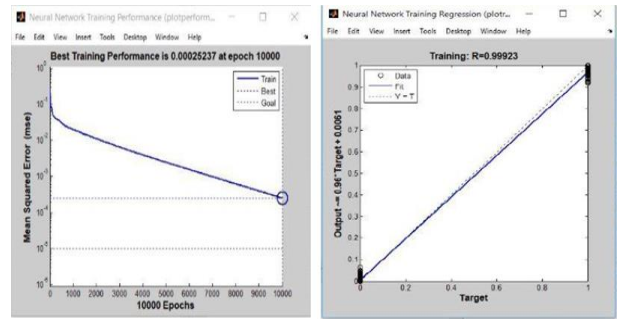


Figure 14. Mean square error and training goal of row No. 4

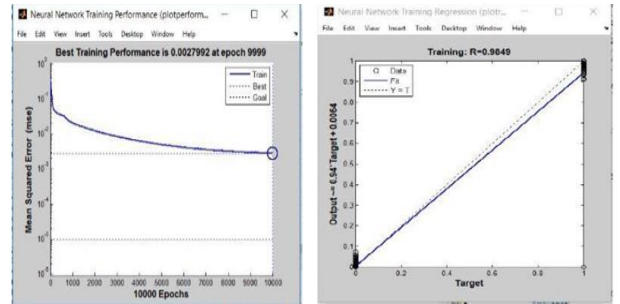


Figure 15. Mean square error and training goal of row No. 5

Figure 15 shows the result of the fifth time of training with 250 hidden neurons. The mean square error was high (approximately  $2.7 \times 10^{-5}$ ) and the target were approaching 0.9849. Again, these results are not as good as the result obtained in Figure 15.

The value in the shadow row number 4 was the best result, according to the results in Table 2, indicating that the number of epochs should be variable while the number of hidden layers should be fixed. The following step entails setting the hidden layer count to 225 and attempting to alter the number of epochs as indicated in Table 3.

Table 3. The result at a fixed number of hidden neurons and variable epochs

No.	Input neuron	Hidden neuron	Output neuron	Training goal	epochs
1	7	225	10	0.99846	10000
2	7	225	10	0.99994	20000

The training on 10000 epochs repeated the training goal from (0.9923) to (0.99846) as shown in Table 3 because the internal weight of the hidden neural approach was stable. After changing the number of epochs to 20000 times, the network terminated the training at 16200 attritions with a training goal-reaching of 0.99994 as shown in Figure 16.

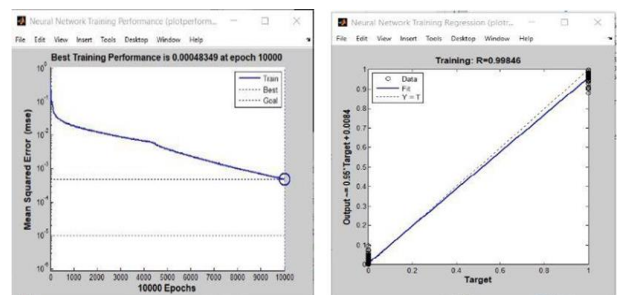


Figure 16. Mean square error and training goal of row No. 1 from Table 2

Figure 16 shows that the mean square error was still high compared to the neural network's goal, but the training is better than the final training due to the weight's stability.

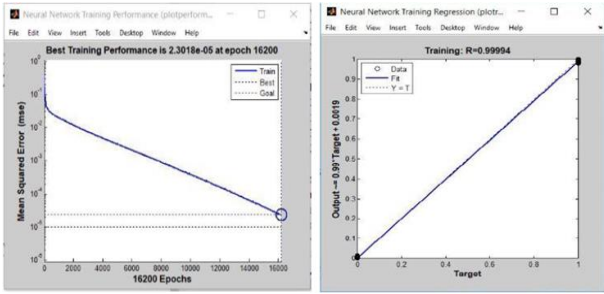


Figure 17. Mean square error and training goal of row No. 2 from Table 2

After repeating training with 20000 epochs, the mean square error became closer to the specific value ( $2.3 \times 10^{-5}$ ) and close to  $1 \times 10^{-5}$  while the value was 0.99994 as shown in Figure 17. This represents the best result achieved with the neural network. With this result, the best way of training is to start with a fixed number of epochs at 20000 times and change the number of hidden layers as shown in Table 4.

Table 4. The third step of fixing the internal weight

No.	Input neuron	Hidden neuron	Output neuron	Training goal	epochs
1	7	300	10	0.99994	16912
2	7	350	10	0.99994	15257
3	7	400	10	0.99994	14805
4	7	450	10	0.99994	14332

After achieving these results at 450 hidden neurons and 14332 attritions, the neural network's weight reached the best value.

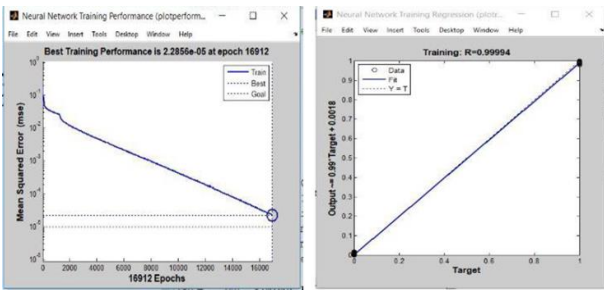


Figure 18. Mean square error and training goal of row No. 1 from Table 3

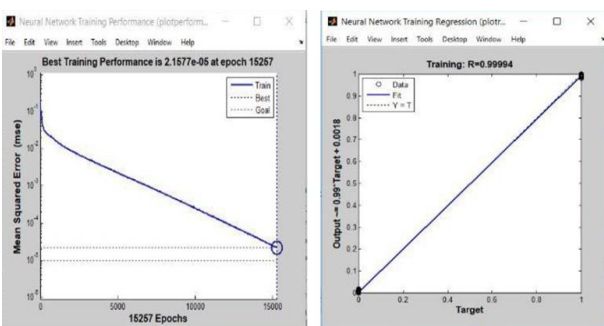


Figure 19. Mean square error and training goal of row No. 2 from Table 3

Figure 18 shows the result of the first time of training with 300 hidden neurons. The mean square error is approximate  $2.28 \times 10^{-5}$  and the target is approaching 0.99994.

Figure 19 shows the result of the first time of training with 350 hidden neurons. The mean square error is approximate  $2.1 \times 10^{-5}$  and the target is approaching 0.99994.

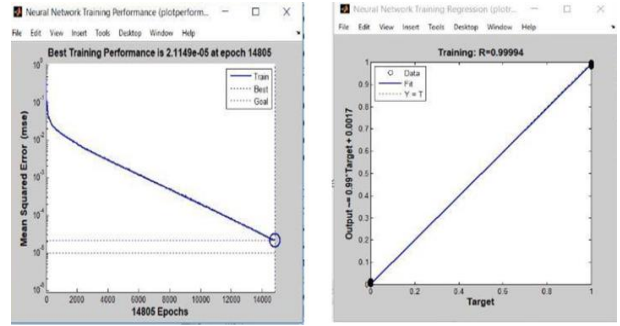


Figure 20. Mean square error and training goal of row No. 3 from Table 3

Figure 20 shows the result of the third time of training with 400 hidden neurons. The mean square error is approximate  $2.11 \times 10^{-5}$  and the target is approaching 0.99994.

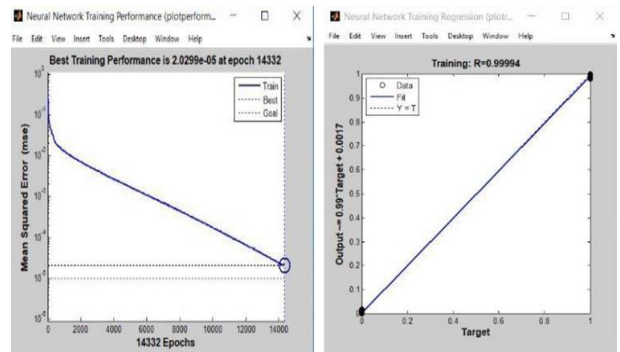


Figure 21. Mean square error and training goal of row No. 4 from Table 3

Figure 21 shows the result of the fourth time of training with 450 hidden neurons. The mean square error was around  $2.029 \times 10^{-5}$  and the target was approaching 0.99994. It should be noted that this is not the best approach to get the best results because there are many other ways, such as momentum term and LM that make neural networks achieve the best result with the minimum number of attrition and hidden neurons.

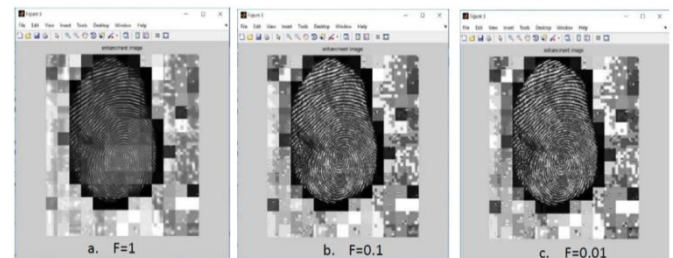


Figure 22. The effect of (f) factor on image enhancements

To summarize what was presented in the current paper, the algorithms that have been used in the present study depend on several factors, and these factors directly affect the result of the training and testing of a system. For instance, factor (f) is



a parameter for image enhancement; whenever the value of (f) is small, the output image enhancement will be better, as shown in Figure 22.

## 5. CONCLUSION AND FUTURE OUTLOOK

In fact, the fingerprint automatic recognition system depends on the accuracy of the input image's features extraction process. Many factors can make the fingerprint systems work incorrectly; among them are poor image quality, dust, devices used in the system, and the person's expertise in using the system; all of these reasons have a significant impact on the recognition process's efficiency.

There are various approaches to satisfy the aim of this study, meaning that the adopted approach may not be the best way but is used for the meantime. The NN depends on features entered into it and does not reach a stable state from the first training, requiring more than one training to achieve its goal. In future works, authors should focus on online fingerprint recognition systems. Furthermore, the effect of the following parameters ought to be studied:

- i. Thinning and enhancement of fingerprint images using other methods of image processing.
- ii. Stability of the system against an increasing number of input images.
- iii. Dynamic of fingerprint databases whenever new fingerprints are deposited.
- iv. Utilizing the deep learning approach to taring the methods such DNN or CNN approaches.

## REFERENCES

- [1] Malarvizhi, N., Selvarani, P., Raj, P. (2020). Adaptive fuzzy genetic algorithm for multi biometric authentication. *Multimedia Tools and Applications*, 79(13): 9131-9144. <https://doi.org/10.1007/s11042-019-7436-4>
- [2] Paramasivan, S.K. (2021). Deep learning based recurrent neural networks to enhance the performance of wind energy forecasting: A review. *Revue d'Intelligence Artificielle*, 35(1): 1-10. <https://doi.org/10.18280/ria.350101>
- [3] Luo, S., Gu, Y., Yao, X., Fan, W. (2021). Research on text sentiment analysis based on neural network and ensemble learning. *Revue d'Intelligence Artificielle*, 35(1): 63-70. <https://doi.org/10.18280/ria.350107>
- [4] Ahmed, A., Hasan, T., Abdullatif, F.A., Mustafa, S.T., Rahim, M.S.M. (2019). A digital signature system based on real time face recognition. In 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), pp. 298-302. <https://doi.org/10.1109/ICSEngT.2019.8906410>
- [5] Nainan, S., Kulkarni, V. (2019). Synergy in voice and lip movement for automatic person recognition. *IEIE Transactions on Smart Processing & Computing*, 8(4): 279-289. <https://doi.org/10.5573/IEIESPC.2019.8.4.279>
- [6] Cao, K., Jain, A.K. (2018). Automated latent fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4): 788-800. <https://doi.org/10.1109/TPAMI.2018.2818162>
- [7] Deshpande, U.U., Malemath, V.S., Patil, S.M., Chaugule, S. (2021). Latent fingerprint identification system based on a local combination of minutiae feature points. *SN Computer Science*, 2(3): 1-17. <https://doi.org/10.1007/s42979-021-00615-7>
- [8] Kalka, N.D., Beachler, M., Hicklin, R.A. (2020). Lqmetric: A latent fingerprint quality metric for predicting Afis performance and assessing the value of latent fingerprints. *Journal of Forensic Identification*, 70(4): 443-463.
- [9] Cavero, I., Guillon, J.M., Holzgreffe, H.H. (2017). Reminiscing about Jan Evangelista Purkinje: A pioneer of modern experimental physiology. *Advances in Physiology Education*, 41(4): 528-538. <https://doi.org/10.1152/advan.00068.2017>
- [10] Grzybowski, A., Pietrzak, K. (2015). Jan Evangelista Purkynje (1787-1869): First to describe fingerprints. *Clinics in Dermatology*, 33(1): 117-121. <https://doi.org/10.1016/j.clindermatol.2014.07.011>
- [11] Dhaked, D., Yadav, S., Mathuria, M., Agrawal, S. (2019). User identification over digital social network using fingerprint authentication. In *Emerging Trends in Expert Applications and Security*, 11-22. [https://doi.org/10.1007/978-981-13-2285-3\\_2](https://doi.org/10.1007/978-981-13-2285-3_2)
- [12] Win, K.N., Li, K., Chen, J., Viger, P.F., Li, K. (2020). Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Generation Computer Systems*, 110: 758-771. <https://doi.org/10.1016/j.future.2019.10.019>
- [13] Li, Q., Nguyen, V.H., Liu, J., Kim, H. (2017). Multi-feature score fusion for fingerprint recognition based on neighbor minutiae boost. *IEIE Transactions on Smart Processing & Computing*, 6(6): 387-400. <https://doi.org/10.5573/IEIESPC.2017.6.6.387>
- [14] Honig, A., Etter, R., Pepperman, K., Morello, S., Hannigan, R. (2020). Site and age discrimination using trace element fingerprints in the blue mussel, *Mytilus edulis*. *Journal of Experimental Marine Biology and Ecology*, 522: 151249. <https://doi.org/10.1016/j.jembe.2019.151249>
- [15] Zhao, J., Hansen, B.J., Wang, Y., Csepe, T.A., Sul, L.V., Tang, A., Fedorov, V.V. (2017). Three-dimensional integrated functional, structural, and computational mapping to define the structural 'fingerprints' of heart-specific atrial fibrillation drivers in human heart ex vivo. *Journal of the American Heart Association*, 6(8): e005922. <https://doi.org/10.1161/JAHA.117.005922>
- [16] Rim, B., Kim, J., Hong, M. (2021). Fingerprint classification using deep learning approach. *Multimedia Tools and Applications*, 80(28): 35809-35825. <https://doi.org/10.1007/s11042-020-09314-6>
- [17] Zhao, X., Zhang, Z., Xu, L., Gao, F., Zhao, B., Tian, O.Y., Zhang, Y. (2021). Fingerprint-inspired electronic skin based on triboelectric nanogenerator for fine texture recognition. *Nano Energy*, 85: 106001. <https://doi.org/10.1016/j.nanoen.2021.106001>
- [18] Hu, N., Ma, H., Zhan, T. (2020). Finger vein biometric verification using block multi-scale uniform local binary pattern features and block two-directional two-dimension principal component analysis. *Optik*, 208: 163664. <https://doi.org/10.1016/j.ijleo.2019.163664>
- [19] Ratha, N.K., Chen, S., Jain, A.K. (1995). Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition*, 28(11): 1657-1672. [https://doi.org/10.1016/0031-3203\(95\)00039-3](https://doi.org/10.1016/0031-3203(95)00039-3)