IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 27, No. 6, December, 2022, pp. 923-931
Journal homepage: http://iieta.org/journals/isi

Finger Veins Verification by Exploiting the Deep Learning Technique

Nawar Ali Ibrahime Al-Obaidy[1*], Basil Shukr Mahmood[2], Ahmed Mamoon Fadhil Alkababji[2]

[1] Technical Engineering College of Mosul, Northern Technical University, Mosul 41002, Iraq
[2] Engineering College, Mosul University, Mosul 41002, Iraq

Corresponding Author Email: nawar.ali@ntu.edu.iq

https://doi.org/10.18280/isi.270608

Received: 1 September 2022
Accepted: 8 November 2022

Keywords:
recognition, finger vein, fusion, deep learning

ABSTRACT

Finger vein verification has recently gained the attention of many researchers as one of the most interesting biometrics. This paper proposes a deep learning model called the Deep Fingers Vein Learning (DFVL). to improve recognition accuracy by training a Convolutional Neural Network. The final model consists of the following layers: three convolutional & ReLU, pooling, fully connected, soft-max and classification. All this after the hand image goes through the basic stages of determining the region of interest by operations within the preprocessing. The effect of changing parameter values was examined, analyzed, and discussed. The best accuracy results recorded by tuning the network parameter is 81.7%. This percentage was increased to 89% after using the five-finger fusion (Correct match of three or more fingers).

1. INTRODUCTION

Recently, the use of biometric technologies has spread as a suitable option for many security, health, and other applications. These techniques depend on capturing and analyzing human behavioral and physiological characteristics and extracting features from them to achieve person identification. Relying on biometrics has become a powerful alternative to using passwords, plastic and smart cards, keys, personal identification numbers, and other things that need to be remembered or avoided from being lost or stolen.

Finger biometrics are among the oldest types of biometrics used for identification. One of its types has been used and is still used in paper and electronic applications, which is the fingerprint [1, 2]. In addition to many other finger biometrics such as the finger inner knuckle [3, 4], finger outer knuckles [5, 6], Finger Geometry [7, 8], Finger Texture (FT) [9-11], and the finger vein (FV) [12, 13] which is discussed in this work. Moreover, many studies have applications for the fusion of more than one biometric such as [14, 15].

As a new physiological biometric, Finger vein recognition is one of the methods for personal recognition. It has advantages, including (1) It cannot be imitated or stolen because it is under the skin. (2) The body must be alive. (3) It has unique patterns for each person, even among twins. (4) It cannot be repeated because it does not leave any trace during the authentication process. (5) During puberty, it remains relatively stable, so it will not need to be re-enrollment. (6) It is not affected by skin roughness or cracks, or changes in texture due to diseases such as diabetes or old age.

Figure 1 shows the simple stages of the biometric system of finger vein authentication. The first stage is enrolment, which consists of capturing and storing an image of the finger, or the whole hand. Implementing one of the two methods used for capturing vein pattern images, Light Reflection or Light Transmission. Implicitly, next would be the segmentation to crop the finger image. Second, as preprocessing, the region of interest (ROI) is cut from the image, which from the vain pattern is extracted. Finally, represents the resulting image as binary information.

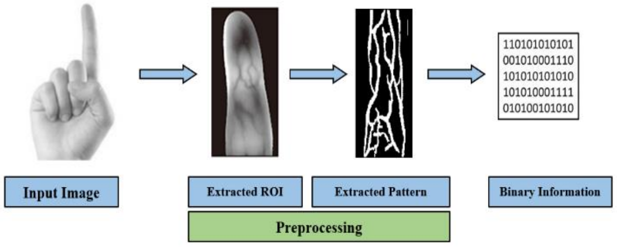


Figure 1. The finger vein authentication

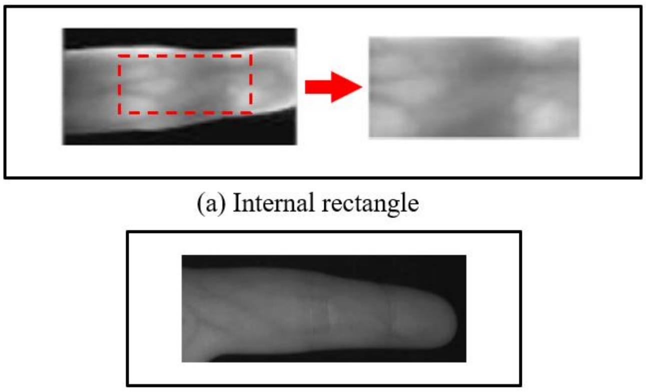
Table 1 shows the distinction of the finger vein from some other types of biometric features in the most important determinants of choosing the appropriate biometrics.

Table 1. Comparison among some biometric modalities [16]

Biometric	Long-term stability	Data size	Cost	Accuracy	Security level
Finger vein	High	Med	High	High	High
Finger-print	Low	Small	Low	Med	Low
Face	Low	Large	High	Low	Low
Iris	Med	Large	High	High	Med
Voice	Low	Small	Med	Low	Low
Hand Geometry	Low	Large	High	Low	Low

This study aims to implement personal recognition using the finger vein features for the five fingers. In addition to voting among them to achieve an appropriate balance between accuracy and security level. The main contributions here are:

- Using an efficient deep learning model with a big dataset for training results in a robust recognition system.
- Using the full inner area of the finger avoids neglecting certain areas of the finger that may have important features, as well as the boundaries of the finger that may add distinction to one finger from another based on its geometry. This might occur due to using an internal rectangle in most research to determine the region of interest. Figure 2 below shows what was mentioned in this paragraph.
- Fusion the matching results of the five fingers features to increase accuracy and security level.



(a) Internal rectangle

(b) Full area

Figure 2. Diversity of extracted areas

The remaining sections are distributed as follows: Section 2 states the related works, Section 3 clarifies the design methodology, Section 4 discusses the proposed model tools, Section 5 contain the sample experimental test, and Section 6 provides the conclusions.

2. RELATED WORKS

The finger vein as a biometric modality was mentioned in many works, which studied and analyzed the properties of this topic in various ways and techniques. Among those discussed are:

In 2014, Lu et al. [17] created a new identification system depending on finger veins with two cameras and diverse fusion techniques. The system proposed took the utility of the finger veins' 3-D structure. Compared with the familiar system of the individual camera, the used system could generate more distinctive information. This resulted from fusing information from two simultaneous captured images, enhancing matching accuracy. The database was collected from 109 persons from four different continents. It contained 6,976 images (as 3,488 images of right and left, respectively) from 436 finger images (Four fingers per person). The captured fingers were the index finger and middle finger from each hand [17]. In 2017, Lee et al. [18] introduced a novel method of finger vein authentication for Anti-Spoofing Using a Laser with a 2-axis Scanner. A laser beam was reflected by a micro-mirror to implement a unified raster scan. No contact between the finger and the detection sensor. The obtained transitional images of the vein were compared with the other of a LED. On the other hand, depending on fleck images in occlusion and perfusion, the patterns of blood flow were obtained. A Gaussian filter was applied, followed by quantitative analysis. This work divided the blood-flow peak curvature by that of the finger vein, to find the ratio of normalized curvature [18]. In 2018, Carrera et al. [19] implemented a finger vein system utilizing textural features. Textural features were presented in this work as significant characteristics that could be utilized in the biometric systems of finger-vein. Gray-level concordance matrices were adopted to obtain the Textural features from the coefficients of wavelet detail included in the images of finger veins. A standardized database of finger veins was used to evaluate the performance of this biometric system. The images adopted in this work were obtained from the SDUMLA-HMT database, which contains finger vein images for 106 persons of both genders. This database was collected from vein images acquired from only three fingers of each hand per person during six sessions. In this paper, only index finger images were used for authentication [19]. In 2020, Zhang et al. [20] submitted a new local descriptor depending on the physiological characteristics of a finger-vein. An unsupervised style was implemented to obtain the finger-veins directional characteristic. Next, to extract the orientation and physiological responses, the filter banks of Gabor were created based on the introductory information. The image was divided into overlapping blocks and non-overlapping cells to generate the histogram in the output. This measure enabled the feature to keep up with local fluctuations in images. In this work, various databases were adopted containing six images for some fingers per person. These databases did not provide samples for all five fingers of the hand. For example, finger vein samples in the first database were obtained from the index and middle fingers only of each person's left hand. While in the second, images of three fingers (index, middle, and ring) only of the person's both hands were provided [20]. In 2020, Yang et al. [21] proposed a convolutional neural network to overcome the problems of poor-quality finger vein images caused by variations in vein patterns or the thickness of finger skin. The proposed network performed the role of identification and anti-fraud by implementing an approach of multitasking learning. In this work, a new database containing images of pivotal rotation for fingers depending on scenarios of real life was created. There are shortcomings in the proposed system in terms of recognition accuracy, improvement of image quality, and data measurement. The vein images used in this work were collected from a variety of databases that included a limited number of fingers, some of which included only one finger (index finger), and some of them only took three fingers (index, middle, and ring fingers) [21]. In 2021, Mustafa and Tahir proposed a personal recognition system based on a finger vein. A Binary Pattern was used to extract the features, in addition to a process for aligning and speeding up the system performance. A few of the enrolled images were excluded from the competition to match the distance to speed up the system. The enrolled database does not contain the original images to avoid attacks targeting personal information. The used databases contained six images per person for only three fingers of each hand (index, middle, and ring) [22]. In 2021, Shaaban and Mahdi presented an Enhanced extraction method of ROI using machine learning. The problems of the slant finger image have been overcome by finding the midline of the finger. To locate the proximal interphalangeal joint and identify the height of ROI, a sliding window was used. At last, using the internal tangents of finger edges to obtain the ROI. By comparing the gray values of the rows in the finger images, the researchers found that the position of the proximal joint shows a row with a higher gray value than the rest, including the position of the distal joint. The database used in this paper included pictures of the fingers of a hundred people from various countries, with six fingers for each person [23].

It is worth noting that all the methods discussed in the previous paragraphs are not supporting extracting features from the entire five fingers and even did not consider their full regions of interest (ROIs). In this study, these issues are addressed.

3. DESIGN METHODOLOGY

The major steps of this work are represented as a block diagram with two main portions: Train and test. Figure 3 demonstrates these major steps. Each portion contains a set of sub-stages. Firstly, capturing an image of the hand using a scanner or imaging device, or by fetching it from a previously downloaded file stored inside the computer, as in our research (CASIA database). Secondly, the preprocessing stage which is described in detail in Figure 4. Thirdly, collecting the finger images, where the capturing process of finger images may take a few seconds from image acquisition to processing and then subsequent use by the recognition system. So, it is necessary to collect it for the later stages. Then a DL model to train the network using (2,000) finger images, or test it using (1,000) finger images. The model performs classification tasks from finger images. The Penultimate is the decision module where the plaintiff's identity is verified. Finally, the output block indicates the result.

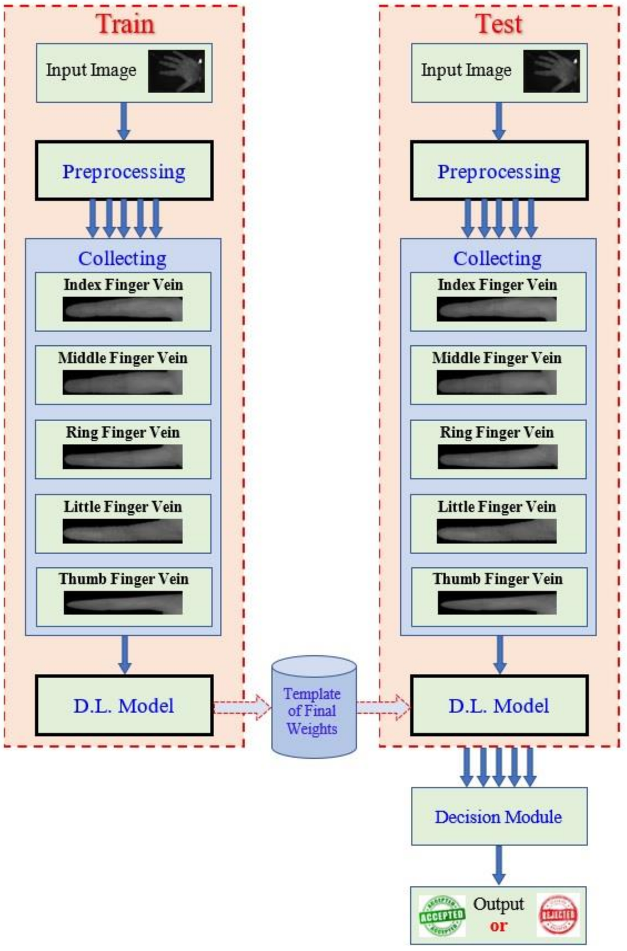


Figure 3. The main suggested steps in this study

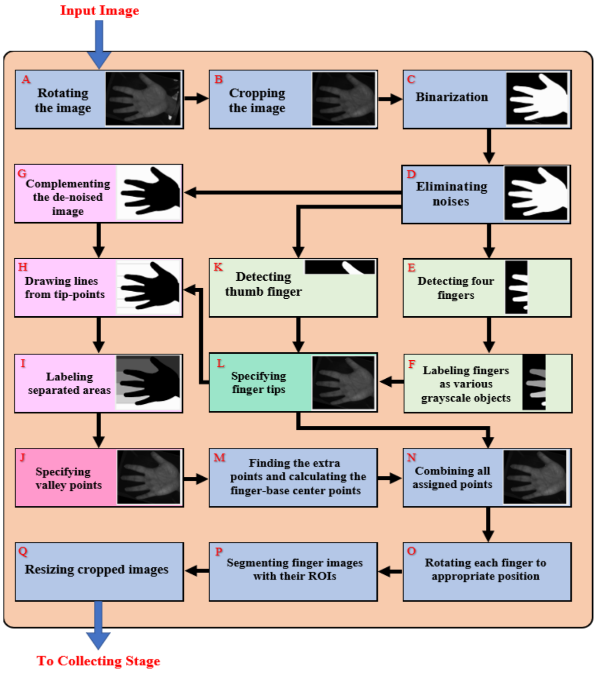

Input Image
A Rotating the image
B Cropping the image
C Binarization
D Eliminating noises
G Complementing the de-noised image
H Drawing lines from tip-points
K Detecting thumb finger
E Detecting four fingers
I Labeling separated areas
L Specifying finger tips
F Labeling fingers as various grayscale objects
J Specifying valley points
M Finding the extra points and calculating the finger-base center points
N Combining all assigned points
Q Resizing cropped images
P Segmenting finger images with their ROIs
O Rotating each finger to appropriate position
To Collecting Stage

Figure 4. The preprocessing block diagram

As shown in Figure 4, the input image is pre-processed as a grayscale image. The preprocessing steps begin with (a) rotating the palm image to the appropriate position and standardizing the size of all fragmented finger images in step (Q) to be properly handled by the software.

The input image enters the preprocessing as a grayscale image. The preprocessing steps begin with (A) rotating the palm image to a suitable position. Followed by (B) cropping the spare part from the hand image that does not contain any required features (The part of the arm attached to the palm). Next (C), a threshold T is used to translate the grayscale image to a binary one depending on Eq. (1) [24-26].

$$BI\ (x, y) = \begin{cases} 1 & \text{if } GI\ (x, y) > T \\ 0 & \text{if } GI\ (x, y) \leq T \end{cases} \qquad (1)$$

where, *BI*(*x*, *y*) Represents the binary image obtained, *GI*(*x*, *y*) denotes the grayscale image, and *T* is the threshold.

To keep the area of the finger objects undiminished, an adaptive value for the threshold is adopted. In step (D) the binary noise that appears with the edges of the hand area is removed. This done by keeping the largest white space that represents the palm and fingers (the hand) and deleting all other white spaces after selecting them and calculating their size. After this step, three actions are taken. Firstly, in (K), is to crop the area of the hand image to (2/7) of the image height to specify the thumb finger as an object. Secondly, in (E) experimentally, the area of the hand is cropped to (3.5/10) of the image width to specify the remaining four fingers as objects. Step (F) which follows the previous actions is to convert the four fingers into various grayscale objects. And by scanning, the tip points of the thumb and the four fingers are identified as the farthest point from the right in each object. In (L) the fingertips are marked on the original image of the hand by merging this image with the black background of the binary image due to the purity of its black color. Thirdly in (G), complementing the de-noised image by drawing lines in (H) based on the fingertips points that are specified. In (I) the aforementioned lines are used to make the areas between the fingers as various grayscale regions to specify the valley points in the next step. In the same way, mentioned previously, these points are marked on the original image of the hand as shown in (J). In step (M), The extra points located at the edge of the hand (near the index finger and little finger) are found. Then the finger-base center points are calculated depending on the coordinates of valley points for each finger.

Later in (N), all points that are found are collected in one picture. The image is rotated in step (O) until the desired finger is in suitable horizontal straightness, depending on the tip and center point of the finger. In step (P), the fingers are segmented with their extracted region of interest. Finally, the size of all fragmented finger images is standardized in step (Q) to be handled correctly by the program.

4. THE PROPOSED MODEL TOOLS

The processing of the hand images produces images of the finger veins five times as much as the original images. This results in a large number of FV images, which require technology with the ability to process data sets and huge databases. Among the most important of these techniques is deep learning (which clarified in [27, 28]). Our Deep Finger Vein Learning (DFVL) model in Figure 5 provides an example of deep learning.

Unlike the neural networks used with FT patterns, multiple feature extraction layers were required. This is because the patterns of finger veins are not so clear in the employed images, and It needs more analysis.

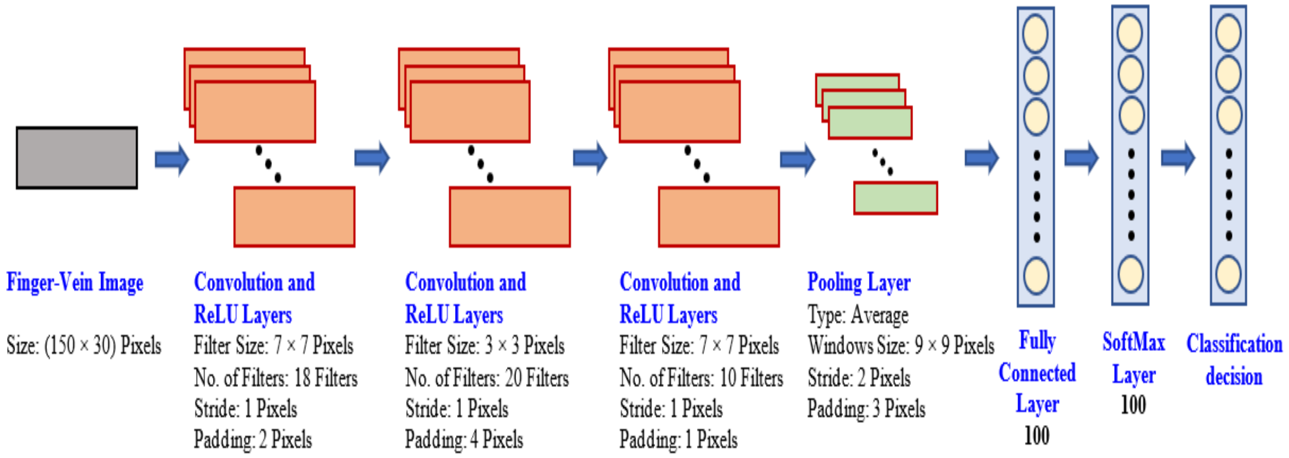


Figure 5. The proposed deep learning model

4.1 Convolution layer

The Convolution Layer is the first layer of a CNN to pull out features from the image. It takes the image and the kernel or filter as inputs of its mathematical formula. Convolution is a kernel or filter of a certain size, and each kernel includes a set of weights. It slides around the image and calculates the weighted sum. At each position, the kernel weights are multiplied by the corresponding image values and added, plus a bias term. Each time this will cause the results to be summed up in a single pixel [29, 30].

4.2 ReLU layer

The Rectified Linear Unit (ReLU) layer, is implemented as a non-linear activation function for the convolutional layer. It removes the negative values from feature maps by setting the output value to zero when the input is less than zero. In doing so, it retains positive values only. The overfitting problems are reduced, by the ability of the convolutional layer to output non-linear feature maps using ReLU [29, 30].

4.3 Pooling layer

The pooling operations reduce the size of feature maps after the convolutional layer. This layer aims to reduce the spatial size of the image to reduce the complexity of the network. To perform the pooling operation, a window of a certain size is selected to slide across the input image. Every time a pooling function is applied to the input elements lying in that window, a single-pixel result. There are several approaches to pooling. The most commonly used approaches are average pooling and max-pooling [31, 32].

4.4 Fully connected layer

The output from the final convolution or pooling layer is fed into this layer. It is used for inference and classification. This layer is called "fully connected" or sometimes "densely connected" because all possible connections from layer to layer exist. This means that each entry of the input vector affects all the outputs of the output vector. But that does not mean that all weights affect all outcomes [30, 31].

4.5 SoftMax layer

The classification probabilities of each input image are calculated using the SoftMax activation function. It is usually used to indicate how close the current input is to a particular class. After applying it each component will be in the interval (0,1), and the sum of all the outputs to be equal to 1 [33].

4.6 Classification layer

The classification layer is performed at the end of the network to make the final decision of recognition or classification. The decision is made in this layer based on the probabilities that are defined in SoftMax for each entry. The rule that is known as the winner-takes-all is implemented in this layer [33-36].

5. SAMPLE EXPERIMENTAL TEST

5.1 Databases descriptions

The database used in this study is one type of the Chinese Academy of Sciences' Institute of Automation Multi-Spectral (CASIA MS) database, which was described in [34]. A type of FV Among the spectral sensors used with wavelength lighting of 940 nm was implemented. The data used contains 600 palm images led to the extraction of 3,000 finger samples.

5.2 Computer descriptions

The DFVL training and testing operations were applied in a DELL Laptop, Intel(R) Core (TM) i7-5600U CPU, 2.60GHz processor speed, 16 GB computer memory, NVIDIA display card, GeForce GTX 1080 graphics processor.

5.3 Practical experiments

A set of values is tested for the convolution layer parameters, such as filter/kernel size, number of filters, stride, and padding. The pooling layer's parameters, such as type, window size, stride, and padding, are also checked with different values. The studied parameters are modified or changed (after 100 training epochs) by considering eight cases, as shown in Table 1. One parameter is checked in each case while leaving the values of the remaining parameters constant. Then the parameter values with the best accuracy result for the current stage are copied to the beginning of the next stage, and the following parameter values are changed and checked in the same way.

Thus, all parameters are checked to obtain the best accuracy result for the eight stages.

Table 1. Checking of DFVL parameters tuning and resultant accuracy observation

Case No.	Max Epochs	Parameters of the convolution layer-1				Parameters of the pooling layer				Accuracy (%)
		Filter size	No. of filters	Stride	Padding	Type	Windows size	Stride	Padding	
1	100	3 × 3	2	1	1	Max	3 × 3	3	1	40.0053
	100	7 × 7	2	1	1	Max	3 × 3	3	1	46.1461
	100	10 × 10	2	1	1	Max	3 × 3	3	1	43.3433
	100	15 × 15	2	1	1	Max	3 × 3	3	1	20.0310
	100	20 × 20	2	1	1	Max	3 × 3	3	1	10.0100
2	100	7 × 7	2	1	1	Max	3 × 3	3	1	46.1461
	100	7 × 7	4	1	1	Max	3 × 3	3	1	54.0541
	100	7 × 7	6	1	1	Max	3 × 3	3	1	51.5516
	100	7 × 7	12	1	1	Max	3 × 3	3	1	55.4555
	100	7 × 7	14	1	1	Max	3 × 3	3	1	55.7558
	100	7 × 7	16	1	1	Max	3 × 3	3	1	56.5566
	100	7 × 7	18	1	1	Max	3 × 3	3	1	60.8609
	100	7 × 7	20	1	1	Max	3 × 3	3	1	59.4595
3	100	7 × 7	18	1	1	Max	3 × 3	3	1	60.8609
	100	7 × 7	18	2	1	Max	3 × 3	3	1	51.0511
	100	7 × 7	18	3	1	Max	3 × 3	3	1	44.0440
	100	7 × 7	18	4	1	Max	3 × 3	3	1	41.2412
	100	7 × 7	18	5	1	Max	3 × 3	3	1	22.8228
4	100	7 × 7	18	1	1	Max	3 × 3	3	1	60.8609
	100	7 × 7	18	1	2	Max	3 × 3	3	1	63.7638
	100	7 × 7	18	1	3	Max	3 × 3	3	1	50.7750
	100	7 × 7	18	1	4	Max	3 × 3	3	1	58.9590
	100	7 × 7	18	1	5	Max	3 × 3	3	1	60.7608
5	100	7 × 7	18	1	2	Max	3 × 3	3	1	63.7638
	100	7 × 7	18	1	2	Ave	3 × 3	3	1	66.0661
6	100	7 × 7	18	1	2	Ave	3 × 3	3	1	66.0661
	100	7 × 7	18	1	2	Ave	5 × 5	3	1	68.9690
	100	7 × 7	18	1	2	Ave	7 × 7	3	1	71.4715
	100	7 × 7	18	1	2	Ave	9 × 9	3	1	72.1722
	100	7 × 7	18	1	2	Ave	11 × 11	3	1	70.6707
7	100	7 × 7	18	1	2	Ave	9 × 9	3	1	72.1722
	100	7 × 7	18	1	2	Ave	9 × 9	4	1	70.2703
	100	7 × 7	18	1	2	Ave	9 × 9	5	1	67.6677
	100	7 × 7	18	1	2	Ave	9 × 9	2	1	73.7738
	100	7 × 7	18	1	2	Ave	9 × 9	1	1	73.3734
8	100	7 × 7	18	1	2	Ave	9 × 9	2	1	73.7738
	100	7 × 7	18	1	2	Ave	9 × 9	2	2	72.7728
	100	7 × 7	18	1	2	Ave	9 × 9	2	3	73.9743
	100	7 × 7	18	1	2	Ave	9 × 9	2	4	71.4715
	100	7 × 7	18	1	2	Ave	9 × 9	2	5	71.5716

At the end of the eight applied cases, the parameters that scored 73.9743% as the highest accuracy are:

- Convolution layer: Filter size=7x7 pixels, number of filters=18, stride=1 pixel, and padding=2 pixels.
- Pooling layer: Type is average, window size=9×9 pixels, stride=2 pixel, and padding=3 pixels.

The accuracy resulting from adding a second convolutional layer and tuning its parameters was monitored in Table 2. This was done similarly to Table 1, with the table abbreviated to show the row of the best accuracy in each case.

It is worth noting that by duplicating the convolution and ReLU layers sequentially and as shown in Table 2, the accuracy remains around the same value of 73.9%. A third convolutional layer has been added, and their parameters have been tested in Table 3 for better accuracy.

It can be seen from Table 3 that the accuracy increased to 80.7% when repeating convolution and ReLU layers three times in succession.

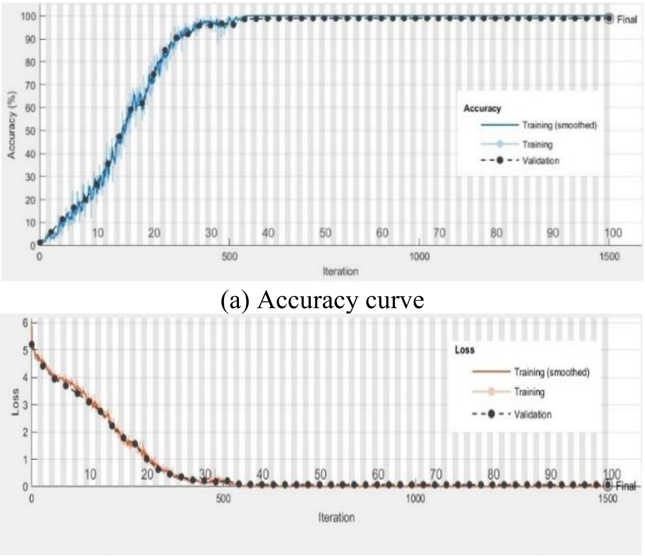
Table 2. Tuning the parameters of the 2nd convolutional layer

Cases	Max Epochs	Parameters of the convolution layer2				Accuracy (%)
		Filter size	No. of filters	Stride	Padding	
1	100	3 × 3	2	1	1	42.6000
2	100	3 × 3	20	1	1	70.9000
3	100	3 × 3	20	1	1	70.9000
4	100	3 × 3	20	1	4	73.9800

Table 3. Tuning the parameters of the 3rd convolutional layer

Cases	Max Epochs	Parameters of the convolution layer3				Accuracy (%)
		Filter size	No. of filters	Stride	Padding	
1	100	7 × 7	2	1	1	52.7000
2	100	7 × 7	10	1	1	80.7000
3	100	7 × 7	10	1	1	80.7000
4	100	7 × 7	10	1	1	81.7000

5.4 Training progress monitoring



(a) Accuracy curve

(b) Loss error curve

Figure 6. Training progress monitoring

The training consisted of using 66% of the finger images for training purposes, which were randomly selected (2,000 images for five finger veins). The following parameters have been used to train the DFVL network: Adaptive moment estimation (Adam) optimizer, momentum value of 0.9, fixed learning rate value of 0.0003, validation frequency value of 30, the mini-batch size value of 64, and maximum epoch of 100. Figure 6 shows two training curves for images of the five-finger veins of the hand. Two curves can be seen, the first in (a) representing the relationship between successful accuracy and frequency. The second in (b) is for small batch losses and their relationship to redundancy. The two curves indicate that the training process is successful.

5.5 DFVL testing

The testing part is the procedure that follows the training and is applied to check the performance of the system and clarify the extent of its success.

Table 4. Testing results of the proposed DL model

Model Details		Accuracy %
Without fusion	Single-Layers of convolution and ReLU	73.97%
	Double-Layers of convolution and ReLU	73.98%
	Triple-Layers of convolution and ReLU	81.70%
Fusing the results of 3 matched fingers		89%

It can be observed from Table 4 that the accuracy of 73.97% was obtained when testing the proposed model with a single convolutional layer, and slightly improved to reach 73.98% when the convolution and ReLU were doubled. In contrast, it jumped to 81.7% when convolution and ReLU layers became doubling three consecutive times. Finally, the summation function was applied to the matching results of the five fingers of each sample. The voting result is equal to one when the sum of matchings is greater than 2. That is, matching with the target has been achieved for three or more fingers out of five fingers.

5.6 Comparisons

The authors suggested the performance result (89%) as a balancing case between the security requirements and the accuracy rate. Table 5 illustrates that an accuracy rate of (99%) was achieved when only one finger was adopted to verify the claim. However, it was not preferable to apply this case, due to the possibility of weakening the security level (the system may be vulnerable to deception). In other words, if the system refuses an acceptable person is better than allowing a fraudulent person to enter.

Table 5. Effect of the No. of matched fingers on the accuracy

No. of matched fingers	1	2	3	4	5
FV Accuracy %	99	97	89	60.5	29

The increase in accuracy for multi-finger recognition comes from the nature of statistics:

$$P_f = \prod_{i=1}^{5} P_i \tag{2}$$

where, P_f is probability of five fingers recognition P_i is the probability (i.e., accuracy) of the linear (i).

$$P_o = 1 - \prod_{i=1}^{5} \overline{P_i} \tag{3}$$

where, P_o is probability of any one of the five fingers P_i is the probability of an unrecognizing finger (i).

Example: Assume the recognition of any finger=99%, then the accuracy of recognizing all fingers=(0.99%)5 ≈ 95%

And the accuracy of recognizing only one out of the five fingers=1–(1-0.99)5=99.99%≈100%

We made another comparison to examine the effect of implementing three training optimizers on the performance of our DFVL model, as shown in Table 6. These optimizers are adaptive moment estimation (Adam), root mean squared propagation (RMSProp), and stochastic gradient descent with momentum (SGDM).

Table 6. Evaluation of various deep learning optimizers

Max Epochs	Optimizer	Achieved Accuracy %
100	RMSProp	71.3000
100	SGDM	76.1000
100	Adam	81.7000

The proposed approaches of the most recent previous networks were simulated and evaluated for our used data set. In these comparisons, four deep learning networks were used: DFCN [37], DDFL [38], RDL [39] and XCM [40]. Table 7 clarifies these comparisons and the surpassing of proposed DFVL on those approaches.

Table 7. Effecting of various approaches on our dataset

Reference	Deep learning model	Accuracy (%)
Ibrahim et al. [37]	Deep Fingerprint Classification Network (DFCN)	70.1000
Al-Nima et al. [38]	Dual Deep Finger Learning (DDFL)	73.8000
Najeeb et al. [39]	Re-enforced Deep Learning (RDL)	39.9000
AL-Hatab et al. [40]	X-axis Classification Model (XCM)	37.3000
Proposed approach	Deep Finger Vein Learning (DFVL)	81.700

5.7 System working mechanizim

The following flow chart illustrates the sequence of operations that the system performs to ensure that the claim person (user) is the same person stored in the database.

The steps of the flowchart in Figure 7 are detailed in the following:

1. Aske user to put his / her hand on the biometric sensor and enter the PIN (Personal Identification Number).
2. Store the images of the user's five-fingers on the vector X.
3. Recall the finger samples of the person concerned from the database and put them in the vector Y.
4. Initialize the voting counter as Vote = 0.
5. Initialize the finger counter as n = 1.
6. Is the claim finger (n) = database finger (n)?
   - If YES increment voting count by 1 and then go to step 7
   - Else go to the step 7.
7. Is the No. of checked fingers less than 5?
   - If YES increment finger count by 1 and then go to step 6.
   - Else go to the step 8.
8. Is the voting count greater than or equal 3?
   - If YES print Accepted Claim.
   - Else print Rejected Claim.

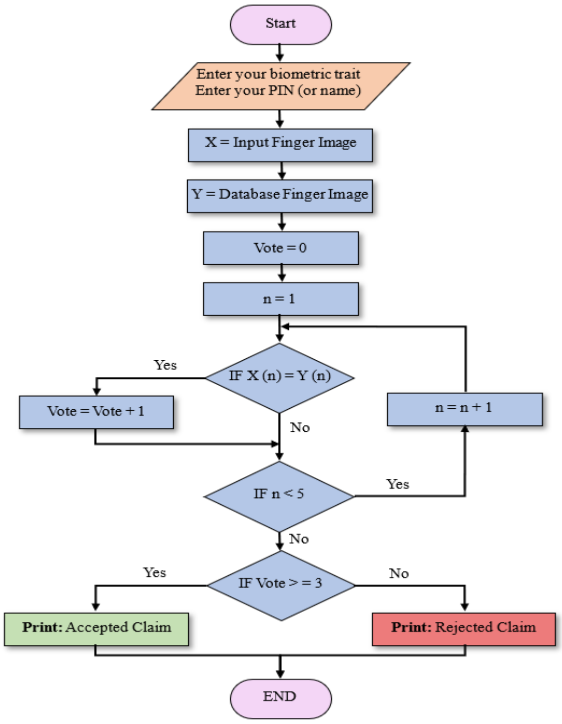


Figure 7. Flowchart of system working mechanizim

6. CONCLUSIONS

In this paper, multiple processes were performed for verifying finger vein images. it consisted of: firstly, image preprocessing operations were implemented to rotating, cropping, binarization, and removing the noise. The fingertips points, valley points, center points of the finger base, and extra points were identified to extract the region of interest to segment the fingers using the entire space. In addition to standardizing the size of the images to be compatible with the program. Secondly, since the properties of the finger veins required further analysis, a DFVL model with a multi-layer was designed and implemented. The network parameters were tested, starting with a single convolutional layer with a pooling layer and monitoring its effect on the accuracy. Then a second convolutional layer was added. Finally, the process was repeated by adding the third convolutional layer. In addition, the FV pattern of the five fingers of each person was stored in the system database. Therefore, the hand's five fingers could contribute together to give a personal recognition decision. Conditional IF rule was adopted to vote on a person's claim, where the claim is accepted if any three fingers (without a specific sequence) match it with the sample stored in the database. In addition, if any finger has an accident, the other fingers will still be valid for implementing the voting for recognition. Therefore, the system could be considered a multi-object FV-based biometric system for the five fingers. Accordingly, the best performance was benchmarked to 89% accuracy, considering security requirements. The hand images used in this research were downloaded from CASIA (spectral 940) databases.

ACKNOWLEDGMENT

This work is supported by the Computer Engineering

Department at the College of Engineering at the University of Mosul in cooperation with Northern Technical University; thanks to the Chinese Academy of Sciences' Institute of Automation (CASIA) for providing the database CASIA-MS-PalmprintV1.

REFERENCES

[1] Peter, M.M.V., Priya, M.V., Petchammal, M.H., Muthukumaran, N. (2018). Finger print based smart voting system'. Asian Journal of Applied Science and Technology, 2(2): 357-361.

[2] Adam, E.E.B. (2021). Evaluation of fingerprint liveness detection by machine learning approach-a systematic view. Journal of ISMAC, 3(1): 16-30.

[3] Anitha, M.L., Rao, K.R. (2015). Extraction of region of interest (ROI) for palm print and inner knuckle print. International Journal of Computer Applications, 124(14).

[4] Anitha, M.L., Rao, K.R. (2016). Fusion of finger inner knuckle print and hand geometry features to enhance the performance of biometric verification system. International Journal of Electrical and Computer Engineering, 10(10): 1351-1356. https://doi.org/10.5281/zenodo.1127210

[5] Zhai, Y., Cao, H., Cao, L., Ma, H., Gan, J., Zeng, J., Wang, J. (2018). A novel finger-knuckle-print recognition based on batch-normalized CNN. In Chinese conference on biometric recognition, pp. 11-21. https://doi.org/10.1007/978-3-319-97909-0_2

[6] Heidari, H., Chalechale, A. (2020). A new biometric identity recognition system based on a combination of superior features in finger knuckle print images. Turkish Journal of Electrical Engineering and Computer Sciences, 28(1): 238-252. https://doi.org/10.3906/elk-1906-12

[7] Shawkat, S.A., Al-badri, K.S.L., Turki, A.I. (2019). The new hand geometry system and automatic identification. Periodicals of Engineering and Natural Sciences (PEN), 7(3): 996-1008.

[8] Angadi, S., Hatture, S. (2018). Hand geometry based user identification using minimal edge connected hand image graph. IET Computer Vision, 12(5): 744-752. https://doi.org/10.1049/iet-cvi.2017.0053

[9] Al-Nima, R.R.O., Dlay, S.S., Woo, W.L., Chambers, J.A. (2015). Human authentication with finger textures based on image feature enhancement. Intelligent Signal Processing Conference, London, UK: The Institution of Engineering and Technology. https://doi.org/10.1049/cp.2015.1784

[10] Omar, R.R., Han, T., Al-Sumaidaee, S.A., Chen, T. (2019). Deep finger texture learning for verifying people. IET Biometrics, 8(1): 40-48. https://doi.org/10.1049/iet-bmt.2018.5066

[11] Al-Nima, R.R., Han, T., Chen, T., Dlay, S., Chambers, J. (2020). Finger texture biometric characteristic: A survey. arXiv preprint arXiv:2006.04193. https://arxiv.org/abs/2006.04193

[12] Yahaya, Y.H., Leng, W.Y., Shamsuddin, S.M. (2021). Finger vein biometric identification using discretization method. In Journal of Physics: Conference Series, 1878(1): 012030. https://doi.org/10.1088/1742-6596/1878/1/012030

[13] Wang, Y., Chen, T. (2021). Finger vein recognition system based on convolutional neural network and android. In Journal of Physics: Conference Series, 2078(1): 012053. https://doi.org/10.1088/1742-6596/2078/1/012053

[14] Perumal, E., Ramachandran, S. (2015). A multimodal biometric system based on palmprint and finger knuckle print recognition methods. International Arab Journal of Information Technology (IAJIT), 12(2): 118-128.

[15] Alay, N., Al-Baity, H.H. (2020). Deep learning approach for multimodal biometric recognition system based on fusion of iris, face, and finger vein traits. Sensors, 20(19): 5523. https://doi.org/10.3390/s20195523

[16] Al-ogaili, H., Shadhar, A.M. (2022). The finger vein recognition using deep learning technique. Wasit Journal of Computer and Mathematics Sciences, 1(2): 1-11.

[17] Lu, Y., Yoon, S., Park, D.S. (2014). Finger vein identification system using two cameras. Electronics Letters, 50(22): 1591-1593. https://doi.org/10.1049/el.2014.1956

[18] Lee, J., Moon, S., Lim, J., Gwak, M.J., Kim, J.G., Chung, E., Lee, J.H. (2017). Imaging of the finger vein and blood flow for anti-spoofing authentication using a laser and a MEMS scanner. Sensors, 17(4): 925. https://doi.org/10.3390/s17040925

[19] Carrera, E.V., Izurieta, S., Carrera, R. (2018). A finger-vein biometric system based on textural features. In International Conference on Information Technology & Systems, pp. 367-375. https://doi.org/10.1007/978-3-319-73450-7_35

[20] Zhang, L., Li, W., Ning, X., Sun, L., Dong, X. (2021). A local descriptor with physiological characteristic for finger vein recognition. In 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, pp. 4873-4878. https://doi.org/10.1109/ICPR48806.2021.9412203

[21] Yang, W., Luo, W., Kang, W., Huang, Z., Wu, Q. (2020). Fvras-net: An embedded finger-vein recognition and antispoofing system using a unified cnn. IEEE Transactions on Instrumentation and Measurement, 69(11): 8690-8701. https://doi.org/10.1109/TIM.2020.3001410

[22] Mustafa, A.A., Tahir, A.A. (2021). A new finger-vein recognition system using the complete local binary pattern and the phase only correlation. International Journal of Advances in Signal and Image Sciences, 7(1): 38-56. https://doi.org/10.29284/ijasis.7.1.2021.38-56

[23] Shaaban, H., Mahdi, H.S. (2021). Enhance region of interest extraction method for finger vein images based on machine learning. Artificial Intelligence & Robotics Development Journal, 1(1); 13-25.

[24] Samavi, S., Kheiri, F., Karimi, N. (2005). Binarization and thinning of fingerprint images by pipelining. In 3rd conference on Machine Vision Image Processing and applications-MVIP, University of Tehran, Iran (Vol. 2).

[25] Mukherjee, A., Kanrar, S. (2011). Enhancement of image resolution by binarization. arXiv preprint arXiv:1111.4800. https://arxiv.org/abs/1111.4800

[26] Al-Nima, R.R.O., Al-Obaidy, N.A., Al-Hbeti, L.A. (2019). Segmenting finger inner surface for the purpose of human recognition. In 2019 2nd International Conference on Engineering Technology and its Applications (IICETA), pp. 105-110. https://doi.org/10.1109/IICETA47481

[27] Çinar, A., Yildirim, M. (2020). Classification of malaria cell images with deep learning architectures. Ingénierie

des Systèmes d'Information, 25(1): 35-39. https://doi.org/10.18280/isi.250105

[28] Fenanir, S., Semchedine, F., Harous, S., Baadache, A. (2020). A semi-supervised deep auto-encoder based intrusion detection for IoT. Ingénierie des Systèmes d'Information, 25(5): 569-577. https://doi.org/10.18280/isi.250503

[29] Sajja, V.R., Kalluri, H.K. (2020). Classification of brain tumors using convolutional neural network over various SVM methods. Ingénierie des Systèmes d'Information, 25(4): 489-495. https://doi.org/10.18280/isi.250412

[30] Abu-Jamie, T.N., Abu-Naser, S.S., Alkahlout, M.A., Aish, M.A. (2022). Six fruits classification using deep learning. International Journal of Academic Information Systems Research (IJAISR), 6(1): 1-8.

[31] Indolia, S., Goswami, A.K., Mishra, S.P., Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. Procedia Computer Science, 132: 679-688. https://doi.org/10.1016/j.procs.2018.05.069

[32] Yani, M. (2019). Application of transfer learning using convolutional neural network method for early detection of terry's nail. In Journal of Physics: Conference Series, 1201(1): 012052. https://doi.org/10.1088/1742-6596/1201/1/012052

[33] Wang, M., Lu, S., Zhu, D., Lin, J., Wang, Z. (2018). A high-speed and low-complexity architecture for softmax function in deep learning. In 2018 IEEE asia pacific conference on circuits and systems (APCCAS), Chengdu, China, pp. 223-226. https://doi.org/10.1109/APCCAS.2018.8605654

[34] Li, X., Chang, D., Ma, Z., Tan, Z. H., Xue, J.H., Cao, J., Guo, J. (2020). Oslnet: Deep small-sample classification with an orthogonal softmax layer. IEEE Transactions on Image Processing, 29: 6482-6495. https://doi.org/10.1109/TIP.2020.2990277

[35] Stephen, O., Sain, M., Maduh, U.J., Jeong, D.U. (2019). An efficient deep learning approach to pneumonia classification in healthcare. Journal of healthcare engineering, 2019: 4180949. https://doi.org/10.1155/2019/4180949

[36] Manne, R., Kantheti, S., Kantheti, S. (2020). Classification of skin cancer using deep learning, convolutional neural networks-opportunities and vulnerabilities-a systematic review. International Journal for Modern Trends in Science and Technology, 6(11): 101-108.

[37] Ibrahim, A.M., Eesee, A.K., Al-Nima, R.R.O. (2021). Deep fingerprint classification network. TELKOMNIKA (Telecommunication Computing Electronics and Control), 19(3): 893-901. http://doi.org/10.12928/telkomnika.v19i3.18771.

[38] Al-Nima, R.R.O., Hasan, S.Q., Esmail, S. (2020). Exploiting the Deep Learning with Fingerphotos to Recognize People. International Journal of Advance Science and Technology, 29(7): 13035-13046.

[39] Najeeb, S.M.M., Al-Nima, R.R.O., Al-Dabag, M.L. (2021). Reinforced deep learning for verifying finger veins. International Journal of Online & Biomedical Engineering, 17(7).

[40] AL-Hatab, M.M., Al-Nima, R.R.O., Marcantoni, I., Porcaro, C., Burattini, L. (2020). Comparison study between three axis views of vision, motor and pre-frontal brain activities. Journal of Critical Reviews, 7(5): 2598-2607.

NOMENCLATURE

DFVL	Deep Fingers Vein Learning
FT	Finger Texture
FV	Finger Vein
ROI	Region of Interest
FVIS	Finger Vein Identification System
3D	Three-Dimensional
MEMS	Micro Electro-Mechanical System
LED	Light-Emitting Diode
HOPGR	Histogram of Oriented Physiological Gabor Responses
FVRAS-Net	Finger-Vein Recognition and AntiSpoofing Network
CNN	Convolutional Neuaral Network
CLBP	Complete Local Binary Pattern
BI	Binary Image
GI	Grayscale Image
T	Threshold
Max	Maximum
Ave	Avearge
CASIA MS	Chinese Academy of Sciences' Institute of Automation Multi-Spectral
Adam	Adaptive Moment Estimation
RMSProp	Root Mean Square Propagation
SGDM	Stochastic Gradient Descent with Momentum
DFCN	Deep Fingerprint Classification Network
DDFL	Dual Deep Finger Learning
RDL	Re-enforced Deep Learning
XCM	X-axis Classification Model