A New Refined-TLBO Aided Bi-Generative Adversarial Network for Finger Vein Recognition

Hossam L. Zayed [1], Heba M. Abdel Hamid [1,]*, Yasser M. Kamal [2], and Abdel Halim A. Zekry [3]

[1] Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Benha, Egypt;
Email: hossam.zayed@bhit.bu.edu.eg (H.L.Z.)
[2] Computer Science Department , Faculty of Computing and Information Technology, Arab Academy for Science Technology and Maritime Transport (AASTMT), Egypt; Email: dr_yaser_omar@yahoo.com (Y.M.K.)
[3] Electronics and Communications Department, Faculty of Engineering, Ain Shams University, Egypt;
Email: aaazekry@hotmail.com (A.H.A.Z.)
*Correspondence: heba.hassan@bhit.bu.edu.eg (H.M.A.H.)

Abstract—**Finger vein recognition is a biometric authentication scheme for analyzing human finger vein patterns. Over the past few decades, Convolutional Neural Networks (CNN) have been widely used for finger vein recognition. However, the conventional issues of CNN are remaining unsolved which are translation invariance and the lack of considerations of position and orientation, thus unable to obtain a large recognition rate. In addition, pre-processing for all kind of finger vein images lead to extra overhead and increases the time for finger vein recognition. In this paper, we proposed a Bi-Generative Adversarial Network (Bi-GAN) with Teaching Learning Based Optimization (TLBO) for finger vein recognition. GAN is an architecture that can use CNNs and are really powerful in learning the underlying data distribution. Further, GAN has been applied previously for this application, but still, it has some serious issues such as hyper-parameters selection, and insufficiency for large feature extraction. Our proposed Bi-GAN with TLBO approach is involved four processes: (1) Image Quality Assessment (IQA), (2) Preprocessing, (3) Feature extraction, (4) Feature matching. We extract texture and soft biometric trait features by Bi-GAN and the parameters are optimized using the TLBO algorithm. In feature matching, we used Canberra Coefficient (CE). Experiments are conducted on the SDUMLA public database that exhibits the efficiency of the proposed Bi-GAN and TLBO in finger vein recognition. The results proved that the proposed approach is superior as analyzed to the CNN, GAN, and Bi-GAN approaches and gives the improvement in the accuracy of recognition.**

Keywords—**finger vein recognition, bi-generative adversarial networks, teaching learning based optimization, image quality assessment, texture, and soft biometric trait features extraction**

I. Introduction

Finger Vein Recognition (FVR) is an arising research topic among researchers in the field of Biometric Authentication [1, 2]. Finger Vein (FV) is a biometric trait that is widely used in several applications with the target of personal identification. FV picture databases are acquired in light fluctuation, uneven illumination, and geometric variation (translational and rotational variation) of the finger [3, 4]. Hence the quality of the individual sample is assessed and improved from the point contrast, edges and corners blur removal, brightness change, irregular shading elimination, etc. There are several processes such as image quality assessment, Region of Interest (ROI) selection or segmentation, normalization, noise reduction, image contrast level enhancement, feature extraction, and feature matching are focused on FVR [5, 6]. From the set of processes, the extraction of features is an important step to be considered for FVR. A finger vein is misshapen. Extracting features for such parts of the human is a difficult undertaking [7]. Due to this effect, texture features are extracted from the regions of the finger vein. Convolutional Neural Network (CNN) is invoked for the extraction of features [8]. In the point of computational resources during feature extraction, CNN is a bottleneck. To overwhelm the issues (high computation and large training samples requirement) of CNN, authors in [9]. A lightweight two-channel Deep Convolutional Neural Network (DCNN)is presented which consists of three convolutional layers, but the design of the topological structure was not effective to recognize finger veins. The natural features of veins are very important such as directionality, continuity, width variability, smoothness, and solidness. These features represent vein patterns heuristic information (Anatomical Structure) since it is affected while classifying finger vein specimens [10]. Furthermore, several processes (thinning, denoising, bifurcation and detection, branch tracking, and morphological dilation) are considered for finger vein model extraction [11]. These processes do not adopt low-quality images. FV codes are employed to improve the quality of the pictures in order to achieve high performance, which enhances the recognition performance only in non-vein patterns [12].

A. Motivation

Several features for example Mean, Energy, Contrast, Inherent Connectivity, and Smoothness are important

Manuscript received October 18, 2022; revised November 28, 2022, accepted December 11, 2022; published May 10, 2023.

doi: 10.12720/jait.14.3.431-443

which counteract noise and boundary irregularities. When using conventional methods of feature extraction for FVR, we must face some issues such as less robustness to noise, misalignment, and too high total computational cost. The initial FV image must be of high quality. Otherwise, it produces a high error rate and less accuracy. For accurate recognition, certain processes are very important: image preprocessing (image quality assessment, ROI selection, enhancement, and normalization), feature extraction, and feature matching for classification. Initial selection of ROI is very important because its boundary is differing as indicated by the size and shape. Similarly, if the ROI is large, then the process of FVR is needed to apply many times. Furthermore, a background image of the FV and foreground information is required for accurate FVR. The extracted features on both the image foreground and background enhance the performance of FVR.

Our main research objectives in this paper are as follows:

(1) To design an accurate and fast FVR which shows good performance for all finger vein images

(2) To find out the most suitable FVR approach which is implemented by the aforementioned motivation for real-time biometric-based authentication applications.

In this research, we presented two novel contributions to meet the above two objectives:

- We first proposed an Image Quality Assessment (IQA) technique that reduces the burden of image preprocessing. The vast majority of the previous works were focused on preprocessing for all types of images of FV. This will increase the time for recognition of images of FV. In the SDUMLA database, a portion of the images is acquired at poor quality due to light fluctuation, and translational and rotational variation in a finger. To evaluate the image quality, four IQA metrics are measured.
- If the quality is not satisfied, then we apply preprocessing methods including Normalization, denoising, and Contrast Enhancement by Pixel Rate Normalization, Adaptive Wavelet Shrinkage with Guided Filter, and Adaptive CLAHE algorithm. Then we apply ROI selection in which Adaptive Thresholding is implemented to separate the FV part.
- Then we suggest a new technique for the extraction of features which is called a Bi-Generative Adversarial Network (Bi-GAN). This technique is comprised of the following: integrating a Teaching Learning-based Optimization (TLBO) algorithm and Bi-GAN for extracting the texture and soft biometric trait features. TLBO is utilized for hyper-parameters selection since in GAN, it is a hard issue.
- We execute the feature matching for the test image and the trained images. The Canberra Coefficient is applied for features matching, whose performance is superior to the conventional distance methods for example Euclidean Distance, Manhattan Distance, Minkowski Distance, etc.

B. Paper Layout

The remaining portions of the paper are structured as follows: In Section II, we give the “Literature Review” on the recent FVR approaches. In Section III, we portray the absolute problems determined for this study. In Section IV, we illustrate the proposed system model and algorithms proposed. In Section V, we present the experiment results and discussion of the proposed and the previous related approaches in the FVR. In Section VI, we provide the paper’s conclusions and future directions.

II. Literature Review

In this part, we first give the traditional FVR schemes listed. Then the conventional machine learning-based FVR schemes are included, and finally, CNN-based FVR schemes are listed.

A. Traditional Finger Vein Recognition Schemes

A novel local discriminative feature learning strategy was proposed in [13]. The target of this research is to learn feature mapping to improve the Discriminative Local Features Ability. To achieve this objective, Multi-directional Pixel Difference Vectors (MDPDV) were suggested, which are utilized to extract the features at a pixel level. Then Discriminative Binary Descriptor (DBD) is introduced for feature fusion. The MDPDV performance is distinctive because of the image quality. The sharpness of FV picture is improved by a modified un-sharp mask and log Gabor filter [14]. A modified repeated line tracking algorithm is for feature extraction. The extracted features consisted of inherent data of the finger vein, which works on the improvement of sharpness. However, the SDUMAL database consists of different images which vary as per the image noise level, low contrast, and irregular shading. For this purpose, sharpness is improved.

Anatomical structure analysis is introduced in [15], which eliminates the artifacts in both high-quality and low-quality image regions. Then, thresholding strategies are proposed in vein pattern labeling, but it doesn’t easy to decide the absolute threshold value for different vein patterns. One of the most important shortcomings in this research direction is securing the heuristic knowledge of the finger vein structure and its characteristics. To isolate the curvature points, some specific feature extraction techniques are required to identify the given test image as either Genuine or Imposter. A user-specific threshold with a tri-branch vein structure was developed in [16]. In this paper, vein pattern anatomical structure is considered to avoid failures in FVR. The vein branches near the bifurcation point are extracted. For matching, an exact threshold is applied. This two-level matching-based FVR is not suited for low-quality FV images (comprised of distorted vein patterns).

B. Conventional Machine Learning Schemes

A portion of the machine learning algorithms (Support Vector Machine (SVM), Fuzzy logic, and Neural Network (NN)) [17−19] has been suggested. In this sub-section, we listed the previous algorithms of machine learning.

The authors of [20] split the image into two layers: the foreground layer and the background layer. The foreground layer consists of texture information and the background layer consists of intensity information. After the layer separation, soft traits and primary features are extracted. In hybrid matching, SVM is utilized for primary biometric characteristic extraction and Manhattan distance is utilized for soft biometric characteristic extraction. The FV image is comprised of side noise, low contrast, and shades. This is brought on by changes in light fluctuation, rotation variation, and finger translation. Hybrid matching by SVM and Manhattan distance is required more computational time, and Manhattan produces a positive result (genuine) and does not indicate trivial deviations from the train and test images. Thus, it leads to large misclassification.

C. CNN-Based FVR Schemes

CNN is one of the best deep-learning algorithms for medical applications [21]. It is applied in [22] for the fundamental goal of FVR in biometric identification. The major purpose of using CNN is to attain a large performance even with the given test image at good quality. The vein of a finger is misshapen so texture attributes are more significant in FVR.

The conventional problems of CNN are position invariance and orientation, which are addressed in [23]. A lightweight two-channel deep CNN is presented, which contains 3-convolutional layers. A mini-ROI is chosen from the test image to achieve a good performance. ROI selection is very simple because it considers the innermost edges as an ROI segmented picture and the topological structure does not provide effective results due to SVM for features classification. SVM uses a black-box model which is not easy to interpret.

Ong and William *et al.* [24] proposed two feature descriptors for texture feature extraction: (i) Local Hybrid Binary Gradient Contour (LHBGC), (ii) Hierarchical Local Binary Pattern (HLBP) extracts both magnitude and sign components of the FV image whereas HLBP used uniform texture patterns (LBP) from the veins of a finger. Furthermore, a multi-instance-based FVR approach is introduced for score-level and feature-level fusions, which rely on certain rules such as Max, Min, and Sum. When concatenating multiple features into a single code vector (0 and 1), it doesn't show an effective solution for FVR because it leads to any important information loss. FV is normalized with 150 × 60 pixels size and is based on the upper and lower FV boundaries. The window size for ROI must be adaptive since the fixed pixel size will lead to data loss and extra overhead in FVR. A Multiscale Uniform Local Binary Pattern block is applied [25] to extract local texture features, followed by using a block-based 2-Directional 2-Dimensional Principal Component Analysis ($(2D)^2$PCA) technique to preserve the local data of the images of finger vein.

III. Problem Description

In this part, we provide a short description of the problems identified in the current works of FVR [26, 27]. CNN model-based FVR was presented in [28], which is pre-trained using ImageNet. The CNN filters and five labels (line, smooth, similar, circle, and noise) are defined in the first convolution layer. In convolution layer 2, edges, corners, and color conjunctions are extracted. In convolution layer 3, more complex variations and texture features are defined. In convolution layer 4, significant variations (class-specific variations) are extracted. In convolutional layer 5, we concentrated on the whole image (extract the significant pose variations). The CNN with CO model's recognition accuracy depends only on the CNN filters. For precise recognition, a large number of FV features are required such as contrast, mean, and energy. Some important preprocessing steps are required such as Denoising, ROI selection, enhancement, and normalization.

GAN was introduced first. FVR in [29], the authors of this paper have addressed the severe issues of CNN for feature representation in FVR. Two problems of CNN are addressed here as follows: (1) In CNN, the fully connected layer is restricted to use the number of FV pictures, and it increases the execution time (extraction of feature and matching), (2) Feature extraction by CNN is not good because it suffers from low-quality images of finger veins. Therefore, U-net-based GAN is introduced for the extraction of features in FVR. In GAN, the hyper-parameter selection is undeniably challenging, and the predefined values do not apply to large-scale databases. Further, a single GAN can extract only the minimum size of features.

Yu and Shanjuan *et al.* [30] proposed a vein-points-based classification for FVR. Vein-points based method is divided into two kinds: (1) Vein-points based, (2) Non-vein points based. The point grouping methods were proposed such as point grouping assisted anatomy structure analysis-based vein extraction and point grouping assisted Gabor filter. These extracts both points (vein and non-vein) and then these points are arranged into several groups and similarity measurement is accomplished between the multiple groups. All points grouping by the feature values is dubious because grouping given the returns of any feature extraction method (e.g., LBP, and CNN) is effective for FVR. The processing time is large since the FV patterns of the whole patterns are extracted and processed for recognition.

Yang and Hui *et al.* [31] proposed an anchor-based manifold binary pattern approach. It is involved with several processes including image acquisition, extraction of ROI, image restoration, image quality enhancement, extraction of features, and matching. To locate the match between the trained and test images, the asymmetric graph is constructed, which can support large-scale datasets. Before Pixel Difference Vector (PDV) extraction, preprocessing is required and also computation of PDV is very time-consuming. The local binary features-based k-means clustering (codebook) is not effective since it requires a large database. For small-scale datasets, this mechanism is not suitable. The problems stated in this segment are resolved by this new refined TLBO-based Bi-Generative Adversarial Network for FVR. A detailed description of the work is given in the following section.

IV. System Model

Our proposed FVR system is involved four main phases: IQA, Preprocessing, feature Extraction, and Matching. Fig. 1 indicates the block diagram for the proposed FVR system.

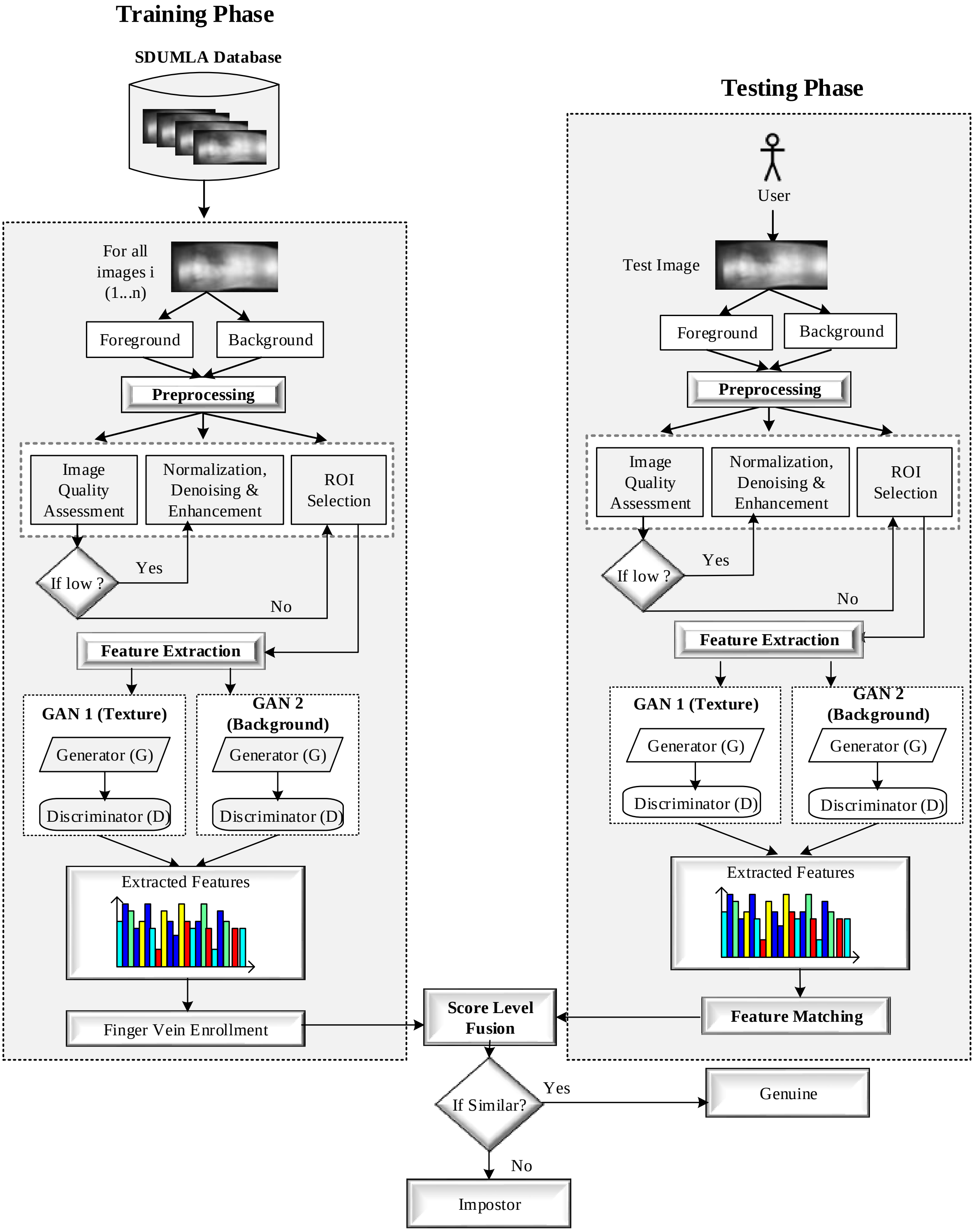


Figure 1. System architecture.

A. System Overview

- **IQA:** The FV image quality is measured because this factor directly affects how well recognition works. When the image's quality is low or too moderate, a step of preprocessing is invoked to resolve the problems.
- **Preprocessing:** This stage of the FVR system is extremely crucial. The major steps consisted is followed normalization, contrast enhancement, denoising, and ROI selection.
- **Feature Extraction:** The identification of vein patterns (attributes) for improving recognition rates and speeding up processing is a crucial step in FVR applications.
- **Feature Matching:** Several distance measures are available for different feature matching and a portion of them are suitable for increasing identification accuracy.

B. IQA

In this step, image quality is assessed before the FVR. In FV image databases, a portion of the pictures is captured at poor quality. This is due to rotational, translation variation of the finger, and light fluctuation. The FV picture is a size of $M \times N$ for quality assessment. To assess the image'squality , four metrics are considered that are Image Brightness I_b, Image Region Contrast I_{rc}, Edge Blur Degree e_{bd} and Noise Degree n_d . We individually compute the aforementioned four metrics for every image of the FV.

- **Image Brightness $\boldsymbol{I_b}$:** we define the i^{th} darker/brighter image as follows:

$$d_i^{(I_b)} = I_o(x,y) + \delta\tau_i \tag{1}$$

$$\delta = \begin{cases} +1\ (bright\ color\ distortion) \\ -1\ (dark\ colordistortion) \end{cases} \tag{2}$$

where τ is the scalar variable, and it is set to $\delta = 10$.

- **Image Region Contrast $\boldsymbol{I_{rc}}$:** we define the i^{th} image region contrast as follows:

$$d_i^{(I_{rc})} = \sum_{k=1}^{n}\left[\left(I_k^{max} - I_k^{min}\right)/\left(I_k^{max} + I_k^{min}\right)\right] \tag{3}$$

where I_k^{max} and I_k^{min} are the maximum and minimum contrast values of the image I.

- **Edge Blur Degree $\boldsymbol{e_{bd}}$:** we define the i^{th} distorted blurred image as follows:

$$d_i^{(e_{bd}:)} = \sum_{(k,I)} I_o(x-k, y-L)\ g(k,L) \tag{4}$$

where g represents the two-dimensional distribution with kernel size μ.

- **Noise Degree $\boldsymbol{n_d}$:** we define the i^{th} image noise degree as follows:

$$d_i^{(n_d:)} = \sum_{(k,I)} I_o\ min\ \sigma_k \tag{5}$$

where $min\ \sigma_k$ represents the kth noise deviation of the image sample block. I_o represents the original image.

The pseudocode for IQA is given below:

Algorithm 1: Steps for IQA

```
(1)  INIT()
(2)  $D_i \leftarrow$ Initialize database images
(3)  For all $D_i$
(4)  Compute $I_b$, $I_{rc}$, $e_{bd}$, $n_d$ by (1), (3), (4), and (5)
(5)  Compute $Q_s(i)$, $Q_s(i)\{x,y\}$
(6)  If ($Q_s \leq \gamma$)
(7)  Go to preprocessing
(8)  else
(9)  Go to feature extraction
(10) End if
(11) End for
(12) While $T_i \leftarrow$ Initialize test image
(13) Do the steps (4), (5), (6)
(14) If (TRUE)
(15) Go to feature extraction
(16) else
(17) Go to preprocessing
(18) End if
(19) End while
(20) EXIT()
```

Finally, Quality Score Q_s is computed for image i by the sum of four factors.

$$Q_s(i) = I_b + I_{rc} + e_{bd} + n_d \tag{6}$$

$$Q_s(i)\{x,y\} = \begin{cases} d_i^{\mu}\{x,y\}\ (\{x,y\}\epsilon\ \omega) \\ I_o\{x,y\}\ (otherwise) \end{cases} \tag{7}$$

$$\mu = \begin{cases} \boldsymbol{I_b} \in (\mathbf{0},\mathbf{1}) \\ \boldsymbol{I_{rc}} \in (\mathbf{0},\mathbf{1}) \\ \boldsymbol{e_{bd}} \in (\mathbf{0},\mathbf{1}) \\ \boldsymbol{n_d} \in (\mathbf{0},\mathbf{1}) \end{cases} \tag{8}$$

If Q_s is less than the threshold γ, image enhancement operation (preprocessing steps) is required. If it surpasses the threshold level, the output of this step is forwarded to the next step, i.e., normalization, denoising, and contrast enhancement. Before IQA, we separate the image into two layers such as foreground and background. We used Image Layer Separation (ILS) method [20] that easily separates the foreground layer and background layer, which are also referred to as smooth and high gradient layers, respectively. In the foreground layer, texture contents are presented, whereas the background layer consists of the distribution of intensity.

C. Preprocessing

It is an initial step of FVR, and it is necessary to get an accurate picture of the FV. Before preprocessing test image is separated into two layers (Foreground Layer and Background Layer). Fore ground layer is made up of texture information and the background layer consists of the intensity distribution. Three processes are involved that are follows in this step:

1) Normalization

The test picture is normalized with a change in the scope of pixel values. The normalization procedure is crucial since some pixel values had a large range and because of this, they dominated others. The desired image intensity range is 0–255 ($Min = 0$ and $Max = 255$). In this paper, we proposed pixel-based image normalization (RGB image) as follows:

$$I_o = (I - Min) \frac{255}{Max - Min} \tag{9}$$

When the image consists of M rows and N columns, normalization runs for M rows and N columns.

2) Denoising

In this noise, removal is executed. Noises in vein images are eliminated using Hybrid Denoising Filters: Adaptive Wavelet Shrinkage with Guided Filter. Two algorithms are combined in this hybrid denoised filter: adaptive wavelet transform shrinkage thresholding and guided filter. Firstly, we apply adaptive wavelet transform shrinkage thresholding for denoising and then used a guided filter for further denoising. In wavelet transform, two elements are used such as decomposition and reconstruction. In this step, we execute DWT for decomposing the noisy image into a sequence of images of several spatial resolutions. Let's take the two-dimensional image that has been decomposed into two-degree directions. It results in four different frequency bands:

(1) Low Frequency (LL)
(2) Horizontal High Frequency (LH)
(3) Vertical High Frequency (HL)
(4) Diagonal High Frequency (HH)

Then the LL is decomposed into four different sub-bands such as LL2, HL2, LH2, and HH2 for the 2nd level wavelet decomposition. Over the various frequency sub-bands algorithm, the thresholding shrinkage algorithm is implemented for noise elimination. In the wavelet-transformed image, the structural information is preserved, which is computed by the energy function. To get the denoised image, energy must be low in the local area. Hence the threshold for shrinkage is calculated by the energy function. We consider two thresholding shrinkage components: y, u . Then the appropriate thresholding shrinkage function is calculated by the following:

$$SF_{j,k}^2 = \frac{1}{\omega^2} \sum_{M=-\omega}^{M=\omega} \sum_{N=-\omega}^{N=\omega} d_{M,N}^2 \tag{10}$$

$$\tilde{d}_{j,k} = \begin{cases} d_{j,k} \left(1 - y \times \frac{\lambda^2}{SF_{j,k}^2}\right), if \ SF_{j,k}^2 \geq y \times \lambda^2 \\ 0 \qquad\qquad\qquad , \ \text{Else} \end{cases} \tag{11}$$

where $SF_{j,k}^2$ represents the thresholding shrinkage function which is implemented on the local region and the window size is $\omega \times \omega = 5$ to compute the local window of $SF_{j,k}^2$. $d_{j,k}$ represents the 2nd wavelet coefficients boundary function, and the y, u are the constant values by 0.1 and 0.3

$$C^2 = \left[\frac{Median\,(|Y_{i,j}|)}{0.6745}\right] \quad [Y \in HH] \tag{12}$$

where $\lambda^2 = (4C^2 \log \omega)$, and C^2 represents the noise invariance for the central pixel of the local window. After we constructed and denoised the wavelet domain, we define the Guided Filter to further enhance the denoised image and it is utilized to preserve the image edges. It is an absolute and linear noise filtering technique that encompasses the spatial data from the denoised image. It has an $O(N)$time complexity and may be implemented for both RGB and gray images. In a denoised image, the guided filter is applied on the individual R, G, and B channels, which is structured as follows:

$$S_i = x_k^R I_i + z_k, \qquad \forall i \in W_k \tag{13}$$

where I is the image consists of color vectors 3×1, x_k represents the coefficients 3×1, and S_i, z_k are the scalar functions. Then the guided filter is implemented on the RGB image as follows:

$$x_k = \left(\sum\nolimits_k + \varepsilon IM\right)^{-1} \left(\frac{1}{|W|} \sum\nolimits_{i \in W_k} I_i p_i - \beta_k \bar{p}_k\right) \tag{14}$$

where $\sum_k$ 3×3 for the covariance matrix I in W_k and IM represents the identity matrix 3×3. The steps involved in the noise reduction by the hybrid denoised filter are as follows:

Algorithm 2: Steps for hybrid denoised filter

```
(1) Begin
(2) Initialize the input FV image
(3) Convert the input (noisy) image into the Frequency
    Domain using Wavelet Transform
(4) Execute the Adaptive Wavelet Shrinkage
    Thresholding over each sub-band
(5) Apply Inverse Wavelet Transform to get the initial
    denoised image $A$
(6) Utilize a guided filter for the denoised image $A$ to
    obtain the more denoised image $B$
(7) Improve the denoised image $B$
(8) Return the output $B$ (fully denoised)
(9) End
```

3) Contrast enhancement

In this step, we apply an Adaptive CLAHE algorithm for the image contrast. Initially, the denoised image is considered, which is changed into an equally sized rectangular block. On each block, histogram equalization is applied which consisted of three operations including: 1) Histogram Creation; 2) Clipping; 3) Redistribution. CLAHE algorithm is different from the conventional histogram equalization method, and it shows the best result for improving the image contrast level, but it is restricted by the Strong Cast Shadow while we process the dark images. The image luminance is improved when the clip point increases. Furthermore, the uniform blocks required lower clip points and the non-uniform blocks required higher clip points. Image distribution of the finger vein is a non-uniform block, so it requires higher clip points. In addition, in an adaptive CLAHE algorithm, Gamma correction is additionally used to adjust the contrast values dynamically for each block. Despite the use of Gamma correction and dynamic clip point selection, our proposed contrast enhancement technique is called adaptive CLAHE filtering. We describe these two operations in detail:

- **Block-Based Dynamic Clip Point:** We define the clip point as follows:

$$\beta = \frac{m}{n}\left(1 + \frac{\alpha}{100} Max_s\right) \tag{15}$$

where m represents the number of pixels in each block, n represents the dynamic range in this block, Max_s is the maximum slope, and α represents the clip point value. In the equation above, the clip point value α must be adaptive, which must be assigned for homogeneous and texture regions for low and high clip point values, respectively. A dynamic clip point value can be calculated as follows:

$$\beta = \frac{m}{n}\left(1 + X\,\frac{V_{max}}{Y} + \frac{\alpha}{100}\left(\frac{\sigma}{Med+c}\right)\right) \tag{16}$$

where σ represents the block standard deviation, Med is the median value, c represents the small value, and V_{max} represents the maximum value of the block, and Y is the image dynamic range.

- **Dual Gamma Correction:** We define the dual comma correction as follows:

$$T(V) = V_{max}\left(\frac{V}{V_{max}}\right)^{\aleph} \tag{17}$$

where $T(V)$ represents the low-intensity pixel values, and the $\aleph$ represents the pixel value in smaller [32].

4) ROI selection (segmentation)

In this process, ROI is selected because of Adaptive Thresholding, which considers the Orientation Angle of 8-neighbors boundaries such as Upper, Lower, Left, Right, and Diagonal (d_1, d_2, d_3, d_4). The adaptive thresholding method is providing high effectiveness, and robustness against image orientation, translation, scale, scattering, irregular illumination, vein posture, and difficult FV structure. However, FVR depends on the single and constant threshold value is difficult for the foreground and the background layer of every image of the FV.

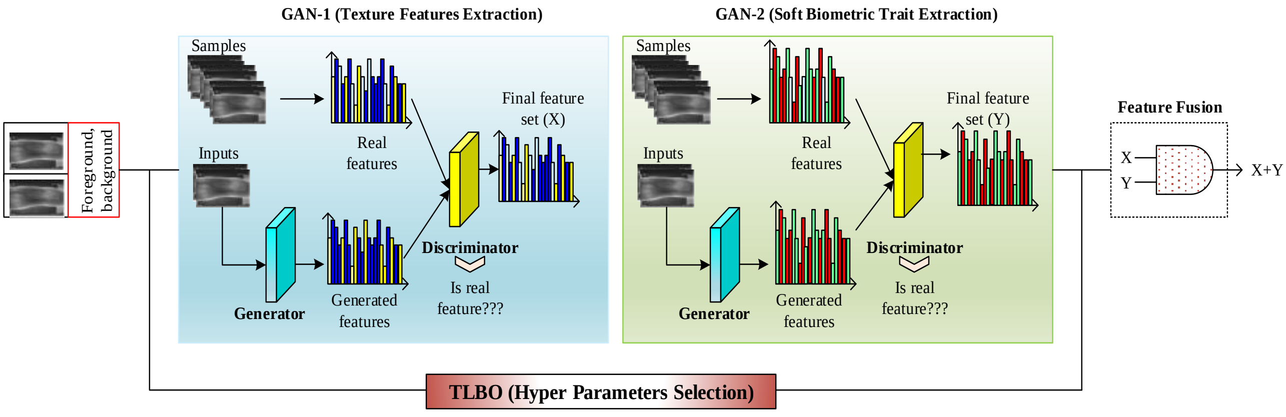


Figure 2. Bi-GAN with TLBO feature extraction.

D. Feature Extraction

For correcting the image orientation, General Adversarial Network (GAN) is proposed.

Since 2018, it is utilized for practical application development in medical image classification, and video gaming applications. In this process, two sets of features are extracted:

(1) Texture features

(2) Soft biometric trait features

In feature extraction, we suggested a Bi-Generative Adversarial Network (Bi-GAN). Bi-GAN alludes to the mix of two GANs. GAN consists of Generator(G) and Discriminator(D). In the first GAN(G1 and D1), we extract the texture attributes and the second GAN(G2 and D2) is used for soft biometric trait features extraction. The extracted attributes are illustrated in Table I.

Other features (geometric and intensity) are also extracted in for accurate identification. In deep learning-based architectures, the choice of parameter selection (Activation Function, Batch Normalization, Drop Out, Max Pooling Number of Convolution Layers, Number of Dense Layers, and Kernel Size) is a difficult task. For parameter selection, we proposed Teaching Learning Based Optimization (TLBO) algorithm. This algorithm outperforms than the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Binary Particle Swarm Optimization (BPSO).

TABLE I. EXTRACTED FEATURES IN THE FINGER VEIN IMAGE

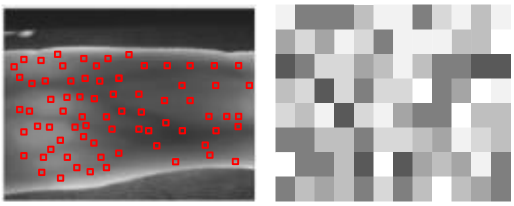
Feature Type	Set of Features
Texture	(1). Angular second moment (2). Contrast (3). Correlation (4). Entropy (5). Energy (6). Uniformity (7). Standard deviation (8). Kurtosis (9). Skewness (10). Coefficient variation (11). Dissimilarity (12). Homogeneity (13). Maxima and Minima
Soft biometric trait	(1). Mean (2). Variance (3). Array of Mean and Variance

Our proposed Bi-GAN with TLBO framework is displayed in Fig. 2 We use a TLBO for hyper-parameters optimization. It can find the optimum set of hyper-parameters for the large dataset. It is executed for making the dynamic solution for the corresponding input. We take the set of hyper-parameters choices illustrated in Table II and the visual and texture features are displayed in Fig. 3. In TLBO, the following actions were undertaken:

(1) The choice of activation function
(2) If normalization is required to use or not?
(3) If max pooling is requiring to use or not?
(4) If dropout is required to use or not?
(5) The number of convolutional layers
(6) The dense layers number
(7) The kernel size

TABLE II. CHOICE OF HYPER-PARAMETERS

Feature Type	Set of Features
Batch Normalization	TRUE, FALSE
Max Pooling	TRUE, FALSE
Number of Convolutional Layers	1, 2, 3
Number of Dense Layers	1, 2, 3
Dropout	TRUE, FALSE
Kernel Size	3,5
Number of Neurons	16, 32, 64, 128, 256, 512, 1024, 4096
Number of Kernels	1, 4, 16, 64, 256



(a) (b)

Figure 3. Feature extraction, (a). Visual features and (b). Texture features.

Algorithm 3: Pseudocode for feature extraction

```
Required: Extract all features in image $i$
for $i \rightarrow f(x,y)$    // $x = num.of\ rows$ , $y = num.of\ columns$
    for all $x = (1-n)$ do
        for all $y = (1-n)$ do
            Compute $Tx, Sbt, O$
        end for
    end for
end for
Return feature vectors set
F $\leftarrow \{f_i(Tx) = (f_1 \dots f_n), f_i(Sbt)(f_1, \dots f_n), f_i(O) = (f_1 \dots f_n)\}$
```

E. Feature Matching

It is the final step of computing the similarity between two features extracted in the testing and training set as illustrated in Fig. 4 where it is evaluated using Canberra Distance. Here we compute the matching score, and the decision reached whether to Genuine or Imposter.

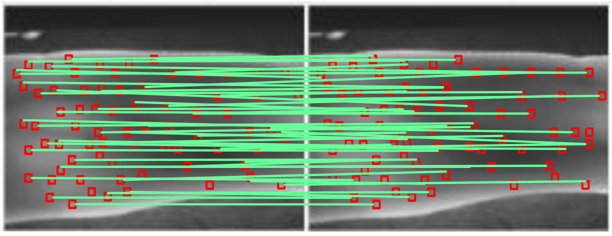


Figure 4. Feature matching.

There are lots of distance measures utilized to evaluate the distance. For example, Euclidean distance, Bhattacharyya distance, chi-Square distance, Manhattan distance, and Minkowski distance. Typically, the FVR framework is desired to attain a high recognition rate at a high speed. For example, the Manhattan distance metric provides a large value for the two similar images which makes the high dissimilarity between the test imagee and training set. Therefore, the feature matching score is computed as follows:

$$M_{sr} = \sum_{i=1}^{n} \frac{|t_i - T_i|}{|t_i| + |T_i|} \tag{18}$$

where n is number of features from $i = 1, 2, \dots,$ n, t_i is the test image i and T_i is the training set features for image i. The M_{sr} value is arranged in ascending order, and the topmost values are indicated high similarity.

V. EXPERIMENTAL RESULTS

In this segment, we describe our experiments for evaluation of the proposed approach and prove the performance concerning robustness, scalability, and efficiency. The hardware and software depictions are illustrated in Table III and the experimental environment and results in MATLAB are illustrated in Fig. 5 where we apply the above operations on FV images of the database and save the extracted features and load them to a graphical user interface(GUI) to be utilized in the classification stage then we enter the user test image to MATLAB using a GUI and apply image separation on it to produce two images(foreground, background layer) as shown in MATLAB then we apply the image quality and preprocessing to give the appeared images in MATLAB window(Noisy image, Denoising image, Cropped, and Normalized image) then extract features to compare it with saved features of database to produce whether the user is an imposter or genuine.

TABLE III. SYSTEM DETAILS

Feature type	Set of features
Processor	Intel (R) Pentium (R) CPU G2030@3.00GHz
RAM	4.00GB
Programming	MATLAB - R2017b

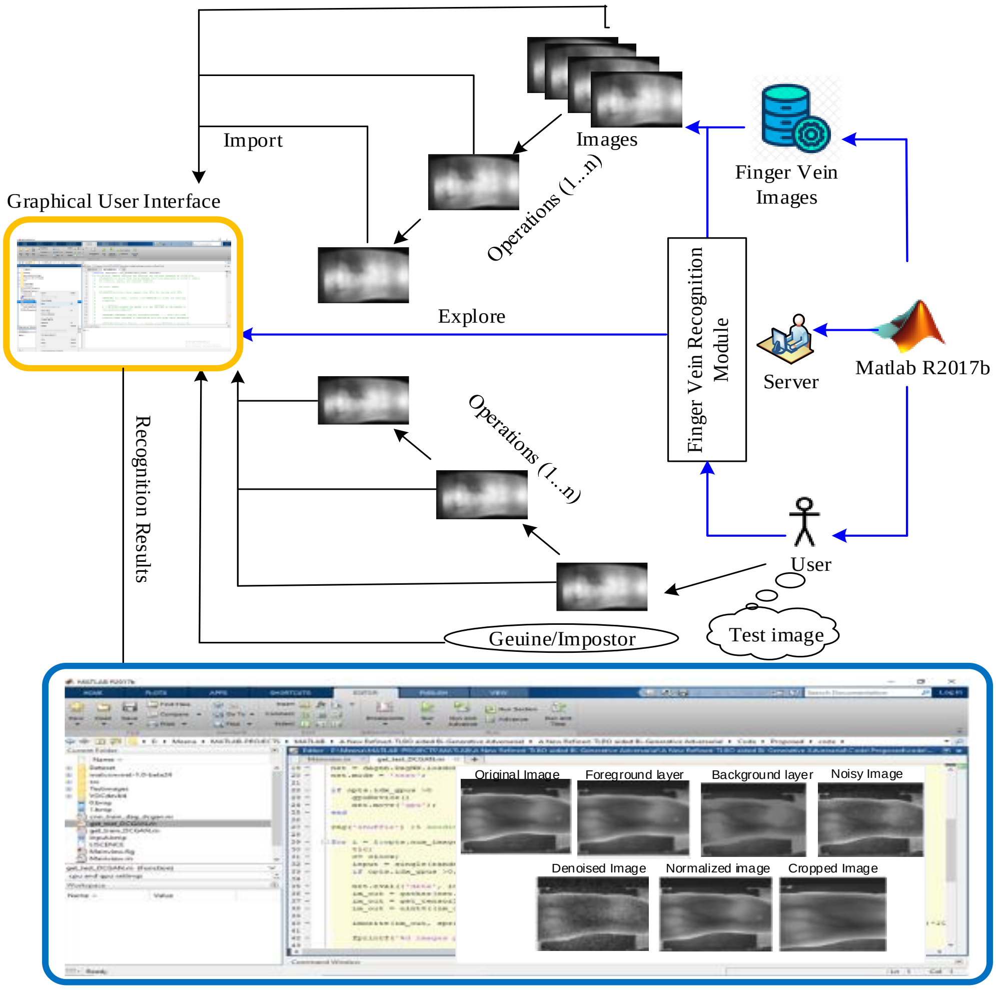


Figure 5. Experimental environment and results in MATLAB.

A. Database Description

In this paper, we investigate the performance of our suggested approach from a publicly accessible dataset called as “SDUMLA Database” [33]. It is collected over 20 days from 34 volunteers. The FV picture-capturing device is manufactured in the Joint Lab (Intelligent Computing, and Intelligent System of Wuhan University, China). 106 subjects of FV images are gained from the device. For every subject, 6 fingers (Index, Middle, and Ring) of both left, and right hands are captured. Therefore, the database is comprised of 3816 images of FV ($106\ subjects\ \times 6\ fingers \times 6\ times$) where Fig. 6 shows a sample of left FV images (Index, Middle, and Ring). The size for each image is 321 × 240 pixels and the acquisition mode for each image is a light transmission mode.

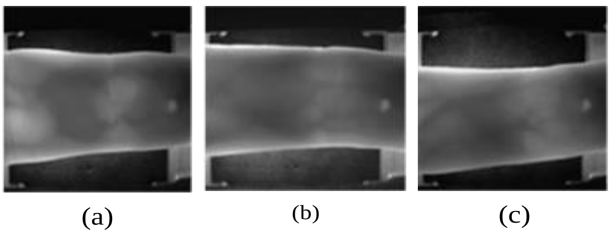


Figure 6. Sample finger (left) Vein images from SDUMLA Database (a). Index, (b). Middle, (c). Ring.

B. Evaluation Criteria

The suggested approach performance is evaluated for the following metrics and their definition as follows:

- Equal Error Rate (EER)

It is commonly utilized as a system evaluation metric for computing the error in recognition. It is computed between the FAR and FRR. The higher error rates indicate the system has obtained poor performance.

- Processing Time (P_t)

It is how much time is taken for processing the recognition of the FV. It is computed by the number of processes that operated to recognize the images of the FV. In this paper, it is defined by the sum of time taken for preprocessing, feature extraction, and matching. It is computed as follows:

$$P_t = p_t + fe_t + fm_t \tag{19}$$

where p_t is preprocessing time, fe_t is the time taken for feature extraction, and fm_t is time for matching.

- False Acceptance Rate (FAR)

It is a unit metric that is used to quantify the FVR system by determining the rate at which genuine images are verified. It is computed as follows:

$$FAR = \frac{\text{Number of false acceptances}}{\text{total number of attempts}} \times 100\% \tag{20}$$

- False Rejection Rate (FRR)

It is a metric that will incorrectly reject the genuine image of a FV and misclassified genuine as an impostor. It is computed as follows:

$$FRR = \frac{\text{Number of false rejections}}{\text{total number of identification attempts}} \times 100\% \quad (21)$$

- Recognition Rate (RR)

It is identified as the total number of correctly classified images of FV divided by the total number of images in the database. It is computed as follows:

$$RR = \frac{\text{Number of correctly classified images}}{\text{Total number of images}} \times 100\% \quad (22)$$

C. Experiments with Different Methods

To evaluate the performance of the suggested approach, some well-known approaches (CNN [28] and GAN [29]) and the Bi-GAN from our suggested approach are considered for experiments here. For fair comparisons, we implemented all the approaches by the SDUMLA database, and previous approaches are described in Table IV.

TABLE IV. MERITS AND LIMITATIONS OF THE PREVIOUS APPROACHES

Recognition approach	Contributions	Merits	Limitations
CNN	1). Extracts very viable features by DCNN 2). In the 1st DCNN layer, line, smooth, similar, circle, and noise features are extracted	1). DCNN is centered around the whole image feature extraction 2). It is supported for large-scale database	1). Insufficient feature extraction 2). Texture features are required for refining recognition accuracy
GAN	1). Predict the pixel's probability of either veins or background 2). Deep feature representation by cycle GAN model	1). High robustness 2). It improves the high finger recognition performance 3). Less processing time compared to CNN 4). Support for large database	1). After feature matching, preprocessing operations are completed, which decreases the recognition accuracy 2). A single GAN is not performed well 3). Optimization of hyper-parameters is necessary

1) Equal Error Rate (EER)

It is the essential metric to compute the recognition. Fig. 7 indicates the performance of EER for a different number of pictures. Among the various approaches, the graph proves that the suggested Bi-GAN with TLBO has obtained a very small EER. In GAN, the discriminator is a component that is used to compare the real and sample images from the dataset. Despite good feature representation, we obtained a good EER in FVR.

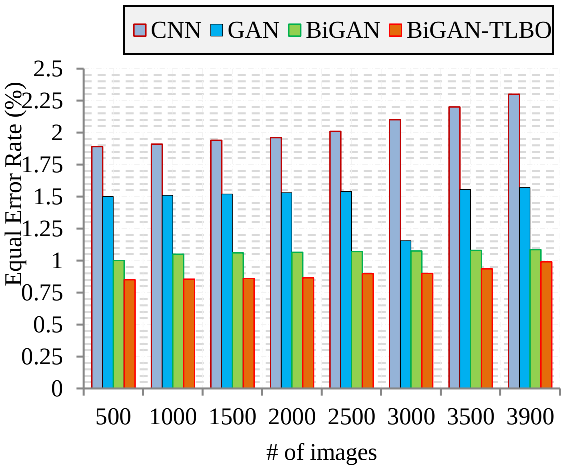


Figure 7. Equal error rate vs. number of images.

We test the EER performance for the whole dataset images. In CNN, 2.03% of EER is attained for all images in the dataset. When compared to CNN, GAN has provided better results, which considers the optimum features for recognition. In CNN, the fully connected layer doesn't scale well for the huge size of the images. It requires further optimization. In GAN, the features from the different images do not suffer to decrease the EER. GAN has also certain drawbacks such as: 1) Hyper-parameters selection; 2) Lack of heuristic knowledge about features; 3) Insufficient for processing the large-scale dataset. Our proposed Bi-GAN-based TLBO mechanism solves these issues. One of the biggest advantages of TLBO is that it improves the proper parameters selection.

2) Processing time

The processing time of the current and the proposed approaches are depicted in Fig. 8. From the graph, it is seen that the CNN has taken significantly more processing time than GAN, Bi-GAN, and Bi-GAN-TLBO.

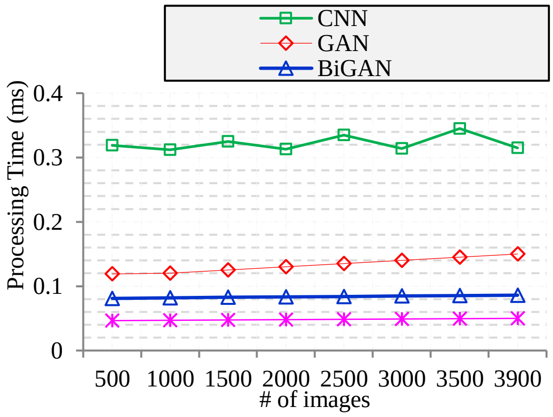


Figure 8. Processing time vs. number of images.

The preprocessing, feature extraction, and feature matching take more time when processing the entire dataset images. In CNN, extraction of features and matching is performed on several convolutional layers. On each layer, different features are extracted, and in GAN extraction of features, and matching is implemented. Then the preprocessing step is taken at the end of feature matching. We extract the two sets of features in parallel and the AND operator is utilized for feature fusion. Our proposed Bi-GAN-TLBO is a promising technique for FVR and it doesn't consume more time.

3) FAR vs. FRR

The curve FAR and FRR in Fig. 9 represents the score or weight to express the result of recognition among the test picture and the training set. For classification, we use the thresholding function in the calculation of the Canberra Coefficient. We take 0.5 as a threshold for the recognition and the picture is higher than the threshold and is considered as the impostor. Otherwise, it is classified as genuine.

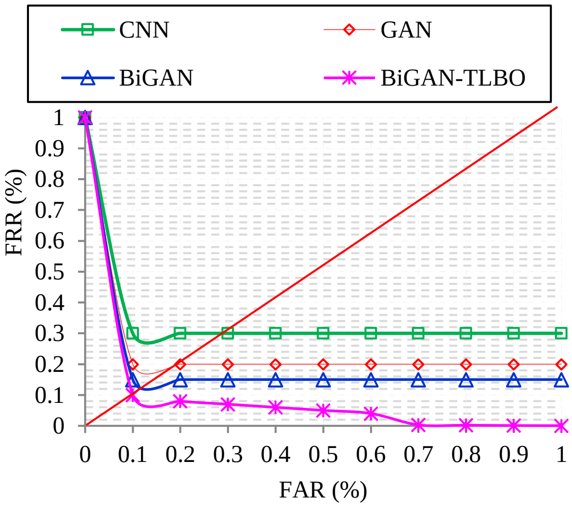


Figure 9. FAR (%) vs. FRR (%).

Despite several reasons, CNN, and GAN achieves poor FAR, and FRR. In some cases, impostor finger vein models satisfied the thresholding function. For that reason, it's a fact that however, the recognition system results in the given finger vein model as "Genuine". To decrease the number of impostor finger vein models that are falsely accepted as genuine, we proposed Bi-GAN-TLBO, which extracts the texture and soft biometric trait features so that depending on the extraction of feature, matching, and recognition method, the performance can differ. Therefore, our proposed technique achieves preferred performance over CNN and GAN.

4) Recognition rate

The recognition accuracy determination is one of the key challenges in FVR. We proposed in this work a robust, efficient, and fast technique that extracts the global and local features from the images of the FV. Fig. 10 shows the recognition rate with the number of images for the proposed approach and other approaches.

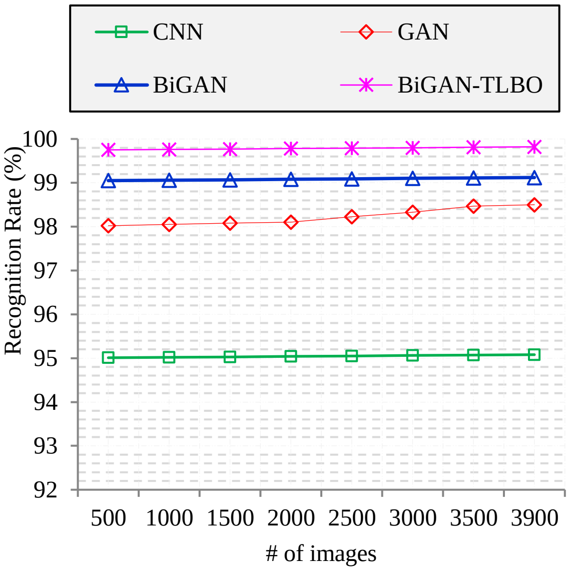


Figure 10. Recognition rate vs. number of images.

When we are looking at the graph in Fig. 10, we conclude that our proposed Bi-GAN-TLBO increases the rate of recognition for varied images. The choice of the similarity method for finding the threshold becomes a serious issue to recognize the different finger veins of the same image and a portion of the matching scores from the impostor are overlapped with genuine images of FV. In our proposed Bi-GAN-TLBO, we center around all aspects for recognizing the pattern of FV according to a person's three fingers (index, middle, and ring). The benefits of proposed strategies are illustrated in Table V and Table VI showing the comparison between the numerical results of previous methods and the proposed method.

TABLE V. SUPPORTED METHODS AND BENEFITS

Processes	Methods	Benefits
IQA	4-IQA metrics	Reduces complexity and processing time
Preprocessing (normalization, denoising, and contrast enhancement)	Pixel Range Normalization, Adaptive Wavelet Shrinkage Thresholding with Guided Filter, and AdaptiveCLAHE	Results in the quality image and thus result from the high recognition rate
Feature Extraction (texture and soft biometric)	Bi-GAN-TLBO	Increases recognition rate and support for large dataset
Feature Matching	Canberra Distance	Reduces EER and FAR

TABLE VI. COMPARISON RESULTS FOR THE PROPOSED APPROACH VS. PREVIOUS APPROACHES

Recognition Approach	EER (%)	Recognition Rate (%)	Processing Time (ms)
CNN	2.14	95.14	Feature extraction – 0.31 Feature matching – 0.05 Sum = 0.36
GAN	1.13	98.45	Feature extraction – 0.0962 Feature matching – 0.0124 Preprocessing time– 0.0089 Sum = 0.1175
Bi-GAN	1.08	99.14	Preprocessing time – 0.0005 Feature extraction – 0.0762 Feature matching – 0.005 Sum = 0.0817
Bi-GAN with TLBO	0.91	99.78	Preprocessing time – 0.0005 Feature extraction – 0.045 Feature matching – 0.001 Sum = 0.0465

As shown in Figs. 7−10, and Table VI, our proposed method is minimizing EER, FAR, and Processing time and maximizes the accuracy of recognition and FRR, which is smaller than the other three methods where our proposed approach has proved that it minimizes the EER up to 0.91%, and augments recognition rate up to 99.78%, respectively. Additionally, decision-making necessitates only 0.0465ms much smaller than the other approaches. This proves that our suggested Bi-GAN-TLBO approach is efficient in maximizing the FV recognition rate while reducing the processing time.

VI. CONCLUSION

In this research, we suggested Bi-GAN-TLBO for the recognition of FV in which four processes are involved such as IQA, preprocessing, feature extraction, and feature matching. We proposed Bi-GAN-TLBO for feature extraction in which features are extracted for recognition by GAN1 and GAN2 respectively. Our proposed Bi-GAN-TLBO approach tolerates the large dataset with the best outcome by TLBO for hyperparameters selection. This approach is tested utilizing MATLAB-R2017b and it is evaluated using the SDUMLA database. We also performed more experiments where we defined the number of convolutional layers of 5 for Bi-GAN TLBO. The performance of our suggested system is highlighted in this study. When compared to CNN, GAN, and Bi-GAN, our suggested approach has proved that it reduces the EER by up to 0.91%, and increases the recognition rate by up to 99.78%, respectively. Additionally, decision-making takes 0.0465 ms, which is substantially less time than CNN, GAN, and Bi-GAN. We developed this approach for implementing real-time medical diagnosis and bio-metric-based security applications. In the future, we have planned to work on the other large-scale datasets and also support other deep learning techniques. Furthermore, other feature-matching techniques are suggested instead of the distance-based feature-matching ones.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Under the supervision of Abdel Halim A. Zekry, Yasser M. Kamal, and Hossam L. Zayed, Heba M. Abdel Hamid programmed and wrote the paper; all authors approved the final version.

REFERENCES

[1] Y. Liu, J. Ling, Z. Liu, J. Shen, and C. Gao, "Finger vein secure biometric template generation based on deep learning," *Soft Computing*, vol. 22, no. 7, pp. 2257–2265, 2017.

[2] D. Ezhilmaran and R. B. Joseph, "Fuzzy based finger vein recognition with rotation invariant feature matching," in *Proc. IOP Conference Series: Materials Science and Engineering*, vol. 263, 2017.

[3] G. Meng, P. Fang, and B. Zhang, "Finger vein recognition based on convolutional neural network," in *Proc. MATEC Web of Conferences*, vol. 128, 2017.

[4] R. W. Ali, J. M. Kassim, and S. N. H. S. Abdullah, "Finger vein recognition using a straight-line approximation based on ensemble learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, pp. 153–159, 2019.

[5] C. H. Hsia, "Improved finger-vein pattern method using wavelet-based for real-time personal identification system," *Journal of Imaging Science and Technology*, vol. 62, no. 3, pp. 304021–304028, 2018.

[6] S. Shazeeda and B. A. Rosdi, "Finger vein recognition using mutual sparse representation classification," *IET Biometrics*, vol. 8, no. 1, pp. 49–58, 2018.

[7] D. T. Nguyen, H. S. Yoon, T. D. Pham, and K. R Park, " Spoof detection for finger-vein recognition system using NIR camera," *Sensors*, vol. 17, no. 10, 2017.

[8] H. G. Hong, M. B. Lee, and K. R. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, 2017.

[9] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Advances in Computer Vision and Pattern Recognition*, vol. 1, pp. 109–132, 2017.

[10] H. Qin and P. Wang, "Finger-vein verification based on LSTM recurrent neural networks," *Applied Sciences*, vol. 9, no. 8, 2019.

[11] R. R. Al-Nima, M. A. Abdullah, M. T. Al-Kaltakchi, S. S. Dlay, W. L. Woo, *et al*, "Finger texture biometric verification exploiting multi-scale sobel angles local binary pattern features and score-based fusion," *Digital Signal Processing*, vol. 70, pp. 178–189, 2017.

[12] M. Rajalakshmi, R. Rengaraj, M. Bharadwaj, A. Kumar, N. N. Raju, *et al.*, "An ensemble-based hand vein pattern authentication system," *CMES — Computer Modeling in Engineering and Sciences*, vol. 114, no. 2, pp. 209–220, 2018.

[13] H. Liu, L. Yang, G. Yang, and Y. Yin, "Discriminative binary descriptor for finger vein recognition," *IEEE Access*, vol. 6, pp. 5795–5804, 2018.

[14] H. Amir and A. R. Dzati, "Sharpness enhancement of finger-vein image based on a modified un-sharp mask with log-gabor filter," *Procedia Computer Science*, vol. 126, pp. 431–440, 2018.

[15] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1892–1905, 2018.

[16] L. Yang, G. Yang, X. Xi, X. Meng, C. Zhang, *et al*, "Tri-branch vein structure assisted finger vein recognition," *IEEE Access*, vol. 5, pp. 21020–21028, 2017.

[17] P. R. B. Joseph and D. Ezhilmaran, "An efficient approach to finger vein pattern extraction using fuzzy rule-based system," *Lecture Notes in Networks and Systems, Springer, Innovations in Computer Science and Engineering*, pp. 435–443, 2019.

[18] X. Qiu, W. Kang, S. Tian, W. Jia, and Z. Huang, "Finger vein presentation attack detection using total variation decomposition," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 465–477, 2018.

[19] M. Wang, D. Tang, and Z. Chen, "Finger vein ROI extraction based on robust edge detection and flexible sliding window," *IJPRAI,* vol. 32, no. 4, pp. 1–19, 2018.

[20] K. Wenxiong, L. Yuting, L. Dejian, L. Dejian, and J. Wei, "From noise to feature: Exploiting intensity distribution as a novel biometric trait for finger vein recognition," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 858–869, 2019.

[21] L. Yang, G. Yang, X. Xi, K. Su, Q. Chen, *et al.*, "Finger vein code: From indexing to matching," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1210–1223, 2019.

[22] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger vein based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2019.

[23] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two streams convolutional network learning," *Neuro Computing*, vol. 290, pp. 100–107, 2018.

[24] T. S. Ong, A. William, T. Connie, and M. G. O. Kah, "Robust hybrid descriptors for multi-instance finger vein recognition," *Multimedia Tools and Applications*, vol. 77, no. 21, 2018.

[25] N. Hu, H. Ma, and T. Zhan,"Finger vein biometric verification using block multi-scale uniform local binary pattern features and block two-directional two-dimension principal component analysis," *Computer Science, Optik*, vol. 208, 2020.

[26] L. Yang, G. Yang, K. Wang, H. Liu, X. Xi, *et al*, "Point grouping method for finger vein recognition," *IEEE Access*, vol. 7, 2019.

[27] H. Liu, G. Yang, L. Yang, K. Su, and Y. Yin, "Anchor based manifold binary pattern for finger vein recognition," *Science China Information Science*, vol. 62, no. 5, pp. 1–16, 2019.

[28] L. Yang, G. Yang, Y. Yin, and X. Xi, "Exploring soft biometric trait with finger vein recognition," *Neurocomputing*, vol. 135, pp. 218–228, 2014.

[29] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 1816–1829, 2015.

[30] L. Yu, X. Shanjuan, and W. Shiqian, "Exploring competitive features using deep convolutional neural network for finger vein recognition," *IEEE Access*, vol. 7, pp. 35113–35123, 2019.

[31] W. Yang, C. Hui, Z. Chen, J-H. Xue, and Q. Liao, "FV-GAN: Finger vein representation using generative adversarial networks," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, 2019.

[32] Y. Chang, C. Jung, P. Ke, H. Song, and J. Hwang, "Automatic contrast-limited adaptive histogram equalization with dual gamma correction," *IEEE Access*, vol. 6, 2018.

[33] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Proc. Chin. Conf. Biometric Recognit*, Berlin, Germany: Springer, 2011, pp. 260–268.