sciendo

TATRA MOUNTAINS
Mathematical Publications
DOI: 10.2478/tmmp-2020-0012
Tatra Mt. Math. Publ. **77** (2020), 109–138

OPENFINGER: TOWARDS A COMBINATION OF DISCRIMINATIVE POWER OF FINGERPRINTS AND FINGER VEIN PATTERNS IN MULTIMODAL BIOMETRIC SYSTEM

Ivan Kovač — Pavol Marák

Slovak University of Technology, Bratislava, SLOVAKIA

ABSTRACT. Multimodal biometric systems are nowadays considered as state of the art subject. Since identity establishment in everyday situations has become very significant and rather difficult, there is a need for reliable means of identification. Multimodal systems establish identity based on more than one biometric trait. Hence one of their most significant advantages is the ability to provide greater recognition accuracy and resistance against the forgery. Many papers have proposed various combinations of biometric traits. However, there is an inferior number of solutions demonstrating the use of fingerprint and finger vein patterns. Our main goal was to contribute to this particular field of biometrics.

In this paper, we propose OpenFinger, an automated solution for identity recognition utilizing fingerprint and finger vein pattern which is robust to finger displacement as well as rotation. Evaluation and experiments were conducted using SDUMLA-HMT multimodal database. Our solution has been implemented using C++ language and is distributed as a collection of Linux shared libraries.

First, fingerprint images are enhanced by means of adaptive filtering where Gabor filter plays the most significant role. On the other hand, finger vein images require the bounding rectangle to be accurately detected in order to focus just on useful biometric pattern. At the extraction stage, Level-2 features are extracted from fingerprints using deep convolutional network using a popular Caffe framework. We employ SIFT and SURF features in case of finger vein patterns. Fingerprint features are matched using closed commercial algorithm developed by Suprema, whereas finger vein features are matched using OpenCV library built-in functions, namely the brute force matcher and the FLANN--based matcher. In case of SIFT features score normalization is conducted by means of double sigmoid, hyperbolic tangens, Z-score and Min-Max functions.

2010 Mathematics Subject Classification: Primary 94A60.
Keywords: biometrics, fingerprint, finger vein, multimodal biometric system.

On the side of finger veins, the best result was obtained by a combination of SIFT features, brute force matcher with scores normalized by hyperbolic tangens method. In the end, fusion of both biometric traits is done on a score level basis. Fusion was done by means of sum and mean methods achieving 2.12 % EER. Complete evaluation is presented in terms of general indicators such as FAR/FRR and ROC.

1. Introduction

Biometric systems are classified as pattern recognition systems which are used to collect and process one or more biological traits leading to identification or verification of one's identity [8]. Furthermore, biometric systems may be categorized into two groups based on the number of traits they use for identity establishment. Namely, they can be unimodal, where such systems utilize only one behavioral or anatomical trait. On the other hand, multimodal systems incorporate several traits for recognition purposes.

Fingerprints are biological traits which are made of pattern of ridges and valleys. For a long time they have been and still are considered as very distinctive and reliable identifiers. They are acknowledged as unique and permanent due to their ability of maintaining uniformity throughout one's lifetime. Fingerprint patterns differentiate even between identical twins [2]. Fingerprint patterns may be categorized into three groups (see Figure 1).

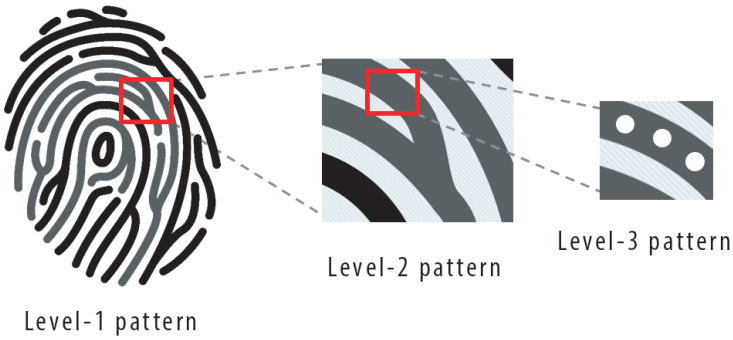

FIGURE 1. Fingerprint features organized in levels [18].

Level − 1 patterns (coarse level) are the most apparent structural patterns defining a global ridge flow. When observed at this level, there are regions where ridges form a specific shape. These regions may be classified into three typologies also known as *loops*, *archs* and *whorls* [8]. Level-1 patterns do not suffice in determining one's identity as they are shared among individuals, however they can considerably speed up the whole authentication process by narrowing the extensive database search if Level-1 patterns do not match.

Level − 2 patterns (local level) represent the local ridge characteristics. Namely such characteristics are denoted as *minutiae points* or *Galton points*. Minutiae indicate locations where ridge lines terminate, merge, split or emerge [9]. Minutiae points are the most prominent features employed by the majority of available commercial or forensic recognition systems. Discriminative strength of minutiae set extracted from a fingerprint has given a rise to minutiae-based matching algorithms where minutiae positions, orientations and sometimes even shape is analyzed to compute a similarity score between two fingerprints [27].

Level − 3 patterns (the finest level) consist of sweat pores, ridge line edges, scars, creases and breaks. Namely, such details can be observed only at very high resolution — at least 1000 ppi. Hence an increased attention is devoted to them because they are deemed rather significant when matching latent fingerprints since normally 20–40 pores are sufficient to recognize a person [27].

Human recognition based on finger veins gained widespread popularity back in 1997–2000 and its commercialization began with Hitachi [28]. Finger vein patterns are acquired in digitized form by means of scanners utilizing near infrared light (NIR), where such illumination is absorbed by oxygenated and de-oxygenated hemoglobin [29].

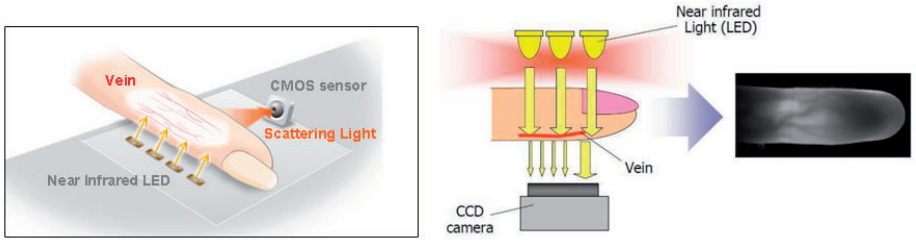

FIGURE 2. Finger vein acquisition principle [19].

Blood vessels located within an individual's finger are intrinsic features used for identity establishment. When compared to fingerprint, finger vein pattern offers several advantages. Due to vein being a part of intrinsic features, it is almost impossible to falsify or replicate it. Note that one of the reasons of its popularity is the user-friendly acquisition. Images can be captured non-invasively using contactless sensors, thus convenience as well as cleanliness is ensured [22]. Besides being very resistant to falsification, finger veins are also considered to be permanent and unique and to offer higher security level than the fingerprint [23].

In this paper we investigate the existing approaches to fingerprint and finger vein multimodal systems and propose our software solution to address this problem. The core of our multimodal automated system named *OpenFinger* is formed by two independent modules for fingerprint and finger vein recognition.

Each module is divided into three smaller sub-modules, namely, each sub-module represents a particular stage in the recognition process. The three stages are: preprocessing, feature extraction and matching. Obtained scores from the aforementioned modules are then fused together within our fusion module.

The paper is organized further as follows. First, in Section 2 we provide a brief analysis of the most important problems in the field. In addition, we focus on some of the existing well-established methods of fingerprint and finger vein recognition as well as some recent published attempts to merge both biometric features into a multimodal system. Our proposed solution is discussed in Section 3 providing a schematic representation and in-depth analysis of internal working of both fingerprint and finger vein pattern recognition modules respectively. Subsequently, we introduce our approach to the fusion of both modules. In the end, we discuss the obtained results as well as contribution of this paper in Section 4.

2. Problem Analysis and Related Work

The main drawbacks and limitations of unimodal systems are lower noise tolerance, accuracy as well as security since they are easier to succumb to spoof attacks [8]. Hence, a robust identification system requires several modalities to address such limitations [1]. This is achieved by fusing the modalities which can be performed at various levels. Namely, fusion may be done at sensor, feature, score and decision level. Each one has its own benefits as well as weaknesses. They are further described in [8] and [25]. There are many situations in everyday life where one biometric feature cannot guarantee a successful authentication, i.e., partial or poor-quality fingerprints. Combining more traits may help to solve this issue by strengthening the match of biometric features to avoid a false rejection and weakening the similarity of the features to avoid a false acceptance. Despite the advantages of multimodal biometrics, we need to ensure the comfort of capturing the biometric trait. There are available sensors that make it possible to capture fingerprint and finger vein images at the same time, i.e., M2-FuseID by MSYS company [17] thus providing the desired trouble-free scanning.

Furthermore, to ensure a reliable user authentication, both fingerprint and finger vein images need to be efficiently enhanced. The problem that arises during fingerprint recognition process is the restoration of image quality in order to reliably extract Level-2 features. Features can be detected in the enhanced images or in the raw images from sensor where they keep their original properties. The main problems with finger vein pattern recognition are the reliable extraction of region of interest, contrast enhancement so that the visibility of veins is at its maximum in order to process the patterns correctly.

In [33], authors argue that fingerprint image enhancement stage is necessary since a reliable minutiae extraction is always heavily affected by low image quality. Gabor filter has proven itself as one of the most suitable techniques to restore the original fingerprint quality as it can adapt its parameters to enhance ridge orientations as well as frequencies. After extensive testing using so called EI (Error Index) indicator they have arrived at the conclusion that correctly tuned Gabor filter rapidly increases the accuracy of minutiae detection. One of its drawbacks is its computational complexity. In our solution we designed an adaptive Gabor filter that runs on GPU what makes its running time very negligible.

Convolutional neural network (CNN) is a type of deep neural network that is primarily used to classify visual data. CNNs can learn spatial hierarchies of features what makes them robust against image alteration due to deformation, translation, scaling or rotation. Some of the most popular CNN architectures are LeNet-5, AlexNet, VGG-16, Inception, ResNet, ResNeXt and DenseNet [6]. More researchers have recently published papers devoted to Level-2 feature extraction from fingerprint images where they used CNNs at the heart of their solutions.

One such solution is called *MinutiaeNet* which is a special CNN for Level-2 feature extraction implemented in Tensorflow framework. MinutiaeNet works in two stages. First stage, so-called CoarseNet, is a deep network which extracts minutiae score map and their orientations. Subsequently, another network called FineNet is employed to refine the candidate minutiae locations and produce final set of detected features. CoarseNet is a residual learning based network and FineNet uses Inception-Resnet network as a core of its architecture. Their solution was evaluated on FVC2004 and NIST SD27 (latent fingerprints) datasets in terms of precision, recall and F1 score. It outperformed well-known algorithms like MINDTCT and VeriFinger.

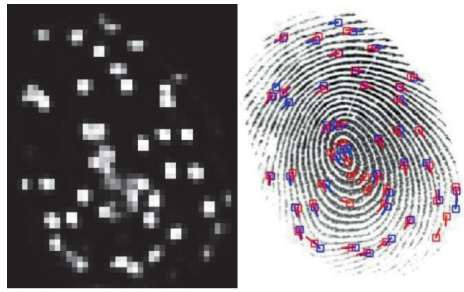

FIGURE 3. Minutiae score map (left) produced by CoarseNet and corresponding final minutiae set (right) produced by FineNet [20].

Another important step in incorporating CNNs into fingerprint recognition was made by authors in [24]. They invented FingerNet, a unified deep network that performs image enhancement and minutiae extraction in a single network. FingerNet normalizes image contrast, estimates orientations, segments ridge pattern and enhances ridge flow by using Gabor filter and finally extracts the minutiae. The network was tested in terms of minutiae extraction accuracy where it achieved minutia location error of 4.4 pixels and angle error of 5.0°. In order to see if fingerprint matching benefits from FingerNet authors compared their solution with well-known algorithms using CMC curve which shows that FingerNet outperforms other solutions.

Ong et al. [21] proposed two-stage multi-instance finger vein recognition system based on minutiae matching by using genetic algorithm and the k-modified Hausdorff distance (k-MHD) measurement. Their recognition rate is as high as 99.7 %. In [11], Ling used deep learning for finger vein recognition. The accuracy of their multi-class classification is 91.67 %, whereas the accuracy of binary classification is 96.27 %. It is worth noting that the low quality dataset may present issue to the neural network. The model might not classify well if the obtained region of interest from preprocessing is of low quality.

In [14] authors introduced a multimodal system based on fusion of finger vein, fingerprint and finger-knuckle-print. The recognition rate of their finger vein unimodal system is 88.68 % when classification is done using SVM. Recognition rate when utilizing KNN classifier is 84.59 %. Their unimodal fingerprint system's recognition rate is at 55.97 % with SVM as classifier and 54.72 % with KNN, respectively. However, they conducted fusion on feature level as well as decision and obtained recognition rate > 94 %. In [7] authors conduct finger vein recognition by means of utilizing convolutional neural network to avoid dependency on image quality. Their recognition rates go beyond 95 %.

In [31] we can learn about a feature-level fusion of fingerprints an finger veins. Distinctive features from both traits are extracted using Gabor filter framework and then transformed into unified feature vector by proposed supervised local-preserving canonical correlation analysis. Unified vectors are classified using nearest neighbourhood method. Recognition accuracy results shows that unimodal fingerprint approach achieved 89.062 % and unimodal finger vein approach was slightly better at 97.187 %. Multimodal approach involved two tests, one with feature-level fusion and another one with score-level fusion. Recognition accuracy of score-level fusion was at 98.75 % being slightly worse than that of feature-level approach with 99.687 %.

3. Proposed Solution

In this section we present our multimodal biometric solution, titled OpenFinger (see Figure 4) which combines fingerprints and finger veins patterns. It is built on top of the *DBOX* open source library introduced by Kádek [13] in his master's thesis. DBOX itself is a unimodal biometric system for fingerprints. For finger vein processing we have developed our own open source library which will be discussed in latter subchapter. Both libraries are developed using technologies such as ArrayFire (GPU computing) [30], OpenCV [5], Caffe (deep neural networks) [10] and Qt.

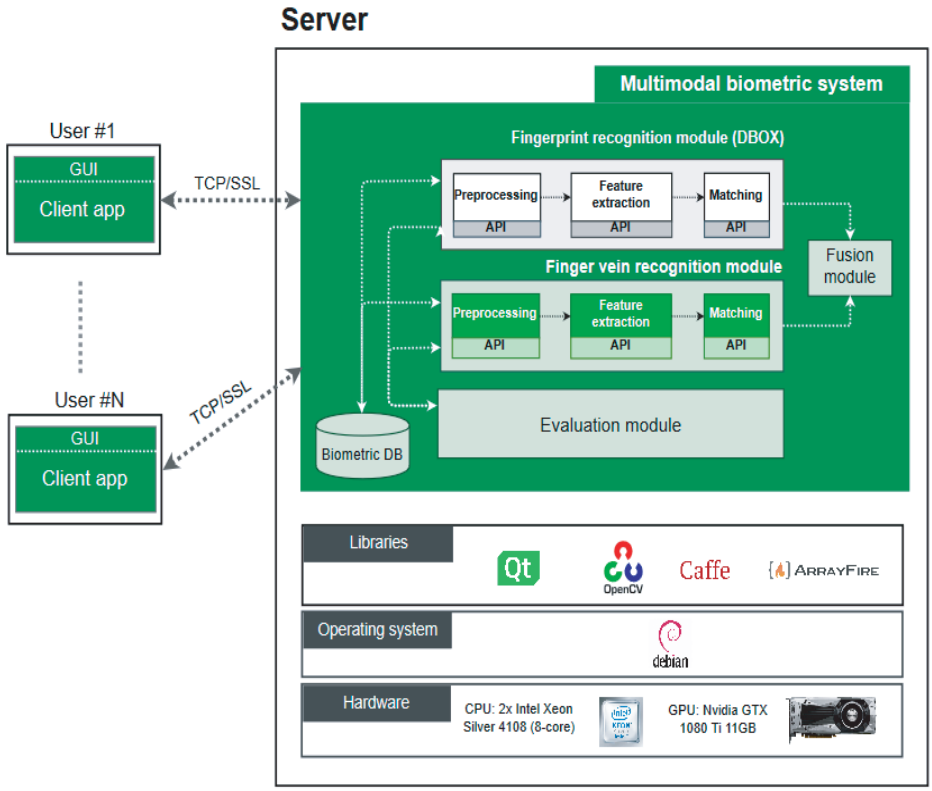

FIGURE 4. OpenFinger system architecture.

The research was conducted on the *SDUMLA-HMT* multimodal database introduced by Shandong University (see [32]). It contains a bundle of biometric data, namely face images, iris images, finger veins, fingerprints and gait videos. These data were acquired from 106 different persons.

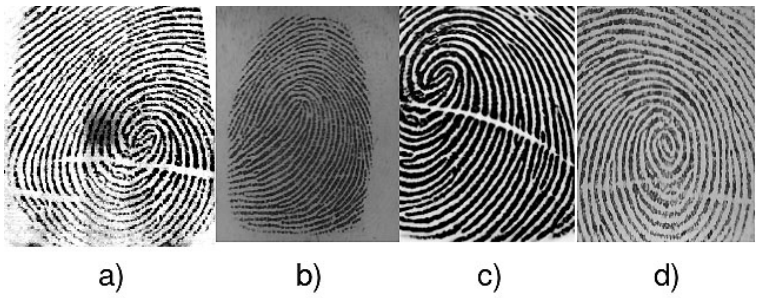

FIGURE 5. SDUMLA-HMT fingerprint images from different sensors: a) URU4000B b) ZY202-B c) FT2BU d) FPR620.

Fingerprint data was acquired from five different sensors, however, note that we have used data from only 4 sensors (see Figure 5). Images from fifth sensor are swipe fingerprints. Images were taken from six fingers, whereas eight impressions were taken from each finger. In total, we have used 20352 images.

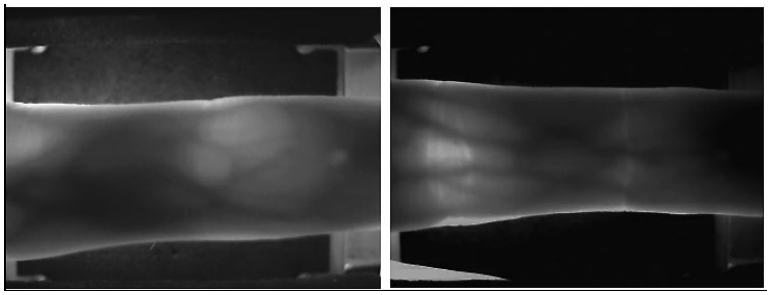

FIGURE 6. SDUMLA-HMT finger vein images.

Finger vein pattern data were acquired from six fingers from each subject, whereas six additional scans of each finger were taken, which gives us 3816 images.

3.1. Fingerprint Recognition

The DBOX solution consists of three libraries where each library represents a particular phase in fingerprint recognition process. First, the image is enhanced in order to ensure reliable extraction of salient features, namely minutiae points. Subsequently, minutiae are extracted using a convolutional neural network and in the end the matching is done using Bozorth3 or Suprema algorithms.

3.1.1. Preprocessing

The first part of the preprocessing stage is segmentation of fingerprint to prevent processing of the irrelevant fingerprint background. However, the contrast is at first enhanced using SUACE (Speeded Up Adaptive Contrast Enhancement) provided by OpenCV. Segmentation is normally done by analyzing the grayscale variance within small image blocks. To distinguish the background, Kádek employed a convolutional neural network for classification, which was trained on blocks of size 19×19. The image was then divided into blocks of $W \times W$, where $W = 8px$. Around each such block, additional larger block of $K \times K$ is cropped, where $W <= K$. In this case $K = 19px$. These larger blocks are then forwarded into the CNN. Classification result for each block is then assigned to the corresponding smaller blocks [13].

Quality map is computed with the help of MINDTCT algorithm developed by NIST. Quality map represents the explicitness of ridge lines in a particular area.

Frequency map is needed since preprocessing phase requires employment of Gabor filters. The local ridge frequency is used to denote the number of ridges per unit length along a hypothetical segment centered at $[x, y]$ and orthogonal to the local ridge orientation $\Theta_{i,j}$ [26]. However, instead of using frequency map, single global frequency value was used for the entire image.

One of the most important steps in preprocessing phase is the local ridge orientation computation. It is inevitable for the correct setting of Gabor filter (see Figure 7). Furthermore, it is needed in determining minutiae orientations in the extraction phase.

Application of Gabor filter is computationally the most demanding step in preprocessing. The filter itself works as a lowpass filter that eliminates high frequencies (e.g., noise) and highlights frequencies corresponding to the density of the ridge lines in the fingerprint image (see Figure b). The definition of an even-symmetric Gabor filter in spatial domain is as follows [13]

$$G(x, y; \Theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\Theta^2}{\sigma_x^2} + \frac{y_\Theta^2}{\sigma_y^2}\right]\right\}\cos\left(2\pi f x_\Theta\right), \quad (1)$$

$$x_\Theta = x\cos\Theta + y\sin\Theta, \quad (2)$$

$$y_\Theta = -x\cos\Theta + y\sin\Theta, \quad (3)$$

where Θ is the filter orientation, f is the cosine wave frequency ($f = \frac{1}{\lambda}$), and σ_x, σ_y being the standard deviations of the Gaussian curve along the x and y axes. Application of filter proceeds by spatially convolving the fingerprint image with the Gabor filter. The orientation image O as well as frequency image F is required during image convolution. Therefore, the image E with applied Gabor

filter G is obtained as follows.

$$E(i,j) \quad = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G\big(u,v;O(i,j),F(i,j)\big)N(i-u,j-v), \tag{4}$$

where N is the normalized fingerprint image and w_x, w_y denote the width and height of Gabor filter mask [26].

DBOX offers possibility of computing Gabor filter application either with the use of CPU or GPU (using ArrayFire framework), whereas if the CPU is used, the image is then split into smaller parts depending on number of computer's threads. Each thread then processes smaller portion of an image.

After application of Gabor filter, the binarization process follows. Every pixel is converted either to white or black. This may be done by using global threshold, however, results have shown that it is better to use adaptive binarization. This is done using OpenCV's adaptive binarization (see Figure 7).

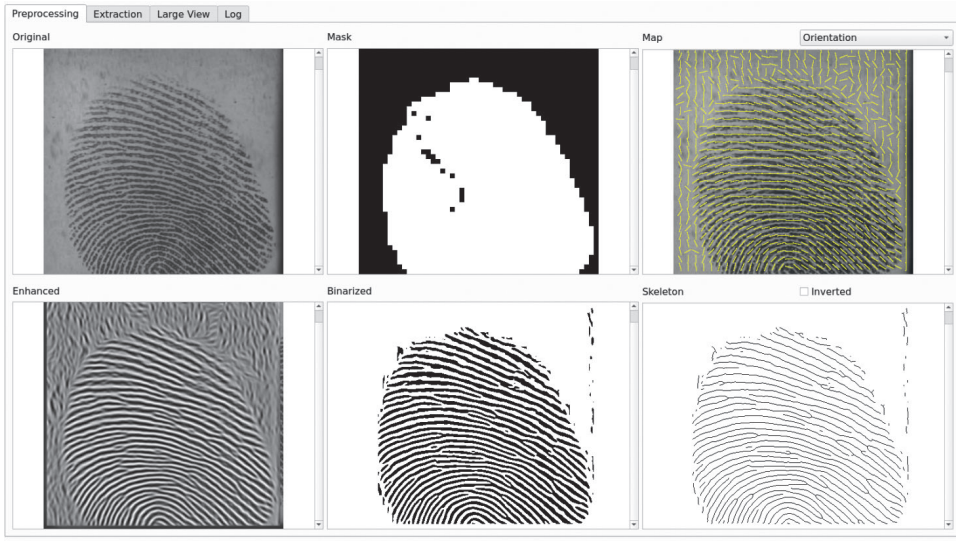

FIGURE 7. Visualization of fingerprint preprocessing stages.

Skeleton creation is the last step in the preprocessing phase where width of ridge lines is reduced to $1px$. Modified Guo-Hall algorithm is implemented in DBOX for this purpose (see Figure 7).

3.1.2. Feature Extraction

After successful preprocessing, the salient features are extracted. This is implemented by using Crossing Number (CN) method. The neighbourhood of each ridge pixel (i.e., black pixel) is scanned in the fingerprint skeleton using a block of size 3×3 (see Figure 8). The CN value is then computed as follows [13].

$$cn(p) = \frac{1}{2} \sum_{i=1}^{8} \left| \text{val}(p_{i \bmod 8}) - \text{val}(p_{i-1}) \right|, \tag{5}$$

where p represents the skeleton pixel, while $p_0, p_1, \ldots p_7$ are its neighbour pixels, respectively. Namely, the value of $cn(p)$ may be one of the following:

- 0 - isolated point,
- 1 - ridge ending point,
- 2 - continuing ridge point,
- 3 - bifurcation,
- > 3 - part of a more complex, undefined minutia point.

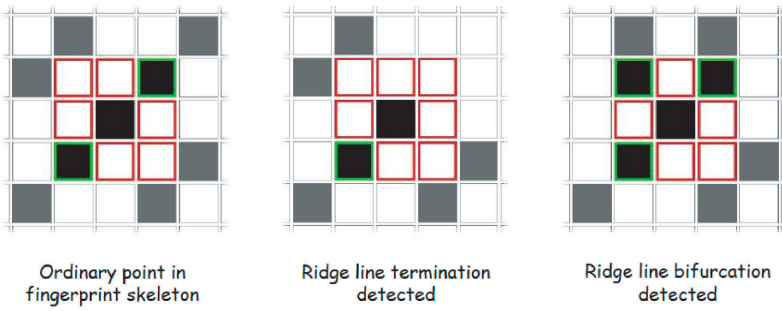

FIGURE 8. Crossing Number method [13].

Furthermore, the CN method is combined with convolutional neural network in order to extract and validate minutiae points, since CN only reveal potential minutiae which require further verification. LeNet based neural network is used to classify image blocks containing minutia candidates. Image blocks are taken from the original fingerprint image from positions where CN method detected ridge line endpoints and bifurcations.

To achieve higher level of robustness, we had decided to train our neural network on images taken from four sensors. When compared to the network utilized in DBOX, its architecture remains unchanged, however it can now successfully classify minutiae on fingerprint images from multiple different sensors.

As shown in Figure 9, we have created an interactive application for image collection for purposes of neural network training. Through such approach we

have managed to acquire roughly 170 000 image data samples out of which 80 % were then used for training. Whereas the remaining 20 % was used for validation. The achieved average accuracy of this network is 97 %.

FIGURE 9. User interface of the application for training data collection depicting minutiae marking.

3.1.3. Feature Matching

Matching is applicable only during identification or verification after the provided fingerprint has been processed. This is the final step within the fingerprint recognition process.

OpenFinger fingerprint matching module offers the use of two algorithms for matching, namely Bozorth 3 and Suprema, whereas Suprema is superior when compared to Bozorth 3 since the latter one requires a large number of matching minutiae in order to consider two fingerprints identical. However, our images varied in quality and sometimes the number of detected minutiae was very small for Bozorth 3. The success of the identification and verification is evaluated according to the preset threshold values. These threshold values are:

- Bozorth 3 - 50 (achieved score higher than 50 signifies a successful match).
- Suprema - 0.15 (score lies in the interval [0,1]).

In our experiments we have used the Suprema algorithm for matching.

3.2. Finger Vein Recognition

Despite many benefits, there exists a number of challenges that need to be overcome. During acquisition, poor illumination or misalignment of the finger position are some of the circumstances that may significantly decrease the recognition rate [22]. Hence a reliable recognition process is needed.

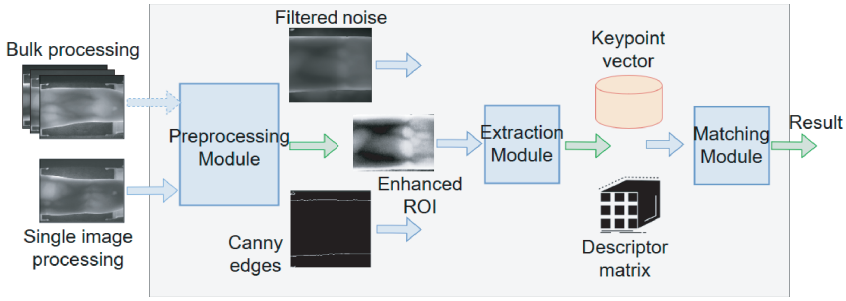

FIGURE 10. OpenFinger finger vein module scheme.

As show in Figure 10, the module is split into three smaller sub-modules (blue colour). Each smaller module represents a stand-alone shared library as well as a particular phase in the recognition process. Furthermore, they are compatible between each other, i.e., the output from one module may be subsequently forwarded as input into the next module. Their communication is based on utilizing the Qt provided signals and slots mechanism.

3.2.1. Preprocessing

Our preprocessing phase consists of the following steps: edge-preserving smoothing using bilateral filter, edge detection by Canny algorithm, contour computation, ROI (region of interest) extraction and finally enhancement of ROI using histogram equalization.

The concept of our preprocessing module is designed in a way which makes it user friendly, meaning it provides an easy-to-use API. As Figure 11 depicts, one might choose between inserting a single finger vein image, an entire bulk, i.e., vector of such images or even a path to a particular image or to a directory containing the images. Furthermore, one can also set additional preprocessing parameters. This allows the user to experiment with the module, thus providing the possibility of achieving better results.

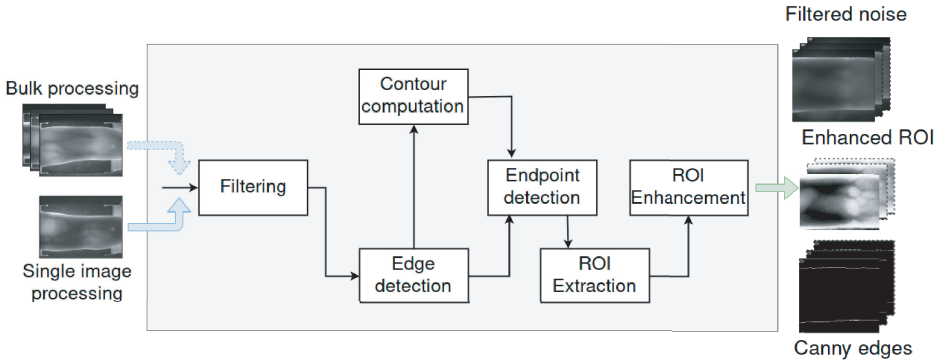

FIGURE 11. OpenFinger finger vein preprocessing module scheme.

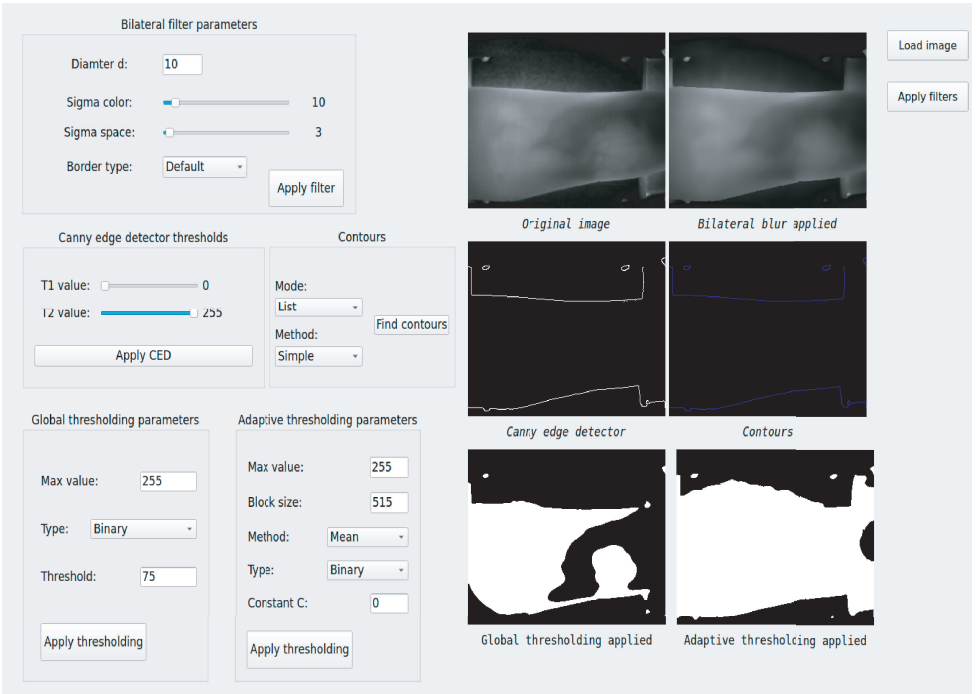

FIGURE 12. Intermediate results of the finger vein preprocessing.

Since finger vein images from SDUMLA-HMT database contain side noise, we slightly crop each image. The original image is of size 240×320 pixels, whereas our cropped image is of size 200×220 pixels. Such resolution was obtained from empirical observation. Furthermore, this significantly improves our endpoint search during ROI extraction.

In order to efficiently detect the finger boundaries, we first employ the bilateral filter, which helps us to remove unwanted noise, while keeping the edges rather sharp. The filter is applied three times in a row since such approach has shown better results when compared to single filter application. We use the OpenCV implementation of the filter.

Bilateral filter uses three parameters, namely diameter of each pixel neighbourhood that is used during filtering, filter sigma in the colour space and filter sigma in the coordinate space. We use value 10 as the diameter and set filter sigma in the colour space to the value of 10 as well, whereas the sigma in the coordinate space is set to 3. Also note that we apply the filter three times in a row (see Figure 12).

One ought to note, why utilization of bilateral filter is to be deemed important. Namely, as shown in Figure 6, finger vein images are mostly of low quality. Certain subset of such images is distinguished with vivid illumination, whereas another subset yields murky and obscure areas. Hence employment of bilateral filter flattens out such irregularities to some extent.

The next step in the preprocessing phase is the employment of Canny edge detector. We use the implementation provided by OpenCV. Since bilateral filter leaves the finger edges sharp, Canny algorithm can reliably detect the finger boundaries.

Prior to utilizing Canny algorithm for distinguishing finger boundaries, we attempted to select the region of interest by means of applying binarization. However, as shown in Figure 12, such approach has proven as unreliable. This happens due to varying levels of illumination in finger vein images.

After we have applied edge smoothing, we employ Canny edge detector (CED) algorithm on our filtered image. The result of CED is a matrix of the same size as our cropped image, however, this matrix contains only values 0 and 255. If the value at (x, y) is 0, then this means that CED did not detect an edge-like pixel at (x, y), hence such pixel is black in the CED matrix. If the value at (x, y) is 255, then this pixel represents edge pixel and such pixel is white, respectively.

OpenCV provides implementation for finding contours, namely it is applied onto Canny matrix and results in a vector of vectors of points, where each vector represents a particular contour, while each point represents an edge pixel in the Canny matrix. Such approach is used to simplify the endpoint search in the ROI extraction step.

FIGURE 13. Neighbourhood scan of potential endpoint.

The most crucial step in the finger vein preprocessing phase is the ROI extraction. First, we find the longest as well as the second longest contour in the contour vector. Presumably, they represent the finger edges. Subsequently, in each of these contours we look for endpoints. The candidate for endpoint is a point situated near the image boundary. These potential endpoints are then filtered by scanning their neighbourhoods. This ought to be done in order to avoid having a random noise pixel as the ROI endpoint. To ensure the candidate is a valid endpoint, i.e., it is a part of the finger edge, we employ an algorithm which scans 50 neighbouring edge pixels. With the help of Canny matrix, such approach ensures a reliable endpoint selection (see Figure 13), as a posteriori results have shown.

After ROI has been extracted, we further submit it to enhancement, namely we conduct histogram equalization (see Figure 14).

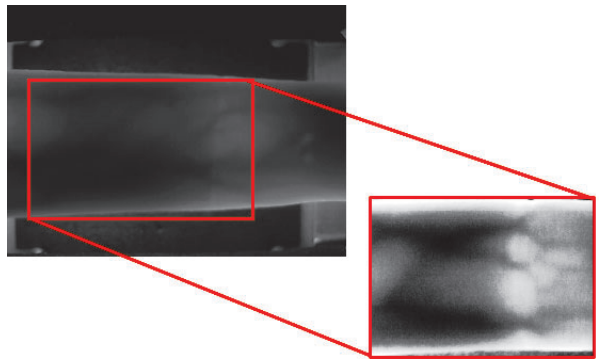

FIGURE 14. Enhanced ROI by means of histogram equalization.

3.2.2. Feature Extraction

Many methods have been proposed on feature extraction in finger vein patterns. However, even though they result in high accuracy, the lack of rotation as well as scale invariance may be considered as drawback. For this purpose, we have decided to use the SIFT [16] and SURF [3] algorithms. Both algorithm implementations are provided by OpenCV.

SIFT performs in five different phases. First, by scanning the entire image, SIFT detects potential points of interest which are invariant to scale and rotation. Subsequently, keypoints are selected based on their stability, namely each candidate location goes through a test of scale and location determination. To each keypoint location, one or more orientations are assigned based on local gradient directions. Around each keypoint, the local gradients are measured at the selected scale and are henceforth transformed into representations, which allow significant degree of local shape distortion as well as change in brightness.

SURF is another approach that deals with detection of scale and rotation-invariant points. To achieve even better performance in computation time and accuracy, their solution is based on Hessian matrix. They rely on determinant of the matrix as the measure for choosing the location and scale.

As shown in Figure 15, we have come across various outcomes when experimenting with these algorithms. The instance depicted in Figure 15 a) represents the outcome with adequately set parameters, obtained a posteriori. On the other hand, the situation shown in Figure 15 b), represents an instance where the edge threshold is set at higher values, which results in more keypoints detected at finger edges. In order to filter out these edge-like features, the edge threshold needs to be set at lower values. When it comes to SIFT, in default, it is set to 10, however such keypoints cannot be considered as a reliable source of information. Thus we have set the edge threshold value at 4. Note that this issue was also addressed in [12]. The situation shown in Figure 15 c) shows how the higher contrast thresholds impacts the number of detected keypoints. When it comes to SURF, we have set the value 170 as the hessian threshold. This means that only features whose hessian value is larger than the threshold are retained. Namely, the larger the value, the less keypoints are obtained. Other parameters are left at default. To ensure higher number of keypoints detected, we use 0.009 as the contrast threshold value. Another important thing to be noted is the drawback of these algorithms, which we have observed during our experiments. Note that the images shown in Figure 15 a) and b) are not the same. Namely, these images are two different scans taken from the same finger. Notice how some of the keypoints detected are not at the same location. Such observation shows how these algorithms are very sensitive to the intra class variations. Nonetheless, this represents an issue that ought to be taken into an account, otherwise matching might result in very few keypoints matched correctly.

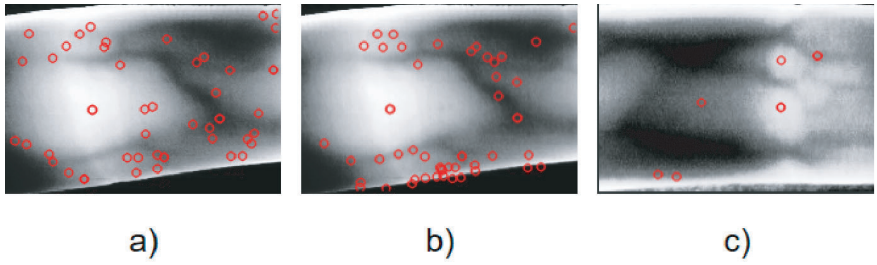

FIGURE 15. Finger vein keypoint detection. a) favourable parameter values; b) higher edge threshold; c) higher contrast threshold.

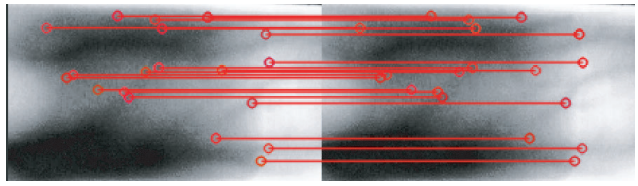

FIGURE 16. Distance-based feature matching.

3.2.3. Feature Matching

When the features are extracted, we employ OpenCV's Brute force and FLANN--based (Fast Library for Approximate Nearest Neighbour) matcher to perform the keypoint matching. According to the OpenCV documentation, the Brute force matcher finds for each descriptor in the first set, the closest one in the second set by trying each descriptor. On the other hand, the FLANN-based matcher is considered to be faster than the Brute force one, however, it will find a good match, although it may not find the best possible one.

Namely, we have found out that SIFT and SURF algorithms are very sensitive to the intra class variations. We have observed that when matching the keypoints extracted from two different impressions of the same finger vein pattern, many of these are false matches, meaning that the Euclidean distance between them is high. Due to such occurrence, we perform thresholding on these keypoints by classyfing two keypoints as matched if the Euclidean distance between them is below the chosen threshold. Note that this issue was addressed by similar approach in [12]. Experimental observations have shown favourable results when addressing the aforementioned issue by means of distance thresholding method.

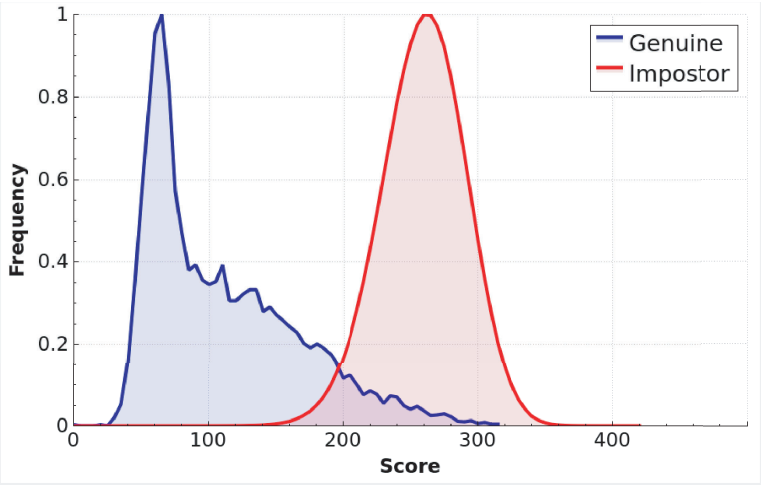

FIGURE 17. Genuine and impostor distribution using SIFT/Brute force approach for finger vein images.

3.2.4. Score Normalization

Match scores from fingerprint module when using Suprema range from 0 to 1. However, note that the distances between matched SIFT-extracted finger vein keypoints range from 0 to ∼500 (see Figure 17). Hence before proceeding with fusion, these match scores need to be normalized.

We have conducted experiments utilizing several normalization techniques. To map the obtained distance-based score to interval [0,1] we use double sigmoid, hyperbolic tangens, min/max and Z-score methods.

The normalized score using double sigmoid is then given as follows [4].

$$s_k^N = \begin{cases} \frac{1}{1+\exp\left(-2((s_k-t)/r_1)\right)}, & \text{if } s_k < t, \\ \frac{1}{1+\exp\left(-2((s_k-t)/r_2)\right)}, & \text{otherwise,} \end{cases} \tag{6}$$

where t is the reference operating point and r_1 and r_2 denote the left and right edges of the region in which the function is linear, i.e., the double sigmoid function exhibits linear characteristics in the interval $(t - r_1,\ t + r_2)$. Value t is generally chosen to be some value from the region of overlap between the genuine and impostor score distribution, whereas r_1 and r_2 are made equal to the extent of overlap between the two distributions toward the left and right of t, respectively [4].

Normalized scores from genuine and impostor distribution shown in Figure 17 are depicted in Figure 18. We have chosen value 230 for t while r_1 and r_2 are

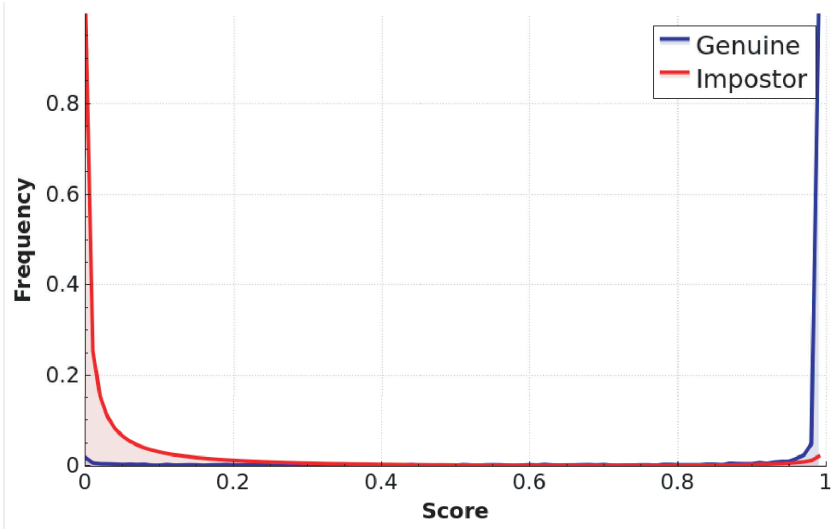

FIGURE 18. Normalization of SIFT/Brute force scores by means of double sigmoid function.

both set to 20. It is important to note that in order to obtain good results, one must select these parameters carefully.

Note that in Figure 18, high frequency of genuine scores is around 1, whereas the frequency of impostor scores is higher around 0. This shows that our finger vein module matched different impressions from the same finger as genuine correctly while different impressions from different fingers were matched as impostors, respectively.

The normalized score by means of hyperbolic tangens function is obtained as follows [4].

$$s_k^N = \frac{1}{2}\left\{\tanh\left(0.01\left(\frac{s_k - \mu_{S_G}}{\sigma_{S_G}}\right)\right) + 1\right\}, \tag{7}$$

where μ_{S_G} and σ_{S_G} are the mean and the standard deviation calculated from the genuine scores. This method is robust because it is insensitive to the presence of aberrant scores [4].

Z-score normalization method is given as follows [4].

$$s_k^N = \frac{s_k - \mu_S}{\sigma_S}, \tag{8}$$

where μ_S and σ_S are the mean and the standard deviation of the set of scores. This technique is not robust because of its two parameters (mean and standard deviation) which are sensitive to outliers [4]. Normalization by means of Z score method is presented in Figure 19.

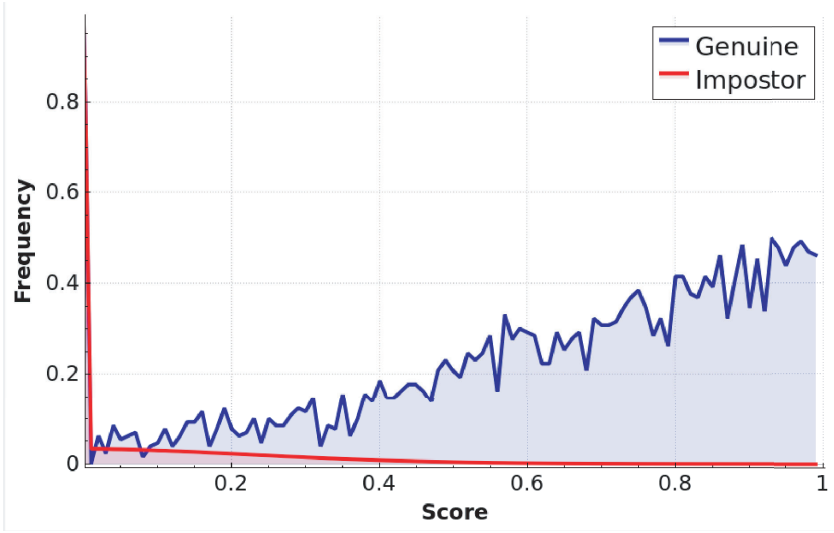

FIGURE 19. Normalization of SIFT/Brute force scores by means of Z-score approach.

The simplest normalization method is the Min − Max. Normalized score obtained by means of Min − Max technique is given as follows [4].

$$s_k^N = \frac{s_k - \text{Min}(S)}{\text{Max}(S) - \text{Min}(S)}, \tag{9}$$

where $\text{Max}(S)$ and $\text{Min}(S)$ are respectively the maximum and minimum values of the scores. This method is not robust because it is sensitive to the presence of outliers [4].

3.3. Score-Level Fusion

Several methods have been proposed on score level fusion approach. However, we have conducted our experiments using min, max, tanH, sum, mean and product fusion methods.

Score fusion using the selected methods is conducted as follows [15].

$$MS_{\text{final}} = \min(MS_{fp}, MS_{fv}), \tag{10}$$
$$MS_{\text{final}} = \max(MS_{fp}, MS_{fv}), \tag{11}$$
$$MS_{\text{final}} = \tanh(MS_{fp}) + \tanh(MS_{fv}), \tag{12}$$
$$MS_{\text{final}} = MS_{fp} + MS_{fv}, \tag{13}$$
$$MS_{\text{final}} = (axMS_{fp} + bxMS_{fv})/2, \tag{14}$$
$$MS_{\text{final}} = MS_{fp} \times MS_{fv}, \tag{15}$$

where MS_{final} represents the final match score while MS_{fp} and MS_{fv} denote the match score from fingerprint and finger vein, respectively. Also note that in (10) a and b represent the weights assigned to each trait. We have used 0.5 for both a and b, respectively.

4. Experimental Results

Well-established indicators of performance of a biometric system are False Accept Rate (FAR/FRR), False Reject Rate (FRR), Receiver Operating Characteristic (ROC) and Equal Error Rate (EER). A false accept occurs when a genuine score falls below the threshold η, whereas false reject is the result of an impostor score exceeding the given threshold η. Hence FAR/FRR represents a fraction of impostor scores that surpass the threshold η while FRR marks a portion of genuine scores that have not surpassed the given threshold η. The ROC curves are mainly used to compare the performance of different biometric systems where FRR is plotted against FAR/FRR in a logarithmic scale. EER denotes the point where the FAR/FRR equals the FRR. Higher EER hence suggests worse performance of the system.

To perform evaluation of our system, we have conducted several tests. Namely, we have tested the performance of each module, where a particular module represents unimodal biometric system. Subsequently we have conducted tests on both modules as part of our multimodal system. Multimodal system tests were conducted using each of the six aforementioned fusion methods. In total there were 4 213 440 impostor fingerprint pairs, 17 640 genuine fingerprint pairs, 7 269 480 impostor and 9 540 genuine finger vein pairs.

The preprocessing time durations of all finger vein images from SDUMLA-HMT database are given in Table 1. Note that this evaluation was done on Lenovo IdeaPad 700-15ISK laptop with Intel Core i5 6300 HQ 2.3 GHz (quad core), 8 GB DDR4 2133MHz and Nvidia GeForce GTX 950M 2 GB. Time duration of preprocessing of single image is ∼0.007 seconds. Note that the preprocessing is done sequentially.

TABLE 1. Time durations of preprocessing stages of all finger vein images from the SDUMLA-HMT database (3816 images).

Phase	Filtering	Edge detection	Contour computation	ROI ext.
Duration (s)	27.5	1.4	0.18	0.31

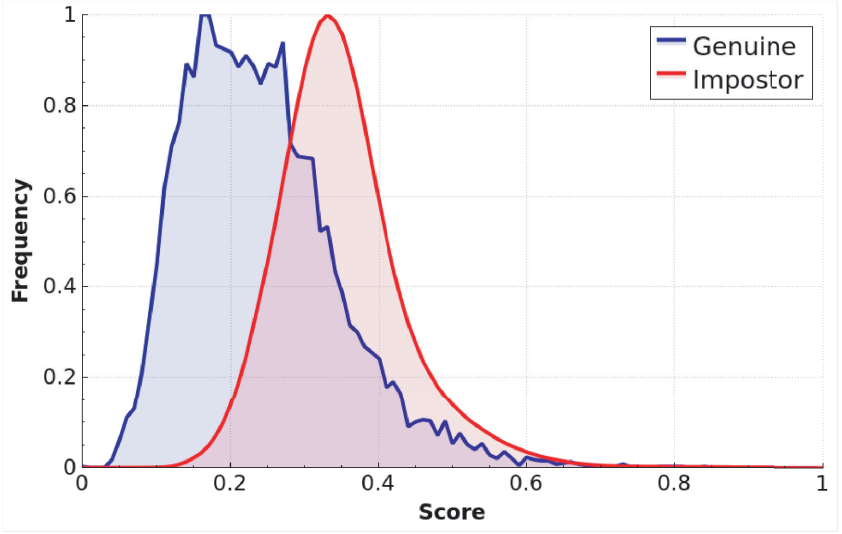

FIGURE 20. Genuine and impostor distribution using SURF/FLANN approach for finger vein images.

In Table 2 we present the results obtained by testing both fingerprint and finger vein recognition modules independently. Namely, we may conclude that both modules as such have obtained noteworthy results. The best recognition rate is provided by fingerprint module - 94.3 %. For finger veins, the combinations of SIFT and brute force matcher provide similar equal error rates ∼6 %. Note that the use of FLANN-based matcher with SURF algorithm obtained rather lower recognition rate. The background cause is discussed in the latter part of this section.

TABLE 2. Equal Error Rates obtained with unimodal systems.

Module	EER	Recognition rate
Fingerprint	**5.7 %**	**94.3 %**
Finger vein (SIFT/BF, double sigmoid)	6.1 %	93.9 %
Finger vein (SIFT/BF, Z-score)	6.12 %	93.88 %
Finger vein (SIFT/BF, Min-Max)	6.1 %	93.9 %
Finger vein (SIFT/BF, TanH)	6.12 %	93.88 %
Finger vein (SURF/FLANN)	26.12 %	73.88 %

Table 3 portrays the results obtained with various fusion methods, whereas the normalization of SIFT/brute force scores is done by means of hyperbolic tangens. In general, this particular combination with sum and mean fusion methods has yielded the best results. The recognition rate goes as high as 97.88 %. FAR/FRR performance plot of this combination is shown in Figure 21, whereas the ROC curve plot is shown in Figure 22. On the other hand, the use of min, max and product fusion methods resulted in slightly lower recognition rates.

Table 3. Equal Error Rates obtained with SIFT/Brute force and TanH normalization combination.

Fusion method	Min	Max	**Sum**	Product	TanH	**Mean**
EER	6.67 %	6.03 %	**2.12 %**	6.49 %	2.79 %	**2.12 %**
Recognition rate	93.33 %	93.97 %	**97.88 %**	93.51 %	97.21 %	**97.88 %**

As shown in Table 4, Table 5 and Table 6, all three SIFT/brute force combinations utilizing Z-score, Min – Max and double sigmoid normalization methods obtained rather notable scores. All three combinations obtained recognition rates >90 %.

Table 4. Equal Error Rates obtained with SIFT/Brute force and Z-score normalization combination.

Fusion method	Min	Max	Sum	**Product**	TanH	Mean
EER	6.03 %	6.04 %	4.8 %	**3.16 %**	4.67 %	4.79 %
Recognition rate	93.97 %	93.96 %	95.2 %	**96.84 %**	95.33 %	95.21 %

Table 5. Equal Error Rates obtained with SIFT/Brute force and Min-Max normalization combination.

Fusion method	Min	Max	Sum	Product	**TanH**	Mean
EER	6.67 %	6.06 %	3.28 %	4.27 %	**2.99 %**	3.23 %
Recognition rate	93.33 %	93.94 %	96.72 %	95.73 %	**97.01 %**	96.77 %

Table 6. Equal Error Rates obtained with SIFT/Brute force and double sigmoid normalization combination.

Fusion method	Min	Max	Sum	Product	**TanH**	Mean
EER	6.2 %	6 %	5.47 %	6.12 %	**5 %**	5.4 %
Recognition rate	93.8 %	94 %	94.53 %	93.88 %	**95 %**	94.6 %

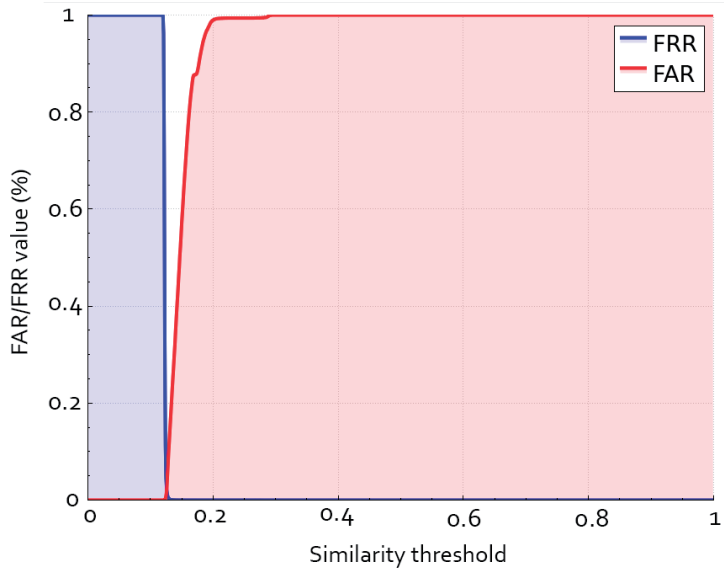

FIGURE 21. FAR/FRR performance plot for combination of SIFT/Brute force normalized by hyperbolic tangens and fused by mean method.

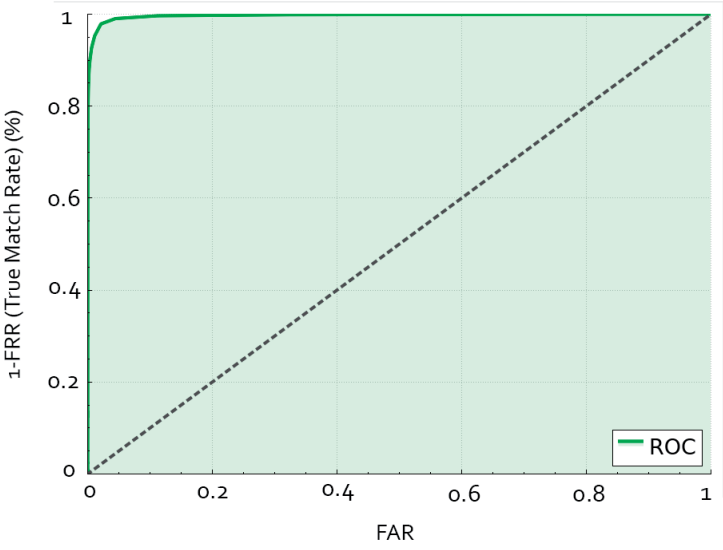

FIGURE 22. ROC curve plot for combination of SIFT/Brute force normalized by hyperbolic tangens and fused by mean method.

We observe that recognition rates are significantly lower when FLANN matcher is used (see Table 7). Such behaviour may also be seen in Figure 20. The overlap between genuine and impostor is greater when compared to Figure 17, where Brute force matcher was used. As we have regarded before, FLANN matcher does not necessarily find the best keypoint match from the matching set. Hence employment of FLANN-based matcher greatly influences the obtained scores.

TABLE 7. Equal Error Rates obtained with SURF/Flann combination.

Fusion method	Min	Max	Sum	**Product**	TanH	Mean
EER	6.68 %	26.12 %	14.05 %	**5.95 %**	11.09 %	13.99 %
Recognition rate	93.32 %	73.88 %	85.95 %	**94.05 %**	88.91 %	86.01 %

During our investigation it was rather difficult to find a similar solution combining fingerprints and finger veins. Moreover, finding experimental results obtained from SDUMLA-HMT database was impossible at the time of our research. Fortunately, Yang and Zhang in [31] proposed a solution incorporating feature-level and score-level fusion of fingerprints and finger veins (see section 2).

Similar to our results, their ROC curves confirm superiority of multimodal systems over unimodal ones. In addition, feature-level fusion significantly outperformed score-level fusion approach. Tests were carried out on the dataset of 640 fingerprints and the same number of finger veins.

Another important outcome of their research was obtaining recognition accuracy rates of 99.687 % for feature-level fusion and 98.75 % for score-level fusion. We achieved similar results despite having dataset consisting of images from multiple sensors. Thus, we may consider our solution as a noteworthy entry into the particular research field.

5. Conclusion and Future Work

In this paper we introduced OpenFinger, our automated multimodal biometric system utilizing fingerprint and finger vein pattern for recognition purposes. It consists of two major modules, where each module is used for processing of particular biometric trait. The extraction of salient features (Level 2 features) in fingerprint module is done by employing convolutional neural network, which is trained on images from four different sensors thus providing sensor interoperability.

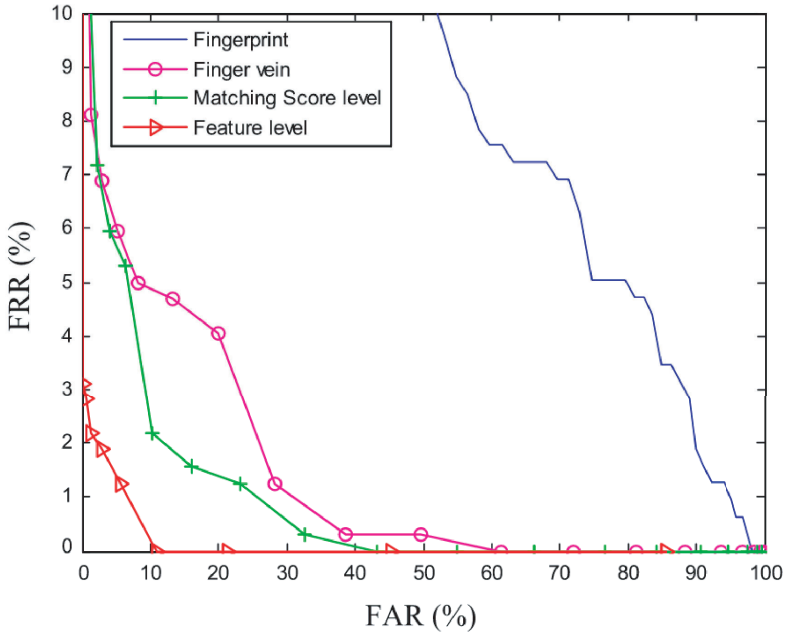

FIGURE 23. ROC curves measured for unimodal and multimodal systems by authors in [31].

On the other hand, finger vein recognition module employs keypoint-based features which are computed by SIFT and SURF algorithms. Prior to extraction, we employ our endpoint-based algorithm for ROI extraction. Despite obtaining high recognition rates, we consider that our finger vein preprocessing algorithm ought to be further improved. ROI extraction may be improved by adapting the algorithm to slanted finger images. In addition, we consider employing convolutional neural network for extraction of salient features from finger vein images. Another future prospect is to administer concurrency for preprocessing phase.

Nonetheless, we may consider our current solution rather successful, as have the a posteriori results shown, due to its ability to keep up with already proposed solutions. Hence the proposed fully automated multimodal biometric system based on fingerprints and finger vein patterns utilizing several normalization as well as fusion techniques along with its complete evaluation and comparison with published pertinent research may be considered as a contribution to the field of biometrics.

REFERENCES

[1] AHMAD, M. I.—WOO, W. L.—DLAY, S.: *Non-stationary feature fusion of face and palmprint multimodal biometrics*, Neurocomputing **177** (2016), 49–61.

[2] BARTŮNĚK, J. S.: *Fingerprint Image Enhancement, Segmentation and Minutiae Detection*. PhD Thesis, Blekinge Tekniska Högskola, Karlskrona, 2016.

[3] BAY, H.—TUYTELAARS, T.—VAN GOOL, L.: *SURF: speeded up robust features*. In: *Computer Vision – ECCV 2006, ECCV 2006.* (A. Leonardis, H. Bischof, A. Pinz eds.), *Lecture Notes in Computer Science Vol. 3951*, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 407–417.

[4] BEN KHALIFA, A.—GAZZAH, S.—ESSOUKRI BEN AMARA, N.: *Adaptive score normalization: a novel approach for multimodal biometric systems*, World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering **7** (2013), 205–213.

[5] BRADSKI, G.: *The OpenCV Library*. Dr. Dobb's Journal of Software Tools, 2000. `http://www.drdobbs.com/open-source/the-opencv-library/184404319`, Accessed on: 13. 5. 2019.

[6] CANZIANI, A.—PASZKE, A.—CULURCIELLO, E.: *An analysis of deep neural network models for practical applications*, Computer Vision and Pattern Recognition, 2017. `https://arxiv.org/abs/1605.07678`

[7] DAS, R.—PICIUCCO, E.—MAIORANA, E.—CAMPISI, P.: *Convolutional neural network for finger-vein-based biometric identification*, IEEE Transactions on Information Forensics and Security **4** (2019), 360–373.

[8] JAIN, A.—FLYNN, P.—ROSS, A. A.: *Handbook of Biometrics*. 1st edition, Springer--Verlag, 2008.

[9] JAIN, A.—ROSS, A. A.—NANDAKUMAR, K.: *Introduction to Biometrics*. 1st edition, Springer-Verlag, 2011.

[10] JIA, Y.—SHELHAMER, E.—DONAHUE, J.—KARAYEV, S.—LONG, J.—GIRSHICK, R.—GUADARRAMA, S.—DARRELL, T.: *Caffe: Convolutional Architecture for Fast Feature Embedding*. Technical Report, Berkeley Vision and Learning Center, 2014. `https://arxiv.org/pdf/1408.5093.pdf`

[11] JIN, L.: *Using deep learning for finger-vein based biometric authentication*. Towards Data Science, `http://web.archive.org/web/20080207010024/`, `http://www.808multimedia.com/winnt/kernel.htm,52019`, Accessed on: 12. 5. 2019.

[12] KAUBA, C.—REISSIG, J.—UHL, A.: *Pre-processing cascades and fusion in finger vein recognition*. In: *International Conference of the Biometrics Special Interest Group (BIOSIG)*, IEEE, Darmstadt, Germany, 2014.

[13] KÁDEK, L.: *Daktyloskopický sieťový systém DBOX - server*. Master's Thesis, Slovak Technical University in Bratislava, FEI ÚIM, 2018.

[14] KHELLAT-KIHEL, S.—ABRISHAMBAF, R.—MONTEIRO, J.—BENYETTOU, M.: *Multimodal fusion of the finger vein, fingerprint and the finger-knuckle-print using Kernel Fisher analysis*, Applied Soft Computing **42** (2016), 439–447.

[15] LATHA, L.—THANGASAMY, S.: *Efficient approach to normalization of multimodal biometric scores*, International Journal of Computer Applications **32** (2011), 57–64.

[16] LOWE, D. G.: *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision **60** (2004), 91–110.

[17] M2SYS: . *M2-fuseID smart finger reader*, M2SYS, `http://www.m2sys.com/wp-content/uploads/pdf/M2-FuseID-web-flyer.pdf`. Accessed on: 10. 5. 2019.

[18] MARÁK, P.—HAMBALÍK, A.: *Fingerprint recognition system using artificial neural network as feature extractor: design and performance evaluation*, Tatra Mt. Math. Publ. **67** (2016), 117–134.

[19] MIURA, N.—NAKAZAKI, K.—FUJIO, M.—TAKAHASHI, K.: *Technology and future prospects for finger vein authentication using visible-light cameras*, Latest Digital Solutions and Their Underlying Technologies **67** (2018).

[20] NGUYEN, D.-L.—CAO, K.—JAIN, A. K.: *Robust minutiae extractor: integrating deep networks and fingerprint domain knowledge*. In: *International Conference on Biometrics (ICB)*, Gold Coast, QLD, Australia, 2018, IEEE. `https://arxiv.org/abs/1712.09401`

[21] ONG, T. S.—TENG, J. H.—MUTHU, K. S.—TEOH, A. B. J.: *Multi-instance finger vein recognition using minutiae matching*. In: *6th International Congress on Image and Signal Processing (CISP)*, Hangzhou, China, 2013. IEEE. pp. 1730–1735.

[22] RADZI, S. A.—KHALIL-HANI, M.—BAKHTERI, R.: *Finger-vein biometric identification using convolutional neural network*, Turkish Journal of Electrical Engineering & Computer Sciences **24** (2016), 1863–1878.

[23] SHAHEED, K.—LIU, H.—YANG, G.—QURESHI, I.—GOU, J.—YIN, Y.: *A systematic review of finger vein recognition techniques*, Information **9** (2018).

[24] TANG, Y.—GAO, F.—FENG, J.—LIU, Y.: *FingerNet: an unified deep network for fingerprint minutiae extraction*. In: *IEEE International Joint Conference on Biometrics (IJCB)*, 2017.

[25] TELGAD, R. L.—DESHMUKH, P. D.—SIDDIQUI, A. M.: *Combination approach to score level fusion for multimodal biometric system by using face and fingerprint*. In: *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, IEEE, Jaipur, India, 2014.

[26] THAI, R.: *Fingerprint Image Enhancement and Minutiae Extraction*. PhD Thesis, School of Computer Science and Software Engineering, The University of Western Australia, 2003.

[27] TURRONI, F.: *Fingerprint Recognition: Enhancement, Feature Extraction and Automatic Evaluation of Algorithms*. PhD Thesis, Università di Bologna, 2012.

[28] WANG, K.—MA, H.—POPOOLA, O. P.—LIU, J.: *Finger vein recognition*. In: *Biometrics* (J. Yang, ed.), Chapter 2, IntechOpen, Rijeka, 2011.

[29] XIE, S. J.—LU, Y.—YOON, S.—YANG, J.—PARK, D. S.: *Intensity variation normalization for finger vein recognition using guided filter based singe scale retinex*, Sensors **15** (2015), 17089–17105.

[30] YALAMANCHILI, P.—ARSHAD, U.—MOHAMMED, Z.—GARIGIPATI, P.— ENTSCHEV, P.—KLOPPENBORG, B.—MALCOLM, J.—MELONAKOS, J.: *ArrayFire: A high performance software library for parallel computing with an easy-to-use API*, AccelerEyes **106**, (2015). `https://github.com/arrayfire/arrayfire`

[31] YANG, J.—ZHANG, X.: *Feature-level fusion of fingerprint and finger-vein for personal identification*, Computer Science, Mathematics Pattern Recognit. Lett. **33** (2012), 623–628.

[32] YIN, Y.—LIU, L.—SUN, X.: *SDUMLA-HMT: A multimodal biometric database*. In: *Biometric Recognition. CCBR 2011.* (Z. Sun et al. eds.) Lecture Notes in Comput. Sci. Vol. 7098, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 260–268.

[33] ZHU, E.—YIN, J.—ZHANG, G.—HU, C.: *A Gabor filter based fingerprint enhancement scheme using average frequency*, (IJPRAI) **20** (2006), 417–430.

Received September 27, 2019

Institute of Computer Science and Mathematics
Slovak University of Technology
Ilkovičova 3,
SK–812-19 Bratislava
SLOVAKIA

E-mail: pavol.marak@stuba.sk
xkovaci1@stuba.sk