

Embedded Vein Recognition System with Wavelet Domain

Chih-Hsien Hsia and Chin-Feng Lai*

Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan

(Received March 15, 2020; accepted June 30, 2020)

Keywords: finger-vein recognition systems, symmetric mask-based discrete wavelet transform, repeated line tracking, support vector machine, embedded system

Advances in computer vision (CV) have led to an increasing market for biometric recognition systems. However, as more users are registered in a system, its expanding dataset will increase the system's response time and lower its recognition stability. As mentioned above, we propose a new high-performance algorithm suitable for embedded finger-vein recognition systems. First, the semantic segmentation based on DeepLabv3⁺ filters out the background noise and enhances processing stability. The adaptive symmetric mask-based discrete wavelet transform (A-SMDWT) and adaptive image contrast enhancement were used in the preprocessing of images, and feature extraction was performed through the repeated line tracking (RLT) method. Next, the histogram of oriented gradient (HOG) of the image was computed, after which a support vector machine (SVM) was then used to train a classifier. Finally, a self-established finger-vein image dataset as well as a public dataset was implemented in the Raspberry Pi platform, which is a low-level embedded system. The experimental results indicated that the proposed system offers advantages such as a high accuracy rate, low device cost, and fast response time. Therefore, the three major issues that were encountered in previous embedded finger-vein image verification systems were mitigated in this work.

1. Introduction

Biometric technologies utilize the distinct biological features of an individual for identification purposes. Therefore, digital technologies can be used to solve underlying problems such as forgetting one's password. Currently available biometric techniques include techniques utilizing the recognition of an individual's veins,^(1–11) face,^(12,13) palmprint/shape,⁽¹⁴⁾ gait,⁽¹⁵⁾ iris,⁽¹⁶⁾ and fingerprint. In general, facial recognition involves the capture of an individual's facial features through a visible light camera. The process of recognition is easily influenced by factors such as the person's facial position or angular movements, wavering light sources, and camera resolution. In addition, wrinkles that develop with age or cosmetic surgery can lead to errors in facial recognition. In fingerprint and palmprint recognition, physical contact is required for print acquisition. However, the secretion of grease and sweat and the presence of dirt on the hands of users can impact the recognition process because

*Corresponding author: e-mail: cinfon@ieec.org
<https://doi.org/10.18494/SAM.2020.2861>

of the acquisition of poor-quality print patterns. Moreover, as friction ridge impressions are external features of the body, they can easily be reproduced by people with ulterior motives. Therefore, regarding the use of friction ridge impressions, there is a risk of finger/palmprints being reproduced. Meanwhile, even though iris recognition offers many features that increase its recognition accuracy, it uses infrared (IR) light to scan an individual's iris and acquire its features, which can cause eye discomfort in the long run. The biological imaging recognition techniques share a common element insofar as they all involve the capture of an individual's external features, which explains why there are various external factors that can influence their reliability. Moreover, the cost and size of a given biometric device also determine whether it will be adopted. In contrast, the veins are a more reliable feature of an individual. Compared with other recognition methods, the vein recognition offers a lower risk of harm or forgery since the veins are located beneath the skin, and this technology allows an individual's features to be acquired using smaller devices. In addition, vein patterns do not change with age, and even identical twins have different vein patterns.^(1–11)

Novel recognition techniques developed in recent years mostly utilize biological imaging.^(1–11,15–32) Given that there are many types of biometrics, a technique's ability to replace others and be accepted by the market is determined by three major criteria, namely, high accuracy rate, low equipment cost, and fast response time. For vein recognition to be widely accepted, the equipment cost must be lowered, although low-resolution cameras will result in poor image quality. In addition, increasing the number of people in a recognition dataset will lead to the inclusion of individuals with similar recognition features, which will lower the system's response time and stability, thus increasing the difficulty of realizing practical application. Certain vein recognition devices currently in use require users to press their fingers directly onto a sensor. However, grease on the fingers may contaminate the sensor and raise sanitary concerns. Therefore, it is important to develop recognition devices that involve contactless acquisition of user data. In this work, we addressed the three important issues mentioned above, i.e., (1) low-cost and contactless devices, (2) high accuracy rate, and (3) real-time processing.

In recent years, the accuracy rates and response times of biometric systems have become important indicators and have led to many studies on vein recognition methods. Wang *et al.*⁽²⁾ proposed a method that combines the Radon transform and eigenvalues. Given that the input device was a capacitive press contact device that might entail sanitary concerns, and that there were only ten subjects collected in the dataset, the method lacked experimental objectivity (that is, the recognition stability and response time could not be determined). Mulyono and Horng⁽³⁾ used a conventional low-cost network camera as their image acquisition device. When near-infrared (NIR) light with a wavelength range of 760–1000 nm is passed through a finger, the hemoglobin beneath the skin absorbs the IR light and creates an image in which vein patterns are visible as shadows. Hence, the device used an IR LED array as its light source. To avoid the impact of light sources nearby, the camera was equipped with an IR filter that allowed IR light with specific wavelengths to pass through while filtering out visible light. However, the drawback of the device was its slower response time. Im *et al.*⁽⁴⁾ utilized a NIR camera to study the use of fixed feature points to improve recognition speed. However, the device is costly.

The algorithm proposed by Miura *et al.*⁽⁵⁾ presented the characteristics of grayscale images and repeatedly tracked the veins of a finger to identify the finger-vein pattern. Although this algorithm showed excellent recognition results, the computing time was long. Zhang *et al.*⁽⁶⁾ utilized curvelet extraction to obtain vein patterns. However, if low-resolution images are used, noise will also be extracted during feature extraction, which makes the method unsuitable for low-cost embedded devices. Liu and Song⁽⁷⁾ proposed an embedded platform for implementing a finger-vein recognition system. Bicubic interpolation was used to reduce the spatial resolution and increase its response time. However, this approach decreased the amount of vein pattern information and affected feature extraction results. Moreover, the physical device had a high cost, large memory requirement, and high computational complexity. In Ref. 10, the low-low (LL) bands in a discrete wavelet transform (DWT) are a means of reducing noise and computational complexity; however, the wavering light source during feature point acquisition affects the accuracy. Hsia proposed the utilization of multiple feature points for regions of interest (ROI) positioning as well as a recognition method based on the concept of multi-image quality assessment (MQA)⁽¹¹⁾ in a recent study. As in other similar studies,⁽¹⁰⁾ the device was not implemented using a low-cost embedded system and was therefore costly. Syarif *et al.*⁽¹⁷⁾ proposed an integrated enhanced maximum curvature method that used the histogram of oriented gradient (HOG) feature descriptor to retain image quality. Even though the authors combined the method with a support vector machine (SVM) to train a classifier, the recognition results were poor. Qin *et al.*⁽¹⁸⁾ proposed a capillary-directed convolution for predicting capillary patterns, in which the Hausdorff distance was used to analyze the spatial similarity between vein samples. Nevertheless, this method required a large memory and a long processing time. Yang *et al.*⁽¹⁹⁾ suggested the use of a Gabor filter to enhance the stability of recognition. However, the computational complexity of certain processes through which the finger is distinguished from the background means that these processes take a long time, making them unsuitable for embedded platforms. In Ref. 20, a neural network (NN) was used to perform matching in finger-vein recognition. However, the high complexity of the operation resulted in slow response times. The vein recognition system proposed by Yu *et al.*⁽²¹⁾ used repeated iteration to determine the direction of capillaries and employed the Sobel edge detector to enhance the capillary patterns. Their dataset only had 25 subjects, and the experiment was somewhat subjective owing to the low number of subjects, leading to concerns about the stability of the system for the recognition of more subjects. The two-dimensional edge detection method presented in Ref. 22 generated better results and used image segmentation methods to process finger-vein images. Even though this approach was able to reduce the processing complexity, the accuracy was unsatisfactory for low-resolution images. A modified binary tree model to enhance the performance of vein recognition has also been proposed.⁽²³⁾ However, the overall system also exhibited higher computational complexity. Lu *et al.*⁽²⁴⁾ presented a new vein recognition system that simultaneously acquired and integrated data from two vein images in order to enhance its matching performance. The disadvantage of the system was the high cost of producing the corresponding device. The same group also presented a finger-vein ROI positioning method for reducing computational load.⁽²⁵⁾ The method included edge detection and directional correction techniques that acquire vein regions. However, the method used

the K-nearest neighbors (KNN) approach to build its classifier and thus required considerable physical resources of a deep neural network (DNN) integrated with conventional features and an SVM to enhance the accuracy of input vein images with poor quality. Moreover, the training required by the system consumed a lot of time, making the system unsuitable for real-time applications. Qi and El Yacoubi⁽²⁶⁾ proposed a DNN for representation learning to predict image quality using very limited knowledge. Das *et al.*⁽²⁷⁾ proposed a deep learning (DL) method based on a convolutional neural network (CNN). While the method exhibited a stable recognition rate, its huge computational complexity necessitated the use of high-end graphics processing units (GPUs), making it unsuitable for the development of low-cost embedded platform applications.

To overcome the problems encountered by the aforementioned authors, we proposed an embedded vein verification system that uses NIR and low-cost RGB cameras in conjunction with an embedded system (the Raspberry Pi 3 Model B platform). First, the finger in the foreground is separated from the background on the basis of DL, so that vein images of fingers in a complex external environment can be acquired in a stable manner. Next, the spatial resolution was reduced by means of an adaptive symmetric mask-based discrete wavelet transform (A-SMDWT), and ROIs with localized vein patterns were identified to reduce the noise, computational load, and system response time. Afterwards, biomedical image contrast techniques were used to enhance the vein pattern features, which were acquired through the repeated line tracking (RLT) method. Then, HOG of the image is computed, after which a SVM is used for classification. The completion of these steps yielded the desired vein patterns and increased the speed of the overall system in terms of embedded image recognition and comparison.

The rest of this paper is organized as follows. Section 2 outlines the proposed embedded vein verification system with a line tracking, an HOG, and an SVM. Section 3 provides the experimental results. Finally, conclusions are given in Sect. 4.

2. Low-complexity Vein Verification Technique

After a rigorous literature review and analysis, we modified algorithms for computer vision (CV) based on the techniques discussed in previous studies and implemented the algorithms in the Raspberry Pi embedded platform. We achieved better data processing and performance than those in the literature. As shown in Fig. 1, the research method consisted of four stages, (1) front-end hardware devices (including vein acquisition); (2) preprocessing of images (including image enhancement, noise removal, and normalization); (3) postprocessing of images (including feature extraction); and (4) verification and matching mechanisms.

2.1 Semantic image segmentation

When veins are illuminated by NIR light, the hemoglobin in the red blood cells absorb the light and appear as shadows [Fig. 2(a)]. Finger-vein images were acquired through this principle, as shown in Fig. 2(b). Since the vein images in this work were acquired using a

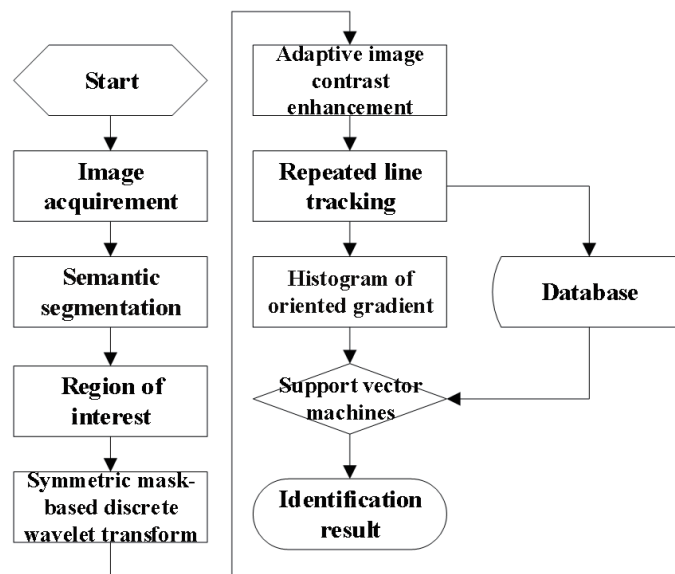


Fig. 1. Flowchart of the proposed system.

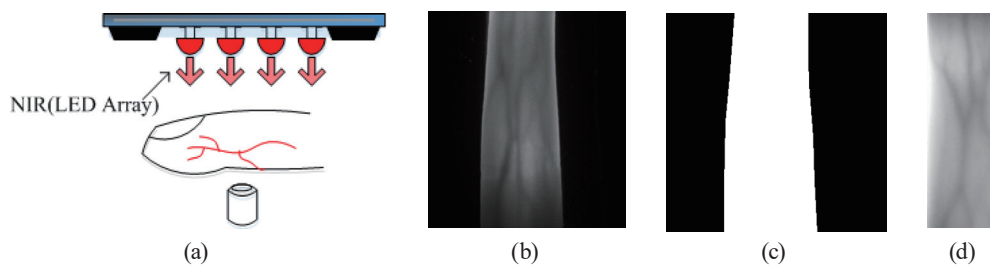


Fig. 2. (Color online) NIR vein image acquisition: (a) schematic of the method of acquisition; (b) image acquired by the system; (c) via DeepLabv³⁺; (d) acquired ROI image (90 × 220).

low-cost RGB camera, the image was blurred, and energy loss from important feature points occurred during feature extraction. Therefore, in the acquisition of biological patterns, it is crucial to enhance the image quality through preprocessing. Furthermore, displacement may occur during vein image acquisition. To retain useful information from the foreground (veins) and overcome the problem of displacement, the semantic segmentation using DeepLabv³⁺ and ROI positioning was established, as shown in Figs. 2(c) and 2(d).

At present, both methods are subject to background noise or exposure. In this work, we propose the framework based on semantic segmentation,⁽³³⁾ as shown in Fig. 3, and a model suitable for finger veins is trained. This method uses a semantic segmentation network, which is described as follows: (1) The encoder is composed of deep convolutional neural networks (DCNNs). The problem of object scaling can be overcome by using atrous convolution, convolution layers of different scales, max pooling layers for feature extraction, and sampling layer by layer. (2) Bilinear upsampling is carried out on the features of the image provided by the encoder; 1 × 1 and 3 × 3 convolution layers are utilized to obtain the final segmentation

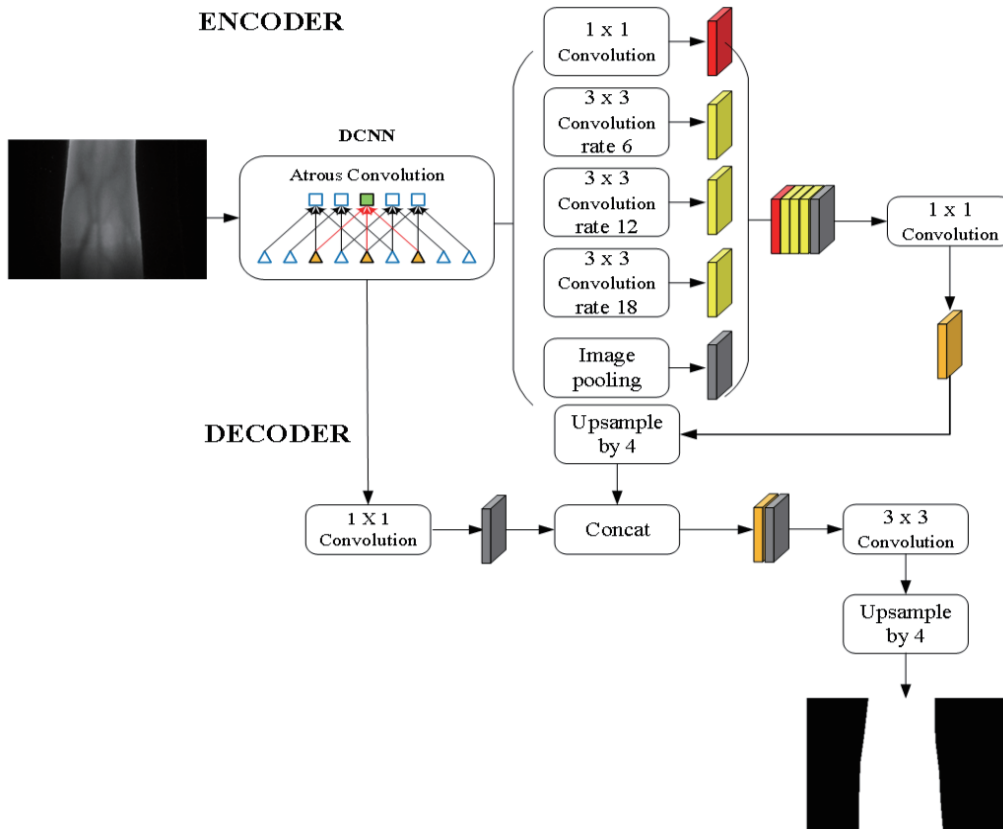


Fig. 3. (Color online) Framework of semantic segmentation.

of finger images through quadrilinear upsampling in a decoder. DCNNs of DeepLabv³⁺ have Mobilenetv2. The average intersection ratio of Xception is higher than that of Mobilenetv2, but Xception comprises 20 times more training parameters than Mobilenetv2. Therefore, Mobilenetv2 has a shorter verification time, so it is selected as the DCNN architecture for this work.

First, an upper point and a lower point on the left side of a finger were identified. The inclination of the finger was determined through these two points. Next, Eq. (1) was used to calculate and correct the angle of inclination, as this significantly reduces the problem of recognition errors caused by finger displacement.

$$angle = \frac{\tan\left(\frac{|x_1 - x_2|}{|y_1 - y_2|} \times 180^\circ\right)}{\pi} \quad (1)$$

We proposed an SMDWT with LL-band matrix coefficient for image preprocessing. Neighboring pixels were used to compute the two-dimensional convolution products, as shown in Fig. 4(a). The A-SMDWT consisted of the 5/3 coefficient⁽³⁴⁾ and the 9/7 coefficient,⁽³⁵⁾ and the complexity of each image was calculated from their standard deviation (SD) and a specific

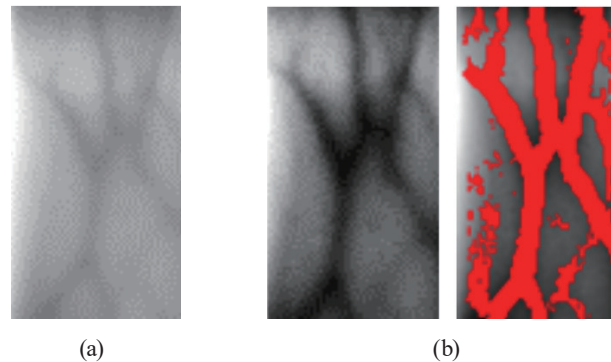


Fig. 4. (Color online) Vein acquired through A-SMDWT: (a) LL-band (45×110); (b) adaptive image contrast enhancement.

threshold. The $9/7$ coefficient is used when the SD of the image exceeds the threshold, as this indicates a high image frequency;⁽³⁶⁾ the $5/3$ coefficient is used when the SD of the image is lower than the threshold, as this indicates a lower image frequency.⁽³⁶⁾ This method not only compiles the effective energy while filtering out noise, but also reduces spatial resolution while decreasing the response time of a system.

When a device is capturing images, the contrast of images may differ from one image to another owing to the placement of the user's finger and the stability of the light source. Therefore, we utilized an adaptive histogram⁽³²⁾ to enhance the dynamic range of the vein images, thereby increasing the contrast and obtaining better results during feature extraction as well as enhancing the accuracy of recognition and matching. An image acquired through this method is shown in Fig. 4(b).

2.2 Feature extraction

From the vein images, it can be seen that the veins appear as black lines after absorbing IR light (i.e., the grayscale level of the veins is lower than that of the surrounding tissue). From the perspective of digital imaging, the veins can be regarded as the valley of an image, in which the degree of darkness determines the depth of the trough. Therefore, vein patterns can be analyzed by detecting the troughs, as shown in Figs. 5(a) and 5(b).

We employed the RLT method, which is a method of tracking vein patterns, as described in Ref. 5. The method tracks the patterns of vein capillaries by detecting the troughs of the image. First, a tracking point was constructed to detect troughs (veins) in nearby pixels; if a trough was detected, the tracking point was specified and the tracking resumed, after which the tracking results obtained were stored in a defined space. If no troughs were present, another tracking point was constructed and vein patterns were obtained from the defined space. The steps are described below.

Step 1: An initial tracking point is specified as (x_s, y_s) and is determined by uniform random numbers. The moving direction attribute is defined as Dlr , Dud and prevents the tracking point from moving into paths with excessive curvature. Dlr , Dud are determined as follows.

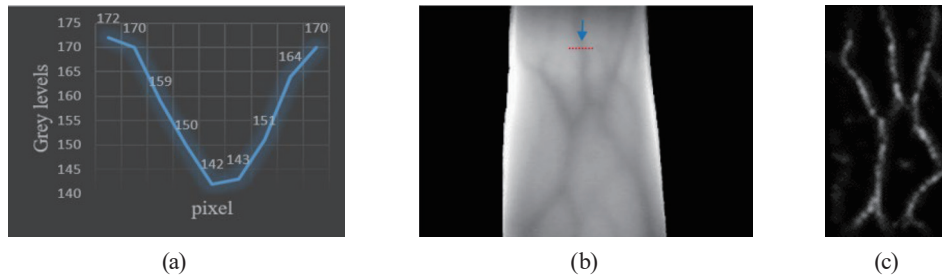


Fig. 5. (Color online) Finger-vein cut-off and distribution of gray levels: (a) distribution of gray levels; (b) position of cut-off area; (c) feature extraction result.

$$D_{lr} = \begin{cases} (1,0), & \text{if } R_{nd}(2) < 1 \\ (-1,0), & \text{otherwise} \end{cases}; \quad D_{ud} = \begin{cases} (0,1), & \text{if } R_{nd}(2) < 1 \\ (0,-1), & \text{otherwise} \end{cases} \quad (2)$$

Here, $R_{nd}(2)$ denotes a uniform random number of $0 - n$.

Step 2: The defined space T_c is initialized and a pixel N_c is defined, which ensures that the tracking point is within the finger and prevents the duplication of previous tracking points. N_c is defined as

$$N_c = \overline{T_c} \cap R_f \cap N_r(x_c, y_c), \quad (3)$$

where $N_r(x_c, y_c)$ denotes a neighboring pixel of the tracking point, as shown in

$$N_r(x_c, y_c) = \begin{cases} N_3(D_{lr})(x_c, y_c), & \text{if } R_{nd}(100) < p_{lr} \\ N_3(D_{ud})(x_c, y_c), & \text{if } p_{lr} + 1 \leq R_{nd}(100) < p_{lr} + p_{ud} \\ N_8(x_c, y_c), & \text{if } p_{lr} + p_{ud} + 1 \leq R_{nd}(100) \end{cases} \quad (4)$$

where p_{lr} and p_{ud} are the selection probabilities, and their optimum values are 50 and 25, respectively, based on test results. $N_3(D)(x, y)$ can be defined as

$$N_3(D)(x, y) = \left\{ (D_x + x, D_y + y), (D_x - D_y + x, D_y - D_x + y), (D_x + D_y + x, D_y + D_x + y) \right\}. \quad (5)$$

Afterwards, the tracking points are stored in the defined space T_c , and V_l is then used to determine if a tracking point has moved. If V_l is positive, then Step 2 is repeated after moving the tracking point. If V_l is zero or negative, then Step 3 is repeated, as this indicates that the current tracking point is not on a vein.

Step 3: Repeat Steps 1 and 2.

Finally, the vein patterns are obtained from the self-defined space. The results of the feature extraction are shown in Fig. 5(c).

The HOG method with the gradient structure characteristic of local shape has a good performance, so it produces good results when applied to vein structures. It is used to obtain the feature descriptors from the RLT images with the ROI as follows. First, the RLT image was segmented for HOG testing; the image was segmented to $N \times N$ pixels ($N = 8$), and 2×2 blocks were grouped, after which the blocks were moved to perform the computation. Then, the features of each block were extracted. The most common way to calculate pixel gradients for each cell is to use Gaussian gradient templates for convolution. Although more complex convolution kernels have been used in Ref. 37, they are not better than using horizontal and vertical convolution kernels. In this work, we used vertical and horizontal directions, $D_x = [-1, 0, 1]$ and $D_y = [-1, 0, 1]^T$, respectively, when computing convolution kernels, as shown in Eq. (6). Next, the histogram division of the processed gradient image was performed. It has been found that the effect is best when 180 degrees of orientation are divided into nine bins. This step utilizes the gradient to obtain a weighted vote on the directions of the histogram. The histograms in each cell are obtained, and the histograms of four cells are strung into a column, after which the combined histograms are normalized.

$$f_x = f \times D_x \text{ and } f_y = f \times D_y \quad (6)$$

$$\text{L2-hys: } f = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}}, \text{ if } v \geq 0.2, v = 0.2 \quad (7)$$

Here, v is a vector that has not been normalized and its maximum value is 0.2, and ε is a small constant to avoid a zero divisor. We used Eq. (7) for normalization, thus acquiring the HOG descriptors of the RLT image.

2.3 Verification matching

The concept of the SVM⁽³⁸⁾ is to establish an optimal objective function for classification by means of modeling. In addition, it uses the principle of structural risk minimization to obtain the so-called optimal classification hyperplane, which serves as the support vector. The objective of obtaining the hyperplane is to ensure that the maximum margin and minimum classification error exist between different classes of data. However, it is difficult to classify data in an actual space. When the data are not linearly separable, a kernel function must be used to map the data from the input space to a feature space, as shown in Fig. 6(a). Figure 6(b) shows the SVM used in this work; it is provided by the OpenCV library. The method consists of two steps.

1) Training:

In this work, the extracted vein pattern features served as input data for training. The training methods mostly employ multiclass SVMs together with radial basis functions (RBFs).

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right) \quad (8)$$

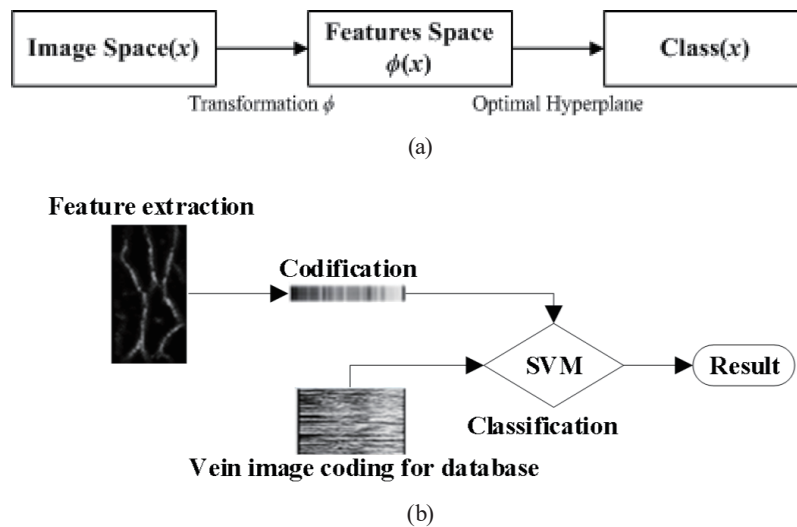


Fig. 6. Vein matching: (a) SVM steps; (b) process of verification matching.

2) Testing:

After training, a classification model is obtained and matching is performed by testing the images, as shown in Fig. 6. The recognition rate is enhanced and optimized by adjusting the parameters.

3. Experimental Results

Figure 7(a) shows the embedded finger-vein verification system developed for this work. To enable the practical use of the system at a low cost, the verification technique was integrated with the interface. An NIR light with a wavelength of 940 nm served as the light source, while a Raspberry Pi RGB camera and NIR filters were used to capture images. Then, the Raspberry Pi platform (Table 1) was used to process the images and the algorithms for verification, thereby establishing the embedded finger-vein system shown in Fig. 7(b).

The finger-vein images used in this work were a public dataset and a private dataset consisting of self-captured images. An 850-nm-wavelength NIR light served as the light source of the public FVUSM dataset⁽²⁷⁾ that comprised a total of 2952 grayscale vein images provided by 123 volunteers (83 males and 40 females) aged between 20 to 52 years. The size of the grayscale images was 640×480 . Each volunteer provided their left and right index and middle fingers. Six vein images were taken for each finger. To ensure that the experimental results are more objective, the left and right index and middle fingers were regarded as being sourced from datasets of different people, thus increasing the number of samples for comparison. There were a total of 492 classes, and each class included six finger-vein images. On the other hand, 940-nm-wavelength NIR light served as the light source of the self-captured images in the private dataset. The images were taken from 32 volunteers (20 males and 12 females) aged between 20 and 25 years. Each volunteer provided their left and right index and middle fingers, and six vein images were taken from each finger. The size of the grayscale images

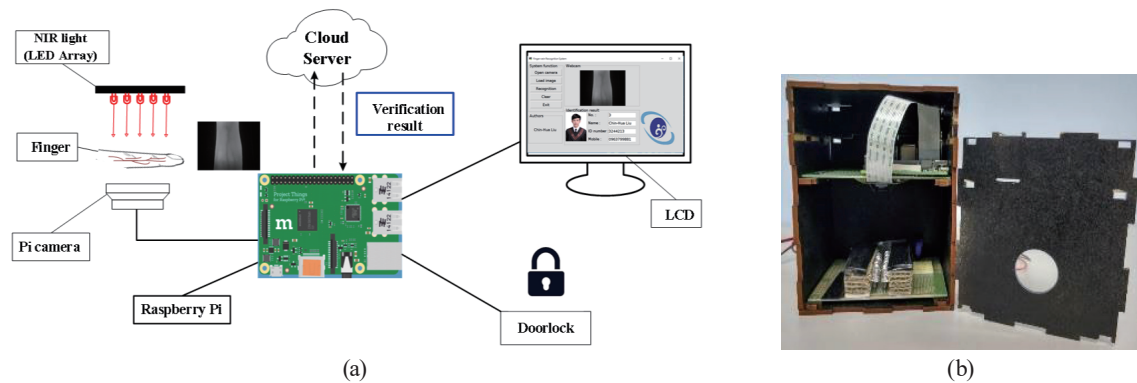


Fig. 7. (Color online) Devices used in this work: (a) system architecture; (b) self-developed finger-vein verification system device.

Table 1
System specifications.

| | |
|----------------------|------------------|
| CPU clock rate | 1.2 GHz |
| Operating system | Raspbian Stretch |
| Memory | 1 GB LPDDR2 |
| Programming language | Python |

was 320×240 . There were a total of 128 classes, and each class included six vein images. The results of the experiment shows the proposed vein verification technique was effective for the FVUSM dataset and our dataset when the data were collected with three images each being used for training and for testing.

We used the equal error rate (EER), an important indicator of system security, as a measure for assessing the verification efficiency. The EER is given by Eq. (9). With regard to the verification system, there are two types of possible errors: the false rejection rate (FRR) and the false acceptance rate (FAR). The former, given by Eq. (10), indicates that the system has incorrectly rejected an authorized user; the latter, given by Eq. (11), indicates that the system has incorrectly accepted an unauthorized user. When the similarity setting is gradually increased from its minimum value, the FAR will gradually decrease from its maximum value and become approximately zero. Meanwhile, the FRR will gradually increase from approximately zero. When the similarity setting reaches its maximum value (for instance, the similarity must be 100% in order for an image to be recognized), the FRR will be at its peak, and the curves formed by the FAR and the FRR will intersect at a point known as the EER, which is the point where both recognition error rates are equal. At this intersection, the sum of the FRR and the FAR is minimum. The performance of the system is most balanced when the similarity is set to the receiver operating characteristic (ROC), as shown in Fig. 8. Therefore, the magnitude of the EER is often used as an indicator of the security performance of verification systems. In this work, the SVM parameters in the OpenCV library⁽³⁹⁾ were adjusted to achieve system optimization. From the results in Table 2, it can be seen that this proposed method achieved a higher accuracy than that in previous works. When three images were used for training and three for testing, the results showed that the proposed finger-vein verification technique

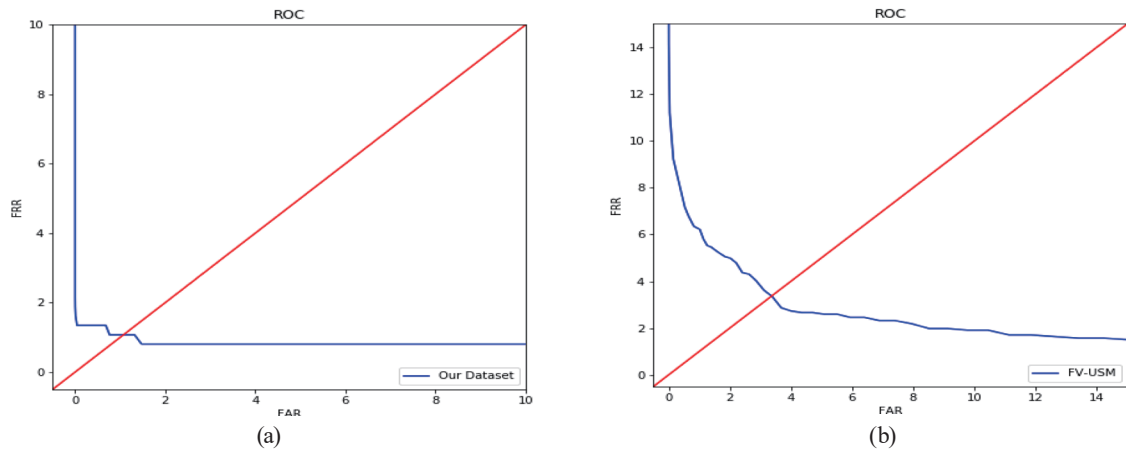


Fig. 8. (Color online) ROC curve: (a) our dataset; (b) FVUSM dataset.

Table 2
EER of various methods.

| Method | EER | |
|--|-----------------|-----------|
| | Our Dataset (%) | FVUSM (%) |
| Liu <i>et al.</i> ⁽⁷⁾ (3 Training / 3 Testing) | 1.51 | 5.28 |
| Lee <i>et al.</i> ⁽¹⁰⁾ (3 Training / 3 Testing) | 3.13 | 10.20 |
| Miura <i>et al.</i> ⁽¹¹⁾ (3 Training / 3 Testing) | 2.50 | 7.20 |
| This work (3 Training / 3 Testing) | 1.06 | 3.30 |
| This work (4 Training / 2 Testing) | 0.20 | 2.91 |
| This work (5 Training / 1 Testing) | 0.06 | 1.62 |

had EERs of 1.06 and 3.30% in our dataset and FVUSM dataset, respectively. EER was then applied as a serious access control system, and the results showed that the error decreased to 0.06 and 1.62% in the dataset if five images were used for training and one for testing.

$$EER = FRR = FAR \quad (9)$$

$$FRR = (FP / FP + TP) \times 100\% \quad (10)$$

$$FAR = (FN / FN + TP) \times 100\% \quad (11)$$

Here, *FP* indicates that the class was falsely predicted as positive; *TP* indicates that the class was correctly predicted as positive; *FN* indicates that the class was falsely predicted as negative.

The matching time of the overall image verification system of this work was shorter than 0.2 s, and the feature extraction time was approximately within 0.5 s. Therefore, it took the system around 0.61 s to completely process one image, as shown in Table 3. By a comparison with the results in previous works under the same development conditions, we found that the proposed method achieved a fourfold decrease in the response time.^(2,5,7)

Table 3
Response times of various methods.

| Method | Extraction (s) | Recognition (s) | Total (s) |
|-------------------------------------|----------------|-----------------|-----------|
| Wang <i>et al.</i> ⁽²⁾ | 3.52 | 1.43 | 4.95 |
| Miura <i>et al.</i> ⁽⁵⁾ | 6.74 | 1.64 | 8.38 |
| Z. Liu <i>et al.</i> ⁽⁷⁾ | 1.63 | 1.27 | 2.90 |
| This work | 0.48 | 0.13 | 0.61 |

4. Conclusions

We proposed a new low-complexity algorithm for real-time applications that can be implemented in a low-cost embedded finger-vein verification system. First, the finger in the foreground is separated from the background using DL, so that vein images of fingers in a complex external environment can be acquired in a stable manner. The A-SMDWT was used to enhance the contrast of images together with the RLT and the HOG methods for feature extraction, and an SVM was used to train a classifier. Finally, the method was implemented using the Raspberry Pi platform. The experiments included the self-capturing of images as well as a data and performance analysis using the public finger-vein image dataset called FVUSM. In comparison with the results of relevant studies, the proposed method achieved a better accuracy and a higher computation speed. The response time of the overall system was 0.61 s, and its EER was approximately 1.06%.

References

- 1 J. Hashimoto: Proc. 2006 Symp. VLSI Circuits (IEEE, 2006) 5–8.
- 2 D. Wang, J. Li, and G. Memik: IEEE Trans. Consum. Electron. **56** (2010) 799. <https://doi.org/10.1109/TCE.2010.5506004>
- 3 D. Mulyono and S.-J. Horng: Proc. 2008 Int. Symp. Biometrics and Security Technologies (IEEE, 2008) 1–8.
- 4 S.-K. Im, H.-M. Park, S.-W. Kim, C.-K. Chung, and H.-S. Choi: Proc. 2000 IEEE Int. Conf. Consumer Electronics (IEEE, 2000) 2–3.
- 5 N. Miura, A. Nagasaka, and T. Miyatake: Mach Vis Appl. **15** (2004) 194. <https://doi.org/10.1007/s00138-004-0149-2>
- 6 Z. Zhang, S. Ma, and X. Han: Proc. 2006 IEEE Int. Conf. Pattern Recognition (IEEE, 2006) 145–148.
- 7 Z. Liu and S. Song: IEEE Trans. Consum. Electron. **58** (2012) 522. <https://doi.org/10.1109/TCE.2012.6227456>
- 8 E. C. Lee, H. Jung, and D. Kim: Sensors **11** (2011) 2319. <https://doi.org/10.3390/s110302319>
- 9 N. Miura, A. Nagasaka, and T. Miyatake: Proc. 2005 Int. Conf. Machine Vision Applications (IAPR, 2005) 347–350.
- 10 C.-H. Hsia: J. Imaging Sci. Technol. **26** (2018) 30402-1. <https://doi.org/10.2352/J.ImagingSci.Technol.2018.62.3.030402>
- 11 C.-H. Hsia: IEEE Sens. J. **18** (2018) 790. <https://doi.org/10.1109/JSEN.2017.2772799>
- 12 C.-H. Hsia, J.-S. Chiang, and C.-Y. Lin: Sci. Iran. **22** (2015) 2081. http://scientiainiranica.sharif.edu/article_3756_1fa54ffe927da5a4d11ad735b254cc5a.pdf
- 13 C.-H. Hsia and H.-T. Li: J. Technol. **30** (2015) 343. <http://www.airitilibrary.com/Publication/alDetailedMesh?DocID=10123407-201512-201601040003-201601040003-343-350>
- 14 J.-M. Guo, C.-H. Hsia, Y.-F. Liu, J.-C. Yu, M.-H. Chu, and T.-N. Le: Expert Syst. Appl. **39** (2012) 11728. <https://doi.org/10.1016/j.eswa.2012.04.081>
- 15 C.-H. Hsia, Y.-J. Dai, S.-L. Chen, T.-L. Lin, and J. Shen: J. Internet Technol. **19** (2018) 451. <https://doi.org/10.3966/160792642018031902013>
- 16 M. Trokielewicz, A. Czajka, and P. Maciejewicz: IEEE Trans. Inf. Forensics Secur. **14** (2019) 1501. <https://doi.org/10.1109/TIFS.2018.2881671>

- 17 M. A. Syarif, T. S. Ong, A. B. J. Teoh, and C. Tee: *Multimed. Tools Appl.* **76** (2017) 8659. <https://doi.org/10.1007/s11042-016-3315-4>
- 18 H. Qin, L. Qin, and C. Yu: *Opt. Eng.* **50** (2011) 057281. <https://doi.org/10.1117/1.3572129>
- 19 J. Yang, Y. Shi, and J. Yang: *Lecture Notes Comput. Sci.* **5994** (2010) 374. <https://doi.org/10.5772/17182>
- 20 A. N. Hoshyar and R. Sulaiman: *Proc. 2010 Int. Symp. Information Technology (IEEE, 2010)* 1020–1023.
- 21 X. Yu, W. Yang, Q. Liao, and F. Zhou: *Proc. 2009 Int. Congr. Image and Signal Processing (IEEE, 2009)* 1–5.
- 22 C. Yu, H. Qing, and L. Zhang: *Proc. 2008 IEEE Int. Conf. Bioinformatics and Biomedical Engineering (IEEE, 2008)* 1876–1879.
- 23 T. Liu, J. Xie, W. Yan, P. Li, and H. Lu: *Neural Comput. Appl.* **26** (2015) 969. <https://doi.org/10.1007/s00521-014-1783-x>
- 24 Y. Lu, S. Yoon, and D. S. Park: *Electron. Lett.* **50** (2014) 1591. <https://doi.org/10.1049/el.2014.1956>
- 25 Y. Lu, X. J. Xie, S. Yoon, J. Yang, and D. S. Park: *Sensors* **13** (2013) 14339. <https://doi.org/10.3390/s131114339>
- 26 H. Qin and M. A. El Yacoubi: *IEEE Trans. Circuits Syst. Video Technol.* **28** (2018) 1677. <https://doi.org/10.1109/TCSVT.2017.2684826>
- 27 R. Das, E. Piciuccio, E. Maiorana, and P. Campisi: *IEEE Trans. Inf. Forensics Secur.* **14** (2019) 360. <https://doi.org/10.1109/TIFS.2018.2850320>
- 28 M. S. M. Asaari, S. A. Suandi, and B. A. Rosdi: *Expert Syst. Appl.* **41** (2014) 3367. <https://doi.org/10.1016/j.eswa.2013.11.033>
- 29 C.-H. Hsia, J.-M. Guo, and C.-S. Wu: *Multimed. Tools Appl.* **76** (2017) 25179. <https://doi.org/10.1007/s11042-016-4296-z>
- 30 J.-M. Guo, Y.-F. Liu, C.-H. Hsia, S.-Y. Su, and H. Lee: *IEEE Trans. Inf. Forensics Secur.* **9** (2014) 953. <https://doi.org/10.1109/TIFS.2014.2355495>
- 31 M.-H. Wang, M.-L. Huang, S.-D. Lu, and Z.-Y. Lee: *Sens. Mater.* **31** (2019) 3437. <https://doi.org/10.18494/SAM.2019.2478>
- 32 K. Nkurikiyeyezu, A. Yokokubo, and G. F. Lopez: *Sens. Mater.* **32** (2020) 703. <https://doi.org/10.18494/SAM.2020.2650>
- 33 L. Liu, Y. Pang, S. W. Zamir, S. Khan, F. S. Khan, and L. Shao: *IEEE Access* **8** (2020) 34019. <https://doi.org/10.1109/ACCESS.2019.2946031>
- 34 C.-H. Hsia, J.-M. Guo, and J.-S. Chiang: *IEEE Trans. Circuits Syst. Video Technol.* **19** (2009) 1202. <https://doi.org/10.1109/TCSVT.2009.2020259>
- 35 C. H. Hsia, J.-M. Guo, and J. S. Chiang: *Signal Process.* **92** (2012) 89. <https://doi.org/10.1016/j.sigpro.2011.06.009>
- 36 C.-H. Hsia: *J. Internet Technol.* **15** (2014) 1083. <https://doi.org/10.6138/JIT.2014.15.7.01>
- 37 N. Dalal and B. Triggs: *Proc. 2005 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2005)* 886–893.
- 38 R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin: *J. Mach. Learn. Res.* **9** (2008) 1871. <https://dl.acm.org/doi/pdf/10.5555/1390681.1442794>
- 39 C.-H. Hsia, S.-C. Yen, and J.-H. Jang: *Sens. Mater.* **31** (2019) 1803. <https://doi.org/10.18494/SAM.2019.2351>