The Neglected Background Cues Can Facilitate Finger Vein Recognition

Pengyang Zhao[a], Shuping Zhao[b], Jing-Hao Xue[c], Wenming Yang[a,*], Qingmin Liao[a]

[a] *Shenzhen International Graduate School / Department of Electronic Engineering, Tsinghua University, China.*
[b] *School of Computer Science, Guangdong University of Technology, China.*
[c] *Department of Statistical Science, University College London, UK.*

Abstract

Recently, finger vein based biometric authentication has attracted considerable attention due to its high efficiency and high security. However, most existing finger vein representation methods focus on vein traits while ignoring background cues, although background cues also convey identity information specific to each individual. In this paper, we leverage background intensity variations in finger vein images as new features to enrich discriminative representation, and accordingly propose a new descriptor named Intensity Orientation Vector (IOV). IOV, scaleable to reflect characteristics of finger tissues, offers additional informative cues for finger vein representation. Furthermore, we propose a new learning scheme named Semantic Similarity Preserved Discrete Binary Feature Learning (SSP-DBFL) for finger vein recognition. Unlike the most bimodal binary feature representation methods, SSP-DBFL preserves high-level semantic similarity in a common Hamming space to exploit the consensus between vein traits and background cues. Specifically, given a finger vein image, we first extract the direction difference vectors (DDV) as the main vein traits and the IOV as the auxiliary background cues . Subsequently, we jointly learn projection functions from these two types of features in a supervised manner, converting the two features into discriminative binary codes with their semantic similarity preserved. Finally, the binary codes are pooled into histogram-based vectors for finger vein representation. Extensive experiments are conducted on five widely used finger vein databases and demonstrate the effectiveness of our proposed IOV and SSP-DBFL.

Keywords: Finger vein recognition, vein trait, background cue, intensity orientation vector, binary feature learning

1. Introduction

Reliable identification of an individual is the essence of personal authentication, which has been widely applied to electronic payment and access control systems. Owning to high efficiency, biometric authentication techniques have gained great attention in the past decade, using various types of traits including face [1], palmprint [2], and finger vein [3]. As an emerging trait, finger vein refers to the blood vessels beneath the finger skin. The finger vein trait meets the growing need for high security in modern society, since it is inside the body.

Since Kono et al. [4] introduced the idea of finger vein authentication in 2000, this technique has attracted much attention [5]. Different from extrinsic traits, such as face and fingerprint, finger veins are invisible under RGB cameras while can only be captured by NIR cameras with a living person. Therefore, finger vein traits have high security against forgery. There is no doubt that the vein patterns or features provide unique identity information for authentication. However, due to the scattering and absorption of NIR light in different finger skin layers, such as epidermis and stratum basale, the captured finger vein images always contain blurred vein regions and background intensity variations. Unlike mainstream biometrics (e.g., face and fingerprint), finger veins make up only a small part of the finger vein image, not to mention blurred vein regions. With limited available vein information, most existing methods focus on the extraction of vein patterns but fail to obtain reliable features in some blurred regions. The work [6] inspires us that the practical optical properties of fingers vary from person to person. This means that the large background in finger vein images also contains some discriminative cues, which are usually neglected or represented ineffectively by most methods.

To surmount the above limitations, we propose a new descriptor named Intensity Orientation Vector (IOV) to reflect the discriminative background cues in a finger vein image. IOV is scaleable to describe the background variations precisely, offering auxiliary informative cues for representation and recognition. Then in order to make full use of both vein traits and background cues in finger vein images, we further propose a new binary feature learning scheme named Semantic Similarity Preserved Discriminative Binary Feature Learning (SSP-DBFL), to jointly learn projection functions from vein traits and background cues simultaneously. As the first feature learning method explicitly considering both vein traits and background cues, SSP-DBFL

*Corresponding author.
Email address: yangelwm@163.com (Wenming Yang)

leverages the consensus and complementarity of these two types of features, by preserving their semantic similarity in the common Hamming space. Specifically, given a finger vein image, we first extract the direction difference vectors (DDV) as the main vein traits and the IOV as the auxiliary background cues. Then, SSP-DBFL jointly learn a series of hash functions from these two types of features to convert them simultaneously into discriminative binary codes. Finally, the histogram-based descriptors are constructed for finger vein recognition. Furthermore, SSP-DBFL adopts discrete optimization to reduce quantification errors, which is also often ignored by other methods. We conduct extensive experiments of finger vein recognition and the results clearly demonstrate the effectiveness of our proposed descriptor (IOV) and method (SSP-DBFL).

This paper presents two main technical novelties and contributions:

- Different from previous finger vein recognition methods which only focused on vein traits, we leverage the background variations as a new auxiliary feature. This offers a novel perspective to represent finger vein images for recognition. Accordingly, we propose a new descriptor named IOV to deliver auxiliary discriminative power by thoroughly reflecting the background intensity variations in finger vein images.

- We propose a new supervised binary-feature learning scheme named SSP-DBFL, to jointly learn compact binary codes for finger vein representation simultaneously from vein traits and background cues. SSP-DBFL effectively preserves semantic similarity between IOV and DDV to exploit the consensus between them. What is more, the objective function of our SSP-DBFL is optimized in a discrete manner, which reduces the quantification error.

The rest of this paper is organized as follows. In Section 2, we briefly review some topics related to our work. The core idea of the proposed descriptor IOV is detailed in Section 3. In Section 4, we present the SSP-DBFL approach and its discrete optimization formulation. Extensive experimental results and some discussions are provided in Section 5. Finally, Section 6 concludes this paper.

2. Related Work

2.1. Finger Vein Recognition

Recently, many effective methods have been proposed for finger vein recognition [5]. These methods can be grouped into four major categories: structure based, local feature based, deep

learning based, and subspace learning based.

Structure-based methods aim to extract vein structures or vein minutiae. As the oxyhemoglobin in veins absorbs more NIR light than other tissues, the vein areas appear darker than the surroundings in the finger vein images. Based on this phenomenon, Miura et al. proposed Local Maximum Curvature (LMC) [7] to extract vein patterns. Since LMC is not robust to illumination variations, Syarif et al. [8] proposed Enhanced Maximum Curvature (EMC) by integrating the Hessian enhancement and Histogram of Oriented Gradients (HOG) to improve the performance. Further, Spatial Curve Filters (SCFs) [9] were designed to extract vein networks with the help of curve length estimation. Yang et al. [10] used a vein orientation map to guide the curvature image to extract reliable vein networks and its backbone. Apart from the extraction of whole vein networks, vein minutiae were extracted in [11, 12] to deal with vein deformations. However, compared with those of fingerprints, the minutiae of finger veins are less. Typical minutiae of finger veins are ending points of veins or tribranshes [12].

Local feature based approaches aim to extract features that reflect local variations or patterns in a small region of finger vein images. Local Binary Pattern (LBP) [13] and its variants Efficient Local Binary Pattern (ELBP) [14] were successfully applied in finger vein representation. In addition, Local Direction Coding (LDC) [15], Local Line Binary Pattern (LLBP) [16, 17] and Gabor filters based method [18] aimed to extract the direction information of each pixel in finger vein images.

Since deep learning shows great potential in various computer vision tasks, various researchers introduced it into finger vein representation and recognition. As a successful attempt, Hong et al. [19] employed a pretrained VGG-16 model to extract finger vein features for recognition. FV-Net [20] was proposed to make the representation more robust to translations, which altered the output of the Convolutional Neural Networks (CNN). Autoencoder was exploited in [21], with the output of the encoder as the finger vein representation. Moreover, Siamese structure based network [22] and Generative Adversarial Network (GAN) [23] were also exploited to finger vein representation to make the representation more discriminative. Recently, Huang et al. [24] and Liu et al. [25] integrated attention mechanisms into networks to boost the recognition performance. By replacing Euclidean distance with cosine distance, arccosine center loss [26] is proposed to make vein features more discriminative. Wang et al. [27] proposed MRFBCNN, integrating bilinear pooling to obtain higher-order features. FVT [28] first introduced Transformer

to finger vein authentication, achieving satisfactory results.

Unlike the above types of methods, subspace learning based methods try to map original features into a low dimensional subspace, which maintains some ideal properties and removes noise during the projection. The representative Principal Component Analysis (PCA) based method [29] aimed to find a subspace with maximum variance. Mei et al. proposed a weighted sparse representation (WSRC) [30] algorithm for finger vein recognition, which weighted each training sample according to its sparse coefficients. Beyond that, some methods that convert the original features into Hamming subspace to learn binary features attracted much attention due to the potential for biometric authentication [31, 32]. For example, Liu et al. [33] proposed a Personalized Binary Codes (PBC) learning method, which incorporated the idea of Linear Discriminant Analysis (LDA) and the constraints of within-class sparsity. Partial Least Squares Discriminant Analysis (PLS-DA) is combined with the joint Bayesian model [34] for compact finger vein feature extraction and recognition. Additionally, a discriminant and sparse feature descriptor [3] is developed for finger vein representation by considering the relationship between different features. To obtain more robust features, Yang et al. [35] proposed sparse reconstruction error constrained low-rank representation (SRLRR), integrating low-rank constraint and sparse representation.

2.2. Finger vein representation

A discriminative feature representation can dramatically boost the recognition performance [2]. The intuitive finger vein representation is to extract binary vein patterns from raw images [36, 7, 8]. Such methods are plagued by translation and rotation of the captured finger. Hence, to handle these deformations, Chen et al. [37] used Scale Invariant Feature Transform (SIFT) to represent a finger vein image. Apart from SIFT, vein minutiae-based representation methods first located ending points or tribranches of vessels, then extracted the minutia features to represent finger veins. What is more, some conventional descriptors such as LBP [13], LLBP [16], Gabor [38, 18] were utilized to represent finger veins from different perspectives. In addition, learning-based methods have emerged in recent years, including manifold learning [39], subspace learning[40, 41, 29, 33], deep learning [20, 22, 42], which gradually become a hotspot in the community. Liu et al. [39] gave a framework for finger vein representation based on manifold learning for the first time. PCA [40, 41] and its variant [29] were also adopted in finger vein representation. However, the unsupervised approaches can not learn from discriminative

label information, which leads to unsatisfactory results. Different from PCA, PBC [33] borrowed from LDA and imposed the sparse constraint on the objective function, resulting in more discriminative and compact binary codes. Since deep learning-based methods show a powerful ability for face representation [43], many researchers proposed various models for finger vein representation. Hu et al. [20] utilized a vanilla CNN to extract finger vein features and proposed a template-like matching strategy to handle translations and rotations in the imaging process. A two-stream convolutional network [22] took raw images and mini-ROIs as inputs for better finger vein representation, which jointly exploited local features and global features in one model. Qin et al. [42] proposed a neural network for finger vein recovery and representation by introducing prior knowledge into the training stage. However, the insufficient samples limit the further improvement of deep learning-based models. Different from the approaches which focus only on the vein traits, our SSP-DBFL exploits discriminative finger vein representation originating from both vein traits and background cues.

2.3. Binary Feature Learning

In recent years, a variety of binary feature learning methods [44, 31, 2, 45] have been proposed for biometric authentication. Different from the hand-crafted binary features (e.g., LBP [13], LLBP [16] and ELBP [14]), the learned binary codes can be more compact and less redundant with a carefully designed objective function [45, 2]. Lu et al. [31] pointed out that quantized binary codes are able to eliminate small variations and noise, thus improving the robustness of the features. On top of that, the high computational and storage efficiency of binary features are also attractive to researchers. For instance, Compact Binary Feature Descriptor (CBFD) [45] was proposed to learn a compact binary feature in an unsupervised manner. CA-LBFL [1] exploited the contextual information of face images by imposing the constraint of the number of 0-1 shifts in binary codes. Fei et al. [2] introduced DDBPD methods to learn a compact and discriminative binary code for palmprint recognition. LCMFC [46] learns a joint representation from direction features and texture features of a palmprint images, automatically determining the importance of each type of features. Additionally, the binary feature learning was also used in multimodal biometric authentication approach [47], aiming to seek a common discriminative Hamming subspace for better representation. Compared with the above binary learning approaches, our SSP-DBFL jointly learns compact binary codes from both vein traits and background cues for finger vein representation, with the aim of exploiting the consensus and

complementarity between these two types of features for more discriminative and comprehensive representation.

2.4. Multi-type Features of Images

Researchers designed various methods to extract different cues in images, such as texture [48] and direction [18]. CNNs [49] rely on various learned filters (i.e., kernels) to gain different cues. However, it is difficult to distinguish the types of features extracted by the learned filters. Gabor filters are designed to extract the features with specific frequency and direction [18]. LBP [13] and its variants exploit the texture information. Pixel Difference Vector (PDV) [45] calculates the difference between the center point and neighboring pixels to describe texture information in a local region. However, the above feature descriptors neglect the background variations in finger vein images that can also provide discriminative cues. Therefore, we propose a new scaleable descriptor named IOV, in order to extract the informative background cues from finger vein images.

3. Leveraging Both Background Cues and Finger Vein Traits

3.1. Motivation

Since the amount of oxyhemoglobin (HbO) in the venous blood is more significant than in other tissues, finger veins can be captured by NIR rays with a wavelength within 780-900nm. The higher absorption of the 780-900nm NIR rays by the HbO leads to vein regions being darker than other regions in finger vein images. Hence for finger vein recognition, most existing studies focused on the extraction of vein features such as lines and directions [18]. However, it was revealed that the tissues or skin layers such as epidermis and stratum basale inside the finger also absorb NIR rays [6], which varies from person to person. Moreover, Kang et al. [50] pointed out that the knuckle region in the finger vein image looks brighter because the synovial fluid absorbs fewer NIR rays, which is also distinct between individuals. That is, these non-vein areas also contain some identity information that can help the finger vein based biometrics. However, unlike HbO which is distributed only in the veins, these tissues or layers are continuously and unevenly distributed in the whole finger, and hence their identity information is reflected in the background intensity variation of the finger vein image. In other words, the large areas neglected by previous

researches, i.e., the background regions, also convey discriminative identity information specific to each individual.

Motivated by this, for the first time, we propose to leverage the background intensity variation in the finger vein image as a new auxiliary feature and introduce a new descriptor called IOV to extract informative background cues for more discriminative and comprehensive finger vein representation.

3.2. Intensity Orientation Vector (IOV)

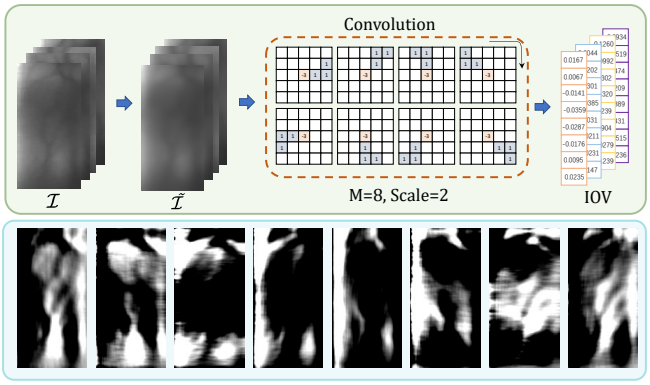


Figure 1: An illustration of the construction of IOV with $M = 8$, $Scale = 2$. Firstly, Gaussian smoothing is applied on the finger vein image $\mathcal{I}$ and obtain $\tilde{\mathcal{I}}$. Secondly, the convolved responses in M orientations are divided by the value of center pixel and concatenated to form IOV. The upper row details the procedure of extracting IOV. The lower row shows the IOV maps in the M orientations.

To reflect the background intensity distribution, we define a new feature descriptor called Intensity Orientation Vector (IOV). As veins and noise may disturb the extraction of IOV, we first apply Gaussian smoothing [50] on raw finger vein image $\mathcal{I}$ to obtain approximated intensity map $\tilde{\mathcal{I}}$:

$$\tilde{\mathcal{I}}(x, y) = \mathcal{I}(x, y) * \mathcal{G}(x, y, \sigma_{\mathcal{G}}), \tag{1}$$

where $*$ denotes the convolution operation, and $\mathcal{G}(x, y, \sigma_{\mathcal{G}})$ is the Gaussian kernel

$$\mathcal{G}(x, y, \sigma_{\mathcal{G}}) = \frac{1}{2\pi\sigma_{\mathcal{G}}^2} e^{-(x^2+y^2)/2\sigma_{\mathcal{G}}^2}. \tag{2}$$

In our work, $\sigma_{\mathcal{G}}$ is set to 9. Then, we form M filters for M orientations, each with scale R, to thoroughly describe intensity variations in $\tilde{\mathcal{I}}$. The mth filter calculates the sum of intensity

differences between the center pixel and the pixels within scale R in the $(m-1)\pi/M$ orientation, as illustrated in Fig. 1. We convolve finger vein images with these filters and arrange the convolved results into a vector that expresses the intensity differences between the center pixel and its neighboring pixels in the M orientations. Next, the stimulus of the intensity difference in each orientation to the intensity value of the center pixel is calculated, which can be written as

$$o_{p_c,d_i} = \frac{\alpha\left(\sum_{p_j \in \wp^R_{d_i}(p_c)}[p_j] - |\wp^R_{d_i}(p_c)| \times [p_c]\right)}{[p_c] + \beta}, \tag{3}$$

where α controls the strength of the intensity differences, which equals to 1 in this paper; β is a small value to avoid the denominator being zero; p_c is the position of the center pixel; $[\cdot]$ denotes the pixel intensity in the corresponding position; $\wp^R_{d_i}(p_c)$ represents the set of pixels which are located within scale R in the d_i orientation from the center pixel; and $|\wp^R_{d_i}(p_c)|$ gives the number of pixels in the pixel set $\wp^R_{d_i}(p_c)$. Finally, we generate the IOV by concatenating the stimulus in each orientation as

$$\mathrm{IOV}_{\mathrm{p_c}} = \left[o_{p_c,d_1}, o_{p_c,d_2}, \ldots, o_{p_c,d_M}\right]. \tag{4}$$

Fig. 1 illustrates the construction of IOV and its resulting maps in M orientations. By eliminating the interference of the finger vein, our IOV depicts the intensity variations of the background thoroughly. That is, IOV can be used to reflect some individual properties of finger tissues and skin layers, which enriches the discriminative representation of finger vein images.

3.3. Direction Difference Vector (DDV)

The vein traits are important for finger vein representation, but they are blurred by Gaussian smoothing during the construction of IOV, Hence in order to effectively extract the vein traits, we adopt the direction difference vector (DDV) in the following way.

Direction feature is widely used in finger vein representation and recognition [18]. To obtain discriminative direction features of finger veins, DDV, which was used as a palmprint descriptor [2], is constructed to extract the vein direction information. Specifically, Gabor filter banks with the direction of $\theta_i = (i-1)\pi/N_\theta$ $(i = 1, \ldots, N_\theta)$ are adopted here to obtain N_θ direction responses. The convolutional result between the ith Gabor filter and a finger vein image $\mathcal{I}$ can be formulated as

$$r_i(x, y) = G_i * \mathcal{I}(x, y), \tag{5}$$

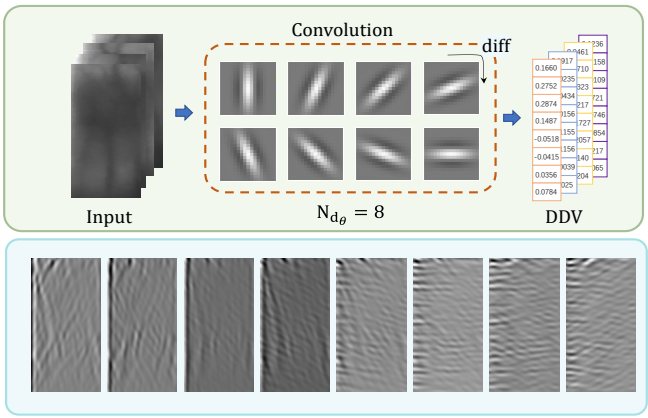


Figure 2: An illustration of DDV with $N_\theta = 8$. The upper row shows the procedure of DDV construction. The lower row demonstrates the DDV maps in the N_θ orientations.

where $r_i(x, y)$ represents the direction response in the direction of $(i-1)\pi/N_\theta$ at the location (x, y); G_i is the Gabor filters with the format of

$$G(x, y, \theta_i) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{1}{2}\left(\frac{x_{\theta_i}^2 + \zeta y_{\theta_i}^2}{\sigma^2}\right)\right\} \cos\left(2\pi f_0 x_{\theta_i}\right), \tag{6}$$

where $\begin{bmatrix} x_{\theta_i} \\ y_{\theta_i} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$; σ denotes the scale of a Gabor filter; ζ represents the aspect ratio of the elliptical Gaussian envelope; and f_0 is the central frequency. Then, the DDV is formed as the concatenation of the response differences between all adjacent directions as

$$\text{DDV} = [(r_1 - r_{N_\theta}), (r_2 - r_1), \ldots, (r_{N_\theta} - r_{N_\theta - 1})]. \tag{7}$$

In our work, we use Gabor filters with 12 directions, i.e., $N_\theta = 12$. The construction of the DDV is illustrated in Fig. 2 (with $N_\theta = 8$ for better visual comparison with the IOV maps in Fig. 1). A DDV indicates changes of direction responses and preserves the significant direction.

4. Semantic Similarity Preserved Discrete Binary Feature Learning (SSP-DBFL)

This section first details the formulation and optimization algorithm of the proposed learning scheme SSP-DBFL, and then introduces how to use SSP-DBFL for finger vein recognition. The flowchart of SSP-DBFL is illustrated in Fig. 3.

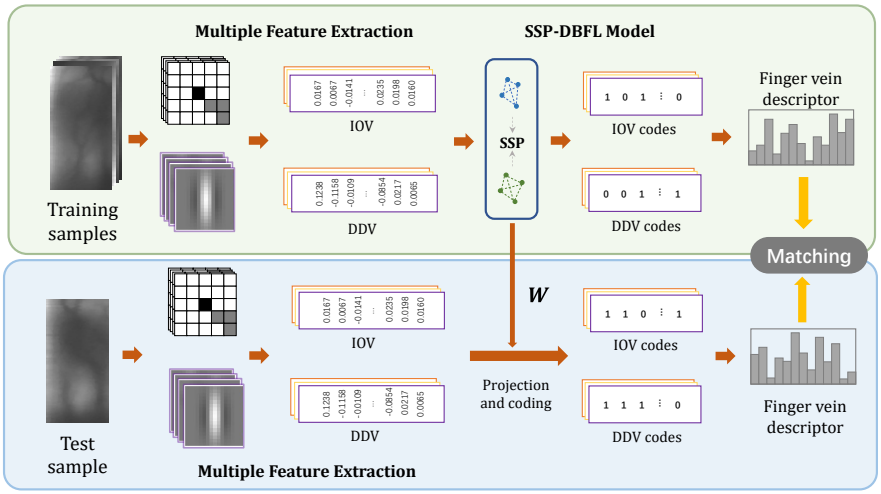


Figure 3: The flowchart of our proposed learning scheme SSP-DBFL. SSP-DBFL aims to jointly learn both vein traits and background cues from finger vein images to exploit their consensus and complementarity. First, we extract the Direction Difference Vector (DDV) for vein traits and the Intensity Orientation Vector (IOV) for background cues from the training samples. Then, we jointly learn projection functions that convert these two features into compact discriminative binary codes. Next, for a test finger vein image, DDV and IOV are extracted and jointly encoded into binary codes. Finally, the histogram-based vectors are constructed for finger vein representation and recognition.

4.1. Notation

We use lowercase letter to represent a scalar (e.g., x). Vector and matrix are indicated by lowercase boldface letter (e.g., $\mathbf{x}$) and uppercase boldface letter (e.g., $\mathbf{X}$), respectively. The $\|\cdot\|_F$ represents the Frobenius norm of a matrix, defined as $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j X_{ij}^2} = \sqrt{\text{tr}(\mathbf{X}^\top \mathbf{X})}$. The l_2-norm of a vector is calculated as $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$. The transposed matrix and inverse matrix denote as $\mathbf{X}^\top$ and $\mathbf{X}^{-1}$, respectively. The trace of a square matrix appears as tr$(\cdot)$, which is the sum of elements on the main diagonal.

4.2. Formulation of SSP-DBFL

Previous research has demonstrated that the binary features are robust to local noise and changes, and it is performance-friendly to biometric authentication [45, 1, 2, 46]. Hence, in this section, we propose our discrete binary learning method called SSP-DBFL that can exploit the consensus and complementarity between finger vein traits and background cues in a common Hamming space.

As aforementioned, DDVs and IOVs of all pixels for each training image are extracted first. Then, DDVs and IOVs of a training image are concatenated into their respective global training matrix. Let $\mathbf{X}^1 = \left[\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_N^1\right] \in R^{d_1 \times N}$ represent the global DDV matrix and $\mathbf{X}^2 = \left[\mathbf{x}_1^2, \mathbf{x}_2^2, \ldots, \mathbf{x}_N^2\right] \in R^{d_2 \times N}$ be the global IOV matrix, where N is the number of total pixels in the training set. SSP-DBFL is designed to jointly learn K hash functions to convert and quantize each $\mathbf{x}_n^v$ into a binary vector $\mathbf{b}_n^v = \left[b_{n1}^v, b_{n2}^v, \ldots, b_{nK}^v\right]$ for DDV ($v = 1$) and DDV ($v = 2$), respectively. In our paper, the kth binary code of the vth feature space b_{nk}^v of $\mathbf{x}_n^v$ can be obtained by the nonlinear sign function sgn$(\cdot)$ as

$$b_{nk}^v = \text{sgn}(\mathbf{w}_k^{v\top} \mathbf{x}_n^v), \tag{8}$$

where $\mathbf{w}_k^v \in R^{d_v}$ is the projection vector of the kth hash function for the vth feature space. As shown in Section 3, $d_1 = N_\theta = 12$ and $d_2 = M = 8$, respectively. The sign function sgn(h) equals to 1 if $h \geq 0$ and -1 otherwise.

Consider the basic assumption discussed in [51]: two types of features originating from the same position should share common underlying structure. In addition, in the common Hamming space, semantic correlation within one modality should be similar to that of the other modality, which contributes to good hash codes [52]. Inspired by above observations, we preserve semantic

similarity of different types of features in the common Hamming space to exploit the consensus and the shared underlying structure between them. Therefore, to make the learned binary features discriminative and comprehensive, we impose three criteria for the learned binary codes to meet:

1. Preserve semantic similarity of two types of features in the common Hamming space.
2. Maximize the inter-class distance and minimize the intra-class distance of binary codes.
3. Minimize the information loss between the learned binary codes and the original features.

Following these three criteria, we formulate our objective function as follows:

$$\begin{aligned}\min_{\mathbf{w}_k^v,\mathbf{b}_i^v|_{v=1,k=1}^{2,K}} \; & J = J_1 + \lambda_1 J_2 + \lambda_2 J_3 \\ & = \sum_{i=1}^{N}\Bigg\{\sum_{j=1}^{N}\|\mathbf{b}_i^{1\top}\mathbf{b}_j^1 - \mathbf{b}_i^{2\top}\mathbf{b}_j^2\|^2 \\ & -\lambda_1\sum_{v=1}^{2}\Bigg(\sum_{\mathbf{x}_j\in\Gamma(\mathbf{x}_i)}\|\mathbf{b}_i^v-\mathbf{b}_j^v\|^2-\sum_{\mathbf{x}_j\in\Lambda(\mathbf{x}_i)}\|\mathbf{b}_i^v-\mathbf{b}_j^v\|^2\Bigg) \\ & +\frac{\lambda_2}{2}\sum_{v=1}^{2}\|b_{ik}^v - \mathbf{w}_k^{v\top}\mathbf{x}_i^v\|^2\Bigg\},\end{aligned} \tag{9}$$

where $\Gamma(\mathbf{x}_i)$ is the subset whose samples come from the class different from that of $\mathbf{x}_i$; $\Lambda(\mathbf{x}_i)$ is the subset whose samples come from the same class with $\mathbf{x}_i$; and λ_1 and λ_2 are two parameters to balance terms. Following the first criterion, in the first term J_1, we calculate the difference between the two types of features in their high-level semantic correlations. As aforementioned, by minimizing J_1, we encourage different types of features to have similar semantic similarity in the common Hamming space, so as to exploit the consensus and underlying structure between them. Following the second criterion, we use J_2 to introduces supervised information into the objective function, so as to reduce the distances between same-class samples while increasing the distances between different-class samples for more discriminative representation. Following the third criterion, J_3 is adopted to reduce the information loss and thus can retain more energy from the original features.

Let $\mathbf{W}^v = [\mathbf{w}_1^v, \mathbf{w}_2^v, \ldots, \mathbf{w}_K^v] \in R^{d_v\times K}$ denote the projection matrix. The elements of binary code matrix $\mathbf{B}^v = [\mathbf{b}_1^v, \mathbf{b}_2^v, \ldots, \mathbf{b}_N^v] \in \{-1, 1\}^{K\times N}$ are obtained as

$$\mathbf{b}_n^v = \mathrm{sgn}(\mathbf{W}^{v\top}\mathbf{x}_n^v). \tag{10}$$

It follows that Eq. (9) can be re-written in a matrix form as

$$\begin{aligned}\min_{\mathbf{W}^v,\mathbf{B}^v|_{v=1}^2} \; & J = J_1 + \lambda_1 J_2 + \lambda_2 J_3 \\ & = \|\mathbf{B}^{1\top}\mathbf{B}^1 - \mathbf{B}^{2\top}\mathbf{B}^2\|_F^2 \\ & - \lambda_1 \sum_{v=1}^{2} \mathrm{tr}(\mathbf{B}^v \mathbf{S} \mathbf{B}^{v\top}) \\ & + \frac{\lambda_2}{2} \sum_{v=1}^{2} \|\mathbf{B}^v - \mathbf{W}^{v\top}\mathbf{X}^v\|_F^2 \\ \text{s.t.} \quad & \mathbf{W}^{v\top}\mathbf{W}^v = \mathbf{I}_{d_v \times d_v}, \quad \mathbf{B}^v\mathbf{1} = \mathbf{0},\end{aligned} \tag{11}$$

where $\mathbf{I}_{d_v \times d_v}$ represents the identity matrix with size $d_v \times d_v$; $\mathbf{S}_{i,j}$ equals 1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same label in the same position across different samples, equals -1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ come from different classes in the same position, and equals 0 otherwise. We further add two constraints. The first orthogonal constraint $\mathbf{W}^{v\top}\mathbf{W}^v = \mathbf{I}_{d_v \times d_v}$ ensures that different projection functions are independent to each other, such that no redundant information is remained between different projections. The second constraint $\mathbf{B}^v\mathbf{1} = \mathbf{0}$ makes sure that the occurrence times of 1 and -1 are equal in each bit, i.e., the variance of each feature bit is maximized. In other words, the second constraint is the pursuit of maximizing the information conveyed by each feature bit.

In previous binary feature learning methods [2, 45, 1], they did not directly optimize the binary code matrix $\mathbf{B}$, but chose to optimize the projection matrix $\mathbf{W}$ with relaxing $\mathbf{B}$ to its sign magnitude $\mathbf{W}^\top\mathbf{X}$. However, this relaxation leads to unignorable quantization errors [53], which degrades the discriminative capability of the representation. In our work, we handle this issue by optimizing in a discrete manner. For easy optimization, we introduce auxiliary variables $\mathbf{Q}^v \in \mathcal{R}^{d_v \times N}$ to substitute the counterpart binary matrix $\mathbf{B}^v$, with a penalty term added to keep their consistency. As a consequence, the objective function can be formulated as follows:

$$\begin{aligned}\min_{\mathbf{W}^v,\mathbf{B}^v,\mathbf{Q}^v|_{v=1}^2} \; & \|\mathbf{B}^{1\top}\mathbf{Q}^1 - \mathbf{B}^{2\top}\mathbf{Q}^2\|_F^2 - \lambda_1 \sum_{v=1}^{2} \mathrm{tr}(\mathbf{B}^v \mathbf{S} \mathbf{B}^{v\top}) \\ & + \frac{\lambda_2}{2} \sum_{v=1}^{2} \|\mathbf{B}^v - \mathbf{W}^{v\top}\mathbf{X}^v\|_F^2 + \gamma \sum_{v=1}^{2} \|\mathbf{B}^v - \mathbf{Q}^v\|_F^2 \\ \text{s.t.} \quad & \mathbf{W}^{v\top}\mathbf{W}^v = \mathbf{I}_{d_v \times d_v}, \quad \mathbf{B}^v\mathbf{1} = \mathbf{0}.\end{aligned} \tag{12}$$

4.3. Optimization Algorithm of SSP-DBFL

The objective function Eq. (12) is not convex due to the sign function sgn(·), resulting in an NP-hard problem. To solve such problems and avoid quantization errors accumulated during the optimization, we derive a discrete optimization algorithm to solve SSP-DBFL as follows.

Step 1: Update $\mathbf{W}^v$ with other variables fixed. The objective function with respect to $\mathbf{W}^v$ can be rewritten as

$$\begin{aligned} J(\mathbf{W}^v) &= \frac{\lambda_2}{2}\|\mathbf{B}^v - \mathbf{W}^{v\top}\mathbf{X}\|_F^2 \\ &= \frac{\lambda_2}{2}\mathrm{tr}(-2\mathbf{B}^{v\top}\mathbf{W}^{v\top}\mathbf{X}^v + \mathbf{X}^{v\top}\mathbf{W}^v\mathbf{W}^{v\top}\mathbf{X}^v) \\ &+ \mathbf{const} \\ \text{s.t.} \quad & \mathbf{W}^{v\top}\mathbf{W}^v = \mathbf{I}_{d_v\times d_v}, \end{aligned} \tag{13}$$

where the constant term **const** is independent of the variable to be optimized. The curvilinear search algorithm [54] is used to solve $\mathbf{W}^v$ with the orthogonal constraint.

Step 2: Update $\mathbf{Q}^v$ with other variables fixed. Since there are two types of features involved in our method (i.e., DDV and IOV), we denote one of them by the superscript v, and the other by $\bar{v}$. Thus, we need to minimize the following formula:

$$\begin{aligned} J(\mathbf{Q}^v) =& \|\mathbf{B}^{v\top}\mathbf{Q}^v - \mathbf{B}^{\bar{v}\top}\mathbf{Q}^{\bar{v}}\|_F^2 + \frac{\gamma}{2}\|\mathbf{B}^v - \mathbf{Q}^v\|_F^2 \\ &= \mathrm{tr}(\mathbf{Q}^{v\top}\mathbf{B}^v\mathbf{B}^{v\top}\mathbf{Q}^v - \gamma\mathbf{B}^{v\top}\mathbf{Q}^v + \frac{\gamma}{2}\mathbf{Q}^{v\top}\mathbf{Q}^v) \\ &- 2\mathrm{tr}(\mathbf{Q}^{v\top}\mathbf{B}^v\mathbf{B}^{\bar{v}\top}\mathbf{Q}^{\bar{v}}) + \mathbf{const}. \end{aligned} \tag{14}$$

The closed-form solution of $\mathbf{Q}^v$ can be obtained by letting the derivative of $J(\mathbf{Q}^v)$ with respect to $\mathbf{Q}^v$ to zero:

$$\frac{\partial J(\mathbf{Q}^v)}{\partial \mathbf{Q}^v} = 2\mathbf{B}^v(\mathbf{B}^{v\top}\mathbf{Q}^v - \mathbf{B}^{\bar{v}\top}\mathbf{Q}^{\bar{v}}) + \gamma(\mathbf{Q}^v - \mathbf{B}^v) = 0. \tag{15}$$

As a result, we can derive the close-form solution of $\mathbf{Q}^v$ as

$$\mathbf{Q}^v = (2\mathbf{B}^v\mathbf{B}^{v\top} + \gamma\mathbf{I})^{-1}(2\mathbf{B}^v\mathbf{B}^{\bar{v}\top}\mathbf{Q}^{\bar{v}} + \gamma\mathbf{B}^v). \tag{16}$$

Step 3: Update $\mathbf{B}^v$ with other variables fixed. The optimization formula with respect to $\mathbf{B}^v$ can be derived as

$$\begin{aligned} J(\mathbf{B}^v) &= \|\mathbf{B}^{v\top}\mathbf{Q}^v - \mathbf{B}^{\bar{v}\top}\mathbf{Q}^{\bar{v}}\|_F^2 - \lambda_1\mathrm{tr}(\mathbf{B}^{v\top}\mathbf{S}\mathbf{B}^v) \\ &+ \frac{\lambda_2}{2}\|\mathbf{B}^v - \mathbf{W}^{v\top}\mathbf{X}^v\|_F^2 + \frac{\gamma}{2}\|\mathbf{B}^v - \mathbf{Q}^v\|_F^2 \\ \text{s.t.} \quad & \mathbf{B}^v\mathbf{1} = \mathbf{0}. \end{aligned} \tag{17}$$

Eq. (17) is equivalent to

$$J(\mathbf{B}^v) = \mathrm{tr}(\mathbf{B}^{v\top}\mathbf{P}^v\mathbf{B}^v + \mu\mathbf{B}^{v\top}\mathbf{B}^v\hat{\mathbf{1}} + \mathbf{B}^{v\top}\mathbf{H}^v), \tag{18}$$

where $\mathbf{P}^v = \mathbf{Q}^v\mathbf{Q}^{v\top} - \lambda_1\mathbf{S}$ and $\hat{\mathbf{1}} = \mathbf{1}\mathbf{1}^\top$; $\mathbf{H}^v = -2\mathbf{Q}^v\mathbf{Q}^{\bar{v}\top}\mathbf{B}^{\bar{v}} - \lambda_2\mathbf{W}^v\mathbf{X}^v - \gamma\mathbf{Q}^v$. To obtain the closed-form solution of $\mathbf{B}^v$, a binary variable $\mathbf{D}^v \in \{-1, 1\}^{d_v\times N}$ is introduced to substitute one of the $\mathbf{B}^v$ in quadratic terms with the constraint $\mathbf{D}^v = \mathbf{B}^v$. In this case, we can reformulate Eq. (18) into the following equation [53]:

$$\begin{aligned} J(\mathbf{B}^v) = &\ \mathrm{tr}(\mathbf{B}^{v\top}\mathbf{P}^v\mathbf{D}^v + \mu\mathbf{B}^{v\top}\mathbf{D}^v\hat{\mathbf{1}} + \mathbf{B}^{v\top}\mathbf{H}^v) \\ &+ \frac{\rho}{2}\|\mathbf{B}^v - \mathbf{D}^v + \frac{\mathbf{G}^v}{\rho}\|_F^2, \end{aligned} \tag{19}$$

where $\mathbf{G}^v \in \mathcal{R}^{d_v\times N}$ measures the difference between $\mathbf{B}^v$ and $\mathbf{D}^v$. As a consequence, we can obtain the closed-form solution of $\mathbf{B}^v$ in a discrete manner, which is calculated as

$$\mathbf{B}^v = -\mathrm{sgn}(\mathbf{P}^v\mathbf{D}^v + \mu\mathbf{D}^v\hat{\mathbf{1}} + \mathbf{H}^v - \rho\mathbf{D}^v + \mathbf{G}^v). \tag{20}$$

Step 4: Update $\mathbf{D}^v$ with other variables fixed. The optimization formula with respect to $\mathbf{D}^v$ can be derived as

$$J(\mathbf{D}^v) = \mathrm{tr}((\mathbf{B}^{v\top}\mathbf{P}^v + \mu\hat{\mathbf{1}}\mathbf{B}^{v^\top} + \rho\mathbf{B}^v + \mathbf{G}^v)\mathbf{D}^v). \tag{21}$$

Similar to Step 3, the closed form solution of $\mathbf{D}^v$ can be obtained as

$$\mathbf{D}^v = -\mathrm{sgn}(\mathbf{B}^{v\top}\mathbf{P}^v + \mu\hat{\mathbf{1}}\mathbf{B}^{v^\top} + \rho\mathbf{B}^v + \mathbf{G}^v). \tag{22}$$

Step 5: Update $\mathbf{G}^v$ and ρ with other variables fixed. The optimization rules of $\mathbf{G}^v$ and ρ are

$$\mathbf{G}^v = \mathbf{G}^v + \rho(\mathbf{B}^v - \mathbf{Z}^v), \qquad \rho = \eta\rho. \tag{23}$$

The above optimization procedures are iteratively repeated until convergence. We summarize the main optimization steps of our SSP-DBFL in Algorithm 1.

4.4. Usage of SSP-DBFL for Finger Vein Recognition

To deal with translations and misalignments in finger vein images, we compress the learned binary feature map of each non-overlapping region into a local histogram-based descriptor and then concatenate them as the final representation. Specifically, after obtaining the projection matrices $\mathbf{W}^v$ through SSP-DBFL, the DDV and IOV of each finger vein image are first converted

Algorithm 1 Major Steps of Semantic Similarity Preserved Discrete Binary Feature Learning.

Input: Training sets $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \ldots, \mathbf{x}_N^v]\,|_{v=1}^2$; Balance parameters λ_1 and λ_2; Penalty parameters γ and μ; Maximum iterations T.

Output: Projection matrix $\mathbf{W}^v$.

Initialize: Random initialize $\mathbf{Q}^v$, $\mathbf{D}^v$; $\mathbf{W}^v$ is initialized by calculating the K eigenvectors according to the top K eigenvalues of $\mathbf{X}^{v\top}\mathbf{X}^v$; $\mathbf{B}^v = \mathbf{W}^{v\top}\mathbf{X}^v$; $\rho = 10^{-3}$; $\eta = 2$.
repeat
  Update $\mathbf{W}^v$ via Eq. (13).
  Update $\mathbf{Q}^v$ via Eq. (16).
  Update $\mathbf{B}^v$ via Eq. (20).
  Update $\mathbf{D}^v$ via Eq. (22).
  Update $\mathbf{G}^v$ and ρ via Eq. (23).
until reach T iterations
return $\mathbf{W}^v$.

into binary feature maps, respectively. Then, we divide each binary feature map into several non-overlapped blocks with size 16×16. For each block, K-means are applied on the training set to learn block-based codebooks [45], and the learned binary feature map of this block is pooled as a local histogram-based descriptor. Finally, we concatenate them to form the final representation $\mathbf{f}$ for finger vein recognition.

For two SSP-DBFL-based finger vein descriptors, the similarity can be measured by the Chi-square distance [55]. Let $\mathbf{f}_i$ and $\mathbf{f}_j$ denote the SSP-DBFL-based descriptors of gallery sample and probe sample, respectively. The Chi-square distance between them is calculated as

$$\mathrm{d}(\mathbf{f}_i, \mathbf{f}_j) = \sum_{l=1}^{N_b} \frac{(\mathbf{f}_{il} - \mathbf{f}_{jl})^2}{\mathbf{f}_{il} + \mathbf{f}_{jl}}, \tag{24}$$

where the subscript l denotes the lth bin in the corresponding feature descriptor; N_b is the number of bins in $\mathbf{f}_i$ and $\mathbf{f}_j$. The smaller the distance, the more likely it is that the two images are from the same category. Fig. 4 illustrates the flowchart of the SSP-DBFL-based finger vein representation and matching.

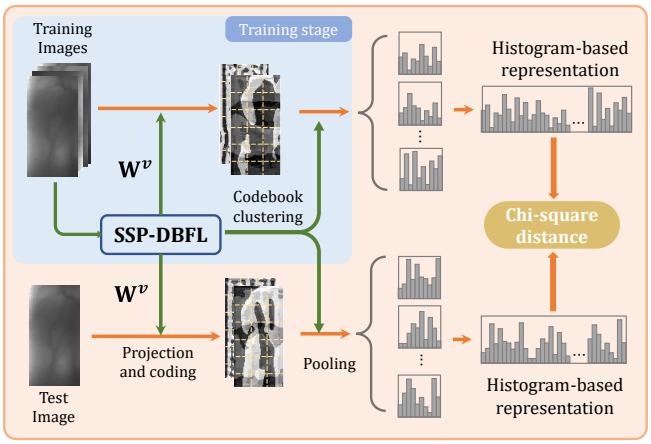


Figure 4: An illustration of SSP-DBFL-based finger vein recognition.

5. Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed IOV and SSP-DBFL.

5.1. Databases

Five popular finger vein databases were adopted in our experiments.

THU-FVFDT2 [56]: 1,220 finger vein images from 610 different fingers. Each volunteer was asked to provide the index and middle fingers of both hands, and two images were captured of each finger. The whole collection was divided into two sessions, whose duration is from three days to one week. ROI images with size 200×100 are also accessible.

SDUMLA [57]: 3,816 images from 106 subjects. For each subject, finger vein images were captured from the index, middle, and ring fingers of both hands. Every volunteer contributed six finger vein images for each finger. Hence, there are 636 classes in this database with six images in each class. The size of raw finger vein images is 320×240, and ROI images are not provided officially.

POLY [18]: 2,520 finger vein images collected in two sessions. There were 105 volunteers participating in two-session collections. All of them were asked to provide the index and middle fingers of both hands and contributed six finger vein images for one finger in each session. The

average duration of two sessions is about 66.8 days. The resolution of raw finger vein images is 513×256. No ROI images are provided.

USM [58]: 5,904 finger vein images from 123 volunteers. The index and middle fingers of both hands were used in two sessions about two weeks apart. In each session, each finger was captured six times, resulting in six raw finger vein images with size 640×480. ROI images are provided.

MMCBNU6000 [59]: 6,000 finger vein images from 100 volunteers. For each volunteer, the index, middle and ring fingers of the left and right hands were provided. Each finger was captured ten times. As a consequence, this database has 600 classes with ten samples per class. The authors released raw finger vein images with size 640×480 and ROI images with size 128×60.

We summarize the basic information of these five databases in Table 1. In the following experiments, we extract ROIs for SDUMLA and POLY databases according to [18]. All ROIs are resized into 144×64.

Table 1: Basic information about THU-FVFDT2, SDUMLA, POLY, USM, MMCBNU6000 databases

Database	#Total Images	#Fingers	#Samples per finger	#Sessions
THU-FVFDT2	1,260	610	2	2
SDUMLA	3,816	636	6	1
POLY	2,520	210	12	2
USM	5,904	492	12	2
MMCBNU6000	6,000	600	10	1

5.2. Ablation Studies

In this section, we evaluate the effectiveness of the proposed IOV. LCMFC [46] is implemented for comparison, because it also extracts bifeatures for biometric representation. Since direction information is important for finger vein recognition, we fix one feature as DDV, and replace the other feature with IOV, PDV, or Texture Feature Vector (TFV). PDV was first proposed by Lu et al. [45], as the sum of the differences between a pixel and its neighbors within R_{PDV} ($R_{PDV} = 3$ in this work). TFV [46] was used to extract texture features. Specifically,

LCMFC aims to jointly learn a projection matrix $\mathbf{P}$ and two coefficients α_1 and α_2 from bifeatures. The learned coefficients α_1 and α_2 represent the relative contributions of the two features to the final representation, respectively. The experiments were conducted on SDUMLA, POLY, and MMCBNU6000, and the resulting recognition performances are listed in Table 2.

Table 2: The identification performance (%) of LCMFC and SSP-DBFL with different auxiliary features.

Methods	Features	SDUMLA	POLY	MMCBNU6000
LCMFC	TFV	96.16	91.03	99.17
	IOV	**98.53**	**98.68**	**99.77**
SSP-DBFL	PDV	97.81	99.40	99.35
	IOV	**99.07**	**99.67**	**99.83**

From the results, we can observe the clear improvement of accuracy when PDV or TFV are replaced by IOV, for both LCMFC and SSP-DBFL on various databases, which demonstrates the effectiveness of IOV. Moreover, we can report that the coefficient (α_2) learned for IOV is always larger than that learned for TFV (while in both case α_1 denotes the importance of DDV). This suggests that IOV provides more discriminative information than TFV.

5.3. Comparison with State-of-the-arts

5.3.1. Finger Vein Identification

In this section, we conduct experiments to verify the effectiveness of our methods for finger vein identification. Finger vein identification aims to classify a query finger vein image to a category with a specific label in the training set. The Accuracy Rate (ARR) is used as the measure to evaluate the identification performance.

For the proposed SSP-DBFL method, we first learn the projection matrices $\mathbf{W^1}$ and $\mathbf{W^2}$ for DDV and IOV, respectively. Then both descriptors DDV and IOV are projected and encoded into K-bit binary codes for finger vein representation. In addition, we also test our SSP-DBFL with different scales of IOV (set to 2 and 4) to verify the impact of the scale on identification performance. The nearest neighbor classifier is adopted to assign labels to the probe finger vein images based on the Chi-square distance metric.

To comprehensively evaluate the effectiveness of our SSP-DBFL, we compare it with various methods including structure based method (LMC [7]), local feature based methods (Gabor [18],

Table 3: The average identification performance (accuracy rate±standard errors) of different methods on five databases. The best is in bold; the second best is underlined; the third best is in italics.

Methods	THU-FVFDT2	SDUMLA	POLY	USM	MMCBNU6000
LMC [7]	96.56±0.0000	97.22±0.0703	97.30±0.0000	97.19±0.0000	87.17±1.2866
Gabor [18]	94.92±0.0000	94.50±0.1764	96.75±0.0000	96.61±0.0000	97.83±0.0491
LDC [15]	87.05±0.0000	**99.21±0.0289**	98.33±0.0000	98.81±0.0000	99.10±0.0131
PWBDC [60]	97.87±0.0000	92.82±0.2083	94.44±0.0000	95.39±0.0000	99.07±0.0192
Hong et al. [19]	64.13±2.0623	74.35±1.8288	88.26±2.0088	87.94±1.0662	96.77±0.7131
FVCAE [21]	76.89±4.8792	93.23±1.0367	95.48±1.0456	90.36±2.3421	96.07±0.3078
ResNet18 [49]	70.90±3.0364	89.87±1.3526	82.60±0.0039	90.86±2.0123	99.44±0.0009
ResNet50 [49]	71.67±2.0371	90.46±1.7245	87.69±0.0021	86.17±1.0268	98.64±0.4122
MRFBCNN [27]	97.92±0.0161	98.16±0.2713	96.93±0.3107	98.05±0.2172	99.13±0.0007
ArcVein [26]	82.68±0.0099	98.85±0.0029	96.61±0.0314	95.02±0.0075	99.23±0.0024
WSRC [30]	98.36±0.0000	79.92±0.2491	74.37±0.0000	89.33±0.0000	98.27±0.1745
ESRC [61]	98.36±0.0000	80.41±0.2127	76.11±0.0000	89.57±0.0000	98.43±0.2084
CA-LBFL [1]	*99.18±0.0443*	97.46±0.5394	*99.14±0.0091*	99.02±0.0937	*99.64±0.0213*
DDBPD [2]	96.07±0.0983	96.00±0.7607	95.71±0.9167	93.17±1.0671	97.80±0.1291
LCMFC [46]	93.47±0.5956	96.16±0.6217	91.03±0.8983	*99.40±0.0978*	99.17±0.0678
SSP-DBFL$_{\text{Scale=2}}$	99.53±0.0072	*98.31±0.4948*	99.39±0.0311	99.40±0.0676	99.83±0.0039
SSP-DBFL$_{\text{Scale=4}}$	**99.84±0.0006**	99.07±0.1421	**99.67±0.0037**	**99.70±0.0095**	**99.83±0.0024**

LDC [15], PWBDC [60]), representative deep learning based methods (Hong et al. [19], FVCAE [21], ResNet18/50 [49], ArcVein [26], MRFBCNN [27]), and subspace learning based methods (WSRC [30], ESRC [61], CA-LBFL [1], DDBPD [2], LCMFC [46]). Since our SSP-DBFL follows the same framework and PDV is not a face-specific descriptor, the works (i.e., LCMFC, CA-LBFL, DDBPD) and PDV can be directly applied to finger vein recognition for comparison. The block size of LCMFC, CA-LBFL, and DDBPD were all set to $[16, 16]$. The parameters of these three methods were carefully tuned to achieve the best performance. For FVCAE, we trained it from scratch and tuned it with the learning rate 0.001 and batch size 256. For Hong et al. and ResNet18/50, we first pretrained them on ImageNet ILSVRC [62] and then finetuned them on each database. The cosine similarity is adopted to measure the similarity between two features for Hong et al., ResNet18/50, ArcVein and MRFBCNN. As with FVCAE,

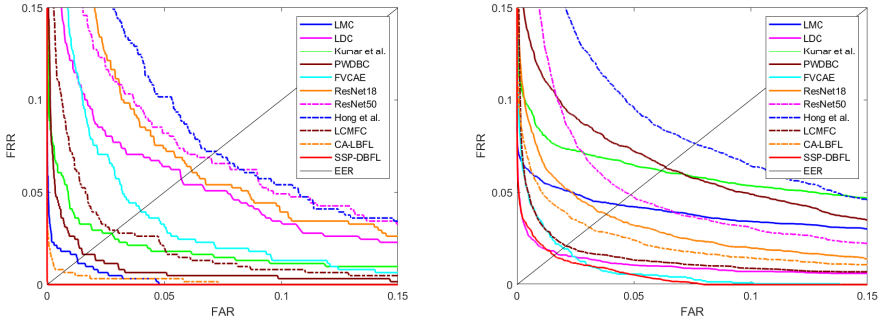
PolySVM is used for classification, and the feature is the output of FV-encoder. In addition, we set the optimal parameters for each database according to Section 5.4.3.

All the methods were repeated 10 times to calculate average ARRs and standard errors. Table 3 lists the experimental results for all methods on the five public finger vein databases. Note that, since all images in the SDUMLA and MMCBNU6000 databases are captured in one session, the training and test samples are randomly selected among all samples with no overlap, while the training samples in other databases are the images from session 1 and the test samples are from session 2.

It can be seen from the results that the performance of most deep learning based methods has a big drop on the THU-FVFDT2 database due to insufficient labeled samples in the training set. Among deep learning based methods, MRFBCNN and ArcVein obtain superior results to others. In addition, compared with the state-of-the-art binary feature learning methods (i.e., DDBPD, CA-LBFL), which only consider a single-type feature of finger vein images, our SSP-DBFL offers superior identification performance to them. LCMFC learns a projection function from two-type features (i.e., direction feature and texture feature). However, the texture features do not provide sufficient discriminative information for finger vein representation (as illustrated in Section 5.2), leading LCMFC to inferior results to our SSP-DBFL. In addition, the identification performance of SSP-DBFL becomes better when the scale of IOV is larger so that an IOV element can incorporate more background cues. In short, the experimental results demonstrate the superiority of SSP-DBFL over the state-of-the-arts for finger vein identification.

5.3.2. Finger Vein Verification

In this section, we evaluate the effectiveness of our SSP-DBFL for finger vein verification. Finger vein verification intends to compare probe samples and gallery samples one by one. There are two types of matches in the verification protocol, genuine and imposter. Genuine match means that the two compared finger vein images come from the same finger, and imposter match means that they come from different fingers. The settings of all the methods involved here are the same as those in Section 5.3.1. We calculate the False Acceptance Rate (FAR) and the False Reject Rate (FRR) and then plot the Receiver Operating Characteristic (ROC) curves (FRR vs FAR) of different methods on the five databases in Fig. 5. From Fig. 5, an observation is that SSP-DBFL consistently obtains the lowest EER, which demonstrates the effectiveness of our methods.



(a) THU-FVFDT2

(b) SDUMLA

(c) POLY

(d) USM

(e) MMCBNU6000

Figure 5: ROC (FRR vs FAR) curves of different methods on the five databases.

5.4. Further Experiments and Discussion

5.4.1. Performance of IOV under Rotation, Scaling and Translation (RST)

In this subsection, we further investigate whether IOV is RST invariant.

Database Construction: As IOV is designed for the extraction of background cues in finger vein images, we construct an experiment database based on THU-FVFDT2. Overexposure and underexposure finger vein images are removed first. Then, we manually select 200 images with clear and rich veins from the remaining images. This operation ensures that the processed images (i.e., after rotation, translation, and scaling) are different from the original images, even under small translations. To obtain probe images with rotation, translation, and scaling, the three operations are applied to the original images separately, and then we crop all processed image at the center with size $[64, 64]$. This avoids setting too many pixels outside the processed image to a constant (such as 0 or 255). Similarly, the original images are also cropped into patches of the same size at the center, serving as the gallery set. Some gallery images and probe images are shown in Fig. 6.

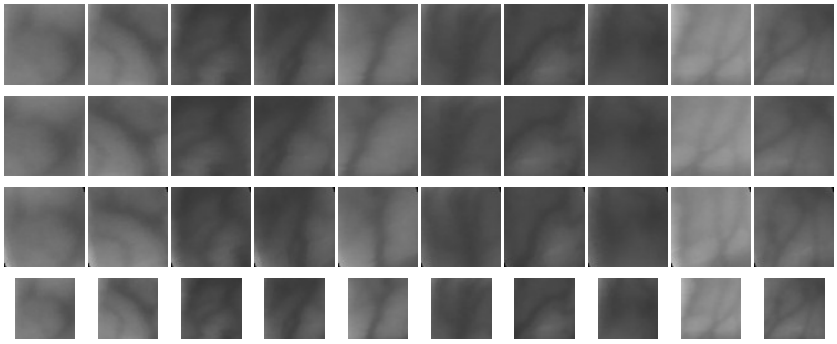
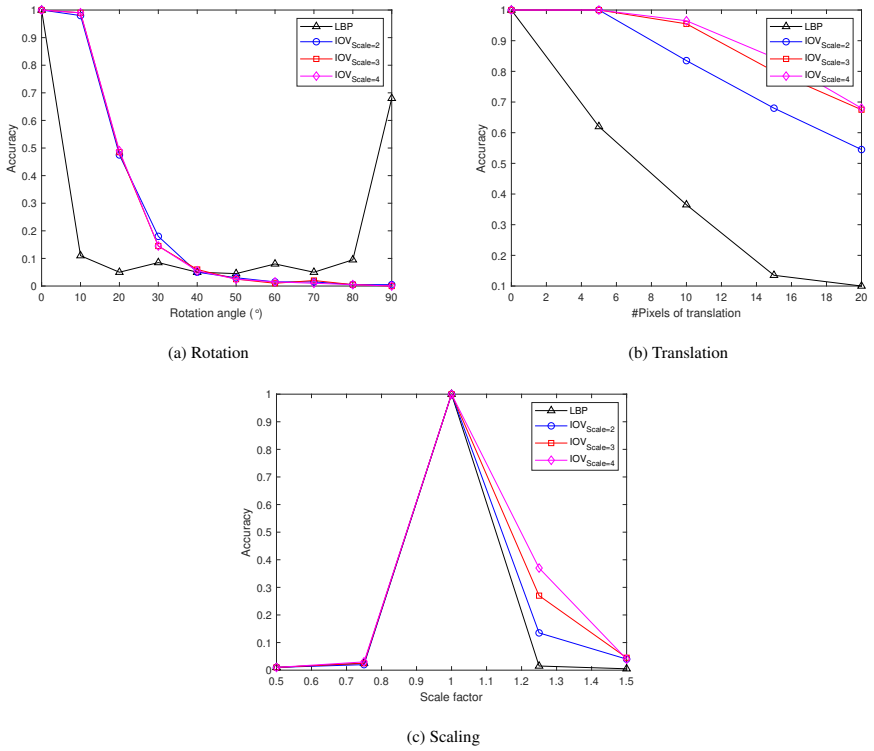


Figure 6: Illustration of gallery images and probe images with rotation, translation, and scaling. Gallery images are in the first row. Translation, rotation, and scaling images are in the second, third, and fourth rows, respectively, as probe images.

Experiments: To represent a finger vein image with IOV, we calculate the histogram of each direction and concatenate them to form the final representation. The nearest neighbor classifier is adopted for classification based on the Euclidean distance. For comparison, LBP is also applied in the same context. For clear demonstration, the classification accuracy under different rotation angles, translation distances (#pixels), and scale factors is plotted in Fig. 7.



(a) Rotation

(b) Translation

(c) Scaling

Figure 7: Classification accuracy under rotation, translation, and scaling.

From Fig. 7a, one can see that using IOV has only a slight performance loss when the rotation angle θ is less than 10°. However, the performance degrades dramatically when θ is larger than 10°. The IOV and LBP obtain similar results when $\theta = 40°$. When $\theta > 40°$, IOV does not perform as well as LBP and gradually loses the discriminative ability. In addition, we can also find that the IOV scale does not have a significant effect on results. Fig. 7b indicates that IOV is robust to translation to some extent, and the robustness becomes better with increased IOV scale. Obviously, IOV performs better than LBP under translation. From Fig. 7c, we can find that IOV does not have the scale-invariant property. When the scale factor is less than 1, neither IOV nor LBP can work effectively. However, However, IOV performs better than LBP when the scale factor is larger than 1, and the performance benefits from a large IOV scale.

5.4.2. Convergence Analysis

In this section, we experimentally analyze the convergence of our proposed SSP-DBFL. We calculate the objective function in Eq. (12) and plot its value versus the number of iterations on the five databases in Fig. 8. From Fig. 8, we can observe that the objective function value decreases dramatically after the first few iterations and then tends to be steady. In general, SSP-DBFL can converge after about 10 iterations. Hence, to ensure the best recognition performance of our method, the iteration numbers are set to 20 for all databases.

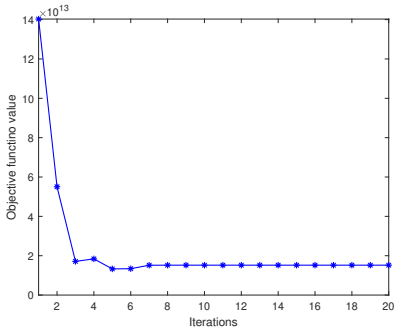
5.4.3. Parameter Analysis

There are four hyper-parameters (λ_1, λ_2, γ and μ) in the SSP-DBFL objective function to balance terms. As usual, the optimal parameters vary with different databases, so do their sensitivities. Hence, in this section, to analyze their effects, we evaluate the performance of SSP-DBFL with different parameters on the five databases. To balance the performance and running time, the scale of IOV is set to 2 for parameter analysis. All the parameters vary within a discrete set $[10^{-6}, 10^{6}]$ with common ratio 10. The variability of identification accuracy of SSP-DBFL with different parameters on each database is shown in Fig. 9.

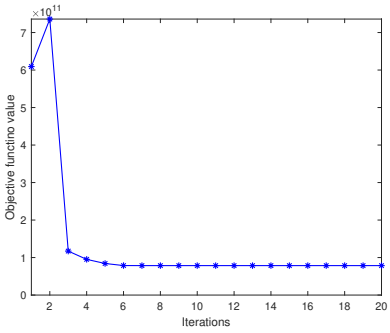
Firstly, to analyze the influence of μ, we set $\lambda_1 = \lambda_2 = 0.1$ and $\gamma = 1$ to evaluate the accuracy versus μ. The effect of different μ is shown in the left-hand column of Fig. 9. We can see that the performance of SSP-DBFL on all databases is relatively stable and satisfactory when μ is within the range of $\left[10^{-6}, 1\right]$, while it decreases in most cases when μ is larger than 10^2.

Secondly, we study the sensitivities of λ_1 and λ_2 by observing the accuracy versus different combinations of λ_1 and λ_2 with optimal μ and $\gamma = 1$. The results are shown in the middle column of Fig. 9. From the results, one can see that a too small λ_2 or a too large λ_1 will degrade the identification performance. We also find that the influence of λ_2 on SSP-DBFL varies with databases. For example, the suitable range of λ_2 is $[10^3, 10^6]$ for THU-FVFDT2 database, but for other databases, λ_2 varying within $[10^{-2}, 10^6]$ leads to only slightly change in the performance. In addition, the reasonable range of λ_1 is located in $[10^{-5}, 10]$ for stable performance.

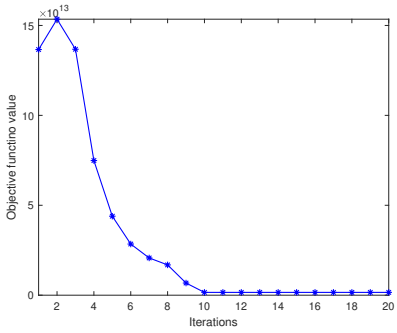
Lastly, the optimal λ_1, λ_2 and μ were fixed to observe the effect of γ. We plot the results in the right-hand column of Fig. 9. One observation is that a small γ poses bad impact on the performance across five databases. But the tolerance for small γ is different for different databases. For example, the performance on MMCBNU6000 is relatively stable until γ becomes smaller than 0.1, while the performance on THU-FVFDT2 decreases dramatically when $\gamma < 10^3$.



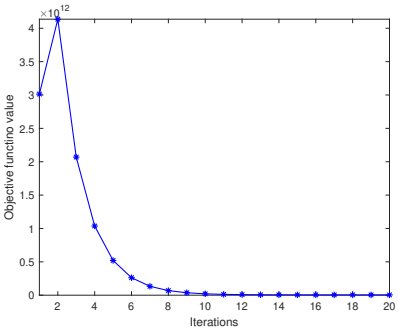
(a) THU-FVFDT2



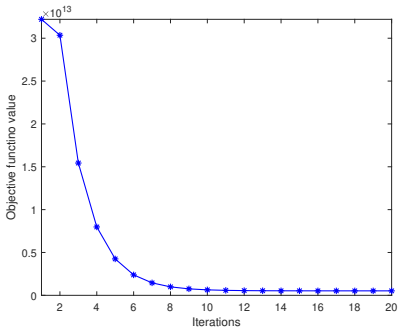
(b) SDUMLA



(c) POLY



(d) USM



(e) MMCBNU6000

Figure 8: Objective function value of SSP-DBFL vs the number of iterations on the five databases.

In addition, the γ that is too large can also impair the performance. Hence, a suitable choice of γ is around 10^3 for our SSP-DBFL to obtain good results on all databases.

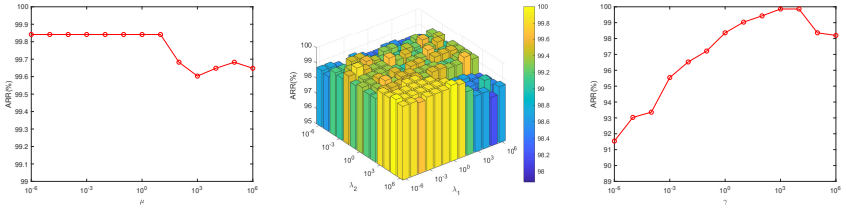
Furthermore, SSP-DBFL aims to learn binary codes with K bits from the two features. To test the effect of the code length, we perform SSP-DBFL on the five databases with different values of K. The relations between accuracy and K are shown in Fig. 10. It can be observed that the accuracy increases with an increase of K, the reason of which is that the binary codes convey less discriminative information with a small K. Since K is smaller than the dimensions of IOV and DDV, we use the largest value of K (8) in this work.

5.4.4. Running Time Analysis

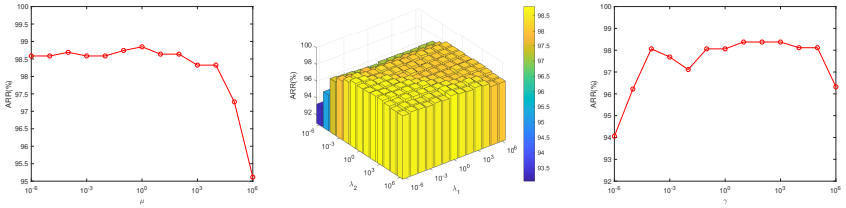
In this section, we evaluate the running time of our SSP-DBFL and other finger vein recognition algorithms. Since SSP-DBFL can be pretrained offline like other learning based methods, the running time of feature extraction and feature matching process is computed. We choose some conventional methods and some deep learning based methods for a comparison on the SDUMLA and POLY databases. The average running time of feature extraction and matching is reported in Table 4. From the results, we can see that the structure based methods like LMC take longer in the matching stage, because these algorithms need to generate more templates to reduce the interference of translations and rotations. In addition, the deep learning based methods such as FVCAE and ResNet 50 spend more time on the inference process than the subspace learning based methods. Our methods take about 84.70ms and 62.71ms for feature extraction on SDUMLA and POLY, respectively. The time consumption of feature extraction of SSP-DBFL is more than that of other subspace learning based methods, because our methods extract two types of features before mapping. However, more importantly, our method has the fastest feature matching speed on the SDUMLA database and ranks second on the POLY database.

6. Conclusion

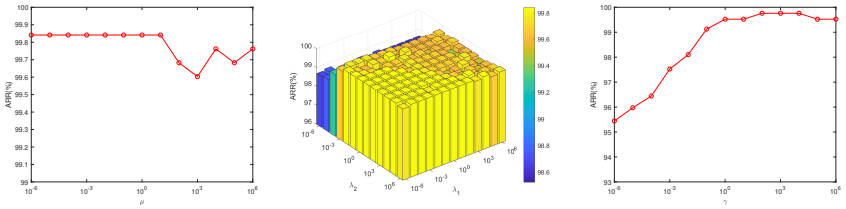
In this paper, we leverage the background information of finger vein images and propose a new descriptor named IOV to describe the background intensity variations for the first time. The scaleable descriptor IOV can thoroughly describe intensity variations and make the finger vein representation more informative and comprehensive. In addition, we propose a discrete binary feature learning method named SSP-DBFL for finger vein recognition, which jointly maps DDV



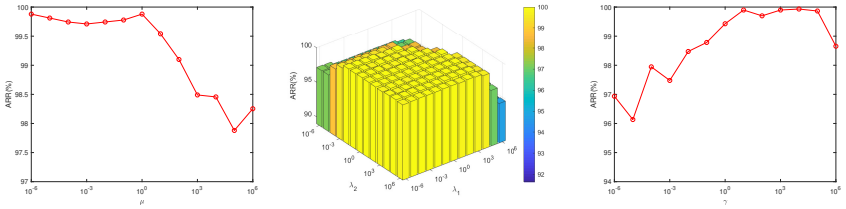
(a) THU-FVFDT2



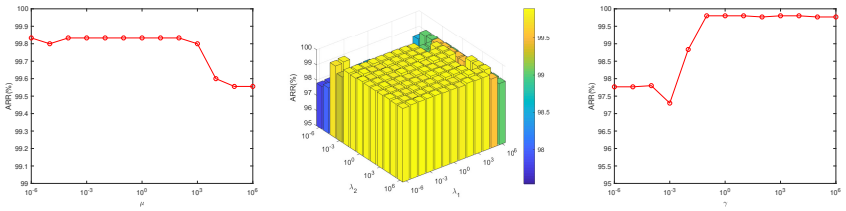
(b) SDUMLA



(c) POLY



(d) USM



(e) MMCBNU6000

Figure 9: Finger vein identification accuracy (ARR (%)) vs parameters λ_1, λ_2, μ, and γ, respectively, on the five databases.

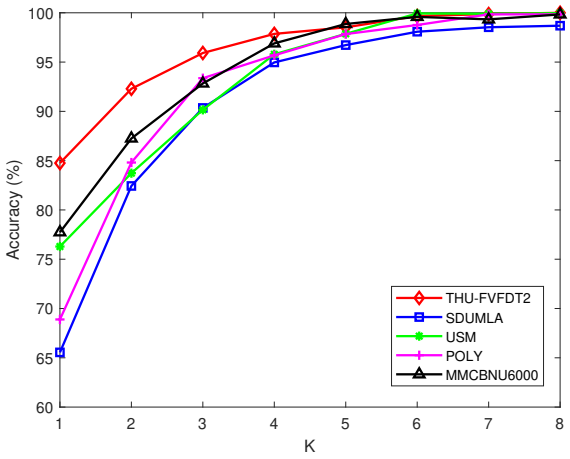


Figure 10: The accuracy of SSP-DBFL vs code length K on the five databases.

Table 4: Running time (ms) of feature extraction and matching for each method on the SDUMLA, POLY databases.

Methods	SDUMLA		POLY	
	Feature extraction	Matching	Feature extraction	Matching
LMC	232.75	898.13	204.39	562.08
Gabor	58.70	347.11	54.18	341.59
LDC	114.98	68.93	121.03	71.26
PWDBC	85.58	55.14	85.07	55.56
Hong et al.	47.57	3.59	51.30	1.74
FVCAE	214.83	178.04	205.22	135.71
ResNet50	146.53	3.09	97.82	1.84
CA-LBFL	67.86	1.70	37.93	2.48
DDBPD	**33.57**	4.50	**37.39**	4.17
LCMFC	79.50	1.47	50.63	**1.45**
SSP-DBFL	84.70	**1.14**	62.71	1.74

for vein traits and IOV for background cues to compact and discriminative codes. To exploit the consensus between these two types of features in the same region, SSP-DBFL preserves the high-level semantic similarity of the two features in the common Hamming space. Extensive experimental results demonstrate the effectiveness of the IOV descriptor and the SSP-DBFL method. Compared with other finger vein recognition methods, our approach obtains better or competitive recognition performance on five widely used finger vein databases. However, the limitation of our system is that the extraction of DDV and IOV slows down the feature extraction process. In future work, we will focus on developing a new descriptor that contains both direction information and background cues with a fast feature extraction process and exploit our system for wider applications in other biometric authentications.

Acknowledgment

This work was partly supported by the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen (No.JCYJ20170817161845824).

References

[1] Y. Duan, J. Lu, J. Feng, J. Zhou, Context-aware local binary feature learning for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (5) (2018) 1139–1153.

[2] L. Fei, B. Zhang, Y. Xu, Z. Guo, J. Wen, W. Jia, Learning discriminant direction binary palmprint descriptor, IEEE Transactions on Image Processing 28 (8) (2019) 3808–3820.

[3] S. Li, B. Zhang, An adaptive discriminant and sparsity feature descriptor for finger vein recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 2140–2144.

[4] M. Kono, A new method for the identification of individuals by using of vein pattern matching of a finger, in: Proc. Fifth Symposium on Pattern Measurement, Yamaguchi, Japan, 2000, pp. 9–12.

[5] K. Shaheed, H. Liu, G. Yang, I. Qureshi, J. Gou, Y. Yin, A systematic review of finger vein recognition techniques, Information 9 (9) (2018) 213.

[6] M. Drahanskỳ, O. Kanich, E. Březinová, K. Shinoda, Experiments with optical properties of skin on fingers, International Journal of Optics and Applications 6 (2) (2016) 37–46.

[7] N. Miura, A. Nagasaka, T. Miyatake, Extraction of finger-vein patterns using maximum curvature points in image profiles, IEICE Transactions on Information and Systems E90-D (8) (2007) 1185–1194.

[8] M. A. Syarif, T. S. Ong, A. B. J. Teoh, C. Tee, Enhanced maximum curvature descriptors for finger vein verification, Multimedia Tools and Applications 76 (5) (2017) 6859–6887.

[9] J. Yang, Y. Shi, G. Jia, Finger-vein image matching based on adaptive curve transformation, Pattern Recognition 66 (2017) 34–43.

[10] L. Yang, G. Yang, Y. Yin, X. Xi, Finger vein recognition with anatomy structure analysis, IEEE Transactions on Circuits and Systems for Video Technology 28 (8) (2018) 1892–1905.

[11] P. Prabhakar, T. Thomas, Finger vein identification based on minutiae feature extraction with spurious minutiae removal, in: International Conference on Advances in Computing and Communications, IEEE, 2013.

[12] X. Meng, J. Zheng, X. Xi, Q. Zhang, Y. Yin, Finger vein recognition based on zone-based minutia matching, Neurocomputing 423 (2021) 110–123.

[13] L. Dong, G. Yang, Y. Yin, F. Liu, X. Xi, Finger vein verification based on a personalized best patches map, in: IEEE International Joint Conference on Biometrics, IEEE, 2014, pp. 1–8.

[14] C. Liu, Y.-H. Kim, An efficient finger-vein extraction algorithm based on random forest regression with efficient local binary patterns, in: IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3141–3145.

[15] X. Meng, G. Yang, Y. Yin, R. Xiao, Finger vein recognition based on local directional code, Sensors 12 (11) (2012) 14937–14952.

[16] B. A. Rosdi, C. W. Shing, S. A. Suandi, Finger vein recognition using local line binary pattern, Sensors 11 (12) (2011) 11357–11371.

[17] H. Liu, L. Song, G. Yang, L. Yang, Y. Yin, Customized local line binary pattern method for finger vein recognition, in: Biometric Recognition, Springer International Publishing, 2017, pp. 314–323.

[18] A. Kumar, Y. Zhou, Human identification using finger images, IEEE Transactions on Image Processing 21 (4) (2012) 2228–2244.

[19] H. Hong, M. Lee, K. Park, Convolutional neural network-based finger-vein recognition using NIR image sensors, Sensors 17 (6) (2017) 1297.

[20] H. Hu, W. Kang, Y. Lu, Y. Fang, H. Liu, J. Zhao, F. Deng, FV-Net: learning a finger-vein feature representation based on a CNN, in: International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 3489–3494.

[21] B. Hou, R. Yan, Convolutional autoencoder model for finger-vein verification, IEEE Transactions on Instrumentation and Measurement 69 (5) (2020) 2067–2074.

[22] Y. Fang, Q. Wu, W. Kang, A novel finger vein verification system based on two-stream convolutional network learning, Neurocomputing 290 (2018) 100–107.

[23] W. Yang, C. Hui, Z. Chen, J.-H. Xue, Q. Liao, FV-GAN: Finger vein representation using generative adversarial networks, IEEE Transactions on Information Forensics and Security 14 (9) (2019) 2512–2524.

[24] J. Huang, M. Tu, W. Yang, W. Kang, Joint attention network for finger vein authentication, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–11.

[25] W. Liu, H. Lu, Y. Wang, Y. Li, Z. Qu, Y. Li, MMRAN: A novel model for finger vein recognition based on a residual attention mechanism, Applied Intelligence (2022).

[26] B. Hou, R. Yan, ArcVein: Arccosine center loss for finger vein verification, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–11.

[27] K. Wang, G. Chen, H. Chu, Finger vein recognition based on multi-receptive field bilinear convolutional neural network, IEEE Signal Processing Letters 28 (2021) 1590–1594.

[28] J. Huang, W. Luo, W. Yang, A. Zheng, F. Lian, W. Kang, FVT: Finger vein transformer for authentication, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–13.

[29] G. Yang, X. Xi, Y. Yin, Finger vein recognition based on $(2D)^2$ PCA and metric learning, Journal of Biomedicine

and Biotechnology 2012 (2012) 1–9.

[30] X. Mei, H. Ma, Finger vein recognition algorithm based on improved weighted sparse representation, in: International Conference on Information Technology and Computer Application (ITCA), IEEE, 2019, pp. 6–8.

[31] J. Lu, V. E. Liong, J. Zhou, Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (8) (2018) 1979–1993.

[32] L. Fei, B. Zhang, J. Wen, S. Teng, S. Li, D. Zhang, Jointly learning compact multi-view hash codes for few-shot FKP recognition, Pattern Recognition 115 (2021) 107894.

[33] H. Liu, G. Yang, L. Yang, Y. Yin, Learning personalized binary codes for finger vein recognition, Neurocomputing 365 (2019) 62–70.

[34] L. Zhang, L. Sun, W. Li, J. Zhang, W. Cai, C. Cheng, X. Ning, A joint bayesian framework based on partial least squares discriminant analysis for finger vein recognition, IEEE Sensors Journal 22 (1) (2022) 785–794.

[35] L. Yang, G. Yang, K. Wang, F. Hao, Y. Yin, Finger vein recognition via sparse reconstruction error constrained low-rank representation, IEEE Transactions on Information Forensics and Security 16 (2021) 4869–4881.

[36] N. Miura, A. Nagasaka, T. Miyatake, Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification, Machine Vision and Applications 15 (4) (2004) 194–203.

[37] Q. Chen, L. Yang, G. Yang, Y. Yin, X. Meng, DFVR: Deformable finger vein recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 1278–1282.

[38] J. Yang, Y. Shi, J. Yang, Finger-vein recognition based on a bank of Gabor filters, in: Asian Conference on Computer Vision, Springer, 2009, pp. 374–383.

[39] Z. Liu, Y. Yin, H. Wang, S. Song, Q. Li, Finger vein recognition with manifold learning, Journal of Network and Computer Applications 33 (3) (2010) 275–282.

[40] J.-D. Wu, C.-T. Liu, Finger-vein pattern identification using principal component analysis and the neural network technique, Expert Systems with Applications 38 (5) (2011) 5423–5427.

[41] T. S. Beng, B. A. Rosdi, Finger-vein identification using pattern map and principal component analysis, in: IEEE International Conference on Signal and Image Processing Applications (ICSIPA), IEEE, 2011, pp. 530–534.

[42] H. Qin, M. A. El-Yacoubi, Deep representation-based feature extraction and recovering for finger-vein verification, IEEE Transactions on Information Forensics and Security 12 (8) (2017) 1816–1829.

[43] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: A unified embedding for face recognition and clustering, in: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 815–823.

[44] X. Xi, L. Yang, Y. Yin, Learning discriminative binary codes for finger vein recognition, Pattern Recognition 66 (2017) 26–33.

[45] J. Lu, V. E. Liong, X. Zhou, J. Zhou, Learning compact binary face descriptor for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (10) (2015) 2041–2056.

[46] L. Fei, B. Zhang, L. Zhang, W. Jia, J. Wen, J. Wu, Learning compact multifeature codes for palmprint recognition from a single training image per palm, IEEE Transactions on Multimedia 23 (2020) 2930–2942.

[47] S. Li, B. Zhang, L. Fei, S. Zhao, Joint discriminative feature learning for multimodal finger recognition, Pattern Recognition 111 (2021) 107704.

[48] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Transactions on Image Processing 19 (6) (2010) 1657–1663.

[49] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[50] W. Kang, Y. Lu, D. Li, W. Jia, From noise to feature: Exploiting intensity distribution as a novel soft biometric trait for finger vein recognition, IEEE Transactions on Information Forensics and Security 14 (4) (2019) 858–869.

[51] Y. Guo, Convex subspace representation learning from multi-view data, Proceedings of the AAAI Conference on Artificial Intelligence 27 (1) (2013) 387–393.

[52] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, X. Gao, Pairwise relationship guided deep hashing for cross-modal retrieval, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 1618–1625.

[53] C. Zheng, L. Zhu, X. Lu, J. Li, Z. Cheng, H. Zhang, Fast discrete collaborative multi-modal hashing for large-scale multimedia retrieval, IEEE Transactions on Knowledge and Data Engineering 32 (11) (2020) 2171–2184.

[54] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, Mathematical Programming 142 (1-2) (2013) 397–434.

[55] A. Satorra, P. M. Bentler, A scaled difference chi-square test statistic for moment structure analysis, Psychometrika 66 (4) (2001) 507–514.

[56] W. Yang, X. Huang, F. Zhou, Q. Liao, Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion, Information Sciences 268 (2014) 20–32.

[57] Y. Yin, L. Liu, X. Sun, SDUMLA-HMT: A multimodal biometric database, in: Biometric Recognition, Springer Berlin Heidelberg, 2011, pp. 260–268.

[58] M. S. Mohd Asaari, S. A. Suandi, B. A. Rosdi, Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics, Expert Systems with Applications 41 (7) (2014) 3367–3382.

[59] Y. Lu, S. J. Xie, S. Yoon, Z. Wang, D. S. Park, An available database for the research of finger vein recognition, in: International Congress on Image and Signal Processing (CISP), IEEE, 2013, pp. 410–415.

[60] W. Yang, W. Ji, J.-H. Xue, Y. Ren, Q. Liao, A hybrid finger identification pattern using polarized depth-weighted binary direction coding, Neurocomputing 325 (2019) 260–268.

[61] W. Deng, J. Hu, J. Guo, Extended SRC: Undersampled face recognition via intraclass variant dictionary, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (9) (2012) 1864–1870.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.