

Article

SIFT-Flow-Based Virtual Sample Generation for Single-Sample Finger Vein Recognition

Lizhen Zhou ^{1,2}, Lu Yang ^{3,*}, Deqian Fu ^{4,*}  and Gongping Yang ⁵ ¹ Department of Information, Zibo Normal College, Zibo 255130, China² Center for Internatoinal Education, Philippine Christian University, Manila 1004, Philippines³ School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China⁴ School of Information Science and Engineering, Linyi University, Linyi 276000, China⁵ School of Software, Shandong University, Jinan 250101, China

* Correspondence: yangluhi@163.com (L.Y.); fudeqian@lyu.edu.cn (D.F.)

Abstract: Finger vein recognition is considered to be a very promising biometric identification technology due to its excellent recognition performance. However, in the real world, the finger vein recognition system inevitably suffers from the single-sample problem: that is, only one sample is registered per class. In this case, the performance of many classical finger vein recognition algorithms will decline or fail because they cannot learn enough intra-class variations. To solve this problem, in this paper, we propose a SIFT-flow-based virtual sample generation (SVSG) method. Specifically, first, on the generic set with multiple registered samples per class, the displacement matrix of each class is obtained using the scale-invariant feature transform flow (SIFT-flow) algorithm. Then, the key displacements of each displacement matrix are extracted to form a variation matrix. After removing noise displacements and redundant displacements, the final global variation matrix is obtained. On the single sample set, multiple virtual samples are generated for the single sample according to the global variation matrix. Experimental results on the public database show that this method can effectively improve the performance of single-sample finger vein recognition.



Citation: Zhou, L.; Yang, L.; Fu, D.; Yang, G. SIFT-Flow-Based Virtual Sample Generation for Single-Sample Finger Vein Recognition. *Electronics* **2022**, *11*, 3382. <https://doi.org/10.3390/electronics11203382>

Academic Editor: Miin-shen Yang

Received: 11 September 2022

Accepted: 17 October 2022

Published: 19 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: finger vein recognition; single-sample; virtual sample; SIFT-flow

1. Introduction

Finger vein recognition is an effective biometric technology which uses subcutaneous finger vein patterns for recognition. Studies have shown that finger vein patterns are unique and stable [1,2]. Compared with other biometric features such as face, fingerprint, and gait, finger veins show the following excellent advantages in applications [3,4]. (1) Internal features: Finger vein patterns are inside the finger, so it is hard to be affected by the external environment and changes in the finger epidermis. In addition, it is very difficult for others to obtain or copy finger vein images. (2) Living body recognition: Due to the special imaging principle, the finger vein image acquisition can only be carried out in the case of living bodies. Therefore, the problem of fake image attack becomes more difficult in the finger vein recognition scenario. (3) Non-contact imaging: When capturing images, fingers do not need to touch the device, making it cleaner and more acceptable. Because of these advantages, finger vein recognition becomes a promising branch of biometrics.

Generally speaking, a finger vein recognition system mainly includes four parts: image acquisition, image preprocessing, feature extraction and matching. From the perspective of feature extraction, finger vein recognition can be divided into the following types:

1. Network-based methods: These methods need to segment vein patterns first and then extract features according to the vein patterns. Related methods mainly include: repeated line tracking (RLT) [5], maximum curvature points (MaxiC) [6,7], mean curvature (MeanC) [8], region growth [9], the anatomy structure analysis-based method [10] (ASAVE), and so on.

2. Local descriptor-based methods: These methods do not require segment vein patterns but directly apply local descriptors to images. Related methods mainly include: local binary pattern (LBP) [11,12], local line binary pattern (LLBP) [13,14], local directional code (LDC) [15], discriminative binary codes (DBC) [16], anchor-based manifold binary pattern (AMBP) [17], and so on.
3. Dimensionality reduction-based methods: These methods have achieved good results in the field of face recognition, so researchers introduced it into finger vein recognition. Commonly used dimensionality reduction techniques include: principal component analysis (PCA) [18], linear discriminant analysis (LDA) [19], two-dimensional principal component analysis ((2D)² PCA) [20,21], and so on. For these methods, multiple images are needed to train the transformation matrix.
4. Deep learning-based methods: Deep learning has been applied to various research fields because of its powerful feature representation ability. Recently, the deep learning-based methods have also achieved remarkable results in the field of finger vein recognition [22–24]. Such methods also require multiple images to participate in training.

The above methods have shown excellent performance for multi-samples finger vein recognition. However, in practical applications, such as identity management systems and attendance systems, often only one image per class can be collected, which leads to the problem of single-sample finger vein recognition. In these cases, due to insufficient intra-class information, the performance of some algorithms will drop significantly, such as network-based methods and local descriptor-based methods. Since a sample cannot obtain the intra-class variations, some algorithms that require supervised learning are not available, such as dimensionality reduction-based methods and deep learning-based methods. Therefore, it is very necessary to solve the single-sample finger vein recognition problem. Furthermore, the single-sample finger vein recognition system requires less storage space and has faster acquisition speed, which will have broader application prospects than the multi-samples recognition system.

In the field of face recognition, some researchers use sample expansion technology to synthesize multiple virtual samples from the original sample, making the single-sample problem into general face recognition. Thus, the face recognition algorithms based on multiple samples can continue to be used in single-sample recognition, which has considerable practical significance. Inspired by this, we propose a finger vein sample expansion method to solve the single-sample finger vein recognition problem. Similarly, the state-of-the-art algorithms widely used in the multiple samples finger vein recognition can continue to be applied in single sample recognition.

Compared with symmetrical face images, finger vein images do not have regular and obvious characteristics. Therefore, we can not directly follow the virtual sample generation method of the face to generate virtual finger vein images. We found that the variations between genuine images are mainly due to the finger's translation, rotation, etc. Many persons have similar habits with their fingers, which lead to similar intra-class variations. Hence, we can capture intra-class variations on a generic set and then use these variations to generate virtual samples for a single sample. Scale-invariant feature transform flow (SIFT-flow) [25–27] can effectively estimate the variations between two images; thus, we adopt it in our paper. Specifically, the SIFT-flow algorithm is used to estimate the variations between genuine images, which is used as the displacement matrix within the class. Then, the key displacements of each class are obtained to form a global variation matrix. After removing the interference displacements and redundant displacements, a final variation matrix is obtained. Finally, the variations matrix is used to generate virtual samples for a single sample on a single sample set. Based on virtual samples, single-sample finger vein recognition has been transformed into multi-sample recognition.

The main contributions of this paper can be summarized as follows. (1) We propose a virtual sample generation method to solve the single-sample finger vein recognition problem. By adding the generated virtual samples, the performance of classical algorithms is improved significantly. (2) In order to obtain effective virtual samples, we learn the intra-class variations on the general data set and then use these variations to generate virtual samples. (3) When learning intra-class variations, we use the SIFT-flow algorithm, which can effectively estimate the displacement between images. The experimental results show that our method can greatly improve the performance of single-sample finger vein recognition.

The rest of the paper is organized as follows. We discuss related work of single-sample recognition in Section 2. In Section 3, we introduce the proposed method of solving single-sample finger vein recognition. We report the experimental protocols and results in Section 4. Finally, the conclusions of our work is given in Section 5.

2. Related Work

Single-sample recognition is an important research branch of biometrics. In particular, single-sample face recognition has attracted many researchers' interests. To solve the problem of single-sample face recognition, many methods have been designed, and the method based on virtual sample generation is one of them [28]. For the virtual sample generation approach, researchers used various technologies to construct multiple virtual images from a single face image and then applied them for recognition. For example, Shan et al. [29] generated 10 face images for each person using a combination of appropriate geometric transformations (e.g., rotation, scaling) and gray-scale transformations (e.g., simulating lighting, artificially setting noise points). Zhang et al. [30] proposed performing singular value decomposition on each image matrix and then generated multiple virtual images for each face image by perturbing the singular values. Wang et al. [31] used face symmetry and sparse theory to synthesize virtual face images for sample expansion. Hu et al. [32] proposed using a single sample to reconstruct a 3D face model and then used the reconstructed model to obtain virtual face images. Xu et al. [33] used the axial symmetry of the face to generate virtual samples.

The research on single-sample finger vein recognition is scant; to the best of the author's knowledge, only Liu et al. [34] proposed a deep ensemble learning method for single-sample finger vein recognition, achieving good results. However, there are many classical algorithms for multiple-sample finger vein recognition; their performance only degrades or fails in single-sample recognition. It will be very meaningful if they can continue to be used in single-sample finger vein recognition. Existing methods cannot achieve this goal. Therefore, in this paper, we propose the method of virtual sample generation to solve the single-sample finger vein recognition problem.

3. The Proposed Method

In this section, we first introduce the SIFT-flow algorithm that will be used in our method and then introduce the proposed SIFT-flow-based virtual sample generation method in detail.

3.1. SIFT-Flow Algorithm

As the SIFT-flow [26] algorithm can effectively estimate the variation of two images, it is widely used in computer vision and computer graphics. For finger vein recognition, we also choose the SIFT-flow algorithm to obtain the displacement matrix between the images.

SIFT-flow uses scale invariant feature transform (SIFT) [35] descriptors to build dense connections between the source and target images. The SIFT descriptor is an excellent local descriptor with illumination and rotation invariance as well as partial affine invariance. The original SIFT descriptor includes two parts: feature extraction and salient feature point detection. SIFT-flow only uses the feature extraction component. The SIFT feature extraction steps are as follows: (1) For each pixel in an image, divide its 16×16 neighborhoods into

4×4 cell arrays. (2) Count the gradient directions of each cell array into 8 main directions, so that a 128 (8 main directions $\times$ 16 cell arrays) dimension feature vector for a pixel can be obtained. The SIFT image is obtained by extracting the SIFT descriptor of each pixel in an image.

In order to obtain the displacement matrix of the two SIFT images, it is necessary to find the best matching pixel for each pixel. The displacement of each pixel can be obtained by the position difference of the pixel with its best matching pixel. The displacement of each pixel consists of a horizontal displacement and a vertical displacement. Liu et al. regarded the matching problem as an optimization problem and design an objective function similar to optical flow. Suppose s_1 and s_2 are the two SIFT images. The objective function for SIFT-flow is defined as:

$$E(w) = \sum_p \|s_1(p) - s_2(p + w(p))\|_1 + \quad (1)$$

$$\frac{1}{\sigma^2} \sum_p (u^2(p) + v^2(p)) + \quad (2)$$

$$\sum_{(p,q) \in \varepsilon} \min(\alpha|u(p) - u(q)|, d) + \min(\alpha|v(p) - v(q)|, d), \quad (3)$$

where $p = (x, y)$ is the coordinate of the current pixel. $w(p) = (u(p), v(p))$ is the displacement vector of the current pixel relative to the matching pixel, which is only allowed to be an integer. $u(p)$ and $v(p)$ represent the displacement in the horizontal and vertical directions, respectively. In addition, ε is the neighborhood of the pixel, and the default value is 4 neighborhoods.

There are three parts of the function: a data term, a small displacement term, and a smooth term. The data item in (1) calculates the difference of two SIFT images. The small displacement term in (2) constrains the displacement vector to be as small as possible, since the best matching pixel should be chosen within the nearest neighborhood. The smoothness term in (3) is used to constrain the translation of adjacent pixels, which should have similar displacements. SIFT-flow uses dual-layer loopy belief propagation as the base algorithm to optimize the objective function. Unlike usual optical flow functions, the SIFT-flow smooth terms allow us to separate the horizontal flow $u(p)$ and vertical flow $v(p)$, which can greatly reduce the complexity of the algorithm.

3.2. SIFT-Flow-Based Virtual Sample Generation (SVSG)

The proposed SVSG method is divided into a training stage and testing stage. A schematic diagram of the virtual sample generation process is demonstrated in Figure 1. (1) Training stage. There are multiple samples for each class on the generic set. First, regions of interest (ROI) are extracted for each finger vein image through efficient preprocessing steps. Then, the displacement matrix for each class is learned using the SIFT-flow algorithm. We extract the key displacements of all displacement matrices, forming a variation matrix. The final global variation matrix is formed after removing the interference displacement and redundant displacement. (2) Testing stage. On the single sample set, there is only one registered sample per class. Using the variation matrix obtained from the generic set, multiple virtual samples are generated for each class. During recognition, the preprocessed input image is compared with the registered samples and virtual samples to obtain the recognition result. An overview of the recognition process is demonstrated in Figure 2.

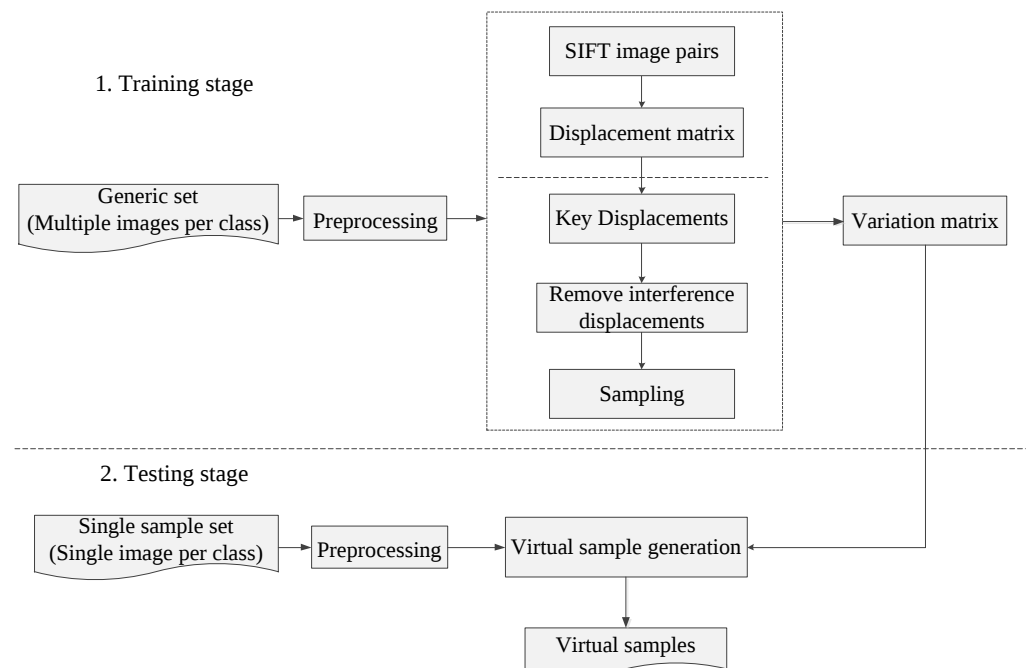


Figure 1. Schematic diagram of the virtual sample generation process.

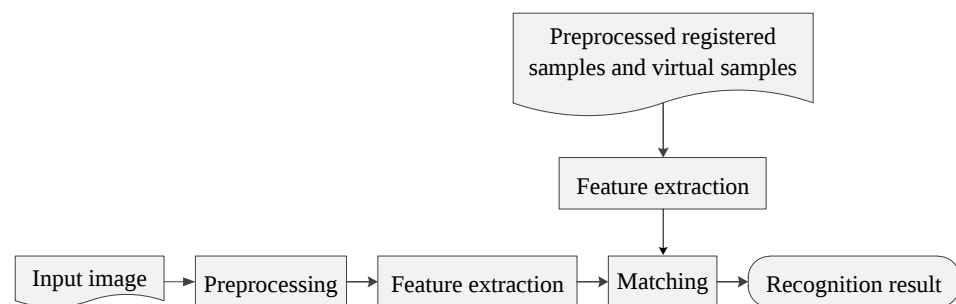


Figure 2. Overview of the recognition process.

3.2.1. Preprocessing

In our work, preprocessing mainly consists of ROI extraction, size normalization, and gray normalization [36].

ROI extraction: The collected finger vein images have complex backgrounds, and the noise in these backgrounds will reduce the recognition performance, so it is necessary to extract the ROI. To obtain the ROI image, we first use the edge detection operator to detect the edge of the finger. Then, the width of the finger area is determined according to the inscribed line of the edge of the finger, and the height of the finger area is detected according to the two knuckles in the finger. Finally, the ROI image can be obtained according to the above width and height

Size and gray normalization: The size of the ROI images obtained using the above steps is different, which will cause trouble for subsequent operations, so we normalize the size of the ROI images. The normalized image size is 80×240 pixels. Then, we use gray normalization to obtain a uniform gray distribution.

3.2.2. Variation Matrix Learning

In this section, we will discuss how to learn the variation matrix from the generic set. The learning process is divided into two steps, which are displacement matrix calculation and global variation matrix calculation. The displacement matrix calculation is for one class, while the global variation matrix calculation is for all classes.

1. Displacement matrix computation.

As mentioned above, the calculation of the displacement matrix is based on the SIFT image pair, so we need to construct the SIFT image pair. In order to ensure that the displacement matrix can cover all displacements within the class, all images are required to participate in forming image pairs. Specifically, for a particular class w , the displacement matrix is calculated as follows:

(1) Construct SIFT image pairs.

For each image within the class w , we obtain the SIFT image using the SIFT descriptor, where the j th SIFT image is represented as $SFimg_j^w$. Then, taking the first SIFT image as the benchmark, the remaining other SIFT images form SIFT image pairs with it: for example, the pair $(SFimg_1^w, SFimg_j^w)$ is formed by the first SIFT image and the j th SIFT image.

(2) Calculate displacement matrix.

In this step, the SIFT-flow algorithm is used to obtain the displacement matrix $disp_j^w$ of each SIFT image pair, which is given as follows:

$$disp_j^w = SIFTFlow(SFimg_1^w, SFimg_j^w) \quad (4)$$

where each matrix consists of displacements in both the X-direction and the Y-direction.

The process of obtaining the displacement matrix between two images is given in Figure 3. Figure 3a presents two genuine images: that is, two images from the same finger. Figure 3b shows the SIFT image pair of the two genuine images, in which the SIFT value of a pixel is represented by a white circle. In Figure 3c, the displacement matrix is given, which consists of horizontal (X-direction) displacement and vertical (Y-direction) displacement. For presentation, the values of the displacement matrix have been normalized to be between 0 and 255. By observation, we can see that the horizontal displacement of different pixels in the same image is consistent, and this feature also applies in the vertical direction.

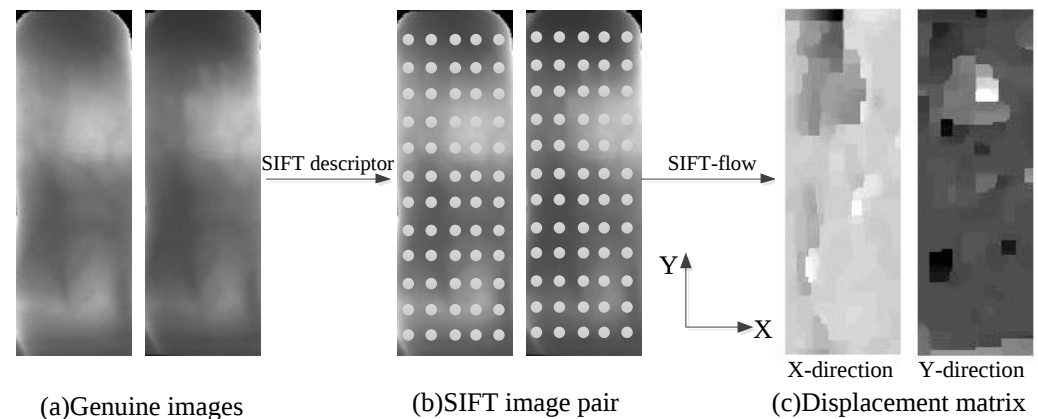


Figure 3. The process of obtaining the displacement matrix.

2. Global variation matrix computation.

Herein, we introduce the steps to obtain the global variation matrix. First, we obtain the key displacement of each displacement matrix and then remove the interference displacement. Finally, the variation matrix is sampled to reduce redundancy.

(1) Obtain key displacements.

Meng et al. [37] pointed out that in finger vein recognition, the displacements of different pixels in two images from the same finger are similar, and Figure 3c also proves this statement. So, we can use the displacement with the most occurrences as the key displacement between two images. First, the frequency of each displacement for the displacement matrix is counted. The displacement

with the largest frequency is used as the key displacement of the matrix, and all key displacements are combined into a variation matrix $keydisp^w$ of class w which can be calculated as:

$$keydisp^w = [f(\max(p(displ_1^w)), \dots, f(\max(p(displ_{m-1}^w)))] \quad (5)$$

where $p(displ_j^w)$ denotes the frequency of all displacements in the displacement matrix $displ_j^w$, and $\max(p(displ_j^w))$ denotes the maximum frequency. Equation 5 gives the displacement with a certain frequency. We calculate the key displacements of all classes in the generic set to form a temporary variation matrix $Vtemp$.

(2) Remove interference displacements.

There are two kinds of interference displacements considered in this paper. The first is the displacement with too large value and small frequency, which is also quite different from the adjacent value. These displacements are caused by occasional large movements of a finger during acquisition and are not universal. If these displacements are used to generate virtual samples, they are likely to adversely affect recognition. The second is that the displacement value is 0 or too small, and these displacements indicate that there is almost no difference between the two images. If such displacements were used to generate virtual samples, they would not be helpful for identification but would create data redundancy. Therefore, we remove the above two displacements.

(3) Sampling.

The existing temporary variation matrix is displacement-intensive. For example, there will be displacements with value x and $x+1$ at the same time, and the virtual samples generated by these two adjacent displacements have almost the same contribution to recognition. In order to avoid data redundancy, for adjacent displacements, we just keep one. Therefore, we sample the remaining matrix according to the step size and use the remaining matrix as the final global variation matrix $V = [v_1, v_2, \dots, v_k]^T$, where $v_i = (\Delta x, \Delta y)$ has two components, representing the displacement in the X-direction and Y-direction.

The process to learning the global variation matrix can be summarized as Algorithm 1.

Algorithm 1: Learning variation matrix.

Inputs: The preprocessed finger vein images on the generic set

Outputs: Variation matrix V

Algorithm:

1. **Initialize** $V = []$, $Vtmp = []$, $i = 1$, $j = 2$
 2. **While** $i \leq n$ **do** \ \ n is the number of classes of generic set
 3. Get SIFT image $SFimg_1^i$ of first image of class i
 4. **While** $j \leq m$ **do** \ \ m is the number of samples per class
 5. Obtain SIFT image pairs $(SFimg_1^i, SFimg_j^i)$
 6. Calculate displacement matrix $displ_j^i$ using Equation (4)
 7. **End while**
 8. Get key displacements of class i
 9. **End while**
 10. Key displacements of all classes form a temporary matrix $Vtmp$
 11. Remove interference displacement of $Vtmp$
 12. Sampling $Vtmp$
 13. The remainder of forms the global variation matrix V
 14. **Return** V
-

3.2.3. Virtual Sample Generation

On the single sample set, there is only one registered sample of per class. We use the variation matrix $V = [v_1, v_2, \dots, v_k]^T$ to generate different virtual samples. Assuming that (x, y) is the coordinate of point p in the registered image I , the coordinate (x', y') of the corresponding point p' in the virtual image can be calculated as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

The translation vector $(\Delta x, \Delta y)$ is a row vector of matrix V . Δx and Δy represent the displacement in the X- direction and Y-direction, respectively. The number of row vectors of V is k , so k virtual images will be generated eventually.

For the newly generated image, we use bilinear interpolation [38,39] to keep its size consistent with the original image. For the unknown point $P = (x, y)$ shown in Figure 4, the four points around it are known, which are $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ and $Q_{22} = (x_2, y_2)$. The pixel value $f(x, y)$ of the point can be calculated by the following equation.

$$f(x, y) \approx \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y_2 - y) + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y - y_1) \quad (7)$$

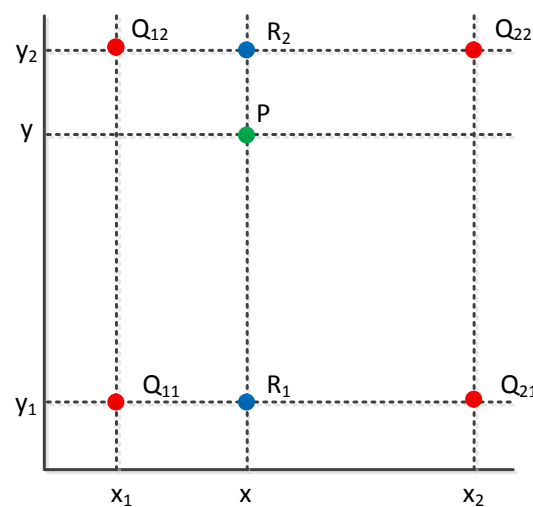


Figure 4. Bilinear interpolation.

The generated multiple samples will participate in the recognition together with the original single sample. Figure 5 shows the process of virtual image generation. In Figure 5a, a registered image is given, and Figure 5b shows the variation matrix. The registered image is transformed with each row of the variation matrix, generating multiple virtual images. Several generated virtual sample images are given in Figure 5c. Since the virtual images are transformed from the registered image, virtual images are very similar to the registered image.

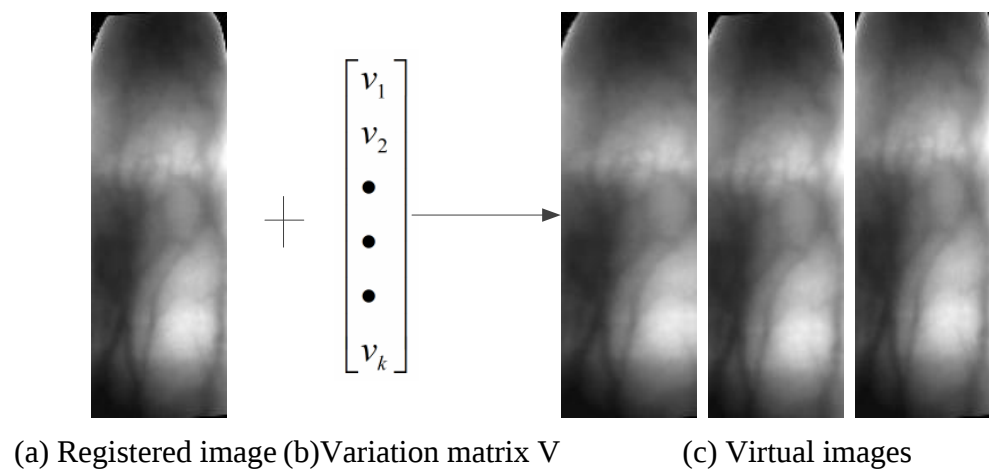


Figure 5. The process of virtual image generation.

4. Experiments

To verify the effectiveness of the proposed method, we conduct experiments on a public finger vein database from Hong Kong Polytechnic University, called HKPU-FV [40]. A total of 156 volunteers participated in the collection. Each volunteer provided six or 12 images from the index and middle fingers. The finger vein image acquisition process is completed in two sessions. Only 105 volunteers participated in collection in the second session, leading to the number of images of each finger being different. We employ finger vein images acquired in the first session. Since the vein patterns of different fingers of the same person are different, there are a total of 312 (156 persons $\times$ 2 fingers) classes and 1872 (156 persons $\times$ 2 fingers $\times$ 6 images) images in our experiments. Several typical finger vein images of the HKPU-FV database are shown in Figure 6.

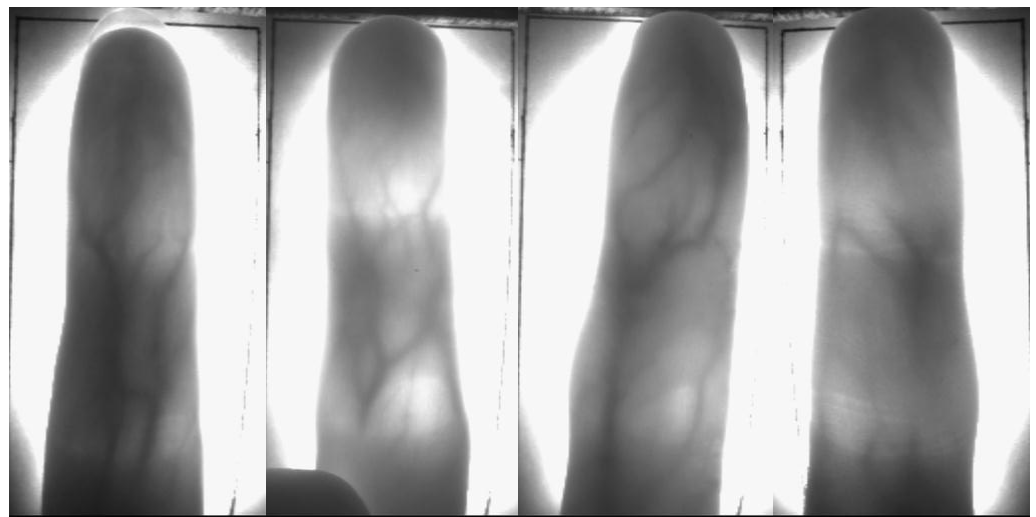


Figure 6. Finger images from the HKPU-FV database.

The database is divided into two non-overlapping subsets: the generic set and the single sample set. The generic set has 50 classes, each class has 6 images, and the images in this dataset are used to train the global variation matrix. The remaining 262 classes constitute the single sample set.

All experiments are implemented in MATLAB2018 on a personal computer with 3.3 GHz CPU and 16.0 GB memory. Four experiments are designed to verify the proposed method: (1) Experiment 1 verifies the effectiveness of the proposed method in solving the single-sample finger vein recognition. (2) Experiment 2 proves that the generated virtual

samples are complementary. (3) Experiment 3 analyzes the interference displacements. (3) Experiment 4 discusses sampling parameters.

4.1. Experiment 1: Effectiveness of SVSG

In order to verify that the proposed method can effectively solve the single-sample finger vein recognition problem, we compare the recognition rates of the classical algorithms and the recognition rates of these methods combined with the proposed SVSG. In this experiment, the first image of each class on the single sample set is used as the registered sample, and the last two images of each class are used as test samples. Two types of classical methods are considered for verification, i.e., the local-based method and network-based method, which are available in single-sample scenarios. The recognition rates are reported in Table 1, and the corresponding CMCs (cumulative match curves) are illustrated in Figure 7.

Table 1. Identification performance of different methods.

Category	Method	Rank-One Recognition Rate
Local descriptor-based methods	LBP	72.33%
	LBP + SVSG	87.02 %
	LDC	75.57%
	LDC + SVSG	86.45%
	LLBP	69.47%
	LLBP + SVSG	85.69%
Network-based methods	MaxiC	55.92%
	MaxiC + SVSG	76.34%
	MeanC	55.73%
	MeanC + SVSG	76.72%

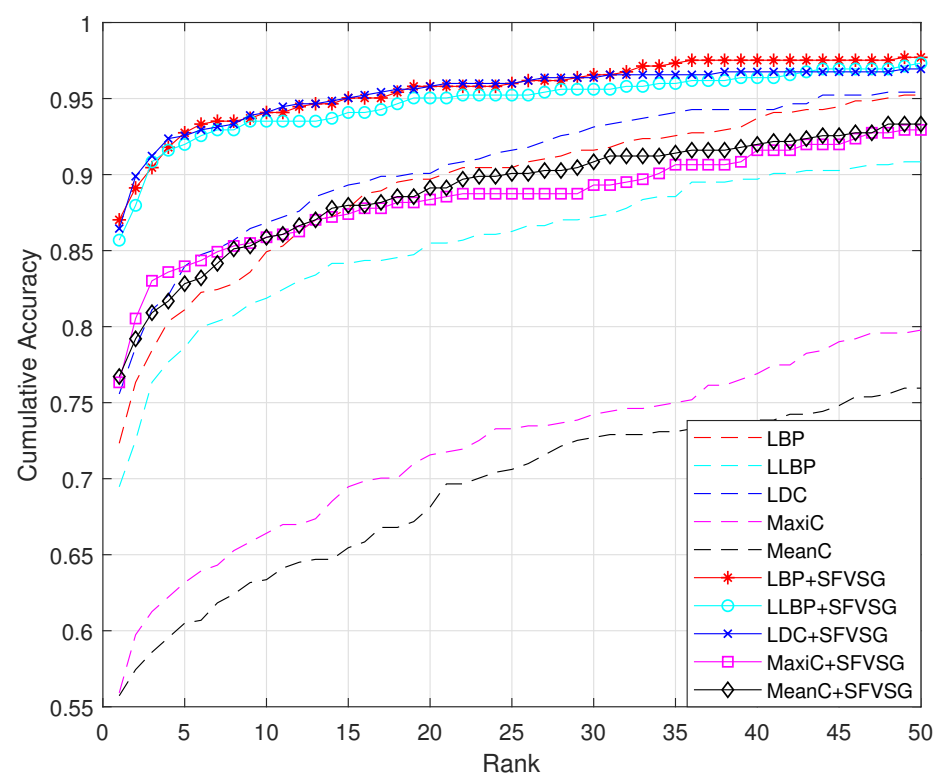


Figure 7. CMCs of different methods.

The experimental results from Table 1 and Figure 7 suggest that the methods combined with SVSG achieve significant improvement in recognition performance compared to the methods used alone. We believe that such a significant improvement is mainly attributed to the distinction and complementary nature of the virtual samples generated by SVSG. The combination of virtual samples and registered samples enriches the information within class, which increases the effectiveness of the successful matching of genuine images.

In single-sample recognition, network-based methods (i.e., MaxiC and MeanC) have poor recognition performance, which may be largely limited by single-sample incomplete vein pattern segmentation and noise. On the other hand, local descriptor-based methods (i.e., LBP, LDC, and LLBP) have better recognition performance than network-based methods, probably because these methods do not need to segment veins and are relatively less affected by single sample.

4.2. Experiment 2: Complementarity of Virtual Samples

The purpose of this experiment is to demonstrate the complementarity of the generated virtual samples. Six virtual samples are generated for each registered sample, and the recognition rates of them are shown in Table 2. In Table 2, for display purposes, we use *Vsample_i* to distinguish different virtual sample images; for instance, *Vsample₁* represents the first virtual sample. In addition, this experiment and subsequent experiments 3 and 4 will adopt the LBP algorithm as the verification algorithm, and the rest of the experimental settings are the same as those of experiment 1.

The data in Table 2 suggests that the highest recognition rate of the virtual sample is 74.62%, and the lowest recognition rate is 56.68%, which means that each virtual sample has a certain distinction. With the combination of virtual sample images, the recognition rate keeps improving. The recognition rate of all samples combined is 87.02%, which is much higher than each virtual sample. The experimental results show that the generated virtual samples are complementary. Specifically, the virtual samples are obtained through transformation; hence, there is a certain complementarity with the registered samples. On the other hand, after sampling, the displacements in the variation matrix are obviously different, so the generated virtual samples are also necessarily different and complementary.

Table 2. Identification performance of different virtual samples.

Template	Rank-One Recognition Rate (%)
Registered sample	72.33%
Virtual sample 1	74.62%
Virtual sample 2	56.68%
Virtual sample 3	68.89%
Virtual sample 4	59.73%
Virtual sample 5	64.12%
Virtual sample 6	60.11%
Vsample1 + Vsample2	75.95%
Vsample1 + Vsample2 + Vsample3	76.15%
Vsample1 + Vsample2 + Vsample3 + Vsample4	85.69%
Vsample1 + Vsample2 + Vsample3 + Vsample4 + Vsample5	85.88%
Vsample1 + Vsample2 + Vsample3 + Vsample4 + Vsample5 + Vsample6	86.45%
Vsample1 + Vsample2 + Vsample3 + Vsample4 + Vsample5 + Vsample6 + Registered sample	87.02%

4.3. Experiment 3: Interference Displacement Analysis

The purpose of this experiment is to determine the interference displacement. Figure 7 shows the projection of the displacement matrix in the X and Y directions. The horizontal axis is a random number between 0 and 1, and the vertical axis represents the value of the displacement.

It can be seen from the figure that the displacement in the X direction is mainly concentrated in the interval $[20, -10]$, and a few points are outside the interval. Correspondingly, the displacement in the Y direction is mainly concentrated in the interval $[-5, 8]$. These displacements outside the above two intervals are the first type of disturbance displacements discussed in Section 3.2.2, that is, displacements of large value and small probability. They are caused by the accidental movement of a finger and are not universal; hence, we removed them. Specifically, displacements that are more than 5 pixels from the boundary and occur only once are removed. In addition, as shown in Figure 8, a large number of points are concentrated at or near the displacement of 0. These displacements indicate that there is almost no displacement between the two images, and they are meaningless for generating virtual samples. Therefore, we also remove these displacements as disturbances. In the specific implementation, we remove all displacements of 0 and displacements less than 3.

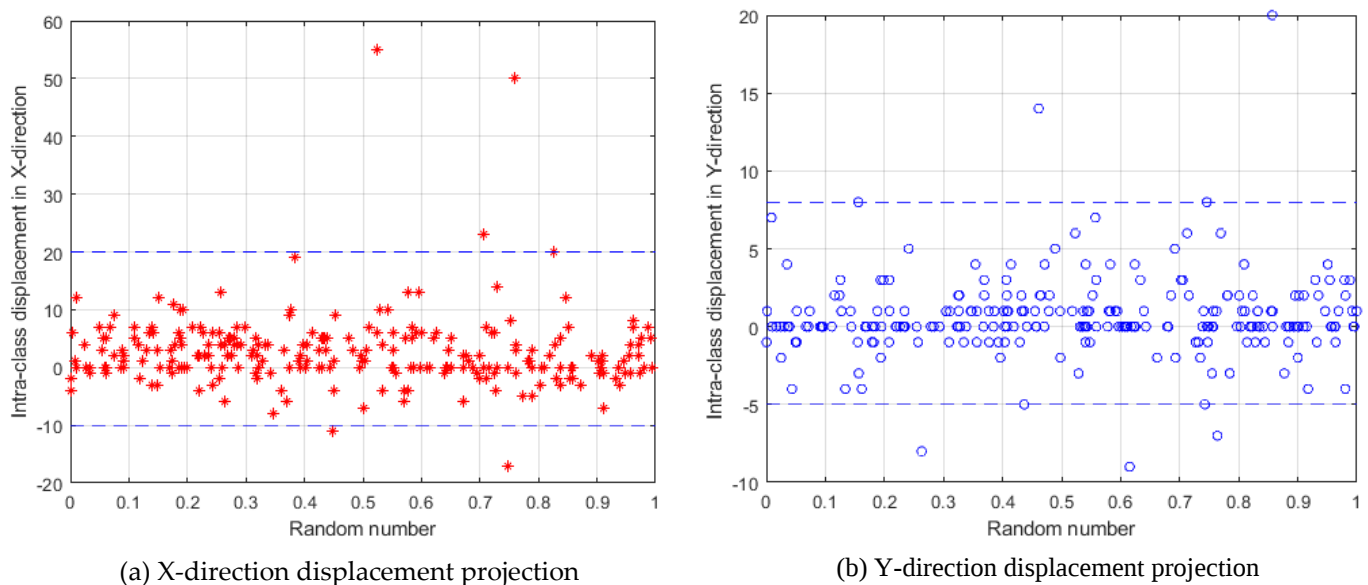


Figure 8. Displacement projection in different directions.

From Figure 8, we can also see an interesting phenomenon: the displacement in the X direction is in the range of $[20, -10]$, and the displacement in the Y direction is in the range of $[-5, 8]$. It means that when collecting images, the amplitude of the finger moving left and right is greater than the amplitude of the up and down movement. In addition, the upper boundary of the X direction is 20, and the lower boundary is -10 , indicating that the magnitude of the finger moving to the right is greater than the magnitude of the finger moving to the left. This may be related to human behavior, which needs to be further explored. These values can guide people when they collect images, reducing intra-class variation of images and increasing the recognition rate.

4.4. Experiment 4: Sampling Step Size

In this experiment, we discuss the effect of the displacement sampling step t on the recognition performance. If the sampling step size t is small, more virtual samples are generated. On the contrary, if the sampling t is large, fewer virtual samples generated. Observing the distribution of key displacements in Figure 8, we found that the smaller the displacement, the more concentrated the points. This indicates that in the actual acquisition,

the images with small finger movement are the majority. Therefore, we consider sampling with sequence step sizes. That is, for dense displacement points, use small steps to obtain more virtual samples. Conversely, for sparse displacement points, a small step size is used.

Since the displacement varies greatly in the X direction, we use the X direction as the benchmark to sample the variation matrix. The recognition rates of different t are shown in Figure 9. It can be seen that recognition rate is largest when t is 5; when t is 4 and 6, the recognition rate is the same. Therefore, we use the three steps with the top three recognition rates to form the sampling sequence $t = \{4, 5, 6\}$. After using this sequence of sampling, a total of six virtual samples are produced, and the experiment proves that the recognition rate is also the highest.

After using this sequence of sampling, a total of six virtual samples are produced. The experimental results in Table 2 prove that the recognition rate of six virtual samples reaches 86.45%, which is higher than using a fixed sampling step.

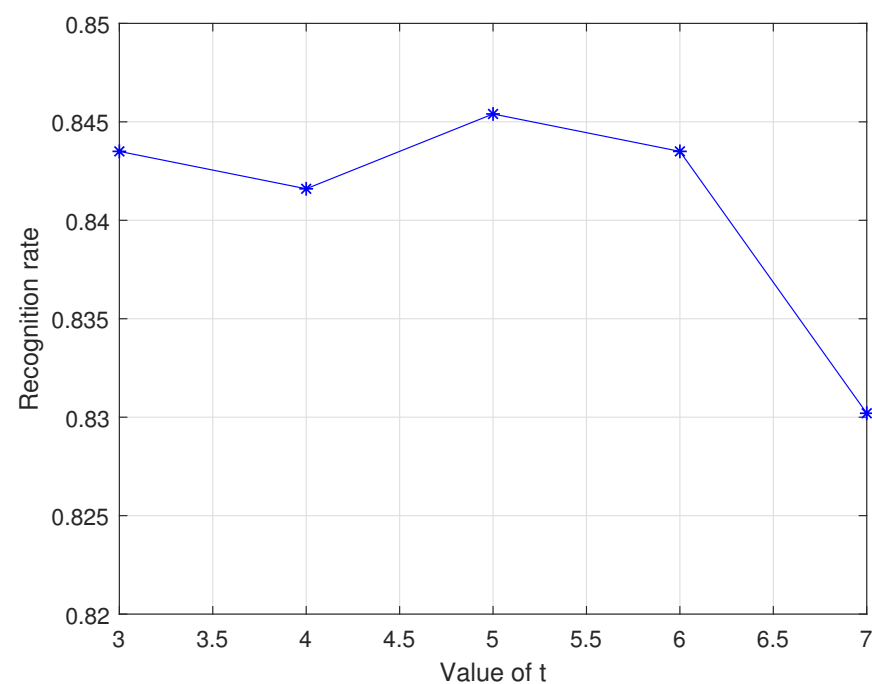


Figure 9. Recognition rates of different t .

5. Conclusions

To address the problem of single-sample finger vein recognition, this paper proposes a SVSG method. Due to the similarity of intra-class variations, we learn the variation matrix on generic set and then use this matrix to generate virtual samples for the single samples on a single sample set. In order to ensure the effectiveness and simplicity of the variation matrix, SVSG also removes the interference displacement and redundant displacement. The results on the public database verify the effectiveness of the method in solving the single-sample finger vein recognition problem. The complementarity between the virtual samples is also verified by the experiment.

Although the proposed SVSG can improve the problem of single-sample finger vein recognition, there is still a gap between our experimental results and the ideal results, which is mainly caused by limited information. In the proposed method, we obtained the intra-class variation matrix through learning, but the activities of some fingers in real life are also unpredictable, which will inevitably lead to some displacements that cannot be learned. The virtual samples are transformed through these displacements, so there is still a gap between the generated virtual matrix and the real collected samples. In the future work, we will dig deeper into the single sample information and look forward to obtaining a better solution to the problem.

Author Contributions: Conceptualization, L.Y. and G.Y.; Data curation, D.F.; Formal analysis, L.Z.; Investigation, L.Z.; Methodology, L.Y.; Resources, D.F.; Supervision, L.Y. and D.F.; Visualization, L.Z.; Writing—original draft, L.Z.; Writing—review and editing, L.Y., D.F. and G.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62076151 and Grant U1903127.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The data presented in this study are available on request from corresponding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yanagawa, T.; Aoki, S.; Ohyama, T. *Human Finger Vein Images Are Diverse and Its Patterns Are Useful For Personal Identification*; Faculty of Mathematics, Kyushu University: Fukuoka, Japan, 2007.
2. Shaheed, K.; Liu, H.; Yang, G.; Qureshi, I.; Gou, J.; Yin, Y. A systematic review of finger vein recognition techniques. *Information* **2018**, *9*, 213. [\[CrossRef\]](#)
3. Liu, Z.; Yin, Y.; Wang, H.; Song, S.; Li, Q. Finger vein recognition with manifold learning. *J. Netw. Comput. Appl.* **2010**, *33*, 275–282. [\[CrossRef\]](#)
4. Sidiropoulos, G.K.; Kiratsa, P.; Chatzipetrou, P.; Papakostas, G.A. Feature Extraction for Finger-Vein-Based Identity Recognition. *J. Imaging* **2021**, *7*, 89. [\[CrossRef\]](#)
5. Miura, N.; Nagasaka, A.; Miyatake, T. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Mach. Vis. Appl.* **2004**, *15*, 194–203. [\[CrossRef\]](#)
6. Miura, N.; Nagasaka, A.; Miyatake, T. Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE Trans. Inf. Syst.* **2007**, *90*, 1185–1194. [\[CrossRef\]](#)
7. Roza, W.A.; Kassim, J.M.; Abdullah, S.N.H.S. Finger vein recognition using straight line approximation based on ensemble learning. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 1.
8. Song, W.; Kim, T.; Kim, H.C.; Choi, J.H.; Kong, H.J.; Lee, S.R. A finger-vein verification system using mean curvature. *J. Pattern Recognit. Lett.* **2011**, *32*, 1541–1547. [\[CrossRef\]](#)
9. Qin, H.; Qin, L.; Yu, C. Region growth-based feature extraction method for finger-vein recognition. *Opt. Eng.* **2011**, *50*, 057208. [\[CrossRef\]](#)
10. Yang, L.; Yang, G.; Yin, Y.; Xi, X. Finger vein recognition with anatomy structure analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 1892–1905. [\[CrossRef\]](#)
11. Lee, E.C.; Jung, H.; Kim, D. New finger biometric method using near infrared imaging. *Sensors* **2011**, *11*, 2319–2333. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Su, K.; Yang, G.; Wu, B.; Yang, L.; Li, D.; Su, P.; Yin, Y. Human identification using finger vein and ECG signals. *Neurocomputing* **2019**, *332*, 111–118. [\[CrossRef\]](#)
13. Rosdi, B.A.; Shing, C.W.; Suandi, S.A. Finger vein recognition using local line binary pattern. *Sensors* **2011**, *11*, 11357–11371. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Liu, H.; Song, L.; Yang, G.; Yang, L.; Yin, Y. Customized local line binary pattern method for finger vein recognition. In *Chinese Conference on Biometric Recognition*; Springer: Cham, Switzerland, 2017; pp. 314–323.
15. Meng, X.; Yang, G.; Yin, Y.; Xiao, R. Finger vein recognition based on local directional code. *Sensors* **2012**, *12*, 14937–14952. [\[CrossRef\]](#)
16. Xi, X.; Yang, L.; Yin, Y. Learning discriminative binary codes for finger vein recognition. *Pattern Recognition* **2017**, *66*, 26–33. [\[CrossRef\]](#)
17. Liu, H.; Yang, G.; Yang, L.; Su, K.; Yin, Y. Anchor-based manifold binary pattern for finger vein recognition. *Sci. China Inf. Sci.* **2019**, *62*, 1–16. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Wu, J.D.; Liu, C.T. Finger-vein pattern identification using principal component analysis and the neural network technique. *Expert Syst. Appl.* **2011**, *38*, 5423–5427. [\[CrossRef\]](#)
19. Wu, J.D.; Liu, C.T. Finger-vein pattern identification using SVM and neural network technique. *Expert Syst. Appl.* **2011**, *38*, 14284–14289. [\[CrossRef\]](#)
20. Yang, G.; Xi, X.; Yin, Y. Finger vein recognition based on (2D) 2 PCA and metric learning. *J. Biomed. Biotechnol.* **2012**, *2012*, 324249. [\[CrossRef\]](#)
21. Hu, N.; Ma, H.; Zhan, T. Finger vein biometric verification using block multi-scale uniform local binary pattern features and block two-directional two-dimension principal component analysis. *Optik* **2020**, *208*, 163664. [\[CrossRef\]](#)

22. Radzi, S.A.; Hani, M.K.; Bakhteri, R. Finger-vein biometric identification using convolutional neural network. *Turk. J. Electr. Eng. Comput. Sci.* **2016**, *24*, 1863–1878. [[CrossRef](#)]
23. Das, R.; Piciuccio, E.; Maiorana, E.; Campisi, P. Convolutional neural network for finger-vein-based biometric identification. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 360–373. [[CrossRef](#)]
24. Yang, W.; Hui, C.; Chen, Z.; Xue, J.H.; Liao, Q. FV-GAN: Finger vein representation using generative adversarial networks. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2512–2524. [[CrossRef](#)]
25. Liu, C.; Yuen, J.; Torralba, A.; Sivic, J.; Freeman, W.T. Sift flow: Dense correspondence across different scenes. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 28–42.
26. Liu, C.; Yuen, J.; Torralba, A. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 978–994. [[CrossRef](#)] [[PubMed](#)]
27. Suni, S.S.; Gopakumar, K. Dense SIFT-Flow based Architecture for Recognizing Hand Gestures. *Adv. Sci. Technol. Eng. Syst. J. ASTES* **2020**, *5*, 944–954.
28. Li, L.; Peng, Y.; Qiu, G.; Sun, Z.; Liu, S. A survey of virtual sample generation technology for face recognition. *Artif. Intell. Rev.* **2018**, *50*, 1–20. [[CrossRef](#)]
29. Shan, S.; Cao, B.; Gao, W.; Zhao, D. Extended Fisherface for face recognition from a single example image per person. In Proceedings of the 2002 IEEE International Symposium on Circuits and Systems (Cat. No. 02CH37353), Phoenix-Scottsdale, AZ, USA, 26–29 May 2002.
30. Zhang, D.; Chen, S.; Zhou, Z.H. A new face recognition method based on SVD perturbation for single example image per person. *Appl. Math. Comput.* **2005**, *163*, 895–907. [[CrossRef](#)]
31. Wang, X.; Huang, W.; Qin, C.; Tian, L. Using weighted average face and symmetrical face to solve problem of single sample per person based on sparse representation. *Appl. Res. Comput.* **2015**, *32*, 1527–1531.
32. Hu, F.; Zhang, M.; Zou, B.; Ma, J. Pose and Illumination Invariant Face Recognition Based on HMM with One Sample Per Person. *Chin. J. Comput.* **2009**, *32*, 1424–1433. [[CrossRef](#)]
33. Xu, Y.; Zhang, Z.; Lu, G.; Yang, J. Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. *Pattern Recognit.* **2016**, *54*, 68–82. [[CrossRef](#)]
34. Liu, C.; Qin, H.; Yang, G.; Shen, Z.; Wang, J. Ensemble Deep Learning Based Single Finger-Vein Recognition. In Proceedings of the International Conference on Cognitive Systems and Signal Processing, Suzhou, China, 20–21 November 2021; pp. 261–275.
35. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
36. Yang, L.; Yang, G.; Yin, Y.; Xiao, R. Sliding window-based region of interest extraction for finger vein images. *Sensors* **2013**, *13*, 3799–3815. [[CrossRef](#)]
37. Meng, X.; Xi, X.; Yang, G.; Yin, Y. Finger vein recognition based on deformation information. *Sci. China Inf. Sci.* **2018**, *61*, 1–15. [[CrossRef](#)]
38. Maeland, E. On the comparison of interpolation methods. *IEEE Trans. Med. Imaging* **1988**, *7*, 213–217. [[CrossRef](#)] [[PubMed](#)]
39. Parker, J.A.; Kenyon, R.V.; Troxel, D.E. Comparison of interpolating methods for image resampling. *IEEE Trans. Med. Imaging* **1983**, *2*, 31–39. [[CrossRef](#)] [[PubMed](#)]
40. Kumar, A.; Zhou, Y. Human identification using finger images. *IEEE Trans. Image Process.* **2011**, *21*, 2228–2244. [[CrossRef](#)] [[PubMed](#)]