IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 November 2022, accepted 30 November 2022, date of publication 5 December 2022,
date of current version 9 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226888

RESEARCH ARTICLE

An Optimization Algorithm to Improve the Accuracy of Finger Vein Recognition

ZHI CHONG WAN[1], **LAN CHEN**[1], **TAO WANG**[1], **AND GUO CHUN WAN**[2]
[1]Department of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China
[2]Department of Electronic Science and Technology, Tongji University, Shanghai 201804, China

Corresponding author: Lan Chen (chenlan@sit.edu.cn)

This work was supported by the Scientific Research and Development Fund of Shanghai Institute of Technology under Project KJFZ2021-109.

ABSTRACT As people's daily behavioral activities become more data-based, how to protect personal information security is a crucial consideration for the whole society. Finger vein recognition is becoming an essential means of identification because of its uniqueness, live detection, security, and many other advantages. Although deep learning can make finger vein recognition have an excellent effect. However, the number of samples needed to build a deep network model is too large, and the current authoritative finger vein database cannot reach the minimum number of samples required. The emergence of Muti-Grained Cascade Forest provides a solution to the problem of insufficient sample data and long training time, which can give a new research avenue in feature extraction. In order to obtain higher accuracy, the deep forest algorithm is introduced in this paper to process the finger vein images. Firstly, the image data in the finger vein image database is pre-processed to prepare for the subsequent feature extraction and matching. Then, the deep forest algorithm is used to find the feature points, and the ORB algorithm is used to match the features to obtain the angular information of each matched pair, and the final identity is determined according to the sparse distribution of angles. The accuracy of finger vein recognition based on the deep forest algorithm is 98.40%. By comparing with other machine learning methods for finger vein recognition, the proposed method has a higher accuracy rate.

INDEX TERMS Deep forest, ROI region extraction, ORB algorithm, finger vein recognition.

I. INTRODUCTION

With the development of artificial intelligence and computer science technology, emerging biometric technologies are gradually replacing traditional identification technologies such as keys and passes. People's requirements for identification are getting higher and higher, and biometric technologies are transforming from the first generation (fingerprint, palm print, voice print) to the second generation (finger vein, face). Compared to other biometric technologies, finger vein recognition has the following advantages: (1)Vein features are hidden under the skin and are not easily stolen. Finger vein images must be captured in a live body, which is difficult to forge, and are not affected by external environments such as skin condition and temperature. (2)Does not change over time, has good stability and uniqueness. (3)The small size and contactless nature of the finger vein capture device allow for a broader range of applications. As a research hotspot, finger vein recognition technology has experienced decades of development, and many advanced finger vein recognition methods have been proposed in recent years.

A. RELATED WORK

According to the different ways of finger vein feature extraction, finger vein recognition methods can be divided into five categories: Vein Pattern Methods, Feature Points Matching Methods, Statistical Characteristic Analysis Methods, Local Features Methods and Deep Learning Methods [1].

Vein pattern methods mainly use various algorithms to extract vein patterns from the pre-processed finger vein

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

images and then compare the geometry or topology of the finger vein patterns to discern whether several finger vein images are from the same finger. Typical methods include repeated linear tracking [2], RLT [3], maximum curvature [4], mean curvature [5], [6], Gabor [7], etc.

Feature points matching methods mainly use minutiae points (endpoints, bifurcation points, midpoints, etc.) or other types of feature points on the finger vein images for matching. The information on these feature points is used to construct the corresponding feature vectors, and then the feature vectors are used for matching. Qin et al. [8] used the endpoint and bifurcation point features on the vein pattern to construct the topology of the pattern based on the extraction of the vein pattern. They demonstrated that the method significantly improved the verification accuracy of the finger vein by experimental data. Liu et al. [9] used the detail points after singular value decomposition for algorithm matching and obtained a excellent experimental result. The SIFT [10] method can extract more feature points from finger vein images. However, the fuzzy vein pattern of the finger vein image is prone to false detection of feature points and does not take into account the vein line distortion caused by finger bending or rotation. Rublee et al. [11] proposed a new algorithm. The ORB algorithm is divided into feature point extraction and feature point description. The feature extraction is based on the FAST algorithm [12], and the feature description is based on the BRIEF feature description algorithm, whose unique binary string form reduces the matching time while saving storage space. The computation time of ORB is only 1% of that of SIFT and 10% of that of SURF. ORB algorithm also has disadvantages, such as the weak ability to handle scale transformation.

Statistical characteristic analysis methods include Principal Component Analysis (PCA) [13], Linear Discriminant Analysis (LDA) [14], and Sparse Representation (SR) [15]. These methods do not require the separation of vein patterns from finger vein images, but use all the information in the finger vein image (vein region + non-vein region) to obtain new feature information. They use the overall structure of the finger vein image as the basis for matching. However, they did not fully consider the local features of the finger vein and had defects in the accuracy of recognition.

Local features methods also do not require image segmentation and mainly feature the grayscale differences between finger vein patterns and the image background. Such methods use a chunking strategy to characterize the changes in the vein image's grayscale distribution, preserving the original image's spatial structure information to some extent. Local features methods are widely used in finger vein recognition, classical local features methods include the local binary pattern (LBP) [16] and the local derivative pattern (LDP) [17]. On the basis of LBP and LDP, researchers have also proposed a number of variants [18]. Zhang et al. [19] proposed the directional binary code, which is a new variant of LBP. Yang et al. [20] proposed the use of LBP-based personalized optimal bit mapping. Experimental results show that the method not only has better accuracy but also has higher reliability and robustness. This method has the drawback that the texture features obtained using this method are susceptible to local grayscale anomalies.

The above four types of methods belong to the traditional finger vein recognition methods, but in recent years, deep learning has been widely utilized in dealing with various problems in bioinformatics and computational biology [21], [22]. Deep learning has shown outstanding performance in image recognition, which has provided new research ideas for finger vein recognition technology. Numerous researchers have also started combining deep learning methods with finger vein recognition technology, further developing vein recognition algorithms. Das et al. [23] proposed a CNN-based finger vein recognition system that achieves stable and accurate performance even for images containing noise. Xie et al. [24] combined CNN and supervised discrete hashing to propose a new finger vein recognition method that significantly reducing the template size. Hou and Yan [25] proposed an arccosine center loss function that significantly improves the feature recognition capability of CNN, but this method is time-consuming and ineffective in feature extraction. Liu et al. [26] designed a shallow network with three convolutional blocks and two fully connected layers, which can be effectively applied to both closed-set architecture (CS-architecture) and open-set architecture (OS-architecture).

To address the problem that finger vein verification methods rely entirely on a priori knowledge and lack the robustness of extracting finger vein features from the original image, Qin et al. [8] separated the striations and backgrounds in finger vein images. They used them as inputs to train a CNN model that can distinguish the vein striations in the images and achieve the extraction and recovery of vein features using limited a priori knowledge. Shaheed et al. [27] described a new model called DS-CNN, where the authors proposed a pre-trained CNN network, Xception, and augmented the data with different geometrical techniques. This method achieved excellent accuracy on both public datasets, SDUMLA and THU-FVFDT2. Boucherit et al. [28] proposed a merged convolutional neural network (merge CNN), which used integrated learning to input images of different quality into the same CNN and finally merged to obtain the optimal CNN structure. The method achieved 99.48% recognition rate in the SDUMLA-HMT database.

Convolutional neural networks (CNNs) have a strong characterization capability, but CNNs require a large number of training samples and take a long time to train, which makes it difficult to meet the requirements of real-time detection [29]. To improve this limitation, Shen et al. [30] proposed a lightweight convolutional neural network model with a triple loss function to train the model, which can recognize new classes without retraining and achieves extremely high accuracy on both publicly available finger vein datasets. In order to satisfy the real-time finger vein recognition, Kuzu et al. [31] proposed a real-time finger vein acquisition system combining convolutional neural network and recurrent neural network, which reduces the recognition time and ensures the recognition accuracy to some extent. Wang et al. [32] proposed a finger vein recognition method using multi-sensory field bilinear convolutional neural network (MRFBCNN), which can better The MRFBCNN can better differentiate the finger vein images with small differences, while using a lightweight neural network to reduce the network parameters and computational complexity.

The proposed GCForest model (Muti-Grained Cascade Forest) solves the problem of insufficient data and long training time for small samples and provides a new research idea for finger vein image feature extraction. However, GCForest has some difficulties in the adjustment of hyperparameters [33]. On this basis, Zhou and Feng [34] proposed a deep forest model, which trained the model by constructing multiple sets of random forests and using the growth of decision trees in each forest, while the trained random forests of each round were then cascaded data processing, i.e., the output of each round was used as the input of the next round, until there was no significant performance improvement in both rounds. Deep forests have far fewer hyperparameters than deep neural networks and can achieve extremely good performance using default hyperparameter values. The many advantages of deep forests make them ideal for applications in finger vein recognition, but there are also some limitations.

B. CONTRIBUTION

In order to improve the accuracy of finger vein recognition as well as to reduce the recognition time for verification, and to solve the problem of small training samples. The main contributions of this article are as follows:

1) We propose a deep forest-based finger vein feature point labeling method, input finger vein image set and corresponding vein feature point coordinate dataset for training, and obtain a model with the function of extracting vein image feature points.
2) After acquiring the feature information of the finger vein image to be recognized by the model, the ORB algorithm is used for matching.
3) In the matching phase, matching is performed using the minimum Hamming distance and the angular distribution is constructed.

C. PAPER STRUCTURE

The remainder of this article is organized as follows. Section II presents the framework and implementation details of the proposed finger-vein-recognition method. Section III shows the experiment results and discussion. Section IV provides the conclusions of the article.

II. PROPOSED METHOD

A. DEEP FOREST

The deep forest algorithm has two main components, multi-grained scanning, and cascaded data processing. Multi-grained scanning reorganizes the raw input data to obtain more feature information. The reorganized data is trained by cascading data processing. That is, the data obtained from the previous layer is used as the input of the next layer until the parameters obtained from the two layers are similar. Therefore, the deep forest can adjust the number of layers of cascade adaptively, which has a good effect in the case of small sample training data [34].

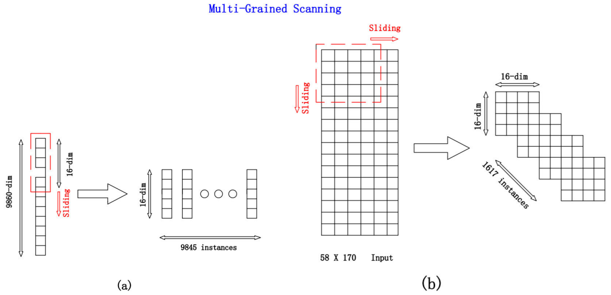

FIGURE 1. Multi-grained scanning.

1) MULTI-GRAINED SCANNING

Multi-grained scanning can obtain more feature information than the raw data and can enhance the subsequent cascade data processing. Suppose there are N $[I_L, I_W]$ pixel size image datasets, then the total pixel size of the whole dataset is:

$$S_{dl} = N \cdot I_L \cdot I_W \tag{1}$$

Then a window of $[W_l, W_w]$ pixel size is used to slice the images of the whole dataset and the step size chosen is $[S_l, S_w]$, then the number of slices per image generated is:

$$n_{section} = \left(\frac{I_L - W_L}{S_l} + 1\right) \cdot \left(\frac{I_W - W_W}{S_w} + 1\right) \tag{2}$$

Then the total number of slices is:

$$n_{all\_section} = N \cdot n_{section} \tag{3}$$

Based on the slice window size, the total number of pixel points is:

$$S_{new\_all} = n_{all\_section} \cdot w_l \cdot w_w \tag{4}$$

These pixel points are the maximum peak of memory consumption when running the code, and then the total pixel points are fed into the random forest for data processing.

Multi-granularity scanning can process one-dimensional data, two-dimensional images, and the resulting data slices will be fed into a random forest for processing. In this paper, the input original image size is 58 × 170 pixels, and a sliding window of [16, 16] is used for scanning with a step size of [2, 2]. As shown in Figure 1(b), the number of slices of [16, 16] pixel size obtained is:

$$n_{section} = \left(\frac{58 - 16}{2} + 1\right) \cdot \left(\frac{170 - 16}{2} + 1\right) = 1617 \tag{5}$$

If the image is converted into one-dimensional data and then intercepted through a sliding window, the result is the same, as shown in Figure 1(a). In order to better reflect the information of vein feature points, this paper will use the processing

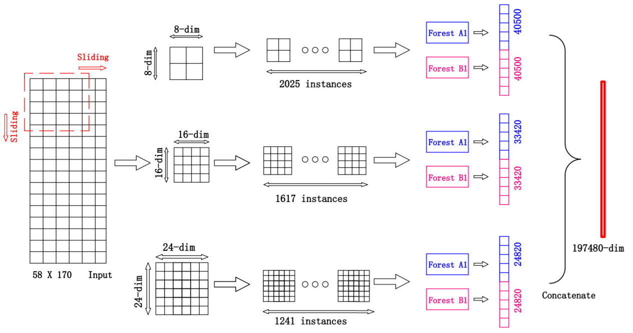

FIGURE 2. Get initial features.

method of two-dimensional images with sliding windows of [8, 8], [16, 16] and [24, 24].

The random forest layer contains completely random forest and ordinary random forest. Each completely random forest contains 500 completely random trees, and each tree selects a feature as its basis for splitting during node splitting, which stops growing based on the fact that each leaf node contains instances of the same class or the number of instances on that leaf node does not exceed 10. Each ordinary random forest also performs the same setup, but with a different treatment on the growth rules of the decision tree. Each ordinary random tree selects k candidate features and filters the split node features by *Gini* value, where k is the square root of the number of input features.

As shown in Figure 2, an image of [170, 58] pixel size is intercepted with a sliding window of [16 × 16] pixel size to obtain 1617 segments. Assuming that 20 classes are needed, then each segment passes through a random forest layer to obtain a class vector of [1 × 20] pixels in size. And then these class vectors are superimposed to obtain the total class vector with the features of all the images in the training set. The subsequent cascade data processing is based on this total class vector.

2) CASCADING DATA PROCESSING

Due to the 20 classes set above, each forest will get a class vector of size [1 × 20]. Then the [1 × 80] size class vector and the initial feature vector of 4 forests will be used as the input of the next layer. At each training layer, the performance of the entire cascade will be estimated on the validation set. The training process will be terminated if there is no significant performance gain. The node state of the forest at this point is the final model. For the classification of an unknown image, the feature vectors of the image are input to the model, and then four sets of [1 × 20] class vector data are obtained. The final class estimates are obtained by averaging these four sets of data and then finding their maximum values, as shown in Figure 3.

3) ALGORITHM FLOW

The algorithm of the deep forest including the training phase and the testing phase.

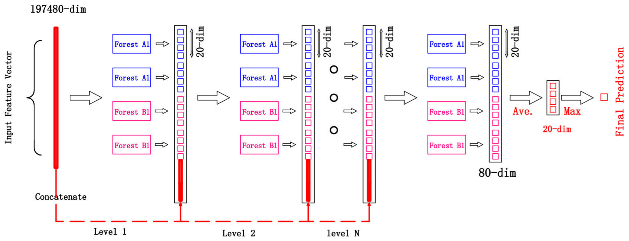

FIGURE 3. Cascading data processing.

Training phase: First, the input set of vein images is randomly selected to prevent overfitting. Then, the vein images are scanned at multiple granularities to obtain fragments with characteristic information. Then, the segments of different sizes are fed into two random forests. These two random forests mainly serve to acquire transformed feature vectors and reorganize all the feature vectors for subsequent cascade data processing. In the cascade data processing stage, four random forests are used for training to obtain the probability of each feature. The output of each layer and the original features are used as input for the next layer. The training is stopped when this layer's output and the previous layer's output are within the set error tolerance.

Testing phase: The image to be recognized is scanned at multiple granularities to obtain its slices. The slices are input to two random forests to obtain feature vectors, and then these feature vectors are reorganized, and input to the model is obtained in the training phase. In this way, each predicted probability value of the category to which the image belongs is obtained. The overall probability is obtained by summing and averaging the probability values of similar categories.

B. EXTRACTION OF FEATURE POINT INFORMATION BASED ON DEEP FOREST

In this paper, we use the deep forest algorithm to extract the feature points of vein images, and the steps are as follows:

1) Classify the coordinates of each feature point in the coordinate data set Data-B by the number of the finger, and construct the corresponding algebra of the coordinates of the feature points belonging to the same finger to obtain the template reference image of each finger.
2) The finger vein image set Data-A and the coordinate data set Data-B are used to train the deep forest algorithm.

The result of feature extraction is shown in Figure 4.

C. FEATURE MATCHING BASED ON ORB ALGORITHM

The ORB algorithm is based on FAST feature point detection and BRIEF feature point description, which satisfies rotation invariance and has had more applications in recent years for image matching. The flow of the ORB algorithm for feature detection and matching is as follows:

1) The eligible feature points on the image are captured using the FAST detection algorithm, and then the image

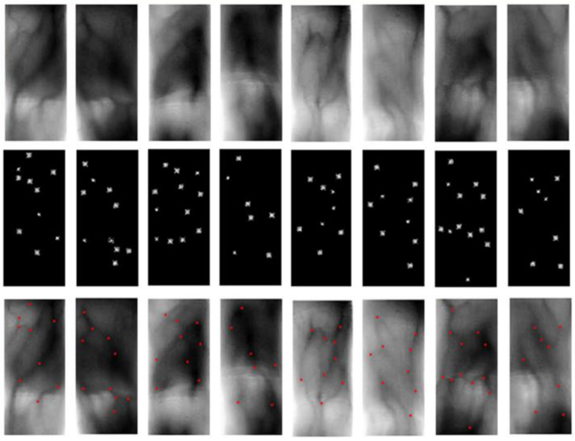

FIGURE 4. Results of finger vein feature extraction based on deep forest.

is computed using the grayscale prime method. In this paper, the model acquires the feature points of the finger vein image, and the FAST test operator finds other features of the image based on these feature points.

2) Generate feature vectors of feature points according to BRIEF algorithm.
3) Calculate the Hamming distance between the template map and the feature vector corresponding to the finger vein image to be recognized, and match the two feature points with the smallest Hamming distance until all features are matched.
4) The final matching probability is obtained according to the sparse distribution of the angle, and if the matching probability is greater than the set threshold, the recognition is considered successful; otherwise, the recognition fails.

A feature point of an image is a point in the image where the absolute value of the gray value of a point and most of its surrounding points exceeds a set threshold. The coordinate points of the feature points acquired by the model are labeled on their original map (i.e., the grayscale value is set to 255). The description of the feature points is then performed using the BRIEF algorithm, which is described using binary encoding. Around the extracted feature points, n pairs of pixel points are sampled and the relationship between the magnitude of the gray value of the point pairs is compared, and then a BRIEF describing the feature vector is created.

$$\tau(A, x, y) = \begin{cases} 1, A(x) < A(y) \\ 0, A(x) \geq A(y) \end{cases} \quad (6)$$

A(x), A(y) denote the grayscale values of x and y of feature point A. Randomly sampling n pairs of points in the domain of A and generating a feature descriptor.

$$f_n(A) = \sum_{1 \leq i \leq n} 2^{i-1} \cdot \tau(A, x_i, y_i) \quad (7)$$

Then a rotation matrix is created based on the principal direction of the feature points, and a binary code containing the angle information is calculated by combining the rotation matrix with the sampled points. Feature points are matched according to the Hamming distance between feature vectors, and the two feature points with the smallest Hamming distance are grouped into a pair. The matching results are shown in Figure 5 and Figure 6.

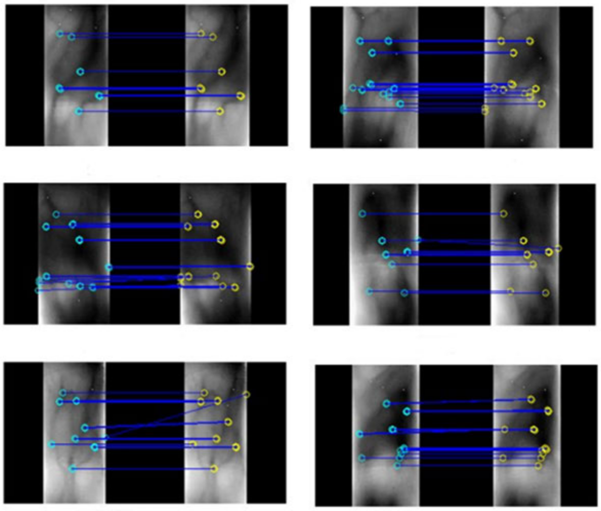

FIGURE 5. ORB algorithm matching between the same finger.

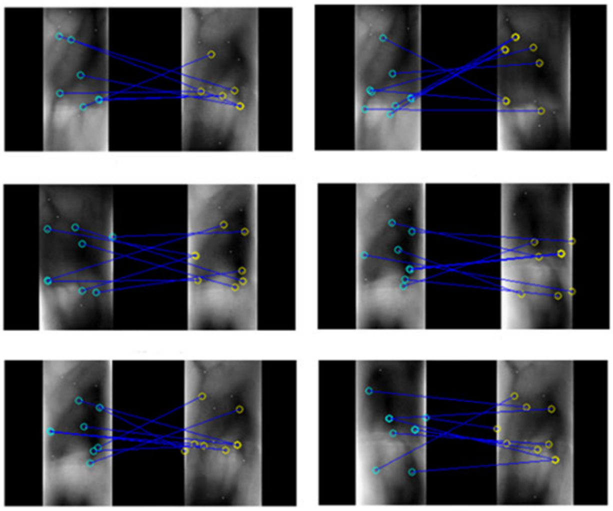

FIGURE 6. ORB algorithm matching between different fingers.

III. EXPERIMENTS RESULTS AND DISCUSSION

A. DATA PREPROCESSING

The finger vein data source used in this paper comes from two parts, which are the finger vein image database of Shandong University (Data1:SDUMLA) and the homemade finger vein database (Data2). Data1 database contains 64 different fingers, each finger has 15 vein images, total 64 × 15 images. Data2 database by using the FV1000 finger vein collector to collect 20 different fingers, each with 15 vein images, for a total of 20 × 15 images. As shown in Figure 7. The

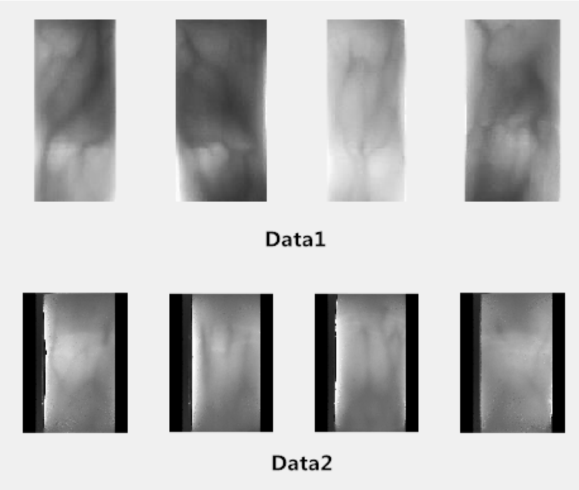

FIGURE 7. Finger vein image database data.

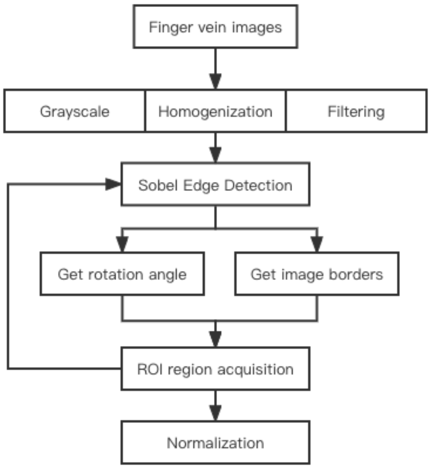

FIGURE 8. Process of ROI region extraction.

finger vein image preprocessing contains several parts such as grayscale, homogenization, filtering process, edge detection, and rotation positioning. The image of Data2 in Figure 7 is directly captured by the collector, so image pre-processing is required to obtain the ROI region image. The extraction process is shown in Figure 8.

B. TRAINING DATA

The clear finger vein images obtained based on ROI region extraction cannot be directly used as data for subsequent training, so feature points in the images need to be acquired to generate the training set data. Common finger vein features are divided into texture, striations, and unchanging local features. The detailed point characteristics of veins are divided into four main categories: endpoints, bifurcation points, midpoints, and quadruple bifurcation points. The feature point extraction methods for finger vein images are as follows:

1) Image processing: Acquisition of binarized images of the vein skeleton. The image is contrast enhanced and then binarized and refined to obtain the vein skeleton of the input image.
2) Extraction of feature points: The refinement map with vein skeleton information is scanned by the line tracing method. Since a line may have multiple bifurcation points, the mirror image assisted method is used to record the location of the branch points to prevent repeated scanning of the line, thus speeding up the acquisition of feature points.
3) Pseudo-feature removal: Among all feature points, some pseudo-features formed by burrs have not been processed. These burrs are usually below 7 pixels in length so that these pseudo-feature points can be processed according to this feature.

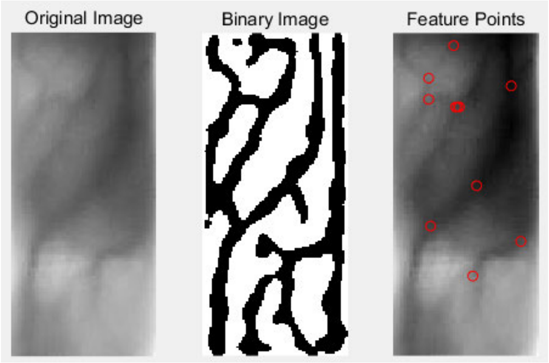

FIGURE 9. Feature points display on vein image data.

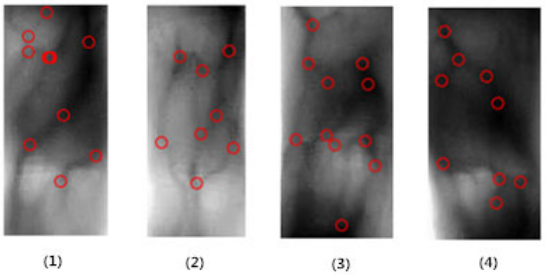

FIGURE 10. Feature points display on the image set.

TABLE 1. Training data description.

Training Data	Description			Format
Finger vein image set Data-A	Data-1	64 × 15 sheets	170 × 58	.bmp
	Data-2	20 × 15 sheets	170 × 58	
Coordinate ingormation set Data-B	Each row in the file represents a finger vein image, and each column corresponds to the coordinates of the feature points on that vein image			.csv

Figure 9 shows the vein skeleton obtained by binarization of the finger vein image and the vein feature points found by refinement. Figure 10 shows the location annotation of the feature points on the original image obtained by the above method for the image set used in this paper. The training data are described in Table 1.

This paper's two vein image sets are processed as follows, as shown in Table 2. For 960 images in the vein image

TABLE 2. Information table of the number of images during the experiment.

Training Data	Training Process		Validation Process
	Images for training	Images for verification (randomly selected)	
Finger vein dataset of SDUMLA	300 Sheets	60 Sheets	660 Sheets
Self-made Finger Vein Dataset	200 Sheets	60 Sheets	100 Sheets

TABLE 3. Definition of the main parameters of the deep forest algorithm.

n_mgsRFtree	Number of decision trees in a random forest during a multi-granularity scan	100
window	The size of the slice window used during the multi-granularity scan	8 × 8 16 × 16 24 × 24
stride	Sliding interval used for data slicing and processing	4
n_cascadeRF	Number of random forests per group in the cascade layer	4
n_cascadeRFtree	Number of decision trees in each random forest in the cascade layer	500
tolerance	Accuracy of cascade growth.	0.8
cascade_test_size	Used for K-fold cross-validation	0.25
min_samples_mgs	Minimum number of samples to perform splitting in random forest training nodes	10
min_samples_cascade	The minimum number of samples in the nodes where random forest training performs segmentation	10

database of SDUMLA, 300 images are used as the training set, and 20% of these 300 images are randomly selected for validating the model's accuracy using cross-validation. The remaining 660 finger vein images are used as the test set. For the self-made vein image set, 200 images are selected for training, 60 for cross-validation in each cascade data processing, and 100 images are used as the test set in the final validation process.

C. PARAMETER SETTING

The more levels the deep network contains, the better the feature learning ability of the deep model. Using a deep forest algorithm to extract finger-vein image feature points is a classification issue. The parameters that affect the classification effect of the algorithm are shown in Table 3.

D. EVALUATION CRITERIA

The following performance metrics are usually used for biometrics: FAR (False Acceptance Rate), FRR (False Rejection Rate), EER (Equal Error Rate), recognition rate, and error rate. In this paper, we will use these metrics to better illustrate the effectiveness of the algorithm.

E. RESULTS

In this paper, we use the deep forest algorithm to obtain the feature point positions of vein images. Then we use the ORB algorithm for feature matching and extract the main direction of feature points by the FAST operator. Use the BRIEF algorithm to binarize and encode the feature points, perform feature matching and obtain angle information according to the Hamming distance data between the reference template image and the image to be recognized. Finally, recognize the results according to the matching probability of angle distribution.

TABLE 4. A set of binary encoding tables extracted by the brief algorithm (Hexadecimal display).

BRIFF algorithm for binarization encoding (hexadecimal display)		
Template images	Vein image to be recognized	
0000000012D15D50	000000001CF91D80	000000001CF91060
000000001D6E4960	000000001CF91E70	000000001CF915D0
0000000012F52840	000000001CF91570	000000001CF91390
0000000012F52630	000000001CF91B10	000000001CF917E0
0000000012F526C0	000000001CF919F0	000000001CF91930
000000001D08B180	000000001CF91B70	000000001CF91180
000000001D08B4E0	000000001CF91360	000000001CF91810
000000001D08B960	000000001CF91F90	000000001D6C1DE0
000000001D6C11B0	000000001CF91450	000000001D6C1270
0000000012D15810	000000001D6C1570	000000001D6C1AE0
0000000012F52D20	000000001D6C13F0	000000001D6C1720
0000000012F52E10	000000001D6C18D0	000000001D6C11E0
0000000012F52660	000000001D6C1630	000000001D6C1ED0
000000001D08BA50	000000001D6C1780	000000001D6C1E70
000000001D08B8A0	000000001D6C1240	000000001D6C1E10
000000001D08BC00	000000001D6C1D20	000000001D6C1DB0
000000001D08B3F0	000000001D6C1D50	
000000001D6C1930	000000001CF918D0	

The set of binarization codes obtained by the BRIEF algorithm is shown in Table 4. The minimum Hamming distance between the image to be recognized and the template image can be obtained according to the binarization encoding, and the features with the minimum Hamming distance are matched correspondingly. And then, the angle information and angle distribution between each pair of feature points is obtained, as shown in Figure 11 and 12. From this, the angle distribution can be discriminated against, whether they are the same finger or not. In order to prevent the misjudgment caused by too few pairs of matching feature points, the threshold value of the judgment parameter is increased and set to 0.8 in this paper.

The recognition rate of the method is 98.40%, FRR is 0.8938%, FAR is 0.7061%, and EER is 2.8%. The EER curve of the method is shown in Figure 13.

F. PERFORMANCE COMPARISON AND DISCUSSION

This paper proposes a finger-vein feature extraction method base on the deep forest. It uses the SDUMLA finger vein database for model training and validation to obtain good recognition accuracy and EER. We use the finger vein recognition results of traditional machine learning methods as the baseline to compare the proposed method with the state-of-the-art CNN-based deep learning methods, all of which use the same finger vein image dataset SDUMLA. The experimental results are shown in Table 5. Compared with the traditional machine learning methods, the deep forest-based finger vein recognition methods are superior in accuracy and EER. The comparison with other advanced CNN-based deep

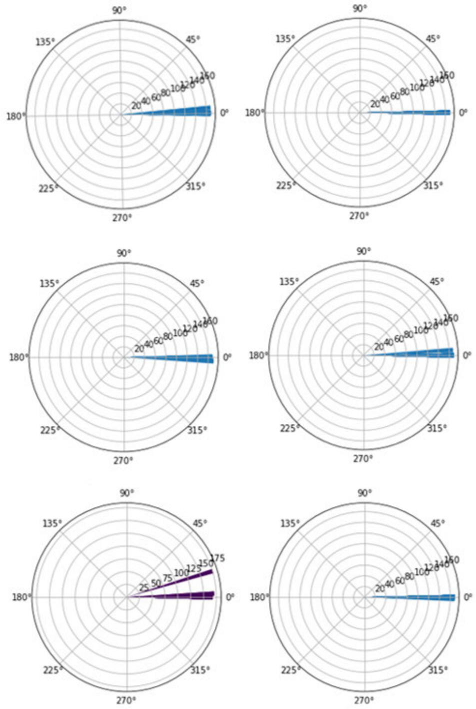

FIGURE 11. Angle distribution between the same fingers.

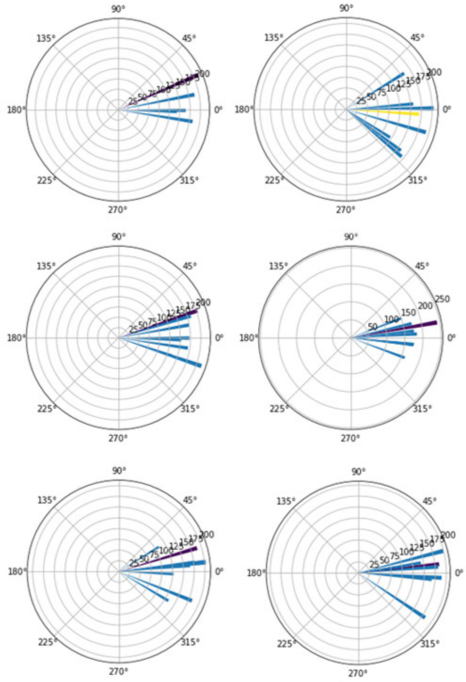

FIGURE 12. Angle distribution between different fingers.

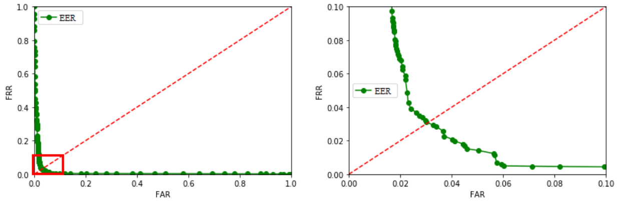

FIGURE 13. EER curve.

TABLE 5. Comparison with machine learning and state-of-art methods.

Method	Acc%	EER %
BP	81.25	-
SVM	86.87	0.14
KNN(k=3)	82.87	-
LCP	94.68	3.08
CNN [23]	93.62	11.66
CNN-CO [35]	-	2.37
CNN(triplet loss) [24]	96.16	7.03
CNN(center loss) [36]	92.07	15.99
TWO-STREAM CNN [37]	-	0.47
VEIN-CNN [31]	-	3.42
Arccosine center loss [25]	98.86	2.60
Arccosine center loss + softmax loss [25]	97.52	2.15
proposed method	98.40	2.8

learning algorithms shows that the deep forest-based algorithm for the finger-vein recognition method has a significant advantage in accuracy and is in the top ranking in EER comparison though not optimal. The deep forest-based method has a more straightforward hierarchical structure than various neural network methods and still performs well with small sample data. Also, due to the characteristics of deep forests, by growing decision trees in multiple sets of random forests, a model with good discriminative ability can be formed. The model will automatically determine the number of layers according to the gain, which is easier to train. Better results can be obtained by adjusting the number of trees in each forest.

In addition to the above advantages, the method proposed in this paper also has some limitations. The deep forest algorithm can get more accurate feature points in the finger vein image, but the limitation of recognition accuracy is still the subsequent matching algorithm. Although the ORB algorithm has a good effect on feature matching, how to use a better matching method to improve finger vein recognition accuracy is still one of the focuses of the subsequent research. Like the difficulty of face recognition in dealing with twins, when there are multiple finger vein skeletons closer to each other in the training data, the possibility of discriminative error will be high. So, many experiments are still needed to judge the feasibility and effectiveness of the method proposed in this paper on finger vein recognition. If the effect of a single model is poor, the idea of Ensemble Learning will be one of the directions of subsequent optimization.

IV. CONCLUSION

In this paper, a deep forest-based finger vein recognition method is proposed. Compared with traditional machine learning models, the method proposed in this paper fully exploits the feature representation capability of deep forest and is applicable to small sample scenarios. The algorithm mainly uses the model to obtain the coordinates of the feature points, and then uses the ORB algorithm to extract the main direction of the feature points by the FAST operator, and uses the BRIEF algorithm to binarize the encoding of the feature points, to match the features and obtain the angle information according to the Hamming distance data between the reference template image and the image to be recognized, and finally to determine the recognition results according to the matching probability obtained from the angle distribution. The experiments show that the method is more effective in vein image extraction and its recognition rate is 98.40%. The proposed method also has some limitations, how to optimize the matching algorithm to get higher recognition accuracy after acquiring finger vein feature points and how to ensure the recognition accuracy when multiple finger vein skeletons are similar. These are the focus directions of further research.

REFERENCES

[1] Z. Zhang and M. Wang, "A simple and efficient method for finger vein recognition," *Sensors*, vol. 22, no. 6, p. 2234, Mar. 2022.

[2] J. Zeng, F. Wang, J. Deng, C. Qin, Y. Zhai, J. Gan, and V. Piuri, "Finger vein verification algorithm based on fully convolutional neural network and conditional random field," *IEEE Access*, vol. 8, pp. 65402–65419, 2020.

[3] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, no. 4, pp. 194–203, 2004.

[4] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vol. E90D, no. 8, pp. 1185–1194, Aug. 2007.

[5] W. Song, T. Kim, H. Kim, J. Choi, H. Kong, and S. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, Aug. 2011.

[6] H. Qin, X. He, X. Yao, and H. Li, "Finger-vein verification based on the curvature in radon space," *Exp. Syst. Appl.*, vol. 82, pp. 151–161, Oct. 2017.

[7] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.

[8] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.

[9] T. Liu, J. B. Xie, W. Yan, P. Q. Li, and H. Z. Lu, "An algorithm for finger-vein segmentation based on modified repeated line tracking," *Imag. Sci. J.*, vol. 61, no. 6, pp. 491–502, 2013.

[10] H.-G. Kim, E. J. Lee, G.-J. Yoon, S.-D. Yang, E. C. Lee, and S. M. Yoon, "Illumination normalization for SIFT based finger vein authentication," in *Advances in Visual Computing*, vol. 7432, G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M. H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 21–30.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[12] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision—Eccv*, vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 430–443.

[13] J. Wang, H. Li, G. Wang, M. Li, and D. Li, "Vein recognition based on (2D) 2FPCA," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 6, no. 4, pp. 323–332, 2013.

[14] J.-D. Wu and C.-T. Liu, "Finger-vein pattern identification using SVM and neural network technique," *Exp. Syst. Appl.*, vol. 38, no. 11, pp. 14284–14289, 2011.

[15] Y. Xin, Z. Liu, H. Zhang, and H. Zhang, "Finger vein verification system based on sparse representation," *Appl. Opt.*, vol. 51, pp. 6252–6258, Sep. 2012.

[16] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[17] E. C. Lee, H. Jung, and D. Kim, "New finger biometric method using near infrared imaging," *Sensors*, vol. 11, no. 3, pp. 2319–2333, 2011.

[18] Z. Zhang and M. Wang, "Multi-feature fusion partitioned local binary pattern method for finger vein recognition," *Signal, Image Video Process.*, vol. 16, no. 4, pp. 1091–1099, Jun. 2022.

[19] B. Zhang, L. Zhang, D. Zhang, and L. Shen, "Directional binary code with application to PolyU near-infrared face database," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2337–2344, Oct. 2010.

[20] G. Yang, X. Xi, and Y. Yin, "Finger vein recognition based on a personalized best bit map," *Sensors*, vol. 12, no. 2, pp. 1738–1757, 2012.

[21] L. Nqk and B. P. Nguyen, "Prediction of FMN binding sites in electron transport chains based on 2-D CNN and PSSM profiles," *IEEE ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 6, pp. 2189–2197, Nov. 2021.

[22] N. Q. K. Le and Q.-T. Ho, "Deep transformers and convolutional neural network in identifying DNA N6-methyladenine sites in cross-species genomes," *Methods*, vol. 204, pp. 199–206, Aug. 2022.

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

[24] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Pattern Recognit. Lett.*, vol. 119, pp. 148–156, Mar. 2019.

[25] B. Hou and R. Yan, "ArcVein-arccosine center loss for finger vein verification," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5007411.

[26] J. Liu, Z. Chen, K. Zhao, M. Wang, Z. Hu, X. Wei, Y. Zhu, Y. Yu, Z. Feng, H. Kim, and C. Jin, "Finger vein recognition using a shallow convolutional neural network," in *Biometric Recognition*, vol. 12878, J. Feng, J. Zhang, M. Liu, and Y. Fang, Eds. Cham, Switzerland: Springer, 2021, pp. 195–202.

[27] K. Shaheed, A. Mao, I. Qureshi, M. Kumar, S. Hussain, I. Ullah, and X. Zhang, "DS-CNN: A pre-trained xception model based on depth-wise separable convolutional neural network for finger vein recognition," *Exp. Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116288.

[28] I. Boucherit, M. O. Zmirli, H. Hentabli, and B. A. Rosdi, "Finger vein identification using deeply-fused convolutional neural network," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 3, pp. 646–656, Mar. 2022.

[29] K. Shaheed, A. Mao, I. Qureshi, M. Kumar, S. Hussain, and X. Zhang, "Recent advancements in finger vein recognition technology: Methodology, challenges and opportunities," *Inf. Fusion*, vol. 79, pp. 84–109, Mar. 2022.

[30] J. Shen, N. Liu, C. Xu, H. Sun, Y. Xiao, D. Li, and Y. Zhang, "Finger vein recognition algorithm based on lightweight deep convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5000413.

[31] R. S. Kuzu, E. Piciucco, E. Maiorana, and P. Campisi, "On-the-fly finger-vein-based biometric recognition using deep neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2641–2654, 2020.

[32] K. Wang, G. Chen, and H. Chu, "Finger vein recognition based on multi-receptive field bilinear convolutional neural network," *IEEE Signal Process. Lett.*, vol. 28, pp. 1590–1594, 2021.

[33] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Freiburg, Aug. 2017, pp. 3553–3559.

[34] Z.-H. Zhou and J. Feng, "Deep forest," *Nat. Sci. Rev.*, vol. 6, no. 1, pp. 74–86, 2019.

[35] Y. Lu, S. Xie, and S. Wu, "Exploring competitive features using deep convolutional neural network for finger vein recognition," *IEEE Access*, vol. 7, pp. 35113–35123, 2019.

[36] D. Zhao, H. Ma, Z. Yang, J. Li, and W. Tian, "Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization," *Infr. Phys. Technol.*, vol. 105, Mar. 2020, Art. no. 103221.

[37] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two-stream convolutional network learning," *Neurocomputing*, vol. 290, pp. 100–107, May 2018.

ZHI CHONG WAN received the B.S. degree in electronic information engineering from Nanchang University, Jiangxi, China, in 2019. He is currently pursuing the degree in control science and engineering with the Shanghai Institute of Technology, Shanghai, China. His current research interests include machine learning and embedded development.

LAN CHEN received the M.S. degree in signal and information processing from Tongji University, Shanghai, China, in 2004, and the Ph.D. degree in astronomical technology and method science from the Shanghai Astronomy Observatory, Chinese Academy of Sciences, Shanghai, in 2010. She is currently a Professor with the Shanghai Institute of Technology, Shanghai, where she is also the Associate Dean of the School of Electrical and Electronic Engineering. Her current research interests include high-speed digital signal processing, digital terminal technology research, and signal simulation technology.

TAO WANG received the bachelor's degree in automation from the Henan Urban Construction Institute, Henan, China, in 2017, and the master's degree in bionic equipment and control engineering from the Shanghai University of Applied Sciences, Shanghai, China, in 2020. His current research interests include video chip development and software development.

GUO CHUN WAN received the M.S. and Ph.D. degrees in transportation information engineering and control from Tongji University, Shanghai, China, in 2005, and 2011, respectively. He became an Associate Professor, in 2002. He joined the Department of Electronic Science and Technology, Tongji University, in 2006. His current research interests include signal and information processing, with the emphasis on error-correcting coding, VLSI architectures, RFID strain sensor, and system on chip (SoC) design for communications and coding theory applications.