International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **1**, February **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 77–90

CURVATURE GRAY FEATURE DECOMPOSITION BASED FINGER VEIN RECOGNITION WITH AN IMPROVED CONVOLUTIONAL NEURAL NETWORK

Jia-Yi Zhao[1], Jin Gong[1], Si-Teng Ma[1] and Zhe-Ming Lu[2,*]

[1]School of Electronic Engineering
Xidian University
No. 2, South Taibai Road, Xi'an 710071, P. R. China
zhaojydy@163.com

[2]School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
*Corresponding author: zheminglu@zju.edu.cn

Received May 2019; revised September 2019

Abstract. *Finger vein recognition (FVR) is a technique for identity authentication based on finger vein images (FVIs) that are acquired using a specific device, which has become one of hot spots in the field of biometrics. The main idea of traditional FVR schemes is to directly extract features from FVIs or finger vein patterns (FVPs) and then compare features among FVIs to find the best match. However, the features extracted from FVIs contain much redundant data, while the features extracted from FVPs are greatly influenced by image segmentation methods. Recently, in order to improve the recognition rate and release the high complexity of image preprocessing, a finger vein recognition method based on deep belief network (DBN) with the features extracted from curvature gray images (CGI) has been proposed. However, the training process of DBN is somewhat time-consuming, and the background information of CGI may affect the recognition rate. In order to further improve the accuracy and speed up the training process, a new FVR algorithm based on the improved convolutional neural network (CNN) and curvature gray feature decomposition (CGFD) is proposed in this paper. First, we calculate the curvature of an FVI using a two-dimensional Gaussian template. Then we extract two gray images from the FVI with different scales and add these two images to obtain a CGI. Unlike the previous method, we further decompose this image into two components named vein curvature gray feature image (VCGFI) and background curvature gray feature image (BCGFI). Finally, using VCGFIs as input, an improved CNN is trained and used to recognize the identity of the input FVI. Experimental results show that our scheme is effective and better than traditional schemes and the previous DBN based method.*

Keywords: Finger vein recognition, Finger vein image, Convolutional neural network, Curvature gray feature decomposition, Vein curvature gray feature image

1. **Introduction.** Vein recognition [1-9] is an emerging biometric technology that uses the veins in hands for authentication. Compared with traditional biometrics such as fingerprint recognition [10], face recognition [11,12] and iris recognition [13], vein recognition has the advantages of non-contact, living body recognition, and high security. Currently, vein recognition mainly includes dorsal hand vein recognition [1,2], palm vein recognition [3,4] and finger vein recognition [5-8]. Compared with the other two, finger vein recognition (FVR) has its own advantages [14]. On the one hand, the acquisition equipment is small, and thus it has a wider application prospect. On the other hand, one can use the

DOI: 10.24507/ijicic.16.01.77

vein information from multiple fingers to further enhance the recognition performance. Furthermore, changes in the external condition of the finger do not easily affect the accuracy of vein recognition, and it is difficult to forge, making it a great advantage in places (such as banks and prisons) where the safety level is high. Therefore, finger vein recognition is receiving more and more researchers' attention. Finger vein recognition is generally divided into four main steps: image acquisition, preprocessing, feature extraction, and matching. The acquisition methods of finger vein image (FVI) can be classified into reflective imaging, transmissive imaging, and lateral transmissive imaging. Image preprocessing includes image enhancement and extraction of region of interest (ROI), which can solve the problems of uneven illumination, low contrast and noise interference to some extent, but it may cause incomplete extraction of vein lines or pseudo feature points, which may affect the recognition accuracy. The final recognition performance mainly depends on feature extraction. Therefore, feature extraction is very important.

Feature extraction techniques in FVR can be mainly classified into three categories: artificially-designed feature based [15-18], statistical characteristics analysis based [19-21] and neural networks based [22-25]. Artificially-designed features based schemes such as scale-invariant feature transformation [15], local binary pattern (LBP) [16,17] and histogram of orientated gradient (HOG) [18] do not require obtaining vein lines, but use ROI images for feature extraction, so they are simple and efficient. Lu et al. [16] adopted a local descriptor named the histogram of competitive orientations and magnitudes, where two types of local histograms are extracted and fused together, i.e., the histogram of competitive orientations and the local binary pattern histogram generated from the image of competitive magnitudes. Lee et al. [17] utilized a weighted local binary pattern with support vector machine. Htwe and Aye [18] first segmented the FVIs to remove the unwanted background and oriented the FVIs to solve the problem of finger displacement, then used the ROI localization method to accurately extract the region of vein vessels and extracted the HOG features. Because the finger vein venous structure has obvious texture and directional characteristics, local features such as LBP and HOG can distinguish well in the recognition process, but local feature extraction usually takes a long time.

The basic principle of statistical characteristics analysis based schemes such as principal component analysis (PCA) [19], linear discriminant analysis (LDA) [20] and sparse representation (SR) [21] is to transform the vein texture image into different subspaces, and generate feature vectors from various coefficients of the subspace to complete vein recognition. Wang et al. [19] combined the traditional 2D-PCA and 2D-FLD (Fisher linear discriminant) technology, Wu and Liu [20] used PCA and LDA to achieve finger vein classification, while Xin et al. [21] successfully applied SR to finger vein recognition tasks. These PCA, LDA and SR based methods can reduce the preprocessing steps and have small space occupation of feature vectors. However, they extract features from a global perspective and insufficiently describe local feature information.

Neural networks based methods such as back propagation (BP) neural network based methods and convolutional neural network (CNN) based methods [22-25] do not require manual extraction of features but adaptively learn valuable feature information based on existing data. Fang et al. [22] designed three convolutional layers to train finger vein features. Liu et al. [23] implemented finger vein recognition based on deep learning and random projection. However, neural network-based methods require a relatively high amount of data. In the case of small sample data, the trained network is not sufficiently effective.

Curvature information has been successfully applied in shape description, corner detection and target recognition. Vein images are commonly with rich lines and sharp contours

that can be described by curvature, which can be used as an important source of identification information. Because the finger vein image contains many curvature features, traditional LBP and HOG descriptors have insufficient ability to extract the curvature information as described in [24]. Thus, recently, Fang and Lu [24] proposed a finger vein recognition algorithm based on deep brief network (DBN) with features extracted by the uniform LBP descriptor from curvature gray images (CGI). In this paper, we denote that method as DBN-CGI. The DBN-CGI method adopted two successive RBM (restricted Boltzmann machine) network layers and one BP (back propagation) network layer, where the BP network is built in the last layer of the DBN. The RBM training process is used for the initialization of the weight parameter of a deep BP network in order to overcome the local optimum problem of the BP network and reduce the long training time due to the random initialization weight parameter. The finger vein image database to be tested is divided into two parts: the training set and the test set. To solve the problems of uneven illumination, low contrast and noise interference to some extent they used the histogram computed from the uniform LBP descriptor processed curvature gray image instead of the original FVI as the input of DBN. However, experimental results show that the training process of DBN-CGI [24] is somewhat time-consuming, and the background information of CGI may affect the recognition rate. The first reason is that DBN is a fully connected network, the training time of such a network is very long, and there is no efficient learning algorithm. The second reason is that the essence of the CGI is an enhanced grayscale image, but is susceptible to the collection environment such as illumination. In order to further improve the accuracy and speed up the training process, a new FVR algorithm based on the improved convolutional neural network (CNN) and curvature gray feature decomposition (CGFD) is proposed in this paper. The contributions of our work mainly lie in three aspects. First, we improve the extracted features by performing curvature gray feature decomposition (CGFD) on the CGI of the finger vein to obtain the corresponding vein curvature gray feature image (VCGFI), which can reduce the influence of background and display the information of the finger vein more, which is more conducive to network learning. Second, we adopt the improved activation function and the intermediate value pooling model, which are better than that are used in the traditional CNN. Third, our algorithm does not need to manually select features, and reduces the influence of subjective factors.

The remainder of the rest paper is organized as follows. Section 2 illustrates our vein curvature feature decomposition method to obtain the vein curvature gray feature image. Section 3 gives our recognition method based on the improved convolutional neural network structure and process. Section 4 shows the experimental results. Section 5 concludes the whole paper.

2. **Proposed Method for Extracting Vein Curvature Gray Feature Image.**

2.1. **Generation of curvature gray image.**

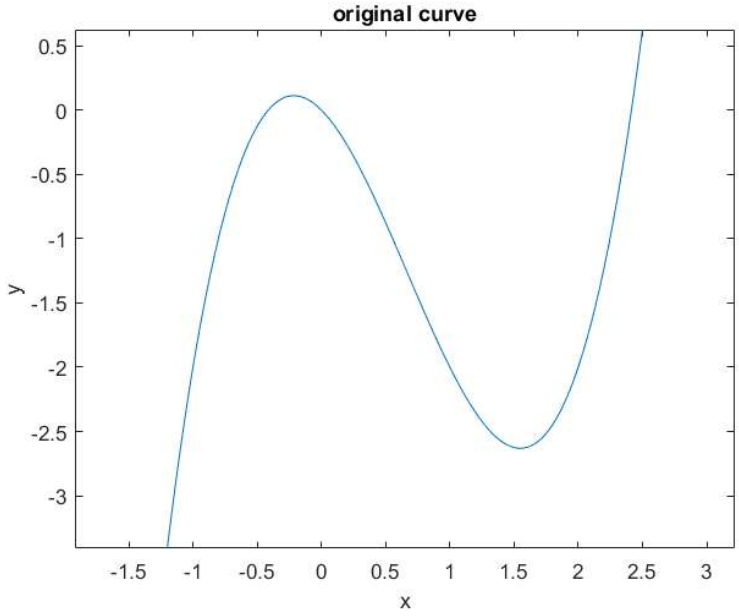
The curvature of a curve is the rate of rotation of the tangential direction of a point on the curve to the length of the arc, which can be defined by differentiation, characterizing the extent to which the curve deviates from the line. Let the equation of the curve be $y = f(x)$, and it has the second derivative, then the curvature at point x can be defined as follows:

$$c(x) = \frac{f''(x)}{(1 + [f'(x)]^2)^{3/2}} \tag{1}$$

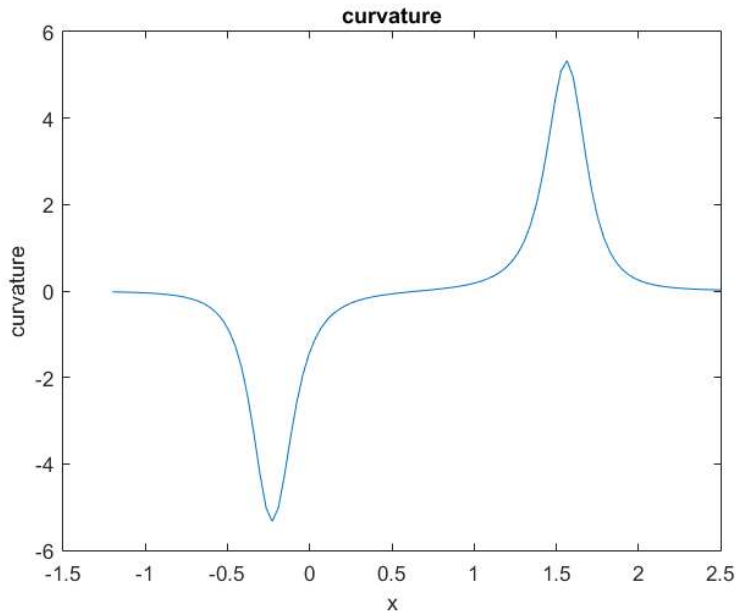
where $c(x)$ is the curvature at x, $f''(x)$ is the second derivative at x, and $f'(x)$ is the first derivative at x. If $f(x)$ is concave at x, then the curvature $c(x)$ is positive; otherwise, if $f(x)$ is convex at x, then the curvature $c(x)$ is negative [28]. A concrete example is

given in Figure 1, where Figure 1(a) shows the original curve and Figure 1(b) shows the corresponding curvature curve. From Figure 1, we can see that the curvature curve shows that if the gray distribution of the vein region is valley-shaped, then the corresponding curvature is positive, and if the gray of the background region is convex, then the corresponding curvature is close to 0 or negative. Therefore, calculating the curvature of the finger vein image can increase the discrimination between the vein region and the background region.

Miuran et al. [28] tracked the center of the finger vein according to the curvature value, and accurately extracted the center point of the finger vein, but the algorithm complexity



(a) Original curve



(b) Curvature curve

FIGURE 1. A concrete example of curvature curve

was high. Since the curvature combines the first derivative and the second derivative information, the image enhancement effect is better than the second derivative. Using the same method in [24], this paper adopts the two-dimensional Gaussian template to calculate the curvature of the finger vein image at two different scales and then combine them to obtain a gray curvature image. Here, the two-dimensional Gaussian template is given as follows:

$$g(x,y) = \frac{\exp\left[-\left(x^2+y^2\right)/\left(2\sigma^2\right)\right]}{\sqrt{2\pi\sigma^2}} \tag{2}$$

where x and y are two independent coordinate variables and σ is the standard deviation. The square template has a side length of $6\sigma+1$. The image matrix row direction is the x-axis and the column direction is the y-axis. Firstly, the first partial derivatives, the second-order partial derivatives and mixed partial derivatives of the template are used to convolution filtering the image, i.e., five partial derivative matrices $\mathbf{P}'_x$, $\mathbf{P}'_y$, $\mathbf{P}''_{xx}$, $\mathbf{P}''_{yy}$ and $\mathbf{P}''_{xy}$ corresponding to the original FVI matrix $\mathbf{I}$ can be obtained. The first-order directional derivatives and the second-order directional derivatives of the two-variables function $z = g(x,y)$ are defined as follows:

$$g'_l = g'_x \cos\theta + g'_y \sin\theta \tag{3}$$

$$g''_{l_1 l_2} = \begin{pmatrix} \cos\theta_1 & \sin\theta_1 \end{pmatrix} \begin{bmatrix} g''_{xx} & g''_{yx} \\ g''_{xy} & g''_{yy} \end{bmatrix} \begin{pmatrix} \cos\theta_2 \\ \sin\theta_2 \end{pmatrix} \tag{4}$$

where θ is the angle between the straight line l and the positive x-axis, g'_x and g'_y are the first-order partial derivatives of $g(x,y)$ respectively, g'_l is the derivative of $g(x,y)$ in the direction of the straight line l. θ_1 is the angle between the positive direction of the x-axis and the straight line l_1, and θ_2 the angle between the positive direction of the x-axis and the straight line l_2. g''_{xx} and g''_{yy} are the second-order partial derivatives of $g(x,y)$. g''_{xy} and g''_{yx} are mixed partial derivatives of $g(x,y)$. $g''_{l_1 l_2}$ is the second-order derivative of the function $g(x,y)$ first along the direction of the line l_1 and then along the direction of the line l_2. Assume the line $l_1 = l_2 = l$ with φ being the angle between the straight line l and the positive x-axis, substituting the 5 partial derivatives of the image matrix as defined in Equation (3) and Equation (4), then the first-order and second-order directional derivatives of the image matrix are given as follows:

$$\mathbf{G}'_\varphi = \mathbf{G}'_x \cos\varphi + \mathbf{G}'_y \sin\varphi \tag{5}$$

$$\mathbf{G}''_{\varphi\varphi} = \begin{pmatrix} \cos\varphi & \sin\varphi \end{pmatrix} \begin{bmatrix} \mathbf{G}''_{xx} & \mathbf{G}''_{yx} \\ \mathbf{G}''_{xy} & \mathbf{G}''_{yy} \end{bmatrix} \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \tag{6}$$

where $\varphi \in \Gamma = \{0°, 30°, 45°, 60°, 90°, 120°, 135°, 150°\}$. Substituting $\mathbf{G}'_\varphi$ and $\mathbf{G}''_{\varphi\varphi}$ into Equation (1), we can obtain the maximum curvature values in eight directions as the final curvature values of the image matrix $\mathbf{I}$ as follows:

$$\mathbf{K}_\sigma = \max_{\varphi\in\Gamma} \frac{\mathbf{G}''_{\varphi\varphi}}{\left(1+\mathbf{G}'^{\,2}_{\varphi}\right)^{3/2}} \tag{7}$$

The standard deviation σ of the two-dimensional Gaussian template determines the size of the template. A smaller template can detect more detailed vein patterns, and a larger template is beneficial for the extraction of coarse vein lines. A concrete example of the results of different standard deviation extractions is given in Figure 2. We can see that the rough vein pattern extracted by the large template is complete, and the details of the small template extraction are more clear, but containing more noise. The curvature feature matrices $\mathbf{K}_\sigma$ extracted when σ equals 1.5 and 2.5 respectively are added to obtain

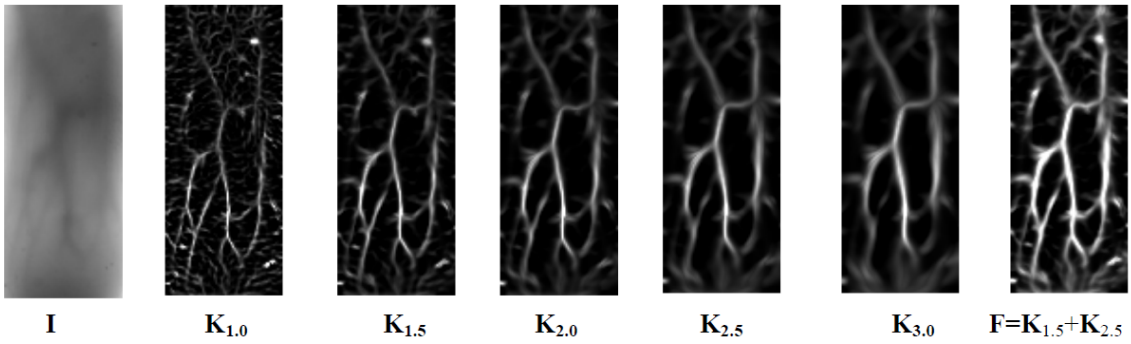


FIGURE 2. Extracting the curvature gray image $\mathbf{F}$ from the original image $\mathbf{I}$ by adding $\mathbf{K}_{1.5}$ and $\mathbf{K}_{2.5}$

the fused image $\mathbf{F} = \mathbf{K}_{1.5} + \mathbf{K}_{2.5}$ in Figure 2, and thus the fusion result contains the full-width rough vein pattern without losing the detail vein pattern. The image $\mathbf{F}$ is called curvature gray image in [24] and in this paper.

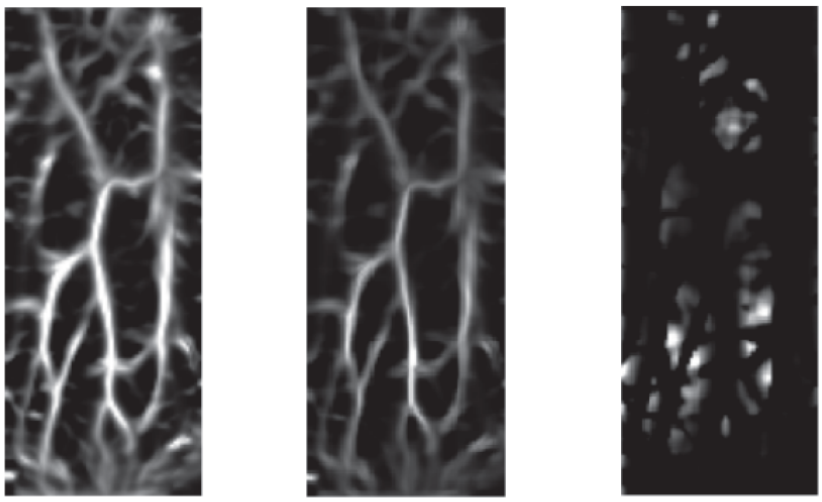
2.2. **Curvature feature gray decomposition.** We have known that calculating the curvature of the FVI can increase the discrimination between the vein region and the background region. If the gray distribution of the vein region is valley-shaped, then the corresponding curvature is positive, and if the gray of the background region is convex, then the corresponding curvature is close to 0 or negative. Thus, the positive and negative values of the curvature gray feature correspond to the vein and background regions of the image. Unlike the ULBP feature extraction method in [24], in order to enhance the vein region, this paper further normalizes the CGI $\mathbf{F}$ according to the positive and negative regions respectively, and then obtains the background curvature gray feature image (BCGFI) $\mathbf{B}$ and the vein curvature gray feature image (VCGFI) $\mathbf{V}$ respectively as follows.

$$\mathbf{B} = \frac{2550\tilde{\mathbf{F}}}{F_{\min 1} + F_{\min 2} + \cdots + F_{\min 10}} \tag{8}$$

$$\mathbf{V} = \frac{2550\tilde{\mathbf{F}}}{F_{\max 1} + F_{\max 2} + \cdots + F_{\max 10}} \tag{9}$$

where $\tilde{\mathbf{F}}$ is the normalized CGI, and $F_{\min i}$ and $F_{\max i}$ are the first ten values after sorting the elements in the normalized CGI from small to large and from large to small respectively. That is, the first 10 values of the two sets of data are taken as the upper limit of the normalization of the positive and negative regions, respectively, to reduce the influence of the exposure mutation at the edge of the image on the curvature result. Here adopting 10 values is based on experiments and experiences. The selection is made by adjusting the number of values until the obtained results are satisfying. Figure 3 gives an example to show the obtained $\mathbf{V}$ and $\mathbf{B}$ based on the image $\mathbf{F}$ in Figure 2. In the next section, we will use the image $\mathbf{V}$ as the input to train the improved CNN.

3. **Proposed FVR Based on Improved CNN.** Aiming at the problem of FVR based on the normal CNN that the recognition rate is reduced when the training samples are few and the training time is short, a finger vein recognition algorithm based on improved convolutional neural network is proposed in this section. First, the improved activation function is used to replace the traditional convolutional network activation function to improve the network generalization ability, and then the improved pooling model is used to reduce the network feature dimension.



F **V** **B**

FIGURE 3. Curvature gray image (CGI) **F** and corresponding vein curvature gray feature image (VCGFI) **V** and background curvature gray feature image (BCGFI) **B**

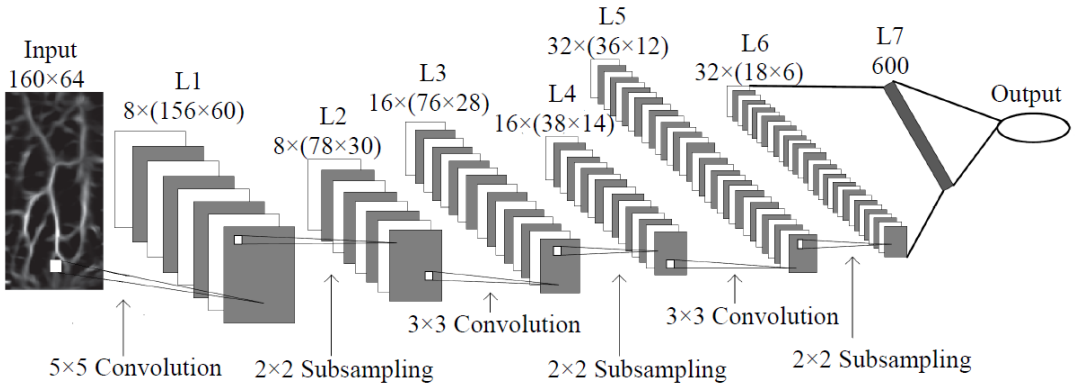


FIGURE 4. Seven-layer convolution neural network model, where L1, L3 and L5 are convolution layers, L2, L4 and L6 are pooling layers, and L7 is the full connected layer

Considering that the size of each image of the finger vein image test set used in this paper is 160×64, the size of each input vein curvature gray feature image (VCGFI) is also 160×40, thus the convolutional neural network model in this paper is shown in Figure 4. It consists of 3 convolutional layers, 3 pooling layers and 1 full connection layer. Each convolutional layer contains a number of convolution kernels of 8, 16, 32, respectively, and the convolution kernel sizes are 5×5, 3×3, and 3×3, respectively. The pooled field size of each pooling layer is 2×2. The convolutional neural network acquires the sample features by layer-by-layer convolution & pooling, and then the nonlinear mapping of the internal multi-layer network enables the model to automatically learn the essential features of the sample from the original samples, thus forming a feature extractor & classifier suitable for the recognition task. Since the convolutional neural network can extract the depth features of the input image well, the complexity of the algorithm is greatly reduced.

3.1. **Improved convolutional layer.** The convolutional layer uses the local connection and weight sharing methods to extract the primary features of the input image. The local connection means that each neuron on the convolutional layer is only connected with partial neurons of the upper layer, while the weight sharing refers to the use of a fixed-size convolution kernel as the connection strength between the current feature map neuron and the upper layer, thereby reducing the number of training parameters. Considering that finger vein image recognition is more complicated than digit recognition, in order to obtain more abundant features, this paper designs a 7-layer convolution network. Let the size of each input picture $\mathbf{V}$ be $m \times m$, the size of convolution kernel $\mathbf{H}$ be $n \times n$, the offset be b_1, the convolutional feature map be matrix $\mathbf{Q}$, and the activation function be $S(t)$, then the convolution obtains the feature of size $(m-n+1) \times (m-n+1)$. The convolution formula is as follows

$$Q_{ij} = S\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(V_{ij}H_{ij}) + b_1\right) \tag{10}$$

where Q_{ij} is the element in the matrix $\mathbf{Q}$, and V_{ij} is the input element corresponding to the convolution kernel $\mathbf{H}$ in the convolution process, not the value of the i-th row and j-th column in $\mathbf{V}$.

3.2. **Improved activation function.** The neural network uses the activation function in order to use its nonlinear factors to improve the expressive power of the network model. The activation function of the traditional neural network is mainly a sigmoid function or a hyperbolic tangent function. The sigmoid function formula is defined as:

$$S(t) = \frac{1}{1+e^{-t}} \tag{11}$$

The hyperbolic tangent function formula is defined as:

$$S(t) = \frac{e^{t}-e^{-t}}{e^{t}+e^{-t}} \tag{12}$$

Both of these functions belong to a saturated nonlinear activation function, and the functions are smooth and monotonously differentiable. The sigmoid function does not have sparsity, and the data needs to be made sparse by the penalty factor, which causes the network to converge too slowly. The Tanh function has zero antisymmetry and does not match the simulated biological neuron features. And the two activation functions have a gradient disappearing, that is, when the function tends to be saturated, the derivative is close to 0, making the network unable to be fully trained. Therefore, the unsaturated activation functions such as ReLU function and the softplus activation function have been proposed. The ReLU function formula is defined as:

$$S(t) = \max(0, t) \tag{13}$$

The softplus activation function formula is defined as:

$$S(t) = \ln(1+e^{t}) \tag{14}$$

When the value is less than 0, the ReLU function forces all values to be 0. When the value is greater than 0, it is equal to itself, so that it has sparsity to avoid the network being over-fitting, and the Softplus function can be regarded as a smoothed version of the ReLU function. These two functions are closer to the model of brain neurons. Combining the advantages of ReLU function and Softplus function, this paper proposes a new activation

function for convolutional neural networks. The formula is defined as:

$$S(t) = \begin{cases} 0 & t < 0 \\ \ln(1 + e^t) - \ln 2 & t \geq 0 \end{cases} \tag{15}$$

It can be seen that the improved activation function not only has the sparseness of the ReLU function, but also has the smoothness of the Softplus function.

3.3. **Improved pooling layer.** The pooling layer simulates the process of feature filtering of the feature map obtained by the convolution operation of the previous layer to form a more abstract target feature, which greatly reduces the feature dimension and has translation invariance. There are two conventional pooling models for convolutional neural networks, namely the average pooling model and the maximum pooling model. The average pooling model refers to calculating the average value of the pooling field and using the value as the result of the local feature map. Similarly, the maximum pooling model takes the maximum value of the pooling field as the value of the local feature map. Let the size of input feature map be $\mathbf{Q}$, the size of pooling field be $p \times p$, the offset be b_2, and the size of each moving step is p, then the pooled feature map is $\mathbf{P}$. The formulas of the average pooling model and the maximum pooling model are

$$P_{ij} = \frac{1}{p^2} \left(\sum_{i=1}^{p} \sum_{j=1}^{p} Q_{ij} \right) + b_2 \tag{16}$$

$$P_{ij} = \max_{i=1,j=1}^{p} (Q_{ij}) + b_2 \tag{17}$$

where $\max_{i=1,j=1}^{p} (Q_{ij})$ indicates the maximum value obtained in the pooling field $p \times p$ of the feature map $\mathbf{Q}$.

Because of the inadequacies of the conventional pooling model, it is impossible to extract the features in the pooled filed completely well. In order to overcome the defects, this paper chooses the intermediate value pooling method. The algorithm is an optimization algorithm based on the average pooling algorithm and the maximum pooling algorithm. The algorithm can take into account the advantages of the above two algorithms, so that the acquired feature error is small and the stability is high. The intermediate pooling algorithm expression is as follows

$$P_{ij} = \frac{1}{2p^2} \left(\sum_{i=1}^{p} \sum_{j=1}^{p} Q_{ij} \right) + \frac{1}{2} \max_{i=1,j=1}^{p} (Q_{ij}) + b_2 \tag{18}$$

Here, we give some explanations. According to the relevant theory, the error of feature extraction mainly comes from two aspects: 1) the error of the estimated variance value caused by the limited size of the neighborhood; 2) the error of the estimated mean value caused by the convolutional layer parameter error. In general, average-pooling can reduce the first type of error and retain more background information, while max-pooling can reduce the second error and retain more texture information. Thus, we use Equation (18) to take care of these two kinds of errors.

3.4. **The cost function of CNN.** Assume that the sample set consisting of m samples is $\left\{ \left(\mathbf{V}^{(1)}, o^{(1)} \right), \left(\mathbf{V}^{(2)}, o^{(2)} \right), \ldots, \left(\mathbf{V}^{(m)}, o^{(m)} \right) \right\}$, and these samples belong to n categories, and $o^{(i)}$ is the category label corresponding to the sample $\mathbf{V}^{(i)}$, then the cost function of a convolutional neural network is defined as

$$J(\mathbf{W}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\| h_{\mathbf{W},\boldsymbol{b}} \left(\mathbf{V}^{(i)} \right) - o^{(i)} \right\|^2 \right) \tag{19}$$

where $\mathbf{W}$ is the connection weight matrix between each layer, $\boldsymbol{b}$ is the offset vector, and $h_{\mathbf{W},\boldsymbol{b}}(\mathbf{V}^{(i)})$ is the output value of the last layer of the network. The network obtains the minimum value of $J(\mathbf{W},\boldsymbol{b})$ by training the parameters $\mathbf{W}$ and $\boldsymbol{b}$. Here, we use the gradient descent algorithm to solve Equation (19), and the optimization formula is

$$W_{ij}^{l} = W_{ij}^{l} - \alpha \frac{\partial}{\partial W_{ij}^{l}} J(\mathbf{W},\boldsymbol{b}) \tag{20}$$

$$b_{i}^{l} = b_{i}^{l} - \alpha \frac{\partial}{\partial b_{i}^{l}} J(\mathbf{W},\boldsymbol{b}) \tag{21}$$

where α is the learning rate. When using the BP algorithm, we first calculate the output value $h_{\mathbf{W},\boldsymbol{b}}\left(\mathbf{V}^{(i)}\right)$ of the last layer of the network, and define the difference between the value and the actual label as $\delta_{i}^{(nl)}$. Then iterate the residuals of each previous layer through the residuals of the last output layer, and then derive the partial derivatives of the two equations Equation (16) and Equation (17). The residual of the last layer of the network is:

$$\delta_{i}^{(nl)} = \frac{\partial J_{i}}{\partial Z_{i}^{(nl)}} = \frac{\partial}{\partial Z_{i}^{(nl)}} \frac{1}{2} \left\| h_{\mathbf{W},\boldsymbol{b}}\left(\mathbf{F}^{(i)}\right) - o^{(i)} \right\|^{2} \tag{22}$$

where $Z_{i}^{(nl)}$ denotes the input weighted sum of the i-th unit of the last layer.

4. **Experimental Results.** To evaluate the performance of our scheme, we use the test image database composed of 3000 finger vein images collected from 1000 persons by the finger vein device developed by our laboratory. Each person has 3 images of size 160×64. In the experiment, the training set and test set are divided according to the 8 : 2 mode. The algorithm implementation platform is MATLAB R2014a, the computer is Inter (R) Core (TM) 2 Quad CPU 2.5GHz with memory 4GB, and the operating system is 64-bit Win7.

In order to verify that the improved activation function of ReLU + Softplus is superior to other conventional activation functions, i.e., Tanh, Sigmoid, ReLU, Softplus, the experiment uses the 7-layer convolutional neural network of this paper, and the pooling layer uses the average pooling model. The number of training times is set to be 500, 1000, 1500, 2000 respectively. The experimental results are shown in Figure 5. It can be seen from Figure 5 that the performance relationship is ReLU + Softplus > ReLU > Softplus > (Sigmoid and Tanh) for any number of iteration times. Thus, the convolutional neural network using the improved activation function of this paper can obtain higher recognition accuracy than other functions. The reason is that ReLU and Softplus are better than other functions, and the combination of ReLU and Softplus can further utilize their own advantages to improve the performance, that is, the improved activation function not only has the sparseness of the ReLU function, but also has the smoothness of the Softplus function.

In order to verify that the convolutional neural network composed of the improved activation function and the intermediate value pooling method is better than the traditional convolutional neural network, Experiment 2 is designed in this paper. The experiment denotes the method using the improved activation function and the intermediate value pooling model as Method One, and denotes the method using the average pooling model and sigmoid activation function algorithm as Method Two, and denotes the method using the maximum pooling model and the sigmoid activation function as Method Three, and denotes the method using the average pooling model and the improved activation function as Method Four, and denotes the method using the maximal pooling model and the improved activation function as Method Five. Here, the finger vein image test database

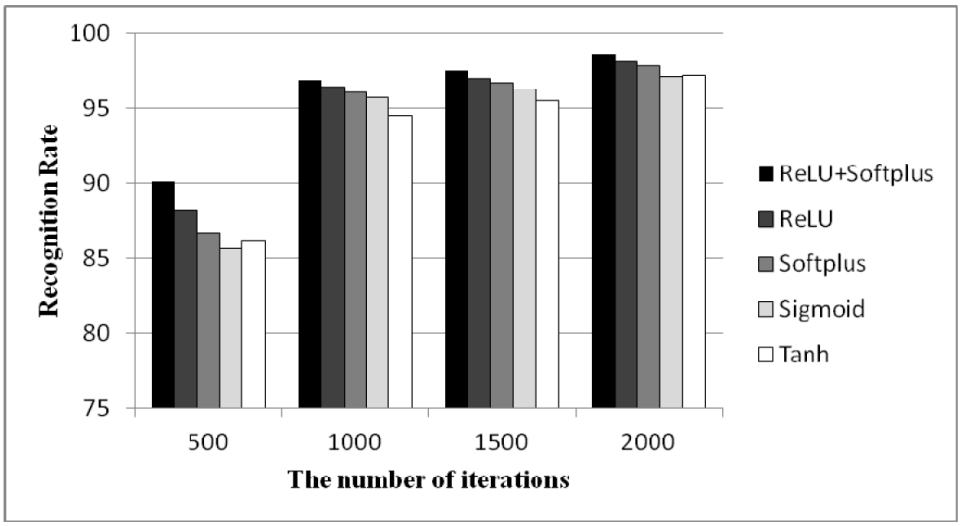


FIGURE 5. Comparisons of recognition rate using different activation functions

TABLE 1. Comparisons of the recognition rate among CNN methods using different settings

Method	Recognition Rate (%)	Improvement of Method One (%)
One	98.4	–
Two	95.7	2.7
Three	96.4	2.0
Four	96.9	1.5
Five	97.4	1.0

is divided into training samples and test samples according to 8 : 2 ratio, the number of training times is 1000. The experimental results are shown in Table 1. It can be seen from Table 1 that the convolutional neural network using the intermediate value pooling model is better than the average pooling model and the maximum pooling model. The reason is that the intermediate value pooling model can reduce two kinds of errors as explained in Section 3.3. Furthermore, compared with the traditional convolutional neural network, the convolutional neural network using the intermediate value pooling model and the ReLU + Softplus activation function has stronger generalization ability and the extracted curvature grayscale features are more conducive to classification.

Now we turn to compare our scheme with some existing schemes, including weighted LBP (WLBP) [17], HOG [18], PCA [19], SR [21], CNN1 [22], CNN2 [23], DBN-CGI [24]. Here, our method, CNN1, CNN2 and DBN-CGI are all trained 1000 times. The comparison results are shown in Table 2. From this table, we can see that our method is superior to other schemes, CNN methods are better than other non-CNN based methods. The first reason is that we improve the features extracted by performing curvature gray feature decomposition (CGFD) on the CGI of the finger vein to obtain the corresponding vein curvature gray feature image (VCGFI), which can display the information of the finger vein more, which is more conducive to network learning. The second reason is that we adopt the improved activation function and the intermediate value pooling model, which are better than that are used in the traditional CNN. In addition, our algorithm does not need to manually select features, and reduces the influence of subjective factors. While the other conventional algorithms only consider extracting part of the texture

information of the finger vein image, and contain subjective factors in feature extraction, resulting in their recognition rate is not as good as our algorithm.

In order to further prove the superiority of the proposed scheme, we adopt another test database to compare our scheme with some existing schemes, including weighted LBP (WLBP) [17], HOG [18], PCA [19], SR [21], CNN1 [22], CNN2 [23] and DBN-CGI [24]. The database is from Malaysian Institute of Technology named FV-USM containing 5,904 finger vein images. During the experiment, the training set and test set are also classified according to the 8 : 2 pattern. Here, our method, CNN1, CNN2 and DBN-CGI are all trained 1500 times. The comparison results are shown in Table 3. From this table, we can see that our method is still superior to other schemes, and deep learning (DL) based methods are better than other non-DL based methods.

Table 2. Comparisons of recognition rate among various methods

Method	Recognition Rate (%)
Weighted LBP (WLBP) [17]	94.2
HOG [18]	94.9
PCA [19]	95.1
SR [21]	95.5
CNN1 [22]	95.8
CNN2 [23]	96.4
DBN-CGI [24]	97.8
Ours	98.4

Table 3. Comparisons of recognition rate among various methods based on FV-USM

Method	Recognition Rate (%)
Weighted LBP (WLBP) [17]	93.8
HOG [18]	94.5
PCA [19]	94.7
SR [21]	95.1
CNN1 [22]	95.4
CNN2 [23]	96.0
DBN-CGI [24]	97.4
Ours	98.0

5. **Conclusions.** This paper presents a finger vein recognition algorithm based on an improved convolutional neural network using the vein curvature gray feature image (VCGFI) as input. It can be seen that the improved convolutional neural network proposed in this paper uses the improved activation function to improve the network generalization ability, and the intermediate value pooling method is adopted in the pooling layer. Experimental results demonstrate that our scheme is superior to many existing schemes. However, there are still some shortcomings in our work. First, our network can only train one finger vein database at a time. Different databases need to be retrained and consume a lot of time. This does not match the actual product type vein recognition system. There is a gap in practicality. Second, the time complexity is still higher than conventional schemes. Third, the improved convolutional neural network we used is essentially a supervised learning process. Training samples must be labeled, but in practice, labeling large amounts of data

is time consuming and laborious, and the cost is too high. Future works will concentrate on applying other CNN models or using better training schemes to further release the training burden and make the scheme suitable for real-time applications.

Acknowledgement. This work is partially supported by the Xidian University National Undergraduate Innovation and Entrepreneurship Training Program.

REFERENCES

[1] J. Wang, G. Wang, M. Li and W. Du, Hand vein recognition based on PCET, *Optik*, vol.127, no.19, pp.7663-7669, 2016.

[2] J. Wang and G. Wang, Hand-dorsa vein recognition with structure growing guided CNN, *Optik*, vol.149, pp.469-477, 2017.

[3] X. Ma and X. Jing, Palm vein recognition method based on fusion of local Gabor histograms, *The Journal of China Universities of Posts and Telecommunications*, vol.24, no.6, pp.55-66, 2017.

[4] X. Yan, W. Kang, F. Deng and Q. Wu, Palm vein recognition based on multi-sampling and feature-level fusion, *Neurocomputing*, vol.151, Part 2, pp.798-807, 2017.

[5] J. F. Yang, Y. H. Shi and G. M. Jia, Finger-vein image matching based on adaptive curve transformation, *Pattern Recognition*, vol.66, pp.34-43, 2017.

[6] W. You, W. Zhou, J. Huang, F. Yang, Y. Liu and Z. Chen, A bilayer image restoration for finger vein recognition, *Neurocomputing*, vol.348, pp.54-65, 2019.

[7] J. Yang, J. Wei and Y. Shi, Accurate ROI localization and hierarchical hyper-sphere model for finger-vein recognition, *Neurocomputing*, vol.328, pp.171-181, 2019.

[8] H. Qin, X. He, X. Yao and H. Li, Finger-vein verification based on the curvature in Radon space, *Expert Systems with Applications*, vol.82, pp.151-161, 2017.

[9] C. Xie and A. Kumar, Finger vein identification using convolutional neural network and supervised discrete hashing, *Pattern Recognition Letters*, vol.119, pp.148-156, 2019.

[10] C. Lin and A. Kumar, Contactless and partial 3D fingerprint recognition using multi-view deep representation, *Pattern Recognition*, vol.83, pp.314-327, 2018.

[11] S. Wu and D. Wang, Effect of subject's age and gender on face recognition results, *Journal of Visual Communication and Image Representation*, vol.60, pp.116-122, 2019.

[12] Y. Chu and L. Zhao, A novel log-based weighted 2DLDA algorithm for face recognition, *ICIC Express Letters, Part B: Applications*, vol.9, no.6, pp.591-598, 2018.

[13] R. Vyas, T. Kanumuri, G. Sheoran and P. Dubey, Efficient iris recognition through curvelet transform and polynomial fitting, *Optik*, vol.185, pp.859-867, 2019.

[14] X. M. Xi, L. Yang and Y. L. Yin, Learning discriminative binary codes for finger vein recognition, *Pattern Recognition*, vol.66, pp.26-33, 2017.

[15] H. Liu, L. Yang, G. Yang and Y. Yin, Discriminative binary descriptor for finger vein recognition, *IEEE Access*, vol.6, no.9, pp.5795-5804, 2018.

[16] Y. Lu, S. Wu, Z. Fang, N. Xiong, S. Yoon and D. S. Park, Exploring finger vein based personal authentication for secure IOT, *Future Generation Computer Systems*, vol.77, no.12, pp.149-160, 2017.

[17] H. C. Lee, B. J. Kang, E. C. Lee and K. J. Park, Finger vein recognition using weighted local binary pattern code based on a support vector machine, *Journal of Zhejiang University Science C*, vol.11, no.7, pp.514-524, 2010.

[18] K. S. Htwe and N. Aye, Finger vein recognition based on histogram of oriented gradients (HOG), *International Conference on Computer Applications (ICCA 2017)*, Yangon, Myanmar, 2017.

[19] J. Wang, H. Li, G. Wang, M. Li and D. Li, Vein recognition based on (2D) 2FPCA, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol.6, no.4, pp.323-332, 2013.

[20] J. D. Wu and C. T. Liu, Finger vein pattern identification using SVM and neural network technique, *Expert Systems with Applications*, vol.38, no.11, pp.14284-14289, 2011.

[21] Y. Xin, Z. Liu, H. Zhang and H. Zhang, Finger vein verification system based on sparse representation, *Applied Optics*, vol.51, no.25, pp.6252-6258, 2012.

[22] 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, 2018.

[23] Y. Liu, J. Ling, Z. Liu, J. Shen and C. Gao, Finger vein secure biometric template generation based on deep learning, *Soft Computing*, vol.22, no.7, pp.2257-2265, 2018.

[24] Z. M. Fang and Z. M. Lu, Deep belief network based finger vein recognition using histograms of uniform local binary patterns of curvature gray images, *International Journal of Innovative Computing, Information and Control*, vol.15, no.5, pp.1701-1715, 2019.

[25] J. Zhang, Z. Wu, G. Cui, P. Fan, J. Cao and M. Ning, Sensor fault diagnosis based on correntropy filter and probabilistic neural network, *ICIC Express Letters*, vol.13, no.8, pp.743-751, 2019.

[26] T. Ojala, M. Pietikainen and D. Harwood, A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition*, vol.29, no.1, pp.51-59, 1996.

[27] L. Wang, R. F. Li, K. Wang and J. Chen, Feature representation for facial expression recognition based on FACS and LBP, *International Journal of Automation and Computing*, vol.11, no.5, pp.459-468, 2014.

[28] N. Miuran, A. Nagasaka and T. Miyatake, Extraction of finger-vein patterns using maximum curvature points in image profiles, *IEICE Transactions on Information and Systems*, vol.90, no.8, pp.1185-1194, 2007.