Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **4**, Number **3**, August **2019**

Finger Vein Recognition Scheme Based on Convolutional Neural Network Using Curvature Gray Image

Jia-Yi Zhao, Jin Gong and Si-Teng Ma

School of Electronic Engineering
Xidian University
Xi'an 710126, P. R. China
zhaojydy@163.com

Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
*Correponding Author: zheminglu@zju.edu.cn

Shu-Chuan Chu and John F. Roddick

College of Science and Engineering
Flinders University
Sturt Rd, Bedford Park SA 5042, South Australia
scchu0803@gmail.com,john.roddick@inders.edu.au

Received May 2019; revised July 2019

Abstract. *Finger vein recognition technology refers to the use of finger vein angiography image authentication technology, which has became one of the hot spots of biometric identification technique. Conventional finger vein recognition technology is based on image features, and its main idea is to extract features of the overall image or features of the vein pattern. Because there are a large amount of redundant data based on features acquired from the whole finger vein image, the time complexity is high, and the features extracted from the vein pattern are greatly affected by the image segmentation algorithm. In order to improve the accuracy of the finger vein recognition algorithm under small samples, a finger vein recognition algorithm based on convolutional neural network using curvature gray images is proposed in this paper. First, we calculate the curvature of a finger vein image using a two-dimensional Gaussian template. Then we extract two gray images of the finger vein image with different scales and add these two images to obtain the final curvature gray image. Using curvature gray images as input, an improved convolutional neural network is trained and used to recognize the identity of the input curvature gray image. Experimental results show that our scheme is effective and better than existing schemes.*
Keywords: Finger vein recognition, Convolutional neural network, Curvature gray image.

1. **Introduction.** Finger vein recognition (FVR) extracts the internal features of the finger and has the advantages of non-contact, in vivo, uniqueness and stability [1]. 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 bank and prison) where the safety level is high. The conventional FVR system mainly includes four parts:

image acquisition, image preprocessing, feature extraction and matching. Finger Vein Image (FVI) acquisition can be divided into reflective imaging, transmissive imaging, and lateral transmissive imaging [2]. Image preprocessing includes image enhancement and region of interest extraction [3, 4], 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 recognition accuracy.

The most important feature extraction techniques in finger vein recognition are mainly divided into three types: artificial design features, statistical characteristics analysis and neural networks. Artificial design features include scale-invariant feature transformation [5], Local Binary Pattern (LBP) [6, 7], Histogram of Orientated Gradient (HOG) [8], Principal Component Analysis (PCA) [9], Linear Discriminant Analysis (LDA) [10] and Sparse Representation (SR) [11], Back Propagation (BP) neural networks and Convolutional Neural Network (CNN) [12, 13]. In 2017, Lu et al. [6] proposed a new fusion description operator based on LBP and Gabor filters. This method has better recognition performance, but the obtained features are up to 190 dimensions, which makes the feature extraction and matching time longer. Because the finger vein venous structure has obvious texture and directional characteristics, local features such as LBP [6, 7] and HOG [8] can distinguish well in the recognition process, but local feature extraction usually takes a long time. Wu and Liu [10]used PCA and LDA to achieve finger vein classification, and Xin et al. [11] also successfully applied SR to finger vein recognition tasks. However, methods such as PCA [9], LDA [10], and SR [11] extract features from a global perspective and insufficiently describe local feature information. Fang et al. [12] designed three convolutional layers to train finger vein features. Lin et al. [13] implemented finger vein recognition based on deep learning and random projection. The main advantage of the neural network-based methods is that they do not require manual extraction of features but adaptively learns valuable feature information based on existing data. However, they require a relatively high amount of data. In the case of small sample data, the network trained is not sufficiently effective.

Because the finger vein image contains a lot of curved texture features, LBP and HOG have insufficient ability to extract curvature information. Thus, this paper proposes a finger vein recognition algorithm based on convolutional neural network and curvature gray image(CGI). The curvature gray image is derived from the computation result between the two-dimensional Gaussian template and the original finger vein image at two scales. We use CGI instead of the original FVI as the input of CNN in order to improve the recognition rate by solving the problems of uneven illumination, low contrast and noise interference to some extent.

2. **Extraction of Curvature Gray Image.** A positive curvature indicates that the curve is concave at that point, and a negative curvature indicates that the curve is convex there [14].The curvature formula is as follows:

$$k = \frac{f''}{(1 + f'^2)^{3/2}} \tag{1}$$

Where k is the curvature of a given point, f'' is the second derivative at that point, and f' is the first derivative at that point. A concrete example is given in Fig. 1 to show the gray and curvature curves at the cross-sectional line of a given finger vein image. The gray curve at the black cross-section and the corresponding curvature curve show 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. In Fig.1, the black cross-section passes through 3 veins,

and thus the corresponding curvature curve has 3 positive peaks. Therefore, calculating the curvature of the finger vein image can increase the discrimination between the vein region and the background region.

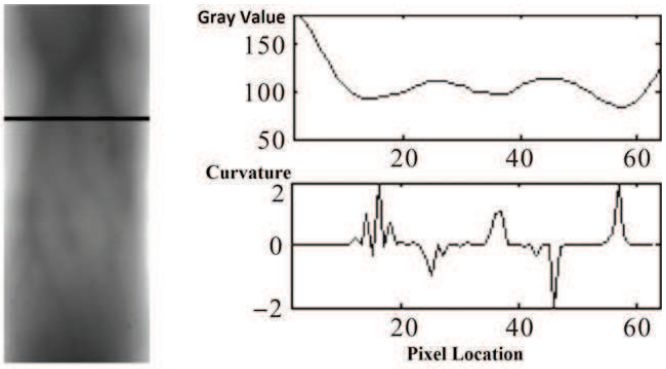

FIGURE 1. Gray (upper-right) and curvature (bottom-right) curves of the cross-sectional line of the vein image (left).

Miura et al. [14] 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 was high. Since the curvature combines the first derivative and the second derivative information, the image enhancement effect is better than the second derivative. In summary, most of the finger vein recognition algorithms currently only extract features for vein regions, ignoring the available information contained in the background region. To this end, this paper uses 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. Thus, the image background area information and the vein area information can be combined. Curvature gray images are then used as input to train the CNN. Here, the two-dimensional Gaussian template is given as follows:

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathtt{exp}[-(x^2+y^2)/(2\sigma^2)] \tag{2}$$

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 $\mathtt{P}'_x$, $\mathtt{P}'_y$, $\mathtt{P}''_{xx}$, $\mathtt{P}''_{yy}$ and $\mathtt{P}''_{xy}$ corresponding to the original FVI matrix $\mathtt{I}$ 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:

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

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

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 Eq. (3) and Eq.(4), then the first-order and second-order directional derivatives of the image matrix are given as follows:

$$\mathtt{G}'_{\varphi} = \mathtt{G}'_{x} \cos\varphi + \mathtt{G}'_{y} \sin\varphi \tag{5}$$

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

In the above formula, φ is taken as from the set $\Gamma = \{0°, 30°, 45°, 60°, 90°, 120°, 135°, 150°\}$. Substituting the direction derivatives $\mathtt{G}'_{\varphi}$ and $\mathtt{G}''_{\varphi\varphi}$ of the image matrix $\mathtt{I}$ into Eq. (1), and obtaining the maximum curvature values in eight directions as the final curvature values of the image matrix $\mathtt{I}$:

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

The standard deviation σ of the two-dimensional Gaussian template will determine the size of the template. The small template can detect more detailed vein patterns, and the large template is beneficial for the extraction of coarse vein lines. The results of different standard deviation extractions are shown in Fig. 2. It can be seen from Fig. 2 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 $\mathtt{K}_{\sigma}$ extracted when σ is 1.5 and 2.5 respectively are added to obtain the fused image $\mathtt{F} = \mathtt{K}_{1.5} + \mathtt{K}_{2.5}$ in Fig. 2, and thus the fusion result contains the full-width rough vein pattern without losing the detail vein pattern.

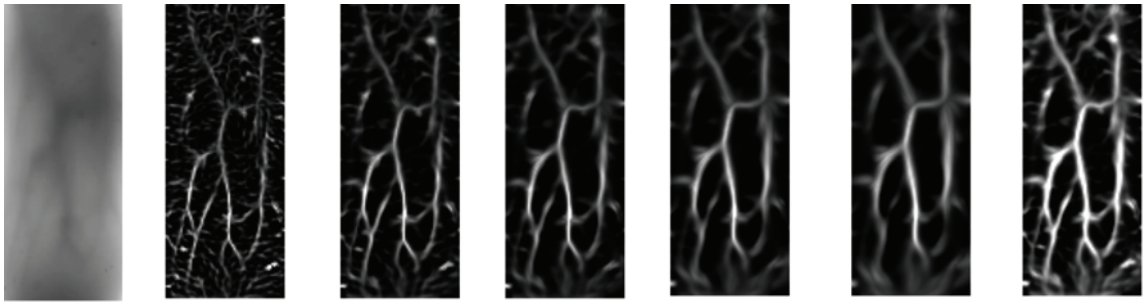

FIGURE 2. Curvature extraction results for different standard deviations σ.

3. **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.

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 curvature gray image is also 160×40, thus the convolutional neural network model in this paper is shown in Fig. 3. 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.

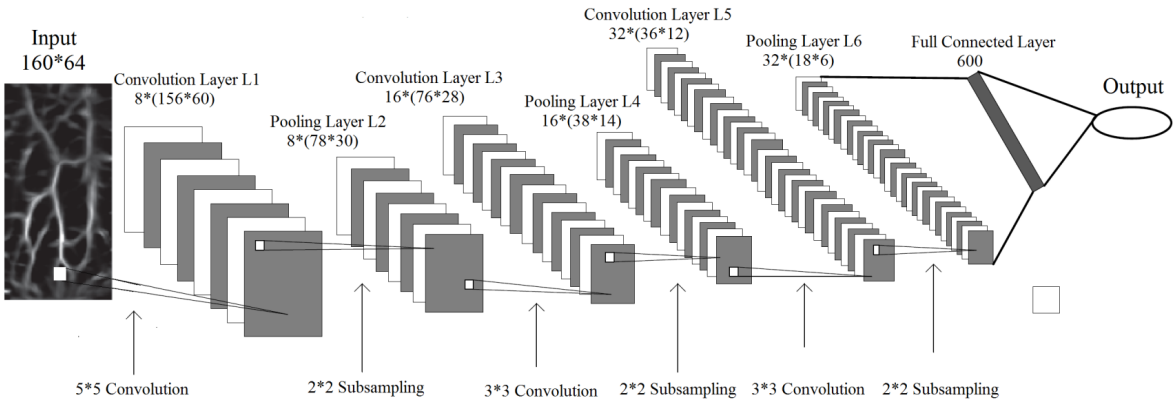

FIGURE 3. Seven-layer convolution neural network model.

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 F be $m \times m$, the size of convolution kernel $\mathtt{H}$ be $n \times n$, the offset be b_1, the convolutional feature map be matrix $\mathtt{Q}$, 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

$$Q_{ij} = S\left(\sum_{i=1}^{n}\sum_{j=1}^{n}(F_{ij}H_{ij}) + b_1\right) \tag{8}$$

Where, Q_{ij} is the element in the matrix $\mathtt{Q}$, and F_{ij} is the input element corresponding to the convolution kernel $\mathtt{H}$ in the convolution process, not the value of the i-th row and j-th column in $\mathtt{F}$.

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:

$$S(t) = \frac{1}{1+e^{-t}} \tag{9}$$

The hyperbolic tangent function formula is defined as:

$$S(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \tag{10}$$

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:

$$S(t) = \mathtt{max}(0, t) \tag{11}$$

The softplus activation function formula is defined as:

$$S(t) = \ln(1 + e^t) \tag{12}$$

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:

$$S(t) = \begin{cases} 0 & t < 0 \\ \ln(1 + e^t) - \ln 2 & t \geq 0 \end{cases} \tag{13}$$

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 $\mathtt{Q}$, 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 $\mathtt{P}$. The formulas of the average pooling model and the maximum pooling model are

$$P_{ij} = \frac{1}{p^2}\left(\sum_{i=1}^{p}\sum_{j=1}^{p} Q_{ij}\right) + b_2 \tag{14}$$

$$P_{ij} = \max_{i=1,j=1}^{p}\left(Q_{ij}\right) + b_2 \tag{15}$$

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

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

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

$$J(\mathtt{W}, \boldsymbol{b}) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\|h_{\mathtt{W},\boldsymbol{b}}(\mathtt{F}^{(i)}) - o^{(i)}\|^2\right) \tag{17}$$

Where $\mathtt{W}$ is the connection weight matrix between each layer, $\boldsymbol{b}$ is the offset vector, and $h_{\mathtt{W},\boldsymbol{b}}(\mathtt{F}^{(i)})$ is the output value of the last layer of the network. The network obtains the minimum value of $J(\mathtt{W}, \boldsymbol{b})$ by training the parameters $\mathtt{W}$ and $\boldsymbol{b}$. Here, we use the gradient descent algorithm to solve Eq.(17), and the optimization formula is

$$W_{ij}^{l} = W_{ij}^{l} - \alpha\frac{\partial}{\partial W_{ij}^{l}}J(\mathtt{W}, \boldsymbol{b}) \tag{18}$$

$$b_{i}^{l} = b_{i}^{l} - \alpha\frac{\partial}{\partial b_{i}^{l}}J(\mathtt{W}, \boldsymbol{b}) \tag{19}$$

Where α is the learning rate. When using the BP algorithm, we first calculate the output value $h_{\mathtt{W},\boldsymbol{b}}(\mathtt{F}^{(i)})$ 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 Eq.(14) and Eq.(15). The residual of the last layer of the network is:

$$\delta_i^{(nl)} = \frac{\partial J_i}{\partial Z_i^{(nl)}} = \frac{\partial}{\partial Z_i^{(nl)}}\frac{1}{2}\|h_{\mathtt{W},\boldsymbol{b}}(\mathtt{F}^{(i)}) - o^{(i)}\|^2 \tag{20}$$

Where $Z_i^{(l)}$ denotes the input weighted sum of the i-th unit of the l-th layer, and $Z_i^{(nl)}$ denotes the input weighted sum of the i-th unit of the last layer.

4. **Experimental Results.** The test image database is 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 mean pooling model. The number of training times is set to be 500, 1000, 1500, 2000 respectively. The experimental results are shown in Fig. 4. It can be seen from Fig. 4 that the traditional convolutional neural network using the improved activation function of this paper can obtain higher recognition accuracy than other functions. 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

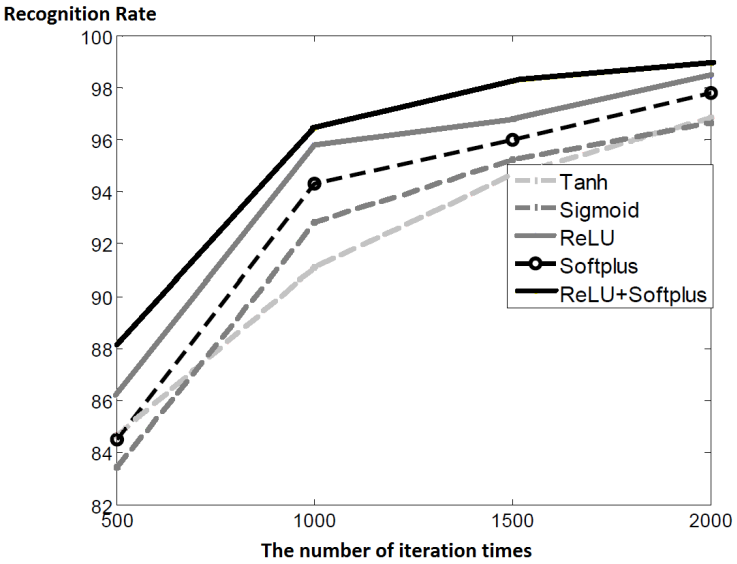

FIGURE 4. Comparisons of recognition rate using different activation functions.

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 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 compared with the traditional convolutional neural network, the convolutional neural network using the intermediate value pooling model and the ReLU+Softplus activation function have stronger generalization ability and the extracted curvature grayscale features are more conducive to classification.

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

Method	Recognition Rate(%)	Improvement of Method One (%)
One	96.8	-
Two	94.7	2.2
Three	95.4	1.5
Four	95.9	0.9
Five	96.4	0.3

Now we turn to compare our scheme with some existing schemes, including weighted LBP(WLBP)[7], HOG[8], PCA[9], SR[11], CNN1[12], CNN2[13]. Here, our method, CNN1 and CNN2 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.

5. **Conclusions.** This paper presents a finger vein recognition algorithm based on an improved convolutional neural network using curvature gray images as input. It can

TABLE 2. Comparisons of recognition rate among various methods.

Method	Recognition Rate(%)
LBP(WLBP)[7]	94.2
HOG[8]	94.9
PCA[9]	95.1
SR[11]	95.5
CNN1[12]	95.8
CNN2[13]	96.4
Our	96.8

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.

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

REFERENCES

[1] Y. Qiu, and Q. Lv, Research on finger vein recognition technology, *Proc. of International Conference on Electrical and Control Engineering*, Wuhan, P. R. China, pp. 2937–2940, 2010.

[2] J. Lee, J. Lim, S. Moon, Y. Park, K. Kim, S.-J. Lee, and J.-H. Lee, Performance comparison of illumination methods for finger-vein imaging and liveness detection, *Microsystem Technologies*, vol. 24, no. 12, pp. 4955—4964, 2018.

[3] A. Banerjee, S. Basu, S. Basu, and M. Nasipuri, ARTeM: a new system for human authentication using finger vein images, *Multimedia Tools and Applications*, vol. 77, no. 5, pp.5857–5884, 2017.

[4] M. Wang, and D. Tang, Region of interest extraction for finger vein images with less information losses, *Multimedia Tools and Applications*, vol. 76, no. 13, pp.14937–14949, 2016.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.886–893, 2005.

[15] 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.

[16] Y. Pang, Y. Yuan, X. Li, and J. Pan, Efficient HOG human detection, *Signal Processing*, vol. 91, no. 4, pp.773–781, 2011.

[17] M. S. M. Asaari, S. A. Suandi, and B. A. Rosdi, Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics, *Expert Systems with Applications*, vol. 41, no. 7, pp. 3367–3382, 2014.