ArcVein-Arccosine Center Loss for Finger Vein Verification

Borui Hou and Ruqiang Yan, *Senior Member, IEEE*

***Abstract*—This article proposes a new loss function termed arccosine center loss, which can learn interclass and intraclass information simultaneously, to improve the discriminative ability of convolutional neural networks for finger vein verification. Specifically, the purpose of arccosine center loss is to reduce intraclass distance and increase the interclass distance through network model training. With the combination of softmax loss and arccosine center loss, the proposed network model can extract features with the interclass dispensation and intraclass compactness, which improves the discriminative ability of learned features. Experimental studies have confirmed the effectiveness and efficiency of the proposed method for finger vein verification.**

***Index Terms*—Biometrics, convolutional neural network (CNN), deep learning, finger vein, metric network.**

I. Introduction

BIOMETRICS, which uses individual physical or physiological characteristics for verification, have been widely used in recent years. As one of the promising biometric methods, finger vein verification has received widespread attention because of its high security [1]. The finger vein verification possesses some advantages, such as living body identification, high specificity, and convenience. It has been confirmed that the finger vein patterns for each finger of each person are unique [2]–[5].

Generally, finger vein verification involves image acquisition, preprocessing, feature extraction, and verification. The image preprocessing usually consists of region of interest (ROI) extraction, normalization, and enhancement. Several preprocessing methods have been proposed to overcome the problem of low quality (e.g., the uneven illumination, noise, and low contrast between the veins and background) in the acquired image. For example, a multiscale method based on Ridgelet transform was utilized for finger-vein image enhancement [6]. A hybrid finger vein image enhancement method based on gray-level grouping and circular Gabor filter was proposed in [7], where gray-level grouping effectively improved the contrast of finger vein image, and then circular Gabor filter was introduced for vein ridge extraction. Combined the advantages of Gabor filters and Retinex filters, a finger vein enhancement method was introduced in [8], where a fuzzy-based fusion method was used to determine the weights in the combination of Gabor and Retinex filtered images. Yang *et al.* [9] designed a Gaussian weight spatial curve transformation method to extract the network pattern of finger vein and achieved improvement in finger vein matching. However, these nonadaptive enhancement methods are not specifically designed for finger vein images and cannot handle noise close to vein structures.

After the image preprocessing, various methods have been applied to extracting hand-crafted finger vein features. For example, Miura *et al.* [10] utilized the repeat line-tracking method to extract global finger vein patterns by calculating the difference between the center pixel and its neighbors in a cross-sectional profile. Inspired by the repeat line-tracking method, the maximum curvature method was proposed to extract the maximum curvature information of the finger vein [11]. Lee *et al.* [12] used the minutia-based alignment and local binary pattern (LBP) method to extract specific finger vein features for verification which reduced the equal error rate (EER) significantly. Meng *et al.* [13] introduced a direction-based local descriptor called LDC to represent the local gradient orientation information, which achieved better performance than the LBP method. Based on the LBP method, a personalized best bit map method was proposed in [14], where only the best bits extracted through LBP were used for matching. Beng *et al.* [15] used the pixel-pattern-based texture feature (PPBTF) method to extract finger vein features, and principal component analysis was applied to further selecting the main PPBTFs before classification.

However, the hand-crafted feature-based methods generally lack distinctiveness and robustness. For example, some of the existing methods have difficulty in extracting features from low-quality images caused by complex lighting conditions, poor quality of infrared light [16], cold weather, fat finger, or poorly designed devices [17]. Moreover, different finger conditions, such as rotation, flip, and shift, add a negative impact on finger vein verification.

The deep learning-based approach, which has the powerful ability to extract features without any prior knowledge, has been successfully applied in finger vein verification to overcome such limitations. For example, Ahmad Radzi *et al.* [18] used a convolutional-subsampling architecture-based convolutional neural network (CNN) model for finger vein verification and achieved a high recognition rate. Hong *et al.* [19] introduced a pre-trained VGG-16 model in finger vein verification, where the classical VGG-16 model was fine-tuned with training images. Similar to Hong *et al.*'s [19] work, Raghavendra *et al.* [20] proposed a pre-trained deep CNN

Manuscript received December 19, 2020; revised January 28, 2021; accepted February 16, 2021. Date of publication February 25, 2021; date of current version March 15, 2021. This work was supported by the Graduate Fellowship Award of the IEEE Instrumentation and Measurement Society. The Associate Editor coordinating the review process was Zheng Liu. *(Corresponding author: Ruqiang Yan.)*

Borui Hou is with the School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: boruihou@seu.com.cn).

Ruqiang Yan is with the School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China, and also with the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: ruqiang@seu.edu.cn).

Digital Object Identifier 10.1109/TIM.2021.3062164

model based on Alex-net, and the existing Alex-net model was extended by seven additional layers to improve the performance of finger vein verification. The deep CNN model was also designed by Wan *et al.* [21] and Qin and El-Yacoubi [22] to represent and/or recover finger vein images. To learn more discriminative features, Hu *et al.* [23] used a deep CNN-based feature matching strategy for finger vein verification. Compared with the state-of-the-art CNN architectures, Das *et al.* [24] designed a CNN with optimized kernel size and small learning rate, where the performance of the model was improved and the test error was reduced. However, the capacity of the widely used finger vein database is not large enough, which makes the deep learning model sensitive to overfitting during the training. In fact, standard deep networks usually do not cover interclass and intraclass information simultaneously, thus making it difficult to perform a robust classification when training data is not enough.

In order to improve the feature learning ability, metric learning, which is proposed to measure the distance between sample features, is introduced in finger vein verification. The purpose of metric learning is to reduce intraclass distance and increase the interclass distance through training. For example, Huang *et al.* [25] adopted contrastive loss in deep CNN to identify finger vein images. In their work, intraclass samples and interclass samples were paired as input data. The training guidance of the proposed method was to maximize the distance of interclass pairs and minimize the distance of intraclass ones. Through the metric learning module, the trained network could learn features with interclass and intraclass information, which improved the ability of finger vein verification. Xie and Kumar [26] introduced a triplet loss-based metric learning model and generated the features as hashing code, which made finger vein verification easier. As an improvement of the contrastive loss model, the input data of the triplet loss model are triple samples, which include two intraclass samples and one interclass sample. These metric learning methods can effectively extract features, including both interclass and intraclass information, thus making improved achievement in the field of finger vein verification. However, it is hard to choose appropriate and paired samples. If the generated input samples are inappropriate, training the model will be hard. Without paired samples, a new metric learning method called center loss was proposed in [27]–[29] and achieved great success in feature extraction. Zhao *et al.* [30] introduced a center loss-based metric learning model, making each class easier to be distinguished. In the center loss, the Euclidean distance between each data and its corresponding category center is measured and constantly updated during training and paired requirement for input data is not necessary.

Inspired by these prior efforts and metric learning methods, this article presents a novel arccosine center loss, which will be used in a deep CNN model for finger vein verification. The softmax loss and newly designed arccosine center loss are applied jointly in the training process of the proposed model with the goal to improve the performance of finger vein verification. The arccosine center loss is designed based on that the cosine distance of features should be close in the same category and far away across different categories. Motivated by center loss and cosine distance, the proposed arccosine center loss learns the corresponding category center and tries to minimize the cosine distance of the features and its centers. Meanwhile, the proposed arccosine center loss is easy to converge and is stable when updating the parameters during training. The proposed method in this article achieves state-of-the-art performance on the publicly available USM database, SDU database, HKPU database, and our own collected database. The main contribution of this article can be summarized as follows.

1) This work attempts to introduce a newly designed loss function, termed arccosine center loss, in finger vein verification. Instead of Euclidean distance between features, cosine distance has been introduced to improve the performance of finger vein verification.
2) A specially designed CNN architecture is proposed for finger vein verification in this article. Compared with other CNN-based methods discussed earlier, the proposed efficient channel attention residual network (ECA-Resnet) model with arccosine center loss improves the practicality of finger vein verification.
3) The proposed method achieves good performance on the publicly available USM database [31], SDU database [32], HKPU database [1], and our own collected database, which proves that features learned from the proposed method, are more discriminative.

The remainder of this article is organized as follows: the theoretical background of ECA-Resnet and arccosine center loss is provided in Section II. Then, the framework of the proposed method is introduced in Section III, followed by experiments and analysis in Section IV. The conclusions are finally drawn in Section V.

II. Theoretical Background

In the field of biometrics, the deep learning method adds a huge boost to feature extraction and representation. The purpose of the deep learning-based method is to obtain the feature representation via directly optimizing the geometric distribution of features in embedding space. In this article, the ECA-Resnet model is used for feature extraction.

A. ECA-Resnet

In this article, the residual network module and ECA module are combined and adopted for feature extraction. A typical depth of the Resnet can be regarded as a combination of convolutional blocks, and each block contains several convolutional layers, batch normalization (BN), or pooling layers.

Suppose F_n represents the output of the nth block and $H_n(.)$ denotes nonlinear transformation. $H_n(.)$ is the designed mapping function, including the operation of convolution, BN, activation function, or pooling. The connection of the outputs of the (*n-1*) blocks and the input of the nth block is shown as

$$F_n = H_n(F_{n-1}) + F_{n-1}. \quad (1)$$

The addition F_{n-1} is copied from the learned *(n–1)*th blocks and called identity mapping. To the extreme, if an identity mapping is optimal, it is easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

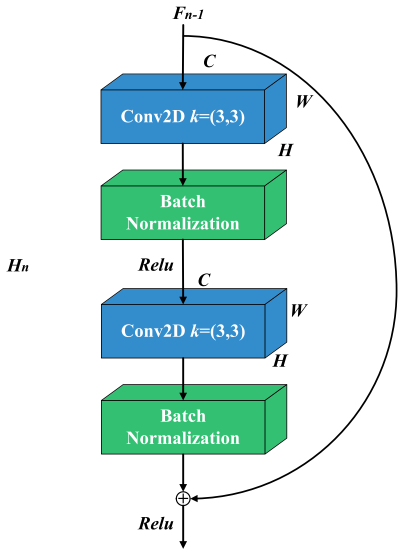

Fig. 1. Overview of Resnet module.

A typical Resnet module is shown in Fig. 1, where the rectified linear units (ReLUs) function works as the activation function in Resnet because it has good capability on solving the vanishing gradient problem in backpropagation for training.

To improve the performance of the deep CNNs, the ECA module [34] is introduced in this study. Given the aggregated feature $f \in R^C$ without dimensional reduction, the ECA module can be expressed as

$$E(f) = w * f \tag{2}$$

where w represents the weight matrix of channel attention and can be learned only considering the interaction between f_i and its k neighbors, that is,

$$w_i = \text{sigmoid}\left(\sum_{j=1}^{k} w^j f_k^j\right), \quad f_k^j \in \Omega_i^k \tag{3}$$

where Ω_i^k is the set of k adjacent channels of f_i. Note that (3) can be implemented by a 1-D convolution with a kernel size of k, and the ECA module can be shown as

$$w = \text{conv}1D(f) \tag{4}$$

where conv1D represents 1-D convolution using the sigmoid function as an active function. Using the given channel dimension C, the kernel size k can be adaptively determined by

$$k = \left|\frac{\log_2 C}{\gamma} + \frac{b}{\gamma}\right|_{\text{odd}} \tag{5}$$

where $|*|_{\text{odd}}$ indicates the nearest odd number of the results. The parameters γ and b are set as 2 and 1 according to the experiments in [34].

An overview of the ECA module and ECA-Resnet block is shown in Fig. 2. After aggregating the convolution features using global average pooling without dimensionality reduction, the ECA module performs a 1-D convolution to learn channel attention.

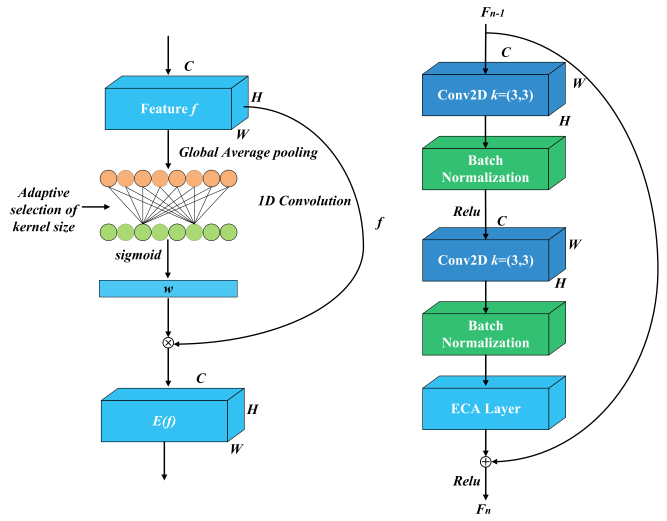

Fig. 2. Framework of ECA module (left) and ECA-Resnet block (right).

B. Softmax Loss

Softmax loss can be considered as the probability of several category selections and the probability sum is equal to 1. The expression of softmax loss is

$$L_s = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{Z_i}^T f_i + b_{z_i}}}{\sum_{j=1}^{l} e^{W_j^T f_i + b_j}} \tag{6}$$

where $f_i \in R^d$ is the feature extracted from the ith sample whose category is z_i. W_j is the jth column of weight, $W \in R^d$, and b_j denotes the bias. The category number and batch size are l and N, respectively. Then, the parameter W_j of loss function can be updated using gradient descent as

$$W_{j+1} = W_j - \alpha \frac{\partial L_s}{\partial W_j} = W_j - \alpha \frac{\partial L_s}{\partial f_i} \frac{\partial f_i}{\partial W_j} \tag{7}$$

where α is the learning rate.

Softmax loss is widely used in deep learning-based classification and tends to maximize the interclass distance, while the constraint of intraclass distance is neglected. In fact, standard deep networks with softmax loss only encourage the separability of features and usually focus on the interclass information and ignore the intraclass information, which may cause misclassification [27]. In order to reduce the risk of misclassification, the center loss and arccosine center loss method are introduced in this article, which will be discussed in Section II-C.

C. Center Loss

Center loss [27] aims to minimize the intraclass distance by learning a feature center of each class and keep the features of different classes separable. The introduction of this effective loss function can improve the discriminative ability of the model. Let $D(\cdot)$ be a metric function to measure the distance between centers and features, and the center loss is defined as

$$L_c = D\big(f_i, c_{l_i}\big). \tag{8}$$

Suppose that $f_i \in R^d$ is the deep feature of the ith sample and $c_{l_i} \in R^d$ represents the center of features of category l_i.

Therefore, $D(\cdot)$ represents the squared Euclidean distance function in center loss and is expressed as

$$D\big(f_i, c_{l_i}\big) = \frac{1}{2}\sum_{i=1}^{N} \| f_i - c_{l_i} \|^2. \tag{9}$$

The center loss mainly aims to reduce the distance between the feature and its corresponding center, and thus, it has effective improvement in minimizing the intraclass distance. However, the center loss still uses the Euclidean distance, which only focuses on the absolute difference between specific features. Such difference is reflected in the numerical values of different dimensions of the individual vector. This means Euclidean distance is sensitive to the absolute value of specific features.

Like the Euclidean distance, the cosine distance is often used for similarity measurement in the field of verification. Focused on distinguishing difference from the direction of the feature vector, cosine distance is insensitive to the absolute value of the specific features. The cosine distance considers the feature vector as a whole, thus avoiding the influence of individual features. The idea of metric loss based on cosine distance has been introduced in [35]–[37] to modify softmax loss and achieved high performance in deep face recognition. Inspired by these prior ideas, the proposed loss optimizes cosine distance between different features to improve the center loss.

D. Arccosine Center Loss

Intuitively, it is a direct method to enhance the discriminative ability of features by using the cosine distance function. The goal of our arccosine center loss method is to obtain the feature representation via directly optimizing the geometric distribution of features in cosine (angular) embedding space. In our proposed method, the cosine distance instead of the Euclidean distance is introduced in measuring the distance between features and centers. Different from center loss, the center here represents a vector, which means the direction of the corresponding category features. Let $c_{l_i} \in R^d$ represents the center of features of category l_i, $C = \{c_{l_1}, c_{l_2}, \ldots, c_{l_n}\}$ represents the center set. Usually, the normalized vector of ν can be denoted as $\tilde{\nu}$: $\tilde{\nu} = (\nu)/(\|\nu\|)$. Let θ_i represents the angle between features and corresponding center, the cosine distance function is defined as

$$L_{\mathrm{ac}} = \sum_{i=1}^{N} (\arccos\theta_i) \tag{10}$$

where

$$\cos\theta_i = \frac{f_i^T * c_{l_i}}{\|f_i\| \|c_{l_i}\|} = \tilde{f}_i^T * \tilde{c}_{l_i}. \tag{11}$$

The cosine value has one to one mapping from cosine space to angular space when the angular is in $[0, \pi]$ [29]. Therefore, the angle between two vectors can be directly used to measure the distance. As indicated in (11), the features and centers need to be $L2$-normalized to get the cosine value or angle.

It is inefficient and impractical if category centers keep updating during the training time. In order to improve efficiency, a mini-batch-based update scheme was proposed in [27]. In each iteration, the centers can be considered as the average of corresponding categories, so that the amount of calculation is greatly reduced. Given a mini-batch with d samples, L_{ac} can be represented as the summation of each sample's loss L_i

$$L_i = \arccos\theta_i = \arccos\big(\tilde{f}_i^T * \tilde{c}_{l_i}\big). \tag{12}$$

The gradients of L_{ac} with respect to $\tilde{f}_i$ is defined as

$$\frac{\partial L_{\mathrm{ac}}}{\partial \tilde{f}_i} = \frac{\partial L_i}{\partial \tilde{f}_i} = -\frac{\tilde{c}_{l_i}}{\sqrt[2]{1 - \big(\tilde{f}_i^T * \tilde{c}_{l_i}\big)^2}}. \tag{13}$$

From (11), $\tilde{f}_i^T * \tilde{c}_{l_i} = \cos\theta_i \epsilon [0.1]$ and the angle θ_i is in $[0, \pi]$, $\sin\theta_i \ \epsilon\ [0.1]$, then (14) can be represented as

$$\frac{\partial L_i}{\partial \tilde{f}_i} = -\frac{\tilde{c}_{l_i}}{\sqrt[2]{1 - \cos{\theta_i}^2}} = -\frac{\tilde{c}_{l_i}}{\sin\theta_i}. \tag{14}$$

The update scheme of centers is defined as

$$\Delta c_j = \frac{\sum_{i=1}^{d} \delta(l_i = j)}{1 + \sum_{i=1}^{d} \delta(l_i = j)} \cdot \frac{\tilde{c}_{l_i}}{\sin\theta_i} \tag{15}$$

where $\delta(l_i = j) = 1$ if l_i is equal to j, otherwise $\delta(l_i = j) = 0$.

The arccosine center loss is introduced to guide the feature learning and reduce the intraclass variation. Moreover, the interclass variation also needs to be considered in feature learning. The combination of the proposed arccosine center loss and softmax loss can improve the classification capability. The softmax loss plays a key role in increasing the interclass distance, while the arccosine center loss works for reducing the intraclass distance. The total loss used in this study is designed as

$$L = L_s + \lambda L_{\mathrm{ac}} \tag{16}$$

where λ is utilized to keep them in balance. The algorithm of the proposed method is shown in Table I. In the experiment part, experiments are conducted to discuss the influence of λ.

III. Methodology

A. Designed Framework

Combining the advantage of arccosine center loss and softmax loss, the proposed deep network model for finger vein verification is shown in Fig. 3, which has the potential to improve the discrimination capacity of the network. Specifically, the ECA-Resnet model is introduced to extract features under the guidance of the proposed arccosine loss and softmax loss, and the pooling layer is performed to lower deep feature dimensions and make it more compact. The target of the proposed network is to minimize the combined loss. The smaller the loss value is, the stronger the discriminate ability of this network will be.

In the proposed network, the image size is 224 × 224 × 3, and the original one channel grayscale image is expanded to the three-channel image by copying it twice as the input.

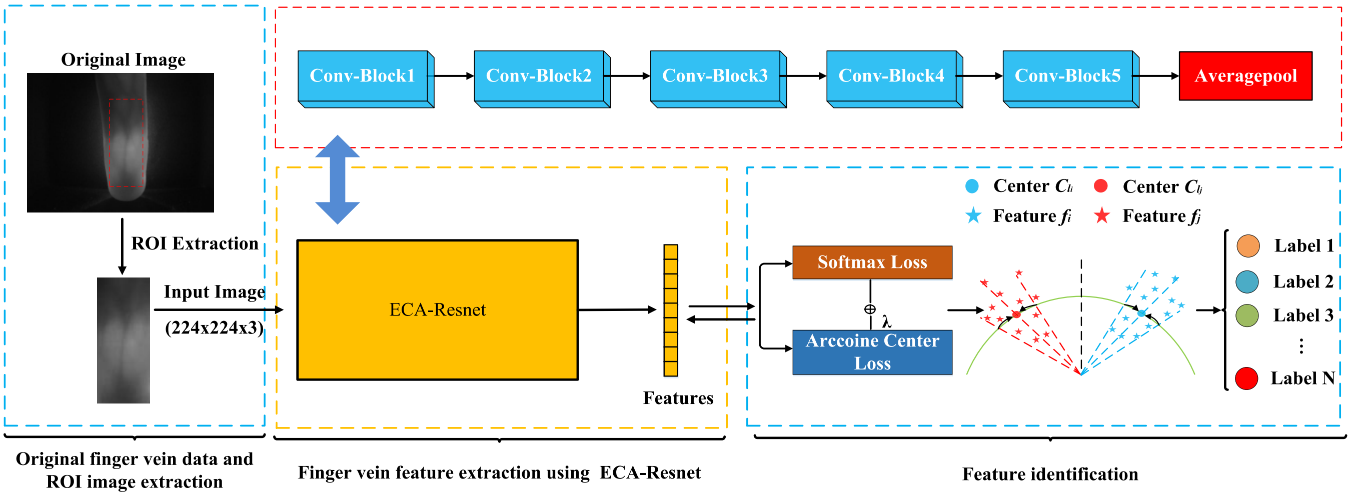

Fig. 3. Framework of our proposed model.

TABLE I
ALGORITHM OF PROPOSED METHOD

```
Algorithm. The algorithm of proposed method
Require: extracted deep features $f_i$, parameters $W$ and $C:\{c_j | j = 1, 2, \ldots, n\}$ in loss layer. Learning rate $\alpha^t$. The number of critic iterations $n$.
1)  Initialize parameters $W$ and $C$.
2)  For $t$ in range $(1, n)$ do:
3)  Compute the joint loss: $L = L_s + \lambda L_{ac}$
4)  Compute the backpropagation error :$\frac{\partial L_i}{\partial \tilde{f}_i} = \frac{\partial L_s}{\partial f_i^t} + \lambda \frac{\partial L_c}{\partial f_i^t}$
5)  Update the parameters $W$ :$W^{t+1} = W^t - \alpha^t \frac{\partial L_s^t}{\partial w}$
6)  Update the parameters $c_j$: $c_j^{t+1} = c_j^t - \Delta c_j^t$
7)  end
```

As the output of fully connected layer, the deep feature size is designed as 512. In this article, the parameters of the proposed network structure are listed in Table II.

B. Training and Testing the Proposed Network

During the training, the stochastic gradient descent algorithm with Nesterov momentum 0.9, the learning rate of 0.01 divided by 10 after 30 epochs, and *L2* weight decay of 5×10^{-4} is applied for training. The batch size is 64 and the maximum epoch number is 100. The main purpose of considering the learning rate and batch size is to keep the balance between the learning ability and training time cost.

IV. EXPERIMENTAL STUDY

A. Databases

The performance of this proposed finger vein verification approach is evaluated by four databases: USM database, SDU database, HKPU database, and our own collected database. The details of the four databases are listed in Table III.

TABLE II
STRUCTURE PARAMETERS OF PROPOSED NETWORK

Layer name	***Output size***	***ECA-Resnet***
		Kernel, filters, stride
Input	224×224×3	/
Conv-Block1	112×112×64	7×7, 64, stride 2
		3×3 max pool, stride 2
Conv-Block2	56×56×64	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$ ECA-layer
Conv-Block3	28×28×128	$\begin{bmatrix}3\times3, 128\\3\times3,128\end{bmatrix}\times2$ ECA-layer
Conv-Block4	14×14×256	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$ ECA-layer
Conv-Block5	7×7 ×512	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$ ECA-layer
	7×7	Average pool
Fc	1×512	/

The ROI image of these original images was extracted for reducing the training cost. In this experiment, the ROI images are extracted using the ROI method applied in [38]. The size of the ROI image is normalized to 224 × 224 × 3 for training, and the original one-channel grayscale image is expanded to the three-channel image by copying it twice as the input.

To improve the performance and prevent overfitting, image transformation, including flip, rotation, shift, shear, and zoom, as listed in Table IV, is used as the data augmentation method to enhance the flexibility of the training images during each training epoch, however, without increasing the number of the training image set. During the training process, the input

TABLE III

DETAILS OF FINGER VEIN DATABASES

Database	***Fingers***	***Per finger***	***Sessions***	***Total***
USM	492	12	2	5904
HKPU	210	12	2	2520
SDU	636	6	1	3816
OWN	1386	6	1	8316

Note: HKPU database consists of 156 subjects [1], but only 105 subjects have been collected in two sessions. In this paper, the finger vein images from 105 subjects (totally 210 classes) which has two sessions are used for experiments.

TABLE IV

DATA AUGMENTATION PARAMETERS

Methods		**Parameters**
Flip	Vertical	True
	Horizontal	False
Rotation (degree range for random rotation)		2.5
Shift (Range scale in [-1,1])	Width	0.05
	Height	0.05
Shear range (Range in [-1,1])		0.05
Zoom range (range in [1- Zoom,1+ Zoom range])		0.05

training images are randomly transformed before going into each training epoch. In this way, the number of training images is not changed.

B. Experiment Setting

Different sets of experiments are designed to evaluate the performance of the proposed method. Both the closed-set protocol and open-set protocol are evaluated. The closed-set protocol, which is more suitable for the assessment of those scenarios that have no new user registered, uses partial samples of all of the classes in the database for training, and the remaining samples for testing. In the open-set protocol, only a part of the classes is used for training, and the remaining classes are used only for testing.

The accuracy (Acc) and EER are introduced to compare the performance of finger vein-based biometric systems. EER comprehensively measures and balances the performance of false rejection rate (FRR) and false acceptance rate (FAR). FAR and FRR are defined as follows:

$$\mathrm{FAR} = \frac{\text{Number of matching scores in false acceptance}}{\text{Number of matching scores}} \quad (17)$$

$$\mathrm{FRR} = \frac{\text{Number of matching scores in false rejection}}{\text{Number of matching scores}}. \quad (18)$$

Actually, there is a threshold that varies from 0 to 1 for FAR and FRR evaluation. When the threshold changes, the FRR and FAR will also change. However, they will change in the opposite direction. The EER is the result when FAR is equal to FRR.

TABLE V

PERFORMANCE OF PROPOSED METHOD ON CLOSED-SET PROTOCOL (%)

λ	Closed-set Protocol							
	USM		SDU		HKPU		OWN	
	Acc	EER	Acc	EER	Acc	EER	Acc	EER
0.01	99.95	0.09	99.15	1.47	99.42	0.95	98.97	2.14
0.05	99.96	0.11	99.48	1.33	99.43	0.99	99.01	1.99
0.1	99.97	0.11	99.22	1.36	99.54	0.69	99.04	2.03
0.5	***99.99***	***0.03***	99.50	0.99	99.59	0.68	99.15	1.92
1	99.98	0.05	***99.56***	***0.72***	***99.60***	***0.55***	99.21	1.90
2	99.97	0.04	99.55	0.89	99.54	0.82	***99.20***	***1.59***
5	99.97	0.04	99.26	1.35	98.98	1.64	99.09	2.23

TABLE VI

PERFORMANCE OF PROPOSED METHOD ON OPEN-SET PROTOCOL (%)

λ	Open-set Protocol							
	USM		SDU		HKPU		OWN	
	Acc	EER	Acc	EER	Acc	EER	Acc	EER
0.01	99.71	0.41	98.62	2.41	98.22	2.39	97.80	4.27
0.05	99.61	0.55	98.65	2.48	98.31	2.44	97.79	4.51
0.1	***99.79***	***0.25***	98.56	2.40	98.49	2.11	97.75	4.26
0.5	99.76	0.21	99.01	2.01	98.76	1.61	***97.90***	***3.97***
1	99.66	0.60	***99.25***	***1.53***	98.70	1.82	97.78	4.04
2	99.53	0.75	99.01	2.07	***99.07***	***1.30***	97.49	4.76
5	99.42	0.89	98.91	2.14	99.01	1.82	97.08	5.52

For verification, a twofold cross-validation method is used in the experiments to obtain a reliable estimate. In the closed-set protocol, the total database is randomly divided into two equal parts for cross validation. In the SDU database and our own collected database, half of the data (three images per class) is regarded as training data and the rest is used for testing. As for the USM and HKPU databases, which have two sessions, one session is used for training and the other session is used for testing. For the division in the open-set protocol, only one-half of the classes are randomly used for training, and the remaining classes are used only for testing.

It should be noted that the experiments are carried out by using Python with Pytorch framework on a workspace computer with the following specifications: Intel Core Xeon CPU E5-2609, 1.70 GHz × 16, RAM 256 GB, and GPU NVIDIA TITAN Xp 12 GB.

C. Parameter Selection

To evaluate the influence of parameter λ on the performance of the proposed network, the parameter λ varies from 0.01 to 5 in the experiments. The average value of all folds measures the overall performance of the proposed method, which are listed in Table V (closed-set protocol) and Table VI (open-set protocol).

As shown in Table V, the performances of the proposed method are good in the finger vein verification on the closed-set protocol. It shows the selection of appropriate parameter λ helps to enhance the verification ability on the closed-set protocol. For example, the proposed method achieves the best performance for the USM database with an

TABLE VII

Performance of Different Loss Functions on Closed-Set Protocol (%)

Loss Function	Closed-set Protocol							
	USM		SDU		HKPU		OWN	
	Acc	EER	Acc	EER	Acc	EER	Acc	EER
Contrastive Loss [25]	99.86	0.42	96.38	6.92	99.60	0.62	98.56	3.45
Triplet Loss [26]	99.00	1.46	96.02	7.31	99.47	0.87	97.08	5.79
Center Loss [27]	99.79	0.47	98.84	2.87	99.47	1.18	99.11	2.27
STNRC Loss [39]	99.89	0.27	96.70	6.30	99.56	0.75	97.05	7.49
Circle Loss [40]	96.43	5.89	91.65	16.09	94.39	8.83	93.84	11.69
N Pairs Loss [41]	94.53	7.71	91.15	15.53	94.78	7.59	93.52	11.88
MS Loss [42]	99.86	0.33	97.92	4.42	99.69	0.54	98.55	3.24
NCA Loss [43]	96.49	5.05	91.82	13.41	97.41	3.69	93.63	10.96
Softmax	99.82	0.36	98.30	3.54	99.05	1.33	98.86	2.40
Softmax + center loss	99.98	0.06	99.47	0.83	96.62	0.73	98.74	1.48
Arccosine center loss	99.90	0.18	98.86	2.60	99.58	0.63	***99.63***	***1.10***
The proposed loss	***99.99***	***0.03***	***99.56***	***0.72***	***99.60***	***0.55***	99.20	1.59

TABLE VIII

Performance of Different Loss Functions on Open-Set Protocol (%)

Loss Function	Open-set Protocol							
	USM		SDU		HKPU		OWN	
	Acc	EER	Acc	EER	Acc	EER	Acc	EER
Contrastive Loss [25]	99.23	1.38	97.38	5.04	98.86	1.35	97.67	4.43
Triplet Loss [26]	97.89	3.51	95.95	7.69	97.48	3.95	95.81	8.04
Center Loss [27]	97.99	2.85	96.35	7.29	98.24	1.88	97.43	4.70
STNRC Loss [39]	99.33	1.11	97.39	5.11	98.61	1.67	97.21	5.87
Circle Loss [40]	95.80	6.79	93.48	11.10	94.01	8.70	92.82	14.51
N Pairs Loss [41]	92.16	11.38	91.43	13.97	90.57	13.08	90.80	17.42
MS Loss [42]	99.46	1.01	97.53	4.79	98.77	1.63	98.03	4.21
NCA Loss [43]	95.17	7.24	93.80	10.41	92.47	9.89	90.86	16.31
Softmax	99.50	0.83	98.37	2.98	97.63	3.30	97.48	4.81
Softmax + center loss	99.78	0.49	99.00	2.01	98.15	2.37	98.24	3.73
Arccosine center loss	99.63	0.67	99.07	2.07	***99.44***	1.83	***98.33***	***3.53***
The proposed loss	***99.79***	***0.25***	***99.25***	***1.53***	99.07	***1.30***	97.90	3.97

Acc of 99.99% and EER of 0.03% when the parameter λ is set as 0.5. Similarly, it achieves the best performance for the SDU database with Acc of 99.56% and EER of 0.72%, and the HKPU database with Acc of 99.60% and EER of 0.55%, when the parameter λ is 1. As for our own database, the proposed method achieves the best performance with an Acc of 99.20% and EER of 1.59% when the parameter λ is set as 2.

As for the open-set protocol, the best performance for the USM database is with an Acc of 99.79% and EER of 0.25% when the parameter λ is 0.1. The proposed method achieves the best performance for the SDU database with an Acc of 99.25% and EER of 1.53% when the parameter λ is 1. Similarly, it achieves the best performance for the HKPU database with Acc of 99.07% and EER of 1.30% when the parameter λ is 2, and for our own database with Acc of 97.90% and EER of 3.97% when the parameter λ is 0.5.

D. Comparison With Other Loss Function

In the study, some commonly used metric learning loss functions, such as the paired-based loss [25]–[30], [39]–[42], and proxy-based losses [43], have been chosen for comparison with the proposed loss function.

The paired-based loss learns features from real sample pairs, such as contrastive loss [25], signal-to-noise ratio contrastive loss [39], triplet loss [26], center loss [27]–[30], circle loss [40], N-pair loss [41], and multi-similarity (MS) loss [42], and so on. The advantage of the pair-based loss is that it is simple and can accurately represent the current model. Nevertheless, the disadvantage of paired-based loss is that the amount of calculation is large and the information is limited. It only uses the current mini-batch information and ignores the performance of the model in the overall data.

The proxy-based loss uses a proxy to represent category features or sample features, such as softmax, and NCA loss [43]. The proxy is a very broad concept, and it can be considered as the feature parameters learning during the training. The advantage of proxy-based loss is that it can learn global information from the performance of the model in overall data. However, it may not be able to accurately express the current model because it learns step by step from historical information.

The experiments' results shown in Tables VII and VIII can be used to compare the performance of the proposed loss function with other loss functions. Note that all the experiments are trained on the same network model with parameter setting being listed in Table II and the parameters of these loss functions are based on their references. The cosine distance is adopted for feature matching.

TABLE IX

PERFORMANCE OF DIFFERENT METHODS ON USM DATABASE (%): CLOSED-SET PROTOCOL

Ref	*Method*	*Acc*	*EER*
[23]	CNN (Large Margin Softmax Loss)	/	**0.06***
[24]	CNN	98.06	3.33
[26]	CNN(triplet loss)	97.54	4.23
[30]	CNN (Center loss & dynamic regularization)	98.17	1.05
[44]	MWCDE	/	0.307*
/	Arccosine center loss	99.90	0.18
/	Arccosine center loss + softmax loss	***99.93***	***0.03***

* *The result is directly cited from reference.*

TABLE X

PERFORMANCE OF DIFFERENT METHODS ON SDU DATABASE (%): CLOSED-SET PROTOCOL

Ref	*Method*	*Acc*	*EER*
[23]	CNN (Large Margin Softmax Loss)	/	**0.46***
[24]	CNN	93.62	11.66
[26]	CNN(triplet loss)	96.16	7.03
[30]	CNN (Center loss & dynamic regularization)	92.07	15.99
[46]	CNN-CO	/	2.37*
/	Arccosine center loss	98.86	2.60
/	Arccosine center loss + softmax loss	***98.51***	***0.72***

* *The result is directly cited from reference.*

TABLE XI

PERFORMANCE OF DIFFERENT METHODS ON HKPU DATABASE (%): CLOSED-SET PROTOCOL

Ref	*Method*	*Acc*	*EER*
[24]	CNN	95.01	4.21
[26]	CNN(triplet loss)	94.53	9.12
[30]	CNN (Center loss & dynamic regularization)	90.08	17.18
[45]	Gabor convolution network	/	1.67*
/	Arccosine center loss	99.58	0.63
/	Arccosine center loss & softmax loss	***99.60***	***0.55***

* *The result is directly cited from reference.*

TABLE XII

PERFORMANCE OF DIFFERENT METHODS ON USM DATABASE (%): OPEN-SET PROTOCOL

Ref	*Method*	*Acc*	*EER*
[19]	CNN(pre-trained VGG16)	93.67	5.96
[22]	Deep representation based feature extraction	/	1.42*
[23]	CNN (Large Margin Softmax Loss)	/	**0.76***
/	Arccosine center loss	99.63	0.67
/	Arccosine center loss & softmax loss	***99.79***	***0.25***

* *The result is directly cited from reference.*

TABLE XIII

PERFORMANCE OF DIFFERENT METHODS ON SDU DATABASE (%): OPEN-SET PROTOCOL

Ref	*Method*	*Acc*	*EER*
[19]	CNN(pre-trained VGG16)	94.57	4.28
[23]	CNN (Large Margin Softmax Loss)	/	**1.20***
/	Arccosine center loss	99.07	2.07
/	Arccosine center loss & softmax loss	***99.25***	***1.53***

* *The result is directly cited from reference.*

As shown in Tables VII and VIII, the proposed method achieves better performance on all of the databases, especially on USM and HKPU databases. This may be attributed to these two databases containing more finger vein images for single subjects. It can be found that the paired-based loss, especially MS loss, achieves better performance than NCA loss. This indicates that paired-based loss is more suitable for finger vein verification.

On closed-set protocol, the proposed arccosine loss performs better performance than center loss and other paired-based losses. Compared MS loss with the proposed loss, the EER is decreased from 0.33% to 0.03% for the USM database, from 4.42% to 0.72% for the SDU database, from 3.24% to 1.59% for our collected database. The combined loss achieves an EER of 0.55% on the HKPU database and it is closed to the best performance with an EER of 0.54% (MS loss).

On open-set protocol, the proposed arccosine loss performed also better performance than center loss and other paired-based losses. Compared MS loss with arccosine center loss, the EER is decreased from 1.01% to 0.67% for the USM database, from 4.79% to 2.07% for the SDU database, from 1.63% to 1.83% for the HKPU database, and from 4.21% to 3.53% for our collected database. The combined loss achieves an EER of 1.30% on the HKPU database and an EER of 3.97% on our own collected database. This demonstrates the optimization on cosine distance works better than Euclidean distance.

E. Comparison With Other Methods on Closed-Set Protocol

In order to evaluate the performance of the proposed method, comparison with several state-of-the-art identification methods [23], [24], [26], [30], [44], [45], [46] on a closed-set protocol of three publicly available databases are conducted and the results are listed in Tables IX–XI.

As shown in Tables IX–XI, the proposed method achieves EER of 0.03% (USM database), 0.72% (SDU database), and 0.55% (HKPU database). This is owing to the effect of the proposed arccosine center loss. The results indicate that the proposed method achieves high accuracy on three databases and can catch up with the best state-of-the-art method.

Compared with the results in [23] (i.e., CNN with large margin softmax loss), the proposed method performs better in the USM database which decreases the EER from 0.06% to 0.03%. While in the SDU database, the proposed method can also catch up with a small gap (0.26%). Similar to [23], the proposed method provided an alternative way for reducing the intraclass distance.

F. Comparison With Other Methods on Open-Set Protocol

The open-set protocol is also adopted in this article to evaluate the performance of the proposed method with various state-of-the-art methods [19], [22], [23] in each database, and the results are summarized in Tables XII–XIV. It is worth noting that the verification challenges with the open-set protocol are greater than those with the closed-set protocol.

TABLE XIV

PERFORMANCE OF DIFFERENT METHODS ON HKPU DATABASE (%): OPEN-SET PROTOCOL

Ref	***Method***	***Acc***	***EER***
[19]	CNN(pre-trained VGG16)	96.02	4.34
[22]	Deep representation based feature extraction	/	2.70*
/	Arccosine center loss	***99.44***	***0.83***
/	Arccosine center loss & softmax loss	99.07	1.30

* *The result is directly cited from reference.*

TABLE XV

ASSESSING THE VALIDITY OF CLOSED-SET PROTOCOL (EER%)

Databases	USM	SDU	HKPU	OWN
Proposed model + softmax loss	0.76	3.98	2.78	2.69
Proposed model + Arccosine center loss	0.53	2.97	1.19	2.27
Proposed model + combined loss	0.33	1.47	1.03	2.43
Proposed model + Data augmentation + softmax loss	0.36	3.54	1.33	2.40
Proposed model + Data augmentation + Arccosine center loss	0.18	2.60	0.63	***1.10***
Proposed model + Data augmentation + combined loss	***0.03***	***0.72***	***0.55***	1.59

TABLE XVI

ASSESSING THE VALIDITY OF OPEN-SET PROTOCOL (EER%)

Databases	USM	SDU	HKPU	OWN
Proposed model + softmax loss	1.61	3.49	3.81	5.23
Proposed model + Arccosine center loss	0.74	4.07	1.88	4.70
Proposed model + combined loss	0.31	2.41	1.47	5.28
Proposed model + Data augmentation + softmax loss	0.83	2.98	3.30	4.81
Proposed model + Data augmentation + Arccosine center loss	0.67	2.07	***0.83***	***3.53***
Proposed model + Data augmentation + combined loss	***0.25***	***1.53***	1.30	3.97

As shown in Tables XII–XIV, the proposed combined loss achieves an EER of 0.25% (USM database), 1.53% (SDU database), and 1.30% (HKPU database). It is worth noting that the proposed arccosine loss performs better than the combination of arccosine loss and softmax loss, which achieves an EER of 0.83% on the HKPU database. The results indicate that the proposed method achieves high accuracy on three databases and can catch up with the best state-of-the-art method.

Compared with the results in [23] (i.e., CNN with large margin softmax loss), the proposed method performs better in the USM database which decreases the EER from 0.76% to 0.25%. While in the SDU database, the proposed method can also catch up with a small gap (0.33%). Compared with the results in [22], the proposed method performs better in the USM database and HKPU database

G. Validity Evaluation and Inference Time

To evaluate the effectiveness of the proposed method, the validity evaluation experiments are conducted on both closed-set and open-set protocols, and the results are listed in Tables XV and XVI. First, the network performances without data augmentation are evaluated in Tables XV and XVI. Then, the performances of the network using data augmentation, arccosine center loss, and combined loss are evaluated with all training tricks to show the attribution of the proposed loss function.

TABLE XVII

INFERENCE TIME OF DIFFERENT DEEP LEARNING-BASED FINGER VEIN IDENTIFICATION METHODS

Ref	[19]	[20]	[21]	[24]	***Proposed Method***
Time(ms)	3.21	1.24	1.83	1.20	**0.38**

It can be observed that the introduction of the proposed model and data augmentation strategies are both effective for finger vein verification in the closed-set and open-set protocols. The method proposed in this article can further improve the performance, thus demonstrating the superiority of this method. In addition, the combination of arccosine center loss and softmax loss not always performs better than arccosine center loss. Considering that our own collected database is larger than other databases, the learned feature may not good enough to represent the image.

The proposed method is also compared with four CNN-based finger vein methods [19]–[21], [24] in terms of inference time, as shown in Table XVII. The inference time is obtained by calculating the average predicting time on a single image of the SDU database. Experiments show that the proposed method reduces by more than 90% of the inference time of [19]. In fact, the introduction of the ECA module has an insignificant influence on inference time.

V. CONCLUSION

In this article, a new metric loss function, termed arccosine center loss, is designed for finger vein verification. The arccosine center loss function inherits the advantage of the center loss and performs more powerful than it. The combination of softmax loss and arccosine center loss is introduced as a training goal to optimize the ECA-Resnet network model for finger vein feature extraction. Compared with other methods, the new loss function focuses on learning discriminative finger vein features in angle space and achieves state-of-the-art results on three publicly available finger vein databases. The experiment results show the effectiveness of the proposed center loss-based deep CNN and it can obtain competitive results as compared with other Euclidean distance-based methods.

In the experiment, the center loss is hard to be restricted and needs to be constrained with appropriate λ value in training. Future works can focus on discussing how to find the suitable parameter λ or designing a new cosine distance-based loss function, which can constrain the features extracted from ECA-Resnet and focus on interclass and intraclass information simultaneously.

References

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

[2] L. Yang, G. Yang, X. Xi, X. Meng, C. Zhang, and Y. Yin, "Tri-branch vein structure assisted finger vein recognition," *IEEE Access*, vol. 5, pp. 21020–21028, 2017.

[3] H. Liu, L. Yang, G. Yang, and Y. Yin, "Discriminative binary descriptor for finger vein recognition," *IEEE Access*, vol. 6, pp. 5795–5804, 2018.

[4] L. Chen, J. Wang, S. Yang, and H. He, "A finger vein image-based personal identification system with self-adaptive illuminance control," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 2, pp. 294–304, Feb. 2017.

[5] S. Joardar, A. Chatterjee, and A. Rakshit, "A real-time palm dorsa subcutaneous vein pattern recognition system using collaborative representation-based classification," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 4, pp. 959–966, Apr. 2015.

[6] H.-B. Li, C.-B. Yu, D.-M. Zhang, and Z.-M. Zhou, "The study on finger vein image enhancement based on ridgelet transformation," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Sep. 2010, pp. 1–4.

[7] J. Zhang and J. Yang, "Finger-vein image enhancement based on combination of gray-level grouping and circular Gabor filter," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, Dec. 2009, pp. 1–4.

[8] K. Shin, Y. Park, D. T. Nguyen, and K. R. Park, "Finger-vein image enhancement using a fuzzy-based fusion method with Gabor and retinex filtering," *Sensors*, vol. 14, no. 2, pp. 3095–3129, 2014.

[9] J. Yang, Y. Shi, and G. Jia, "Finger-vein image matching based on adaptive curve transformation," *Pattern Recognit.*, vol. 66, pp. 34–43, Jun. 2017.

[10] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger vein patterns based on iterative line tracking and its application to personal identification," *Syst. Comput. Jpn.*, vol. 35, no. 7, pp. 61–71, Jun. 2004.

[11] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vol. 90, no. 8, pp. 1185–1194, Aug. 2007.

[12] E. C. Lee, H. C. Lee, and K. R. Park, "Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction," *Int. J. Imag. Syst. Technol.*, vol. 19, no. 3, pp. 179–186, Sep. 2009.

[13] X. Meng, G. Yang, Y. Yin, and R. Xiao, "Finger vein recognition based on local directional code," *Sensors*, vol. 12, no. 11, pp. 14937–14952, Nov. 2012.

[14] 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, Feb. 2012.

[15] T. S. Beng and B. A. Rosdi, "Finger-vein identification using pattern map and principal component analysis," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Nov. 2011, pp. 530–534.

[16] E. C. Lee and K. R. Park, "Image restoration of skin scattering and optical blurring for finger vein recognition," *Opt. Lasers Eng.*, vol. 49, no. 7, pp. 816–828, Jul. 2011.

[17] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1892–1905, Aug. 2018.

[18] S. Ahmad Radzi, M. Khalil-Hani, and R. Bakhteri, "Finger-vein biometric identification using convolutional neural network," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 24, pp. 1863–1878, 2016.

[19] H. Hong, M. Lee, and K. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, p. 1297, Jun. 2017.

[20] R. Raghavendra, S. Venkatesh, K. B. Raja, and C. Busch, "Transferable deep convolutional neural network features for fingervein presentation attack detection," in *Proc. 5th Int. Workshop Biometrics Forensics (IWBF)*, Apr. 2017, pp. 1–5.

[21] W. Kim, J. M. Song, and K. R. Park, "Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (NIR) camera sensor," *Sensors*, vol. 18, no. 7, p. 2296, Jul. 2018.

[22] 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.

[23] H. Hu *et al.*, "FV-net: Learning a finger-vein feature representation based on a CNN," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3489–3494.

[24] 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, Feb. 2019.

[25] H. Huang, S. Liu, H. Zheng, L. Ni, Y. Zhang, and W. Li, "DeepVein: Novel finger vein verification methods based on deep convolutional neural networks," in *Proc. IEEE Int. Conf. Identity, Secur. Behav. Anal. (ISBA)*, Feb. 2017, pp. 1–8.

[26] 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.

[27] Y. Wen, K. Zhang, L. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 499–515.

[28] D. Wu, S.-J. Zheng, W.-Z. Bao, X.-P. Zhang, C.-A. Yuan, and D.-S. Huang, "A novel deep model with multi-loss and efficient training for person re-identification," *Neurocomputing*, vol. 324, pp. 69–75, Jan. 2019.

[29] Z. Li, C. Xu, and B. Leng, "Angular triplet-center loss for multi-view 3D shape retrieval," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8682–8689.

[30] 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.

[31] M. S. Mohd 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 Syst. Appl.*, vol. 41, no. 7, pp. 3367–3382, Jun. 2014.

[32] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Proc. Chin. Conf. Biometric Recognit.* Berlin, Germany: Springer, 2011, pp. 260–268.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[34] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 11531–11539.

[35] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6738–6746.

[36] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[37] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[38] P. Gupta and P. Gupta, "An accurate finger vein based verification system," *Digit. Signal Process.*, vol. 38, pp. 43–52, Mar. 2015.

[39] T. Yuan, W. Deng, J. Tang, Y. Tang, and B. Chen, "Signal-to-noise ratio: A robust distance metric for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4815–4824.

[40] Y. Sun *et al.*, "Circle loss: A unified perspective of pair similarity optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6398–6407.

[41] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[42] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[43] J. Goldberger *et al.*, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, no. 17, 2004, pp. 513–520.

[44] G. Wang, C. Sun, and A. Sowmya, "Multi-weighted co-occurrence descriptor encoding for vein recognition," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 375–390, 2020.

[45] Y. Zhang, W. Li, L. Zhang, X. Ning, L. Sun, and Y. Lu, "Adaptive learning Gabor filter for finger-vein recognition," *IEEE Access*, vol. 7, pp. 159821–159830, 2019.

[46] 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.

Borui Hou was born in Yichang, Hubei, China, in 1992. He received the B.S. degree in measurement and control technology and instrument from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015. He is currently pursuing the Ph.D. degree in instrumentation science and technology with Southeast University, Nanjing.

His academic interests include fault diagnosis, signal analysis, including physiological signal analysis, pattern identification, and computer vision.

Ruqiang Yan (Senior Member, IEEE) received the M.S. degree in precision instrument and machinery from the University of Science and Technology of China, Hefei, China, in 2002, and the Ph.D. degree in mechanical engineering from the University of Massachusetts at Amherst, Amherst, MA, USA, in 2007.

He was a Guest Researcher with the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, from 2006 to 2008, and a Professor with the School of Instrument Science and Engineering, Southeast University, Nanjing, China, from 2009 to 2018. He joined the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2018. His research interests include data analytics, machine learning, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, and dynamic systems.

Dr. Yan is a fellow of ASME in 2019. His honors and awards include the New Century Excellent Talents in University Award from the Ministry of Education, China, in 2009, the IEEE Instrumentation and Measurement Society Technical Award in 2019, and multiple best paper awards. He is also the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT and an Associate Editor of the IEEE SYSTEMS JOURNAL and the IEEE SENSORS JOURNAL.