Infrared Physics and Technology 105 (2020) 103221

Contents lists available at ScienceDirect

Infrared Physics & Technology

journal homepage: www.elsevier.com/locate/infrared

Finger vein recognition based on lightweight CNN combining center loss and dynamic regularization

Dongdong Zhao[a], Hui Ma[a,b,*], Zedong Yang[a], Jianian Li[a], Wenbo Tian[a]

[a] *College of Electronic Engineering, Heilongjiang University, Harbin 150080, China*

[b] *Key Laboratory of Information Fusion Estimation and Detection, Heilongjiang Province 150080, China*

ARTICLE INFO

Keywords:
Center loss
Finger vein identification
Improved regularization
CNNs

ABSTRACT

Finger vein recognition is one of a new biometric recognition technology, which has a wide range of applications in daily life. However, the quality of finger vein images is less than satisfactory because of the disappointing sensor conditions based on infrared light. This problem results in the inaccuracy of finger rein identification. In order to solve this problem and speed up convergence, this paper introduces a new approach of identifying finger veins using the Convolutional Neural Networks (CNNs) with center loss function and dynamic regularization. The proposed method will makes full use of the labels and then ameliorate results. We compare its performance with several popular loss function, such as softmax loss and triplet loss. Experiments were carried out on the datasets of MMCBNU_6000 and FV-USM, whose results show that not only the proposed loss function minimize the error rate and is less time-consuming, but also avoid overfitting.

1. Introduction

The research to finger vein identification can be traced back to 1990s. Researchers found that The distribution of finger vein stripes varies from person to person, even among twins. Therefore, human finger veins was proved to be used as biometric information for identification. The theoretical base for finger vein recognition is derived from the phenomenon that hemoglobin in human blood vessels absorbs specific near-infrared light. Based on that theory, the researchers expose veins on finger to near-infrared light of a specific wavelength. By doing this, they can clearly view the pattern of veins. Compared with other biometrics (fingerprint [1], palmprint [2], face [3] and iris [4]), finger vein recognition technology has some obvious advantages [5] such as support for living body detection, small device size and high security. Today finger vein recognition has been paid more and more attention because of the above-mentioned advantages and many researchers have proposed considerable algorithms for it. At present, the classical research on finger vein recognition mainly includes four parts: image acquisition, image pre-processing, feature extraction and feature matching. The most important part among them is feature extraction, because the quality of extracted feature has a great influence on subsequent matching phase. There are two ways for feature extraction, one is hand-crafted features (LBP [6], SIFT [7]), this feature are forged features based on people's understanding to the characteristics of samples; another way is based on neural network, which can automatically extract representative deep features, this kind of features can't be understood well but performs well.

In order to simplify the procedure of image preprocessing and extract representative features, a finger vein recognition system based on lightweight CNNs was proposed in this paper. In the proposed deep learning method, the CNN was used to extract typical features automatically from images to reduce the dimension and acquire more representative features from the original data; the function of center loss function is fine tuning obtained feature of CNNs and make each classes easier to be distinguished; dynamic regularization was used to accelerate convergence and enhance the ability of generalization of the model. Compared with original model, the proposed loss model outperforms the original one as it will diversify the inter-class features and minimize the displacement of intra-class features. Adding to its merits is the decreased risk of overfitting. So the discriminative ability of the learned features can be highly consolidated.

The main contributions of this paper are summarized as follows:

(1) Lightweight Convolutional Neural Network with fewer convolution layer and fully connected layer is used to obtain the main features of each finger vein automatically in a short time. The gained low dimensional features can represent the original finger vein images to the greatest extent, which makes it easier and faster for

* Corresponding author.
E-mail address: mahui929@126.com (H. Ma).

https://doi.org/10.1016/j.infrared.2020.103221
Received 4 December 2019; Received in revised form 28 January 2020; Accepted 28 January 2020
Available online 08 February 2020
1350-4495/

classification.
(2) Center loss function was proposed to minimize the intra-class distance of the features of finger vein. With the joint of center loss function and softmax loss function, the highly discriminative features can be obtained for finger vein recognition.
(3) We proposed a new way for regularization, the parameter of the regularization is dynamic, which is conducive to efficient convergence and reducing overfitting.

The remainder of this paper is divided into four sections. They are organized as follows: Section 2 discuss related work with finger vein recognition and motivation of this paper. Section 3 explains the principle of proposed method. Section 4 presents the experimental results and analysis. Section 5 summarizes the whole work of the paper and elaborate the future work.

2. Related work and motivation

Researchers have made great progress in finger vein recognition in the past decades. Song *et al.* [8] proposed a new way of extracting finger vein using mean curvature, this algorithm can extract clear pattern from images with blurry veins. In order to reduce the dimension of the images and extract main features, Wu *et al.* [9] applied PCA and linear discriminant analysis in image processing and achieved great results. In 2016, Yang *et al.* [10] proposed a finger vein feature representation method based on adaptive vector field estimation, the experiments show that this method is very effective in improving the matching accuracy of finger vein. In the same year, Xi *et al.* [11] proposed a new Discriminative Binary Codes (DBC) learning method for finger vein recognition, they used the MLA database and the PolyU database for algorithm performance evaluation. The equal error rate of proposed method are 0.88% and 1.44%, indicating that the DBC-based finger vein recognition algorithm is efficient and highly accurate. In 2017, Liu *et al.* [12] proposed two block selection methods for finger vein recognition, experiments show that the proposed method can greatly reduce the dimension of the feature vector while keeping the recognition rate unchanged. You *et al.* [13] analyzed the internal factor of degradation of finger vein image and then proposed a new bilayer restoration to improve the quality of the finger vein images, results show that the proposed method is effective and robust in finger vein image restoration and enhancement. To resolve the sensibility to positional variations of the finger, Kang *et al.* [14] proposed an entirely new system, which applies a 3-D reconstruction method to build the full-view 3-D finger vein image, the results show great robustness of the proposed method.

As we have mentioned in the previous part, the extracted features are hand-crafted. This kind of method achieves high speed of response in recognition phase. Despite of these merits, there are some obvious problems which are listed in the following part.

(1) Hand crafted features can be considered to be shallow features, So it can barely represent the whole image and may contain useless information.
(2) The hand crafted features are sensitive to noise.
(3) Without human intervention, hand crafted features can't be automatically extracted from the original images, which is inconvenient and time-consuming.

Motivated by the success of CNNs for image classification, some researchers used CNNs for finger vein recognition and delivered good results. Fang *et al.* [15] introduced a novel finger vein verification system based on two-stream convolutional network learning, they integrated the information of original image and mini-ROI and then select final network system. Finally, they invented state-of-the-art system which is accurate and cost-saving. In order to address the limitations of available datasets for training, Xie *et al.* [16] presented a light CNN with supervised discrete hashing for finger vein identification. Not only did the proposed method achieve outstanding results over other classical convolutional neural network architecture, but also dramatically reduced the template size. Fayyaz *et al.* [17] brought out a way based on Self-Taught Learning, this method can learn discriminating features automation without doing any special preprocessing and achieve high accuracy with dataset independence. Hou *et al.* [18] presented a new deep-learning method that combines CAE and SVM for finger vein recognition. According to the results of experiments, the EER is lower than that of many state-of-the-art method without any prior knowledge. In order to enhance the safety of finger vein recognition and prevent original biometric templates from being attacked, Yang *et al.* [19] developed a new protection algorithm using the binary decision diagram, this method creates an irreversible version of original finger vein template, which is safer than traditional algorithm. To solve the problems of high complexity and large parameters in CNNs, Zhang *et al.* [20] proposed an adaptive Gabor convolutional neural networks for finger vein recognition. Results show that this method will reduce the number of parameters and the speed of processing row data.

There are some advantages and disadvantages in finger vein recognition with deep learning. Models of deep learning is end-to-end, there is no need to extract hand-crafted features and it has strong ability of anti-interfere. But deep learning need quite a number of samples and more time to train the model. This paper was designed to solve the problems of time-consuming and low recognition rate.

3. Methodology

3.1. Convolutional neural network

CNN is a kind of feed forward neural network with deep structure and convolution calculation. CNN has been developed rapidly because of it's ability of representation learning. It is widely adopted in Computer Version and Nature Language Processing. Hidden layer of CNN contains convolutional layer, pooling layer and fully connected layer. Convolutional layer is used to extract features from input layer. The ability of pooling layer is to add nonlinear factors and reduce the number of parameters. Fully connected layer is responsible for mapping one feature space to another. Besides, some network often add activation function to reinforce nonlinear expression.

3.2. Proposed method

3.2.1. Proposed model with lightweight CNN

Hidden layers differed in terms of their effect. The more layers CNNs contains, the stronger representation learning ability but more time-consuming. On the contrary, if hidden layer contains a few layer, the model will fail to achieve expected result. So it's important to select appropriate layer for CNN. In this paper, hidden layer of CNN includes convolution layer, pooling layer and fully connected layer. The used network contains 3 convolutional layers, 3 pooling layers and 2 fully connected layers, as is shown in Fig. 1. The proposed network is similar to AlexNet network, layer 2 pays more attention to edges and corner. Layer 3 contains more information about complex invariance and similar textures. Layer 4 shows significant variations and is more class-specific. Layer 5 describes that the the finger vein has significant pose variations. The last layer represents the feature vector of entire finger vein. Because of the lightweight structure of CNNs, the model is robust though it contains a few variables. The details of the distribution of variables in the model is shown in Table 1.

3.2.2. CNN with softmax loss function

The last layer of the model is softmax layer, which is used for classification. The input of softmax function is an arbitrary N-dimensional vector and the outputs have the same dimension, the value of output vector elements are mapped between 0 and 1. The elements of

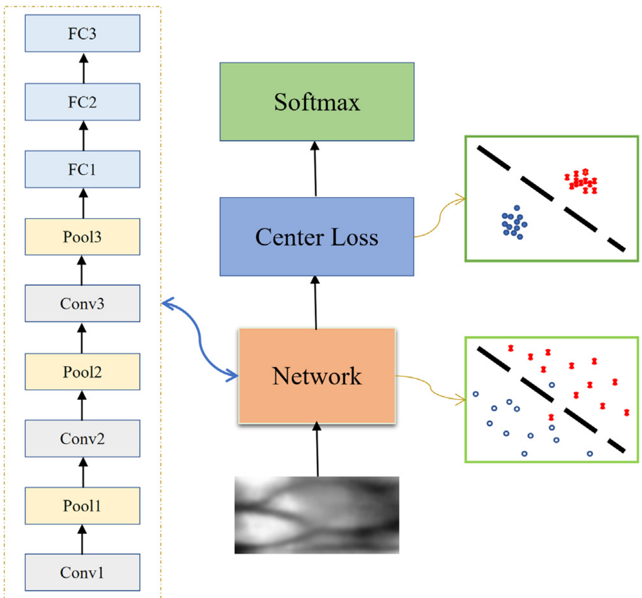


Fig. 1. The structure of proposed model.

Table 1
The parameters of CNN model.

Type	Filter size/stride	Output size	Params
Conv1	3 × 3/1	32 × 60 × 128	288
Pool1	2 × 2/2	32 × 30 × 64	–
Conv2	3 × 3/1	64 × 30 × 64	576
Pool2	2 × 2/2	64 × 15 × 32	–
Conv3	3 × 3/1	128 × 15 × 32	1152
Pool3	2 × 2/2	128 × 7 × 16	–
Fl1	–	256	3760 k
Fl2	–	200	51.2 k
Total	–	–	3732 k

the output vector is shown as follows:

$$s_i = \frac{e^{w_j}}{\sum_{k=1}^{n} e^{w_k}} \qquad \forall\ i \in 1, 2, \cdots, n \tag{1}$$

set $D_j s_i$ is the partial derivative of s_i to w_j, then

$$D_j s_i = s_i(\delta_{ij} - s_j) \quad \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2}$$

The loss function used in the model is the cross entropy loss function.

$$loss = \sum_{i=1}^{n} \hat{y} \cdot \log(y) \tag{3}$$

where $\hat{y}$ represents the expected output, y, the real one. Combining them with softmax function, we can get softmax loss function.

$$L_s = -\sum_{i=1}^{n} \sum_{j=1}^{n} 1\{y_{(i)} = j\} \cdot \log(s_j) \tag{4}$$

where m is the number of samples. s_j is the output value of the softmax layer. Then we can get the partial derivative of the loss function for each input, and each variable w can be updated using gradient descent.

$$w_i := w_i - \alpha \frac{\partial L_s}{\partial w_i} = w_i - \alpha \frac{\partial L_s}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_i} \tag{5}$$

where α is the learning rate.

In order to obtain visualization results, we set the feature vector into 2 dimension using softmax loss function in the dataset of MNIST handwritten digital. The distribution of 2-dimensional feature is shown in Fig. 2, from which we can see that the distance between some samples from the same classification is lager than those from the different classification, this could cause misclassification. In order to reduce the risk of misclassification, we proposed two programs for improvement, which will be discussed in the next section.

3.2.3. CNN with triplet loss function

Triplet loss function is one kind of loss function in deep learning [21]. The trained data includes an anchor, a positive, and a negative, and the similarity calculation of the sample is achieved by optimizing the distance between the Anchor example and the Positive example being less than the distance between the Anchor example and the Negative example.

As is shown in Fig. 3, triplet loss in finger vein recognition is a way to learn good embeddings for each finger vein. In embedding space, finger vein from the same person should be close together and form well separated clusters.

The goal of triplet loss is to make sure that:

- Two examples with the same label have their embeddings close together in the embedding space.
- Two examples with different labels have their embeddings far away.

The core of triplet loss function is to select one sample A as Anchor in training, then we should choose another sample P which has the same label with Anchor. After that, we choose another sample N which has the different label with Anchor. If the two samples are marked with the same label, our goal is to shorten the distance between them. If not, we should shift our goal to extending their distance. Therefore the triplet loss is defined as follows:

$$L_t = \sum_{i=1}^{n} [\max(\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha, 0)] \tag{6}$$

where $\|*\|$ represents Euclidean distance, so $\|f(x_i^a) - f(x_i^p)\|^2$ is the Euclidean distance between Anchor and Positive, we use $d(a\_p)$ to represent it. The same theory, $\|f(x_i^a) - f(x_i^n)\|^2$ is the distance between Anchor and Negative, which is represented by $d(a\_n)$, α is a margin between $d(a\_p)$ and $d(a\_n)$, there would produce loss if $d(a\_p)$ is larger α than $d(a\_n)$, otherwise there is no loss produced. Training the model using the gradient descent method after calculating the loss function and partial derivative, we trained the model using triplet loss in the MNIST dataset. From the distribution of 2-dimension feature (Fig. 4) we can conclude that this kind of loss function does not optimize the model enough. This problem can be attributed to two causes. The first cause is the slow convergence of triplet loss. The second one is the incapability of this kind of loss function to make full use of the information with labels. Considering that, we introduce a newly loss function which can solve this kind of problem and will be introduced in the next section.

3.2.4. CNN with center loss function

In finger vein recognition, even different finger from different identify have similar backgrounds, that would contribute many misidentification because the distance between two different samples is not separable enough. Center loss function was first used in face recognition and achieved incredible results [22], the key part of this paper is to illustrate how to minimize the intra-class variations while keeping the features of different classes separable. Thus, the discriminative ability of the model was improved because of this effective loss function. This kind of practice can minimize the distance between two samples that from the same label and maximize the distance from the different label. The core of center loss function is shown as follows:

$$l_c = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \tag{7}$$

where $c_{yi} \in R^d$ denotes the $y_i th$ class center of deep features. x_i denotes the deep feature of $i th$ sample, m is the number of samples of each batches, Ideally, $y_i th$ is updated as the depth features are learned. To put

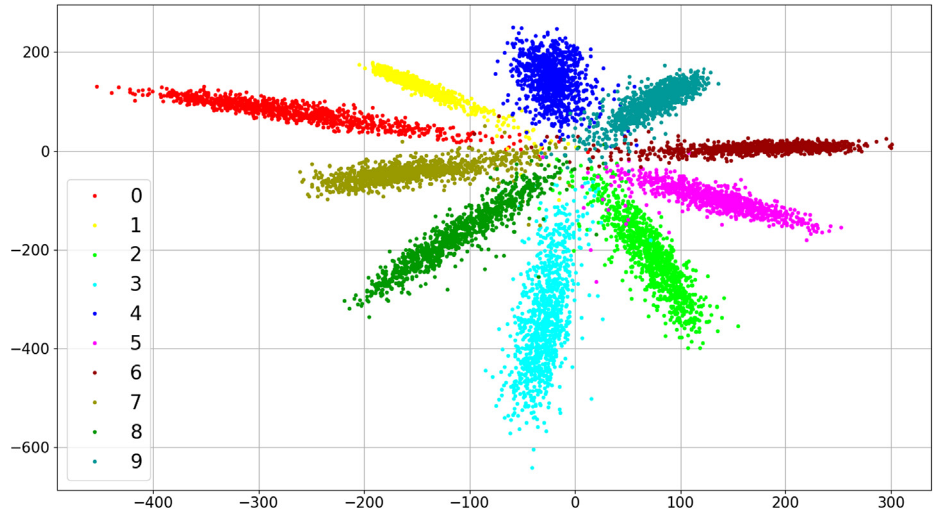


Fig. 2. The distribution of 2-dimension feature of softmax loss function in MNIST.

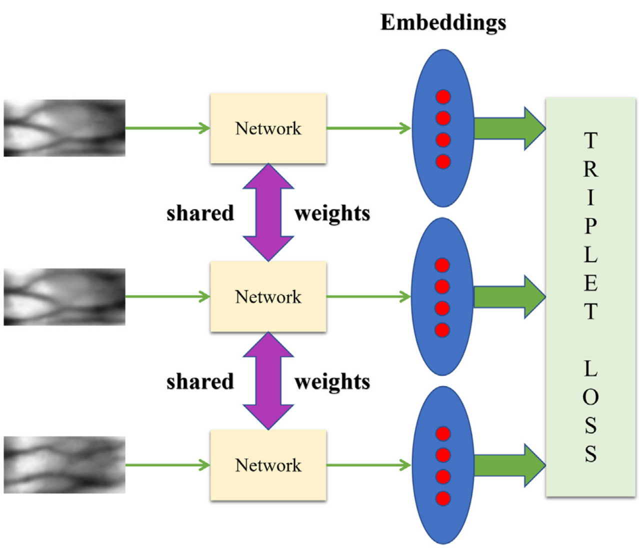


Fig. 3. Mechanism of action of Triplet loss function.

it into this way, the entire training set needs to be taken into account in each iteration, and the characteristics of each class are averaged. But this idea is pie in the sky and totally unrealistic. So we need change the training strategy:

(1) The training samples in the mini-batch are updated instead of the category centers of the entire training set; in each iteration, only the features of the corresponding categories are averaged (some category centers may not be updated).
(2) In order to avoid large interference caused by small sample categories, a factor α is used to control the learning rate of the category center.

In order to train all kinds of feature center vectors, we derive the deep feature of each data sample from l_c and update each center feature using gradient descent as follows:

$$\frac{\partial l_c}{\partial x_i} = x_i - c_{y_i} \tag{8}$$

$$\nabla c_j = \frac{\sum_{i=1}^{m} \delta(y_i = j)\cdot(x_i - c_j)}{1 + \sum_{i=1}^{m} \delta(y_i = j)} \tag{9}$$

where $\delta(x) = \begin{cases} 1, & if\ x\ is\ true \\ 0, & if\ x\ is\ false \end{cases}$, the total loss can be obtained when we

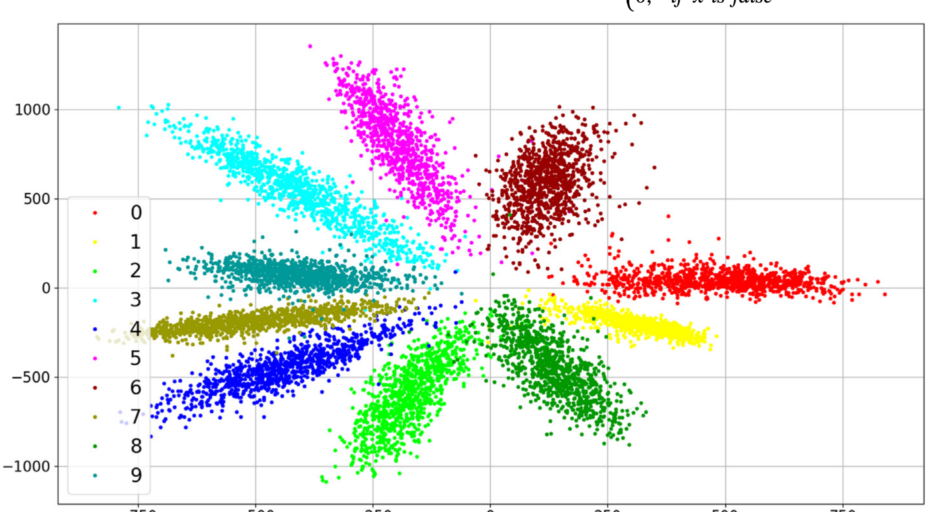


Fig. 4. Distribution of 2-dimension feature using softmax and triplet.

Table 2
The algorithm of proposed method.

Algorithm. The algorithm of proposed method

Input: Training data $\{x_i\}$. Initialized parameters θ_c in convolution layers. Parameters W and $\{c_j \mid j = 1, 2, \dots, n\}$ in loss layers, respectively. Hyperparameter α and learning rate μ_t. The number of iteration $t \leftarrow 0$.
Output: The parameters θ_c.
1: **While** not converge **do**
2: $t := t + 1$
3: Compute the joint loss by $l_t = l_s + \lambda l_c$
4: Compute the backpropagation error $\frac{\partial l_t}{\partial x_i^t} = \frac{\partial l_s}{\partial x_i^t} + \lambda \cdot \frac{\partial l_c^t}{\partial x_i^t}$
5: Update the parameters W by $W^{t+1} = W^t - \mu^t \cdot \frac{\partial l_c^t}{\partial W^t}$
6: Update the parameters c_j for each j by $c_j^{t+1} = c_j^t - \alpha \cdot \Delta c_j^t$
7: Update the parameters θ_c by $\theta_c^{t+1} = \theta_c^t - \mu^t \sum_{i=1}^{m} \frac{\partial l^t}{\partial x_i^t} \cdot \frac{\partial x_i^t}{\partial \theta_c^t}$
8: **end while**

combine center loss and softmax loss, which is shown as follows:

$$l_t = l_s + \lambda l_c = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_{y_j}^T x_j + b_{y_j}}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\| \tag{10}$$

The loss function algorithm flow is as follows in Table 2:

We trained the model in MNIST dataset using center loss and softmax loss, the experiment shows that 2-dimension feature of the same label converged to one point(as is shown in Fig. 5), from which we can see that this kind of loss function did optimized the model. Besides, it's exciting that center loss function made great improvement in MNIST Handwritten digit dataset, but what about finger vein? In the next section we will use the experiments to show if this kind of loss improve the model.

3.2.5. Dynamic regularization

Optimizing network structure is an important topic in neural network research. When the network structure is complex enough, not only will it cause the decline of learning efficiency, but also will lead to overfitting. Small-scale networks are efficiency and have better generalization ability, but too small network would result in underfitting. So how to select network structure is essential. The usual practice is to prune redundant network using penalty.

$$\widetilde{E}(W) = E(W) + \lambda \cdot E_p(W) \tag{11}$$

where $E(W)$ is the error function of input samples, $E_p(W)$ is weight penalty, which can restrain the weight of redundant connections. λ represents penalty coefficient. There is no penalty when λ equals to zero. And when λ is positive infinity, it indicates that the network weights is only determined according to penalty item.

The most frequently used method is L1-regularization, it was widely used in perceptual compression and feature extraction because it can get sparsity results. But traditional regularization may slow down learning speed. To accelerate convergence rate, a dynamic regularization method is proposed. Error function is taken into account in regularization.

$$\widetilde{E}(W) = E(W) + \frac{\lambda}{val(E(W))} E_p(W) \tag{12}$$

where $val(E(W))$ means the value of $E(W)$.

Eq. (12) shows that when $val(E(W))$ is large, regularization coefficient is small. It means regularization has a few influence on model. And when $val(E(W))$ is small, the corresponding regularization coefficient is large, then regularization works. This strategy will solve the problem of low convergence rate brought by regularization.

4. Experiments

We conducted our experiments in Pycharm using python 3.6 and tensorflow at a version 1.10.0. The training system with four cores of 2.30 GHz Intel i5-8300H CPU and an NVIDIA GTX 1050Ti GPU.

4.1. Datasets

Experiments is carried out on two available public databases for testing and verifying the performance of proposed method. This two databases are MMCBNU_6000 and FV_USM, they are from Chonbuk National University and Universiti Sains Malaysia respectively.

4.1.1. Mmcbnu_6000

The first dataset, which is from Chonbuk National University, consists of finger vein images from 100 volunteers. Each participant was asked to provide images of his or her index finger, middle finger, and ring finger of both hands in a conventional office environment. The collection for each finger is repeated for ten times to obtain 60 finger vein images of each volunteer. Hence, MMCBNU_6000 is composed of 6000 images. Each image is stored in "bmp" format with the resolution of 640 × 480. We extracted ROI images (as is shown in Fig. 6, whose resolution is 128 × 60) in each original images and the original data should be rotated and added Gauss noise to expand the datasets.

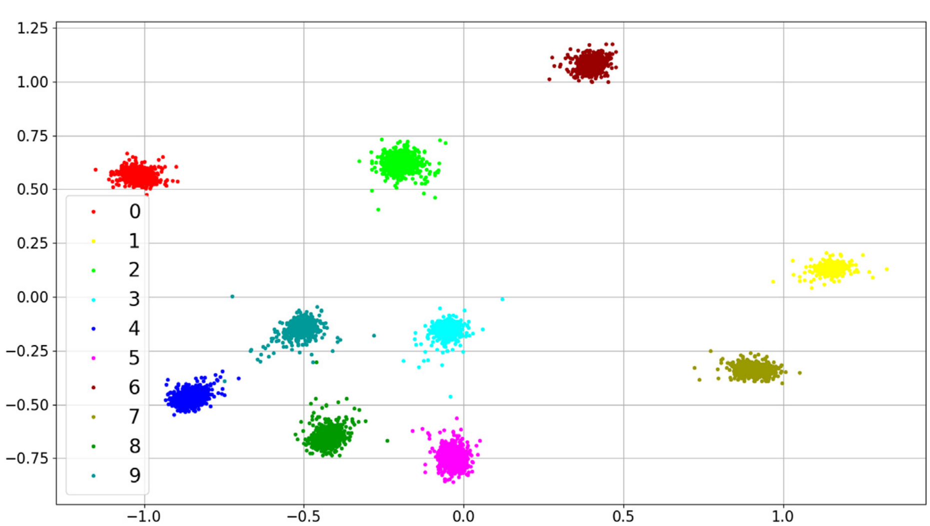


Fig. 5. Distribution of 2-dimension feature using softmax and center loss.

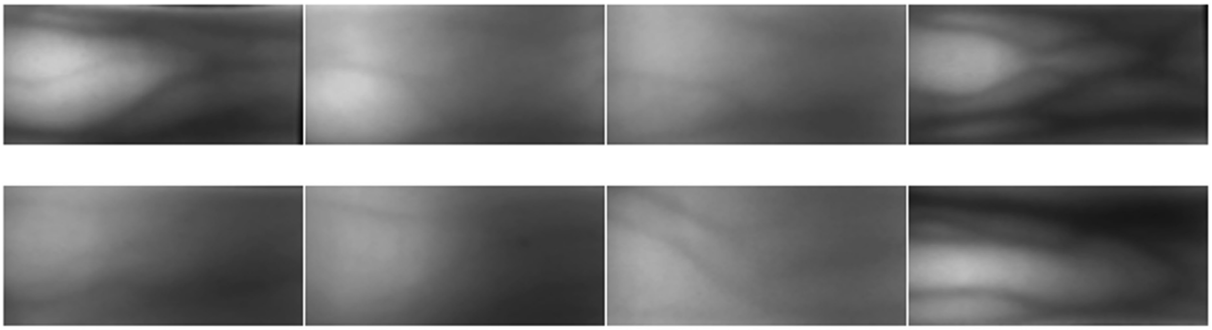


Fig. 6. Samples in MMCBNU_6000.

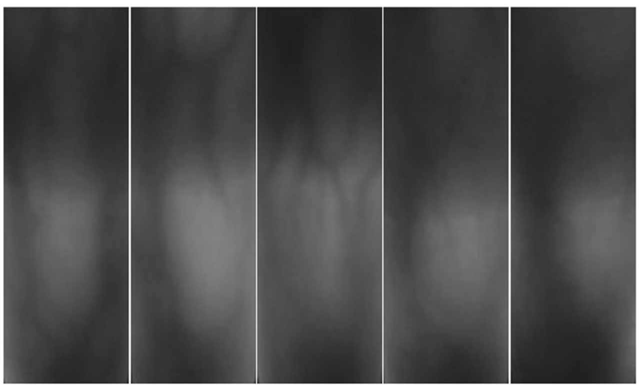


Fig. 7. Samples in FV_USM.

4.1.2. FV_Usm

The second dataset was collected from 123 volunteers comprising 83 males and 40 females, who were staff or students of University of Sains Malaysia. The age of the subject ranged from 20 to 52 years old. Each subject offered the images of four fingers: left index, left middle, right index and right middle fingers. In this way, a total of 492 finger classes were obtained. In our experiments, we categorized each finger from each person as one class, so there are 492 classifications in the dataset of FV-USM (only use the first session in the experiments). The resolution of ROI images is 300 × 100, as is shown in Fig. 7. We rotate and add Gauss noise to expand the datasets, the same as MMCBNU_6000.

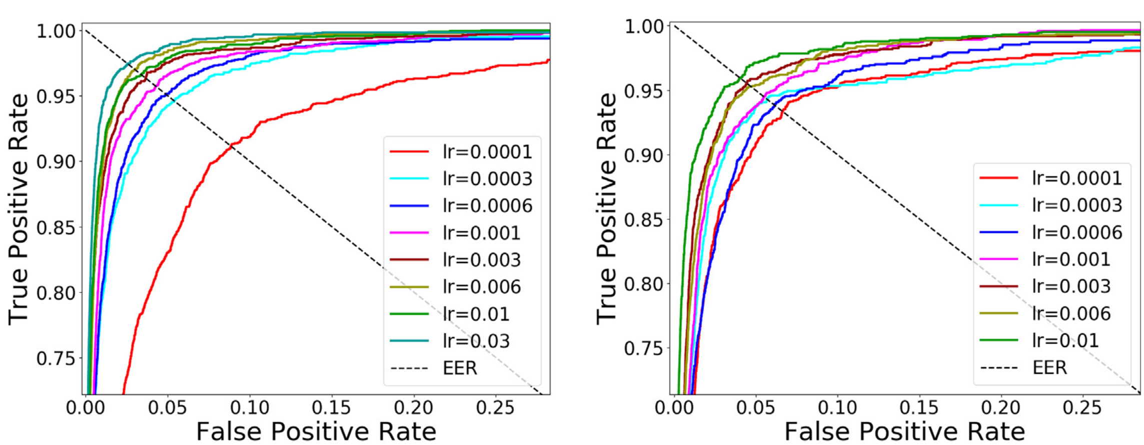
4.2. Parameter determination

4.2.1. Softmax loss function

We randomly selected 20% of each classifications as the validation images, the rest of images were used as the training images, all the experiment complied with this rule. In this experiments, we set different learning rate to find out the appropriate value. Fig. 7 shows that receiver operating characteristic curve [23] of each learning rate varied. From the experiments, we can find that when the learning rate are 0.03 and 0.01 respectively in the datasets of MMCBNU_6000 and FV_USM, their corresponding EER are 2.57% and 4.04%. In the next experiments we choose 0.03 and 0.01 respectively as the learning rate of softmax. Problems would arise from wrong use of learning rate. For instance, if the value of data is large enough, the model will diverge and then lead to infinite loss. If learning rate is too small, the model will convergence slowly. So in the next experiments we use this value as the learning rate. From the experiment, we can conclude that this kind of loss function can't achieve great performance. Two factors contribute to its ill performance, First, this kind of softmax loss can't reduce intra-class distance and increase the inter-class distance, thus there are many miscarriage of justice. Secondly, the model is overfitting, the testing datasets have low accuracy while the training datasets have the accuracy of 100% approximately.

4.2.2. Triplet loss function

From the experiments of softmax loss function, we can conclude that the performance of this kind of loss function is not good enough. The reason is that this kind of loss function can't reduce the distance of intra-class and increase the distance of inter-class. So triplet loss function was introduced to this paper. This loss function can reduce the intra-class distance to less than the inter-class distance. And we can set



(a) (b)

Fig. 8. ROC (Receiver Operating Characteristic) curve of softmax loss, ROC curve in MMCBNU_6000, (b) ROC curve in FV_USM.

different margin in the triplet loss, which is essential to the model, if the value of the margin is too small, the triplet loss may not work, which is similar to softmax loss. The too high value of the margin will make it hard to train the model. Fig. 8 shows the performance of different margin, from which we can see that different finger vein library has different optimal margin. The best margin in MMCBNU-6000 is 0.3, the EER can reach to 0.823%, 1.647% lower than the best EER using softmax loss; 0.9 is the best margin in FV-USM, the corresponding EER is 1.444%, which is 2.596% lower than the EER using softmax loss. In summary, compared with the softmax loss, the triplet loss function performs better, so we can draw a conclusion that triplet loss can optimize the model to some extent.

4.2.3. Center loss function

The experiment of triplet loss shows that it can reduce the error rate of models. The EER was reduced by 2% approximately, but from the experiment of MNIST we can see the intra-class distance was not much decreased. So the center loss was introduced to the finger vein recognition system. In theory, this kind of loss function can indeed reduce the intra-class distance. From the distribution of 2-dimensional feature of MNIST, we can see the each classification convergence to one point, the distance of intra-class is decreased but the distance of inter-class grows. In the experiment of center loss, there are two steps for training the model, the first step was used to update the parameters of loss layer, the second step was used to update the parameters of features. So there are two learning rate in the experiment, the learning rate of loss layer was discussed in Section 4.1. Besides, there are two parameters(α, λ) need to be optimized, α controls the learning rate of features, and the λ controls the proportion of center loss added to total loss. In order to find out the best value of each parameter, we use the thought of coordinate ascent. First, we set λ to a random value (λ is set to 0.0001 in the experiment), then different value of α was set to find out the optimal α, the best α is 0.00001, 0.0001 respectively in MMCBNU-6000, FV-USM), after that, we fix the optimal α and set different λ in the experiment to find out the best value of λ. In the end, the optimal value of this two parameters was found as is shown in Fig. 9, from which we can see that the optimal EER can reach 0.586% and 1.417% in each finger vein library, lower than the EER of triplet loss.

4.2.4. Proposed method

In the experiments, we discovered that the accuracy of training set is always 100% after specific iteration, so there is overfitting in the model to some extent. In order to solve that problem, the dynamic regularization was used in the model. Different value of parameters was set to find out the optimal value, which remains similar to the previous strategy. The result is that this improvement can consolidate the robustness of the model, and their EER are 0.503% and 1.07% respectively in MMCBNU-6000 and FV-USM. The ROC curve of proposed method is shown in Fig. 10.

4.3. Comparison with the state-of-the-art algorithm

This paper has investigated finger vein recognition performance using different loss function. Extra criteria for evaluation was added to explain the advantages of the proposed model. The ROC curve is shown in Fig. 11, all of them show great performance in ROC curve, all of the AUC in each ROC curve is more than 0.99, which is disadvantageous for evaluating the performance of each loss function. the reason for this phenomenon is that the proportional imbalance of positive and negative examples. So in order to better evaluate the performance of each loss function, PR (Precision-Recall) curve was introduced to experiments, which is shown in Fig. 12. The Precision-Recall curve has higher discriminant ability in the situation where the number of negative examples is considerable while the number of positive examples is extremely rare. In addition, CMC curve [24] is introduced in this paper, as is shown in Fig. 14. This kind of evaluating indicator will explain the robustness of the model. To be brief, we evaluated the performance of each loss function (Accuracy, ROC curve, PR curve and CMC curve). Three contrast experiments was introduced to the experiments. From the experiments we can arrive at the conclusion that the performance of softmax loss is far inferior to others and proposed method has higher accuracy than many other state of the art algorithm. The equivalent error rate and accuracy of best result of each loss function was summarized in Table 3, from which we can see the improved center loss function performs as well as state of the art method.

5. Conclusion

In this paper, a finger vein recognition system based on Convolutional Neural Networks with center loss and improved regularization was proposed to enhance the robustness and achieve higher performance. It is well-known that Convolutional Neural Networks are time-consuming in training step. The more layers of the network, the more time spent in training step. Therefore, this paper proposed a lightweight CNN, which is similar to AlexNet network. This guaranteed the real-time of the finger vein recognition system. In order to maximize the distance of inter-class and minimize the distance of intra-class, center loss was applied in the model and achieved remarkable results. Finally, we proposed a improved regularization to reduce the risk of overfitting. It has been experimentally validated that the appropriate training strategies can achieve outstanding performance in finger vein recognition. Experiments show the accuracy of the proposed method can reach to 99.05% and 97.95% in MMCBNU_6000 and FV_USM respectively, the equivalent error rate (EER) is 0.503% and 1.07% in

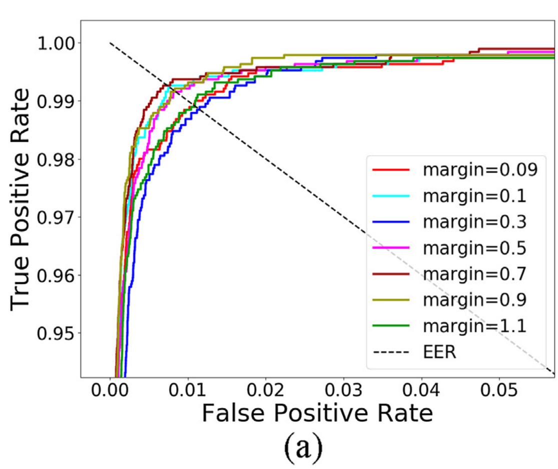


Fig. 9. ROC curve of triplet loss, (a) ROC curve in MMCBNU_6000, (b) ROC curve in FV_USM.

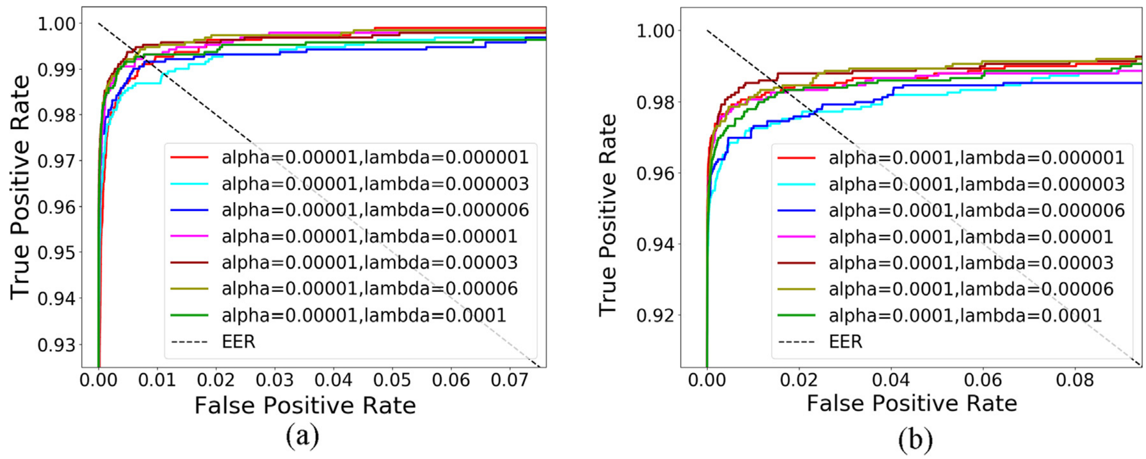


Fig. 10. ROC curve of center loss, ROC curve in MMCBNU_6000, (b) ROC curve in FV_USM.

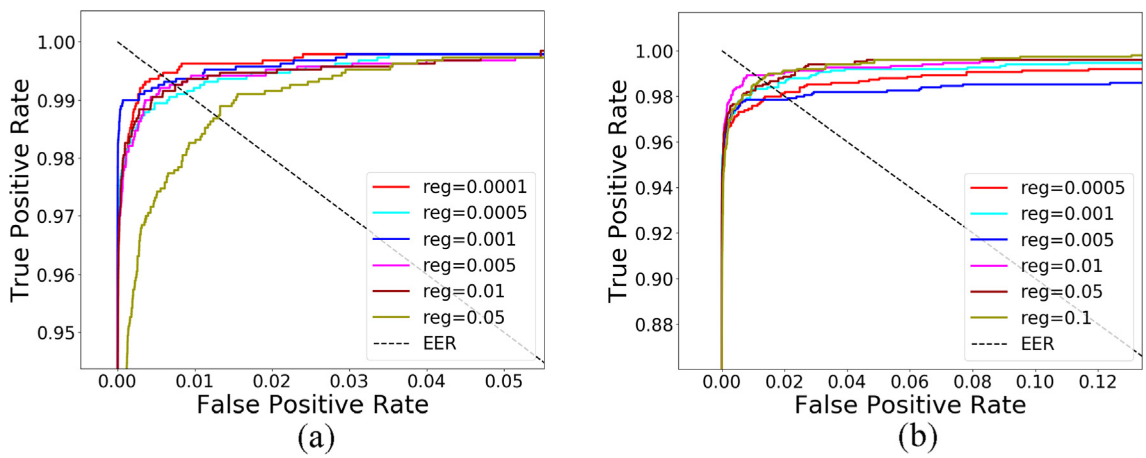


Fig. 11. ROC curve with proposed method, (a) ROC curve in MMCBNU_6000, (b) ROC curve in FV_USM.

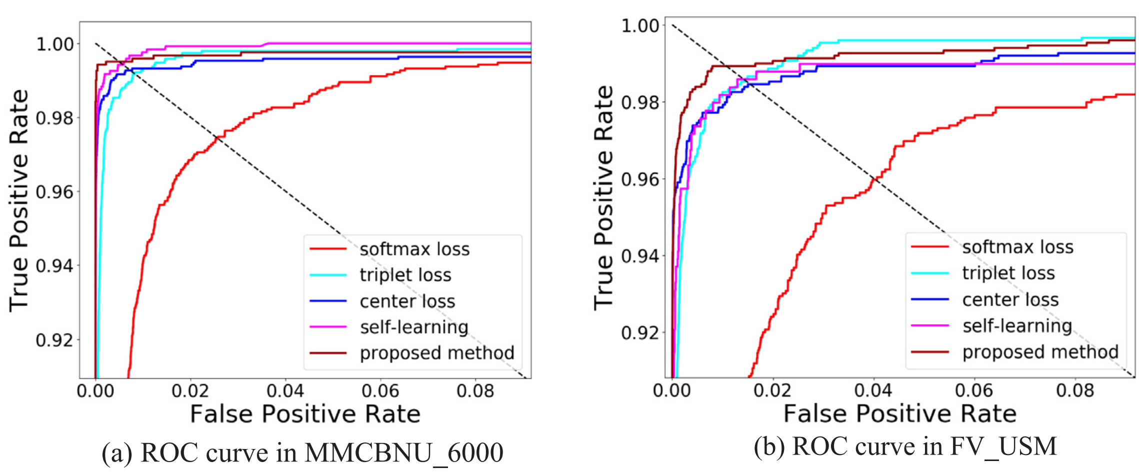
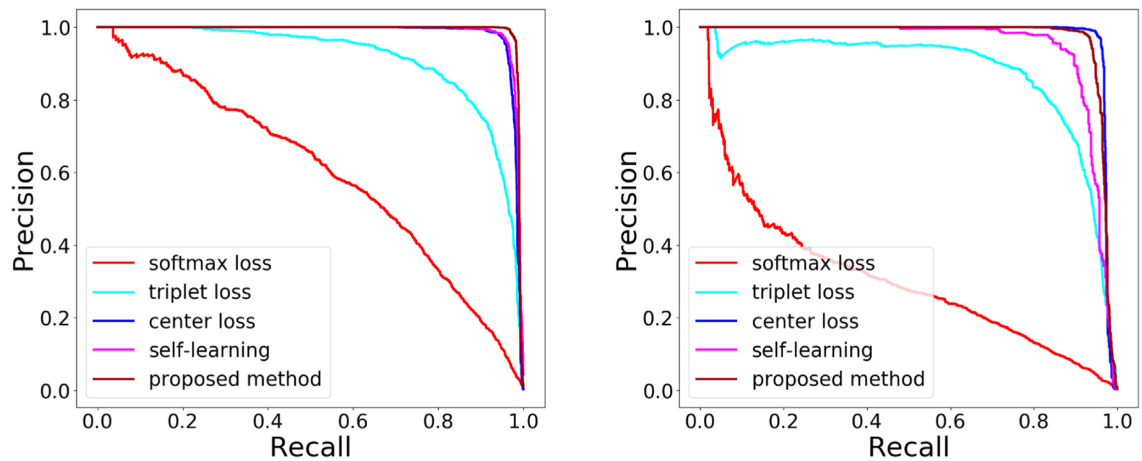


Fig. 12. ROC curve of different method.

MMCBNU_6000 and FV_USM. From the results of experiments a conclusion can be drawn that the proposed method in this paper can approach the level of state of the art method.

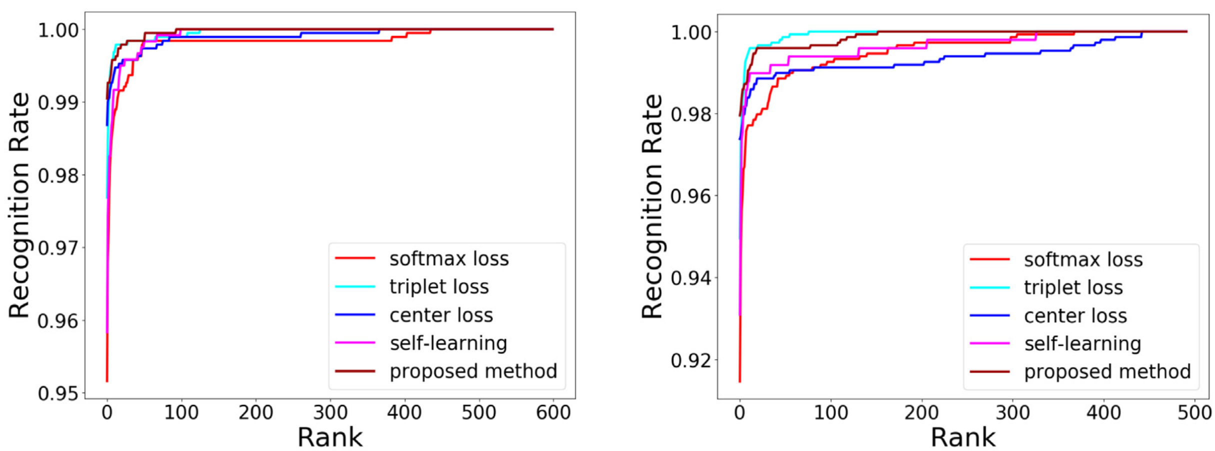
We believe that our method, apart from its standalone utility, provides a useful solution for finger vein recognition. It stands an extraordinarily promising future for a constructive system which uses lightweights CNN, dynamic regularization and center loss for finger vein.

As the proposed system has not been unassailable, there is still a lot of room for advancement, we will go ahead and beef up this model. The improvement work will embark on from three aspects. First, as for the structure of CNN, we will employ contemporary popular model to elevate the accuracy of its performance. Secondly, as the details of this model need to be further enhanced, exponential decay will be introduced to the learning rate. Besides, built on the variables in the training, we will add moving average. Finally, more loss function will



(a) PR curve in MMCBNU_6000 (b) PR curve in FV_USM

Fig. 13. Precision-Recall curve of different method.



(a) CMC curve in MMCBNU_6000 (b) CMC curve in FV_USM

Fig. 14. CMC curve of different method.

Table 3
The performance of different loss function.

Loss function	MMCBNU_6000		FV_USM	
	ACC	EER (%)	ACC	EER (%)
Softmax loss	0.9516	2.570	0.9146	4.040
Triplet loss	0.9542	0.823	0.9496	1.444
Center loss	0.9868	0.741	0.9738	1.547
self-taught learning [15]	0.9583	0.586	0.9309	1.417
Proposed loss	0.9905	0.503	0.9795	1.070

be supplemented in the experiments to compare the corresponding performance and analyse the pros and cons of each loss function. In the upcoming future, more convincing criteria for evaluation will be put forward in work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (No. 61573132), the Key Laboratory Project Fund of Heilongjiang Province (No. DZGC201610) and the Specialized Fund for the Basic Research Operating Expenses Program of Heilongjiang Province (RCYJTD -201806).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.infrared.2020.103221.

References

[1] Ratha, K. Chen, S. Karu, et al., A real-time matching system for large fingerprint databases, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996).
[2] M. Chaa, Z. Akhtar, A. Attia, 3D palmprint recognition using unsupervised convolutional deep learning network and SVM classifier, IET Image Proc. 13 (5) (2019) 736–745.
[3] P. Phillips, H. Moon, S. Rizvi, et al., The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Machine Intell. 22 (10) (2000) 1090–1104.
[4] R.P. Wildes, Iris recognition: an emerging biometric technology, Proc. IEEE 85 (9) (1997) 1348–1363.
[5] J. Hashimoto. Finger vein authentication technology and its future, in: 2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers. IEEE, 2006, pp. 5–8.
[6] E. Lee, H. Jung, D. Kim, New finger biometric method using near infrared imaging,

Sensors 11 (2011) 2319–2333.
[7] H. Kim, E. Lee, G. Yoon, et al., Illumination normalization for SIFT based finger vein authentication, International Symposium on Visual Computing, Springer, Berlin, Heidelberg, 2012, pp. 21–30.
[8] W. Song, T. Kim, H. Kim, et al., A finger-vein verification system using mean curvature, Pattern Recogn. 32 (11) (2011) 1541–1547.
[9] J. Wu, C. Liu, Finger-vein pattern identification using SVM and neural network technique, Expert Syst. 38 (11) (2011) 14284–14289.
[10] J. Yang, Y. Shi, G. Jia, Finger-vein image matching based on adaptive curve transformation, Pattern Recogn. 66 (6) (2017) 34–43.
[11] X. Xi, L. Yang, Y. Yin, Learning discriminative binary codes for finger vein recognition, Pattern Recogn. 66 (6) (2017) 26–33.
[12] B. Liu, S. Xie, D. Park, Finger vein recognition using optimal partitioning uniform rotation invariant LBP descriptor, J. Electr. Comput. Eng. 2016 (3) (2016) 1–10.
[13] W. You, W. Zhou, J. Huang, et al., A bilayer image restoration for finger vein recognition, Neurocomputing 348 (2019) 54–65.
[14] W. Kang, H. Liu, W. Luo, et al., Study of a full-view 3D Finger Vein Verification Technique, IEEE Trans. Inf. Forensics Secur. (2019).
[15] Y. Fang, Q. Wu, W. Kang, A novel finger vein verification system based on two-stream convolutional network learning, Neurocomputing 290 (5) (2018) 100–107.
[16] C. Xie, A. Kumar, Finger vein identification using convolutional neural network and supervised discrete hashing, Pattern Recogn. 119 (5) (2019) 148–156.
[17] M. Fayyaz, M. Hajizadeh-Saffar, M. Sabokrou, et al., A novel approach for Finger Vein verification based on self-taught learning, 2015 9th Iranian Conference on Machine Vision and Image Processing, IEEE, 2015, pp. 88–91.
[18] B. Hou, R. Yan, Convolutional auto-encoder model for finger-vein verification, IEEE Trans. Instrum. Meas. (2019).
[19] W. Yang, S. Wang, J. Hu, et al., Securing deep learning based edge finger-vein biometrics with binary decision diagram, IEEE Trans. Ind. Inf. (2019).
[20] Y. Zhang, W. Li, L. Zhang, et al., Adaptive gabor convolutional neural networks for finger-vein recognition, 2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS), IEEE, 2019, pp. 219–222.
[21] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, CVPR (2015) 815–823.
[22] Y. Wen, K. Zhang, Z. Li, et al., A discriminative feature learning approach for deep face recognition, European Conference on Computer Vision, Springer, Cham, 2016, pp. 499–515.
[23] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2005) 861–874.
[24] V. Štruc, N. Pavešić, The complete gabor-fisher classifier for robust face recognition, EURASIP J. Adv. Signal Proc. 2010 (1) (2010) 847680.