Applied Intelligence
https://doi.org/10.1007/s10489-019-01530-4

A novel method based on deep learning for aligned fingerprints matching

Yonghong Liu[1] · Baicun Zhou[1] · Congying Han[1,2] · Tiande Guo[1,2] · Jin Qin[1]

Abstract

In this study, a novel method based on deep learning for aligned fingerprints matching is proposed. According to the characteristics of fingerprint images, a convolutional network, Finger ConvNet, is designed. In addition, a new joint supervision signal is used to train Finger ConvNet to obtain deep features. Experimental studies are performed on public fingerprint datasets, the ID Card fingerprint dataset and the Ten-Finger Fingerprint Card fingerprint dataset. Furthermore, four performance indicators, the false matching rate (FMR), false non-matching rate (FNMR), equal error rate (EER) and receiver operating characteristic (ROC) curve, are measured. The experimental results demonstrate the effectiveness of the proposed method, which achieved a competitive effect in comparison with conventional fingerprint matching algorithms in fingerprint verification tasks using the FVC2000, FVC2002, and FVC2004 datasets. Moreover, the matching speed of the proposed method was almost 5 times faster than the fastest conventional fingerprint matching algorithms. In addition, it can be used as a fast matching method to filter out many templates with low scores by setting a threshold according to the matching scores and thus accelerate the process in identification tasks.

Keywords Fingerprint matching · Deep learning · Finger ConvNet · Fast matching method

1 Introduction

Of all biological characteristics, the fingerprint is the most commonly used characteristic for biometric verification and identification tasks [1]. Fingerprint matching technology, which is determined by the local ridge characteristics and their relationships, is the earliest and most widely used biometric authentication technology due to the uniqueness and invariability of fingerprints [2]. Fingerprints have gradually been applied in fields such as access control, attendance, finance, public safety, and e-commerce. Meanwhile, they have been used in police system for long time, especially in identifying criminals.

The automatic fingerprint identification system (AFIS) is an identification system that is concerned with some issues including image acquisition, features extraction and matching [3]. Compared with some traditional authentication technologies, AFIS is safer and superior and is used in many information security domains. As shown in Fig. 1, fingerprint features are generally described in a hierarchical order at three different levels: global features for Level 1 (e.g., singular points and type lines), local features for Level 2 (e.g., minutiae), and fine-detail features for Level 3 (e.g., sweat pores) [1, 4]. The local or minutiae features are the most commonly used ones for fingerprint matching.

Fingerprint matching methods are usually classified into three families: correlation-based matching, minutiae-based matching and non-minutiae feature-based matching [1, 5]. For correlation-based matching, Bazen et al. [6] proposed a correlation-based fingerprint verification system using the richer gray-scale information of the fingerprints. Ross et al. [7] presented a matching scheme according to ridge feature maps and proposed a novel fingerprint alignment method without extracting minutiae or the singular points. In Cavusoglu et al. [8], authors introduced orientation field estimations into correlation-based matching algorithms, which reduce the amount of correlations and save resources.

In contrast to correlation-based methods, the representation in minutiae-based matching is a feature vector composed

✉ Congying Han
hancy@ucas.ac.cn

[1] University of Chinese Academy of Sciences (UCAS), Beijing, China

[2] Key Laboratory of Big Data Mining and Knowledge Management, CAS, Beijing, China

Published online: 25 July 2019

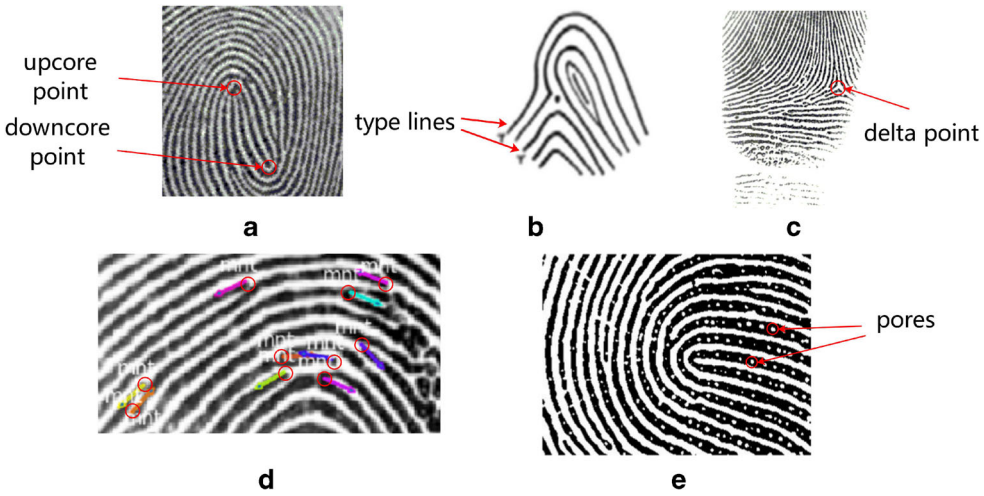

Fig. 1 Fingerprint global and local features. **a** Core points, **b** type lines, **c** delta point, **d** fingerprint minutiae, **e** pores. **a**, **b** and **c** are Fingerprint global features. **d** and **e** Local and fine-detail feature, respectively

of the fingerprint minutiae, and each minutia includes its location and orientation for alignment to maximize the number of matched minutiae. Ghaddab et al. [9] introduced Expanded Delaunay Triangulation consideration (EDT-C), which is a new fingerprint matcher based on minutiae triplets. The proposed matcher uses an extended form of Delaunay triangulation and includes some optimization operations that allow quickly consolidating local matchings and filtering non-matching minutia triplets. Aimed at frequent spurious minutiae, Peralta et al. [10] proposed a filtering approach to remove spurious minutiae to improve the performance of matching algorithms. Li et al. [11] proposed a neighbor minutiae boost-based algorithm and a multi-feature-based score fusion method to combine features effectively to obtain a distinctive similarity score.

Since reliable minutiae extraction of extremely poor-quality fingerprints is difficult and minutiae-based matching approaches do not perform well when the area of the fingerprint sensor is small, designers will consider non-minutiae feature-based matching methods due to more reliable extracted features (e.g., Level 3 features) than minutiae. By extracting the circular region of interest (ROI) at the core points and maximizing their orientation correlation, Kumar et al. [12] developed a robust fingerprint matching system and computed the modified Euclidean distance between the extracted orientation features of the sample and query images for matching. Meanwhile, the Euclidean distance computed between the extracted orientation features of the template and query images is used for matching.

For fingerprint matching algorithms, feature extraction is an indispensable step. In conventional matching algorithms, fingerprint images must be preprocessed because they are usually gray images with much noise, and this process usually includes segmentation, enhancement, orientation extraction, binarization, and thinning. When processing identification tasks, a fingerprint query aims to identify the matched template by operating in a 1:N matching process over a fingerprint database. To improve identification efficiency, many acceleration methods are adopted in fingerprint matching algorithms to pick out very few possible matched templates to process matching algorithms. In addition, acceleration methods usually cost less time than matching algorithms, and such steps can speed up the identification process. For example, fingerprint pattern classification [13] (see Fig. 2), can reduce

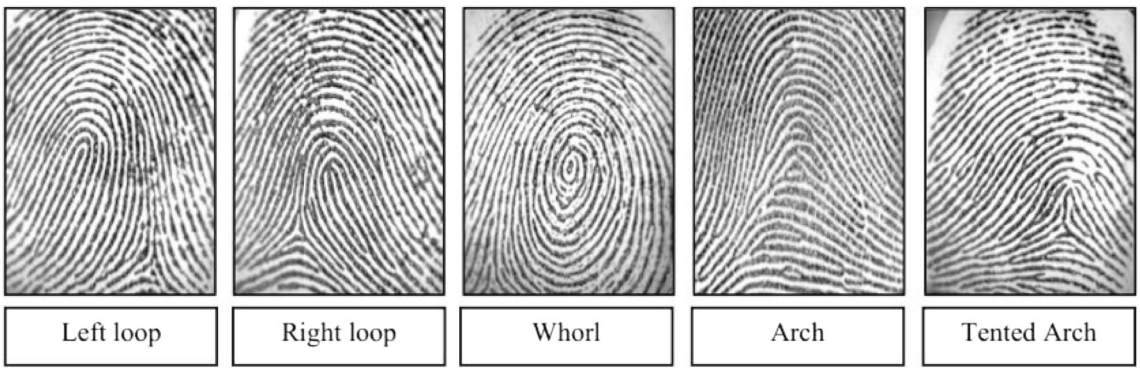

Fig. 2 Types of fingerprint patterns [1]

the matching set and thus reduce the matching time during an identification task. Galar et al. [14] reviewed the fingerprint classification literature considering the problem from a double perspective: feature extraction methods and learning models. Furthermore, taxonomies and classifications for the feature extraction, singular point detection, orientation extraction and learning methods are presented. Galar et al. also devised a deep experimental study following the proposed double perspective in [15].

Nevertheless, with the explosive growth in fingerprint databases (e.g., ID Card fingerprints), automatic recognition systems have become increasingly more difficult than ever. Due to the tedious steps, fingerprint matching algorithms consume more time and resources, such that the real-time capability of algorithms is weakened.

To address the above issues, we propose a novel end-to-end fingerprint matching method for aligned fingerprints based on deep convolutional neural networks (CNNs) [16] and design a CNN architecture according to the framework of VGG-16 [17] and the characteristics of fingerprint images. Specifically, due to moderate numbers of neurons and layers, the proposed CNN architecture contains fewer parameters than other typical network architectures (e.g., VGG-16, GoogLeNet [18] and the residual net (ResNet) [19]). Then, inputting a fingerprint pair into the trained network, we can judge whether they correspond to the same finger or not. This is a fully feed-forward step and complex preprocessing steps are not required for fingerprint matching tasks.

Unlike conventional matching methods that include tedious processes, the proposed method does not contain a preprocessing operation and only computes network layer operations, hence significantly improving the efficiency. By training this Finger ConvNet, we can obtain a CNN model to extract highly discriminative and compact features with good generalization ability. In addition, the process of feature extraction is performed offline. Then, the fingerprint features are used to train a binary classifier for final verification tasks. Due to the high efficiency and fast matching speed of our matching method, it can be used as a fast matching method to speed up the identification process, namely, operating N verification tasks. The experimental results reveal that the proposed method achieves a very fast speed for practical online usage, even on a CPU. Moreover, the proposed method achieves almost 5 times faster speed than the fastest conventional fingerprint matching algorithm.

The contributions of this paper are summarized as follows:

- We analyze the characteristics of fingerprints and design a CNN architecture that is applicable to fingerprint images. We hereby propose a supervised learning method to learn the discriminative and compact fingerprint features.
- We propose a new fingerprint matching method based on CNNs different from conventional matching algorithms. This approach is very efficient for large-scale fingerprint data and achieves high accuracy.
- We use this end-to-end method as a fast matching method to accelerate fingerprint identification. Moreover, extensive experiments demonstrate that this method is quite fast and efficient since it achieves an extremely small error rate with a fairly low penetration rate.

The rest of this paper is organized as follows: Section 2 briefly reviews some related works while Section 3 proposes the end-to-end deep learning method. Experiments for testing and comparison are conducted in Section 4. Finally, Section 5 draws some conclusions.

2 Related work

Conventional fingerprint recognition technologies mainly consist of image acquisition, preprocessing, feature extraction, and matching. For example, minutiae-based matching algorithms [9, 10] need to extract minutiae for alignment to maximize the number of matched minutiae. Luo et al. [20] used ridge information consisting of some sampled minutiae for relative pre-alignment, to prevent misalignment of two fingerprints. Since the fingerprint alignment by the reference minutiae is usually not satisfactory, Jiang et al. [21] proposed a minutia matching method by using both the local and global structures of minutiae. Similarly, for the other two families, namely, correlation-based matching algorithms [7, 8] and non-minutiae feature-based matching algorithms [12], the algorithms also need to extract fingerprint features. However, as the amount of fingerprint data gradually increases, the feature extraction speed of the conventional matching algorithms will be slower due to the complex process of image preprocessing (e.g., orientation extraction) and the performance of algorithms will also decline.

Currently, deep learning techniques have received significant attention and have been proven to be successful in computer vision applications, such as object detection [22], personality recognition [23] and face recognition [24, 25]. In speech recognition tasks [26], the deep learning technique has also achieved a good effect. Recently, deep learning has also been used in the medical field [27, 28]. Fujita et al. [29] proposed a novel approach of a computer-aided diagnosis (CAD) system for the detection of fibrillations and flutters for heart diseases by employing an 8-layer deep

convolutional neural network. Moreover, the proposed model requires only basic data normalization without preprocessing and feature extraction from raw samples. Due to the outstanding learning behavior, CNNs are also used in fingerprint fields. Wang et al. [30] achieved an outstanding effect in fingerprint classification tasks based on the new depth neural network method. A novel CNN method is proposed in [31] to estimate the minutiae score map and minutiae orientation, as well as to refine the candidate minutiae locations. Liu et al. [32] proposed a CNN method based on Faster Region-based Convolutional Neural Network (Faster-RCNN) to detect fingerprint singular points. Moreover, deep learning techniques also show good performance in partial fingerprint matching [33].

3 Learning deep ConvNet for fingerprint matching

In this section, a fast and effective fingerprint matching program is presented for aligned fingerprint images based on convolutional neural networks. Figure 3 shows the overall flowchart of the proposed method: (*a*) train proposed deep ConvNet to obtain a model that can extract highly discriminative and compact features through a classification task; (*b*) input any two fingerprint images, and extract and concatenate features; (*c*) train a 2-way classifier (label 0 and 1 represent "no" and "yes", respectively); and (*d*) input two query fingerprints for final fingerprint verification.

3.1 CNN architectures

Compared with natural images, fingerprints have their own characteristics and performance requirements. Because of the strong expressive ability of VGG-16 [17], we design a convolutional neural network (Finger ConvNet) based on the framework of VGG-16 [17] and fingerprint knowledge, and train the architecture to extract fingerprint features for fingerprint matching. Furthermore, we compare the matching performance with VGG-16 [17], GoogLeNet [18] and ResNet [19] in the experiment section. Finger ConvNet contains five ConvBlocks and each block consists of two convolutional layers followed by batch normalization(BN) [35] and an exponential linear unit (ELU) [36]. Following the last block, the CNN architecture connects two sections: the FeatureBlock and the softmax output layer. The last hidden layer is used as the feature layer, where highly compact and discriminative features are formed. Because of the different sizes of fingerprint images, sampling and crop methods are adopted to fix the aligned images to the same size of 256×256. The input is $256 \times 256 \times k$ for square patches, where $k = 3$ for color images and $k = 1$ for gray

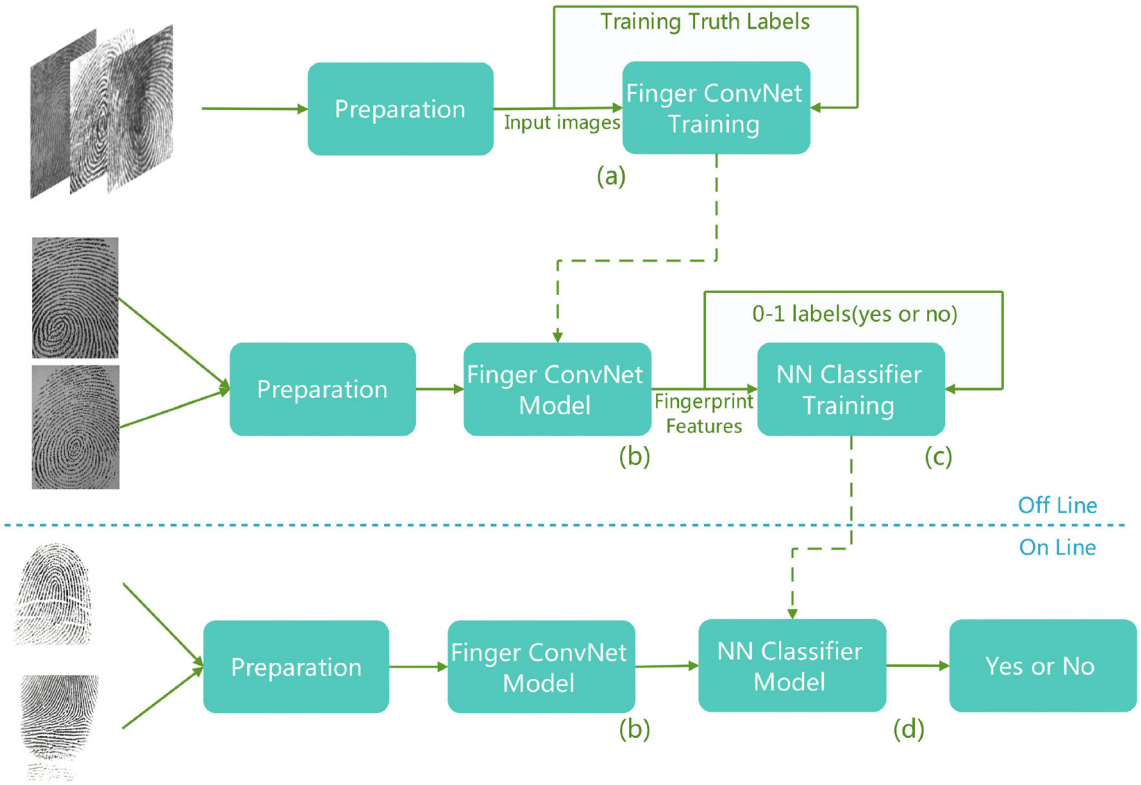

Fig. 3 The overall flowchart of our proposed method

images. In our work, all fingerprint images are converted into gray images.

As shown in Fig. 4, the detailed structure of the Finger ConvNet contains five convolutional blocks, the FeatureBlock and softmax output layer, which takes $256 \times 256 \times 1$ input and predicts n (e.g., $n = 5,000$) subjects (a finger is regarded as a subject, i.e., a class). Each ConvBlock is composed of two convolution layers, BN and the ELU operation, where BN is adopted after each convolution layer and before ELU. The FeatureBlock consists of a fully connected layer and activation, and after the activation, Finger ConvNet has a dropout layer operation [37, 38] (only for training) to avoid overfitting issues. The dimension of the feature layer is fixed at 320, while the dimension of the output layer is decided by the number of subjects.

Compared with VGG-16, the 3rd, 4th and 5th ConvBlocks consist of two convolutional layers followed by batch normalization (BN) and the exponential linear unit (ELU). Between each ConvBlock, there is a 2×2 max-pooling layer except the last two blocks. In addition, Finger ConvNet only contains a fully connected layer.

The operation in each convolutional layer can be expressed as

$$x_j^l = S\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right), \tag{1}$$

where $*$ denotes the convolutional operation, x_j^l and x_i^{l-1} are the j-th current neuron output of the l-th layer and the neuron connecting x_j^l in the previous layer, respectively, k_{ij}^l is the convolutional kernel (filter) in the l-th layer, and b_j^l is the bias of the l-th layer for the j-th output map. S is the non-linear activation function ELU and is operated element-wise, which is shown to perform better than the $sigmoid$ function [16] and rectified linear unit (ReLU) [39] in our works. In the proposed network structure, the filter size of the first ConvBlock is 4×4, and the second and third ConvBlock use 3×3 as the filter size. Then, in the last two blocks, weights are locally shared in every 2×2 region. The stride for the first three and the last two ConvBlocks are 1 and 2, respectively.

The last hidden layer of deep ConvNet is fully connected to the last convolutional layers, and after the operations of BN and activation, the output of Finger ConvNet consists of highly discriminative and compact features. After the operation of activation, the last hidden layer is operated by the dropout layer, and we can avoid overfitting efficiently and increase the network generalization ability. Then, after dropout operation, Finger ConvNet is trained by classifying the feature vectors through a softmax classifier. The Finger ConvNet extraction process is expressed as $f = \text{Conv}(x, \theta)$, where $\text{Conv}(\cdot)$ is the feature extraction function, i.e., the output of the last hidden layer, f is the extracted feature vector, x is the input fingerprint data and θ represents the Finger ConvNet parameters to be learned. Finger ConvNet's output is an n-way softmax predicting the probability distribution over n different subjects,

$$P(y = j|x; \theta) = \frac{\exp(y_i')}{\sum_{j=1}^{n} \exp(y_j')}, \tag{2}$$

where $y_j' = \theta_i^T x = \sum_{i=1}^{320} x_i \cdot w_{i,j} + b_j$ linearly combines the 320-dimensional fingerprint features x_i as the input of neuron j, n denotes the amount of subjects, and $P(y =$

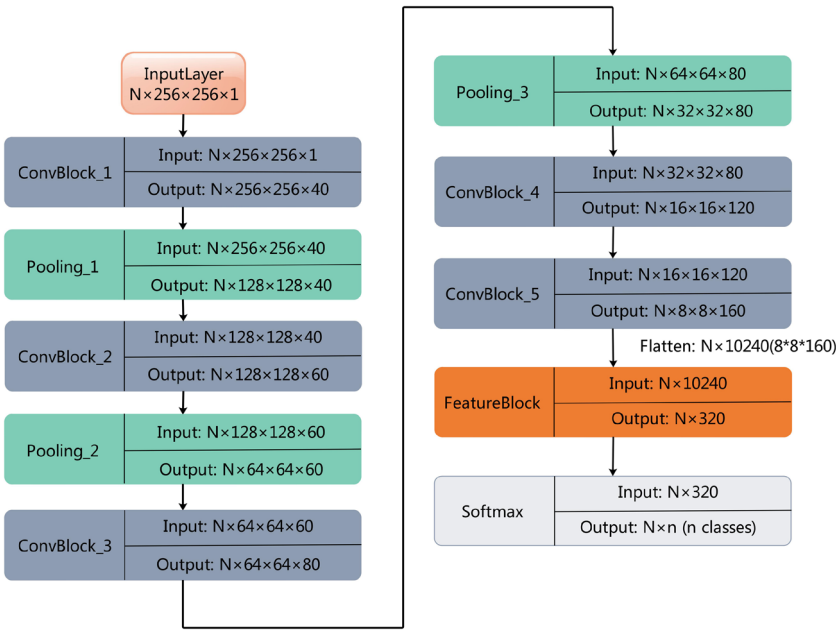

Fig. 4 Illustration of proposed Finger ConvNet. Scalar N means the size of the mini-batch (the number of images in an iteration input into the network for training) and the network contains the map number and the dimension of the input, convolutional blocks, max-pooling layers, the FeatureBlock and the softmax output layer. The last dimension of each layer represents the number of neurons in a layer

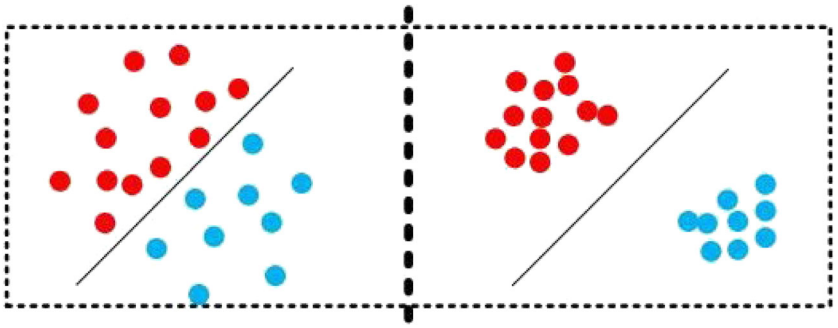

Fig. 5 The left image denotes the possible situation trained by minimizing the categorical cross-entropy loss, for which the intra-class spacing of features is greater than the inter-class spacing, while the right image is the desirable situation we expect [40]

$j|x;\theta)$ is its output, which is the probability of belonging to class j.

3.2 Training

Learning the end-to-end mapping function Conv$(\cdot)$ requires the estimation of network parameters θ, which includes the weights and biases in each layer. For this classification task, it can be achieved by classifying the feature vectors through an n-way softmax layer, which outputs a probability distribution over the n classes. Learning these network parameters is achieved by minimizing the categorical cross-entropy loss $\mathcal{L}_{softmax}$. It takes the formulation

$$\mathcal{L}_{softmax}(f, t, \theta_{ident}) = -\sum_{i=1}^{n} y_i \log p_i = -\log p_t, \tag{3}$$

where f is the extracted fingerprint vector, t denotes the target class and θ_{ident} represents the softmax layer parameters. y_i is a $1 \times n$ vector that denotes the target probability distribution, where $y_t = 1$ for the target class t while $y_i = 0$ for other classes i. p_i is the predicted probability distribution.

For the categorical cross-entropy loss $\mathcal{L}_{softmax}$, from the perspective of clustering, the extracted features are not good. As shown in Fig. 5, in many cases, the intra-class spacing is even greater than the inter-class spacing. We expect that the features are not only distinguishable but also require large differences. Feature learning needs to ensure that the extracted features are universal and of good recognition degree. To correctly classify all the subjects simultaneously, Finger ConvNet must form discriminative features.

Some loss function limits are added when training Finger ConvNet to improve the performance of the network. The first method is to introduce the center loss $\mathcal{L}_{center}$ [40] which was first proposed for face recognition. The center loss $\mathcal{L}_{center}$ is formulated as

$$\mathcal{L}_{center} = \frac{1}{2}\sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2, \tag{4}$$

where c_{y_i} indicates the y_ith class center of deep features, x_i is the extracted features before softmax layers and m is the size of the mini-batch. The equation effectively characterizes the intra-class variations, and it can result in reducing intra-class spacing. By adding the center loss, the trained network can output more cohesive features. We adopt the categorical cross-entropy loss $\mathcal{L}_{softmax}$ combined with the center loss $\mathcal{L}_{center}$ to train Finger ConvNet for learning the objective of the intra-class compactness, which is very beneficial to the discriminative feature learning. Then, our goal is to minimize the joint loss function $\mathcal{L}$; the formulation is given in (5).

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{softmax} + \lambda\mathcal{L}_{center} \\ &= -\sum_{i=1}^{n} y_i \log p_i + \lambda\sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2,\end{aligned} \tag{5}$$

where scalar λ is used as a weight for balancing the two loss functions. When λ is set to 0, the joint loss function $\mathcal{L}$ becomes the conventional categorical cross-entropy loss $\mathcal{L}_{softmax}$.

The second measure is to consider enlarging the inter-class features differences. For face verification tasks, contrastive loss $\mathcal{L}_{contras}$ [24] is proposed to enhance the discriminative power of the deeply learned face features. In the Siamese network [41, 42] of the Caffe framework [43] developed by Berkeley Vision and Learning Center (BVLC),[1] the contrastive loss is used as the loss function and this loss function can effectively contend with the relationship of paired data in networks. Similarly, for fingerprint data, we use contrastive loss to train Finger ConvNet to extract discriminative features. A verification signal is a label that indicates whether two samples in paired data belong to the same class, $y = 1$ for the same class and $y = -1$ for different classes. The verification signal directly adjusts the FeatureBlock of Finger ConvNet and can effectively enlarge inter-class spacing and can reduce

[1] http://caffe.berkeleyvision.org/.

the intra-class variations. The contrastive loss $\mathcal{L}_{contras}$ is expressed as

$$\mathcal{L}_{contras} = \frac{1}{2N}\sum_{j=1}^{N}(yd^2 + (1-y)\max(m-d,0)^2), \quad (6)$$

where y is the verification signal, $d = \|a_j - b_j\|_2$, which denotes the Euclidean distances between two sample features a_j and b_j in paired data. N is the number of paired data in a mini-batch, and N is equal to half of the size of mini-batch. Scalar m is a default margin threshold, and it indicates the expected distance of the feature between two different classes. When $y = 1$, contrastive loss $\mathcal{L}_{contras} = \frac{1}{2N}\sum_{j=1}^{N} yd^2$; if the Euclidean distance of paired data from the same class is large, it shows that the current model is not good. Conversely, when $y = 0$ it means that paired data are from different classes and contrastive loss $\mathcal{L}_{contras} = \frac{1}{2N}\sum_{j=1}^{N}(1-y)\max(m-d,0)^2$, and we expect that the distance of paired data is larger than margin m; thus, it shows that features of different classes could be distinguished better. The joint supervision of categorical cross-entropy loss and contrastive loss is adopted to train Finger ConvNet. The joint loss function $\mathcal{L}$ can be expressed as

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{softmax} + \mu\mathcal{L}_{contras} \\ &= -\sum_{i=1}^{n} y_i \log p_i \\ &\quad +\mu\left(\frac{1}{2N}\sum_{j=1}^{N} yd^2 + (1-y)\max(m-d,0)^2\right), \end{aligned} \quad (7)$$

To be similar to (5), scalar μ is used as a weight for balancing the two loss functions. Finger ConvNet is learned by minimizing the loss $\mathcal{L}$ using stochastic gradient descent (SGD) with the standard back-propagation. To avoid border effects during training, all the convolutional layers have no padding operations.

3.3 Feature extraction

After training the Finger ConvNet through the classification task, we will obtain the end-to-end mapping function Conv(·), which can form discriminative and compact features $f = \text{Conv}(x, \theta)$, where x is the input fingerprint data and θ denotes the learned Finger ConvNet parameters. Since more fingerprint classes are predicted during training, the generalization ability of Finger ConvNet will become stronger. By loading the trained model parameters of Finger ConvNet and inputting aligned fingerprint images, we can extract discriminative fingerprint features (320-dimensional feature vectors). Then, these feature vectors are ready for the final fingerprint verification task.

3.4 Verification tasks

For this task, we can use a learned binary classifier to judge whether they correspond to the same finger or not. In conventional machine learning, there are many methods to achieve classification tasks. Common classifiers include the SVM, k-nearest neighbor (KNN), logistic regression, decision tree, etc. However, these methods usually cause a large consumption of time and resources. Properties of neural networks (NNs) include the following: high classification accuracy, strong learning ability, strong robustness and fault tolerance for noise data. Moreover, NNs can approximate any nonlinear relationship. In our work, we choose a NN as the classifier.

As shown in Fig. 6, our neural network classifier is a simple topology graph that contains an input layer taking the FeatureBlock of Finger ConvNet, a fully connected layer followed by the activation function ReLU and the network output layer, which is a 2-way softmax, $y_i = \frac{\exp(x_i)}{\sum_{j=0}^{1}\exp(x_j)}$ for $i = 0, 1$, where x_i is the total input to an output neuron i, and y_i is its output. It represents a probability distribution over the two classes 0 or 1 (being the same fingerprint or not). The input data are the concatenated features from any two fingerprints (640-dimensional feature vectors), with labels 0 and 1 for different classes and the same class, respectively. The classifier is learned by minimizing the categorical cross-entropy loss $\mathcal{L}_{softmax} = -\log y_i$, where $i \in \{0, 1\}$ denotes the target class. The loss is minimized by stochastic gradient descent, where the gradient is calculated by backpropagation.

4 Experimental results

In this section, the experimental results are shown to evaluate the proposed method and its comparison with other algorithms. In concrete terms, we first present the experimental setups including datasets, training strategies

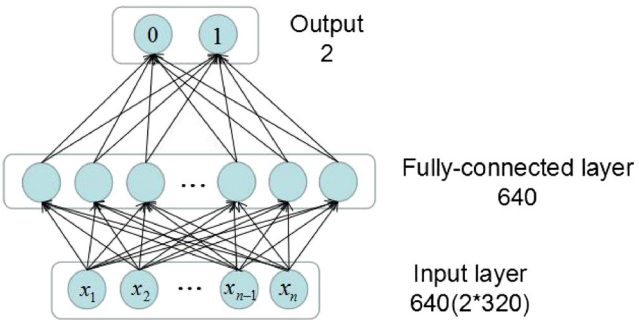

Fig. 6 The structure of NN classifier. It contains three layers: input layer, fully-connected layer and a 2-way softmax. The hidden neurons are ReLUs and the dimension is labeled beside each layer

Table 1 The detailed information of ID Card fingerprints, Ten-Finger Card fingerprints and the public datasets of FVC2000, FVC2002, FVC2004 and NIST sd04, sd14

Databases	Sets	Size	Resolution	Impression	Subject
Lab	ID Card	256 × 360	500 dpi	6 ∼ 10	8122
Database	Ten-Finger Card	640 × 640	500 dpi	1 ∼ 15	20223
FVC2000	DB1a	300 × 300	500 dpi	8	100
	DB2a	256 × 364	500 dpi	8	100
	DB3a	448 × 478	500 dpi	8	100
	DB4a	240 × 320	∼500 dpi	8	100
FVC2002	DB1a	388 × 374	500 dpi	8	100
	DB2a	296 × 560	569 dpi	8	100
	DB3a	300 × 300	500 dpi	8	100
	DB4a	288 × 384	∼500 dpi	8	100
FVC2004	DB1a	640 × 480	500 dpi	8	100
	DB2a	328 × 364	500 dpi	8	100
	DB3a	300 × 480	512 dpi	8	100
	DB4a	288 × 384	∼500 dpi	8	100
NIST	sd04	512 × 512	500 dpi	2	2000
	sd14	832 × 768	500 dpi	2	27000

and parameter setups. Second, we compare different training processes with different training strategies to choose the most suitable model. Finally, we compare our proposed algorithm with other algorithms in terms of fingerprint matching performance on public datasets. Meanwhile, the computation and speed analyses are stated to show the real-time capability of our method.

4.1 Experimental setup

The experimental evaluation of our proposed method and its comparison are performed on lab datasets and the public datasets: FVC2000 [44], FVC2002 [45], FVC2004 [46], NIST sd04 [47], and NIST sd14 [48]. The lab datasets contain two categories, i.e., ID Card fingerprints and Ten-Finger Fingerprint Card fingerprints, which are gray scale fingerprint images collected by the Ministry of Public Security of the People's Republic of China. In Table 1, the detailed information (e.g., size, resolution and subject, etc.) of datasets are reported. For all DB4s of FVC datasets, fingerprint images can be synthetically generated based on the method proposed in [49, 50]; then, the resolution is considered to be approximately 500 dpi or so.

Before training the Finger ConvNet, all fingerprint images are aligned. In our work, each finger (subject) is regarded as a class and an impression represents a fingerprint image. Due to fewer fingerprint images in each subject, the CNN network cannot be trained adequately. Therefore, some rigid transformations (e.g., rotation, translation) are used to achieve data augmentation and increase the diversity of data for training data. More concretely, we randomly rotate fingerprint images by an angle within 15° clockwise or counterclockwise and randomly move images in a parallel manner by a value within 20 pixels. Meanwhile, due to different sizes of these images, the network takes as input the aligned fingerprint images after the cropping or sampling process to fix the aligned images to the same size. In our work, all images are fixed with size of 256 × 256.

To learn a CNN model for feature extraction, we randomly choose 80% of subjects (finger classes) as training data, which include more images in a finger class from the lab database. In addition, we randomly select 10% of images of each training subject to generate the validation data. The residual data and all public datasets are used as test data to evaluate the performance of our approach. Then, Finger ConvNet is trained by classifying the fingerprint images and the transformed images simultaneously.

The weights of each layer are initialized by drawing randomly from a Gaussian distribution with zero mean and standard deviation 10^{-3} and biases are set to 0. We preprocess the data with the per-pixel mean subtraction and use a weight decay with the coefficient 1.0×10^{-4} to prevent overfitting. The learning rate is initialized to 0.01 and changed with iteration steps by an exponent drop.

The TensorFlow[2] (version 1.4.1) [51] framework is used to implement the proposed method, and the experiments are run on a machine with an Intel Core i7-6700 CPU at 3.4 GHz with 32 GB RAM, and an NVIDIA GeForce GTX 1060 GPU. Moreover, when testing our proposed method,

[2] http://tensorflow.org/.

Table 2 The accuracy rate (%) of the validation set for different networks on the ID Card fingerprint dataset and Ten-Finger Card fingerprint dataset

Dataset	VGG-16 [17]	GoogLeNet [18]	ResNet [19]	Proposed CNN
ID Card Data	97.85	94.66	96.18	98.42
Ten-Finger Card Data	95.93	93.61	94.87	96.89

we can even use a CPU to achieve the process. Finally, we conduct extensive experiments of fingerprint matching on our laboratory datasets with fewer impressions in a class and on public datasets. Due to permission issues and privacy concerns, we cannot make the lab datasets publicly available. However, we have made our test code and the trained CNN model publicly available[3] for other researchers to perform comparative studies.

4.2 Architecture of finger ConvNet evaluation

Based on the framework of VGG-16 [17], we design a CNN architecture Finger ConvNet applied to fingerprint data. In this section, we compare the validation set classification accuracy rates between the proposed CNN architecture and some typical network architectures (VGG-16 [17], GoogLeNet [18] and ResNet) under the supervision of softmax loss [19]. We train these networks on the ID Card fingerprint dataset and Ten-Finger Card fingerprint dataset under the supervision of softmax loss.

Table 2 lists the results of the accuracy rate of the validation set for different networks on the ID Card fingerprint dataset and Ten-Finger Card fingerprint dataset. From Table 2, we can note that our proposed CNN architecture outperforms other networks on the classification performance. Moreover, our proposed CNN architecture is designed based on VGG-16 [17]; therefore, it can be seen that the performance of VGG-16 is close to that of our proposed CNN architecture. Thus, we train the proposed CNN architecture to extract fingerprint features for matching.

The depth is important for CNN architecture in computer vison tasks. Based on the framework of VGG-16, the proposed CNN architecture contains five convolutional blocks. To verify whether depth is important for the proposed CNN architecture, we conduct experiments with the proposed CNN architecture containing different numbers of convolution blocks on the ID Card fingerprint dataset. The experimental results illustrated in Fig. 7 reveal that the deep CNN architecture achieves better classification performance than the shallow CNN. The main reason is that a deep neural network contains more expression ability for the variance of the same finger. Moreover, the proposed CNN architecture containing five convolution blocks achieves the best performance; therefore, we choose the CNN architecture for fingerprint matching evaluation.

4.3 Fingerprint classification experiments

We expect to learn discriminative and compact hidden features for final verification tasks, namely, learning the end-to-end mapping function Conv(·), so it is very critical to classify a large number of subjects accurately. This classification task can be achieved by classifying the feature vectors through an n-way softmax layer, which outputs a probability distribution over the n classes.

Due to different sources of fingerprint images, dataset distributions vary from different datasets and the learned models trained by different datasets have different performance outcomes. Moreover, as mentioned in Section 3, in many cases, if we only use the softmax loss as the supervision signal, the extracted features may not be discriminative. To solve this problem, we adopt the softmax loss combined with the center loss or contrastive loss to jointly

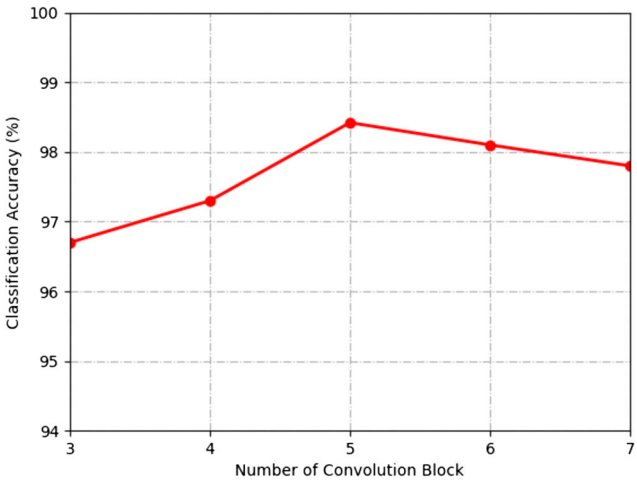

Fig. 7 The accuracy rate (%) of the validation set for different numbers of convolution blocks

[3] https://github.com/lyhucas/TestMatching

supervise the CNN architecture. By implementing contrasting experiments of different training methods or different data compositions and observing the validation set classification accuracy rates, we can select the model that provides the highest performance.

4.3.1 Experiments on different supervision signals

We separately train four kinds of CNN models under the supervision of softmax loss (3), softmax loss combined with center loss (5), softmax loss combined with contrastive loss (7) and softmax loss combined with mix loss.

As mentioned in Section 3, the center loss $\mathcal{L}_{center}$ can result in reducing intra-class spacing while the contrastive loss $\mathcal{L}_{contras}$ can effectively enlarge inter-class spacing and reduce the intra-class variations. Accordingly, we consider the two losses as a new supervision signal to reduce intra-class spacing and enlarge inter-class spacing so that the extracted features are more discriminative and cohesive. As shown in (8), the mix loss $\mathcal{L}_{mix}$ is a combination of center loss and contrastive loss

$$\begin{aligned}\mathcal{L}_{mix} &= \alpha\mathcal{L}_{center} + \beta\mathcal{L}_{contras} \\ &= \frac{\alpha}{2}\sum_{i=1}^{m}\|x_i - c_{y_i}\|_2^2 \\ &\quad +\beta\left(\frac{1}{2N}\sum_{j=1}^{N} yd^2 + (1-y)\max(m-d,0)^2\right), \quad (8)\end{aligned}$$

where scalars α and β are used as weights for balancing the two loss functions. Then, the joint loss function $\mathcal{L}$ can be expressed as $\mathcal{L} = \mathcal{L}_{softmax} + \mathcal{L}_{mix}$.

Figure 8 compares the validation set classification accuracy rates of four kinds of Finger ConvNet models under different supervision signals on different datasets. We train the CNN architecture to classify subjects on the ID Card fingerprints dataset (73,038 images in 6,000 subjects) and the Ten-Finger Card fingerprints dataset (99,714 images

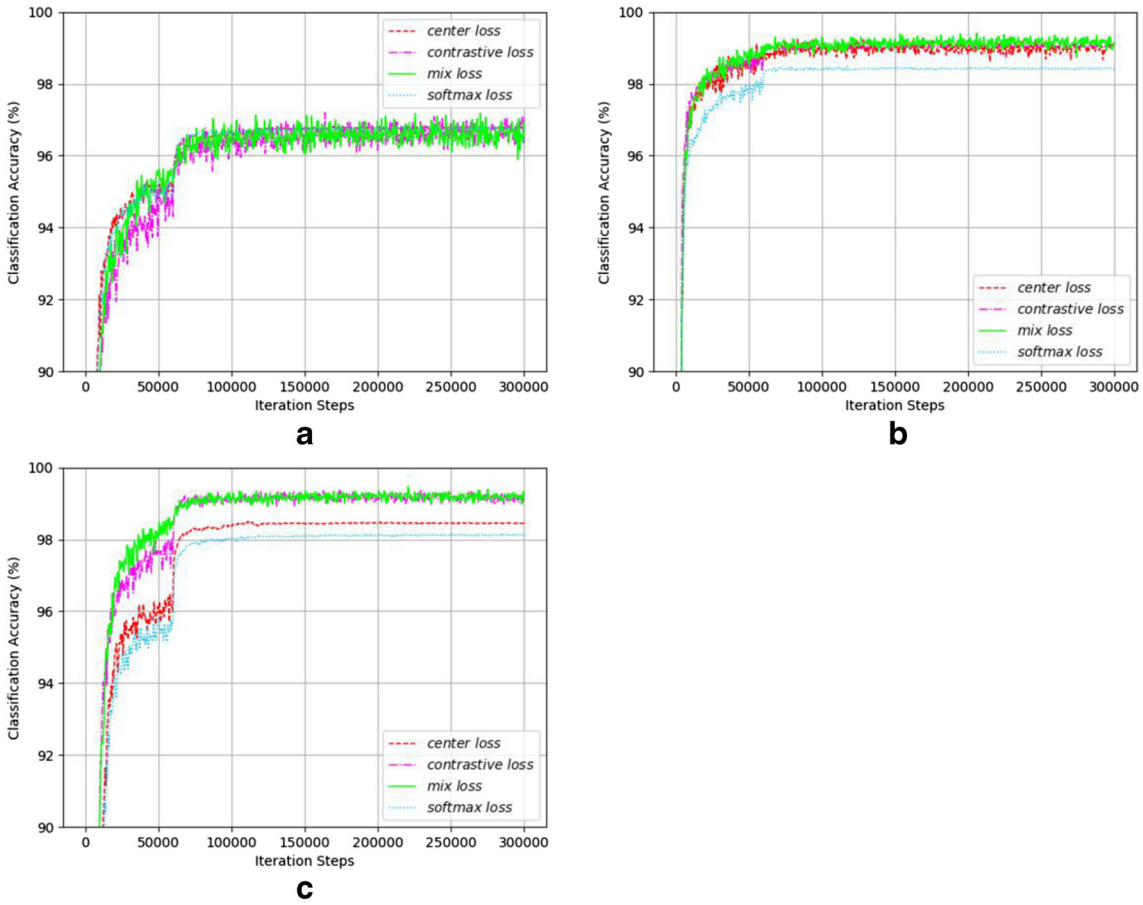

Fig. 8 The accuracy rate curves of the validation set of Finger ConvNet under different supervision signals on different datasets. The blue, red, purple and green markers show the validation set accuracy rates of the Finger ConvNet, in which the supervision signals are softmax loss, softmax loss combined with center loss, softmax loss combined with contrastive loss and softmax loss combined with mix loss, respectively, on the **a** Ten-Finger Card fingerprint dataset, **b** ID Card fingerprint dataset and **c** mixed-data dataset

Table 3 The EER (%) comparisons under supervision signals on the ten-finger card fingerprint dataset

Dataset	Softmax Loss	Center Loss	Contrastive Loss	Mix Loss
Ten-Finger Card dataset	4.191	2.638	1.945	1.633
mixed-data dataset	3.426	2.311	1.564	1.152

in 7,790 subjects). During training, we consider each finger as a class and expect that each class (finger) contains more images. Then, for the mixed-data dataset (149,537 images in 9,573 subjects), we choose these ID Card fingerprints and Ten-Finger Card fingerprints that consist of more images in a subject.

As shown in Fig. 8, joint supervision signals achieve better performance, and higher classification accuracy rates indicate that better hidden features can be learned on the ID Card fingerprint dataset, while the accuracy rate lines under different supervision signals on the Ten-Finger Card fingerprint dataset and mixed-data dataset are not clear. We train corresponding classifiers based on the different Finger ConvNet models trained on the two datasets. Then, we carry out verification tasks and compare equal-error rate (EER) on the Ten-Finger Card fingerprint dataset to determine which supervision signal is the most suitable for Finger ConvNet. EER represents the value where FMR and FNMR are equal. FMR represents the rate of different fingerprints that are considered to be the same by the template fingerprint matcher in a database. FNMR represents the rate of corresponding fingerprints that are erroneously considered different.

Table 3 lists the results of the EER comparisons of these models trained under supervision signals on the Ten-Finger Card fingerprint dataset. Table 3 shows that the joint supervision signal softmax loss combined with mix loss $\mathcal{L}_{mix}$ can achieve better classification performance than other supervision signals for the Finger ConvNet on different datasets. More concretely, the learned CNN architecture under this supervision signal can extract more discriminative features.

4.3.2 Experiments on different datasets

In this section, we train the Finger ConvNet on three different datasets to find the most suitable composition of datasets for learning the network according to the accuracies of the validation set. Using the same methods and datasets as described in Section 4.3.1, we compare the performance of Finger ConvNet learned under different supervision signals on different datasets in Fig. 9.

Figure 9 shows that the Finger ConvNet learned on the mixed-data dataset can achieve better classification accuracy rates than that learned on other datasets. That is, a more discriminative Finger ConvNet can be attained. The main reason is that our choice of these ID Card fingerprints and Ten-Finger Card fingerprints affords more impressions in a subject. Thus, the mixed-data dataset contains more fingerprint data and the samples are more diverse. In addition, Fig. 8 shows that the learned CNN architecture under joint supervision signals can achieve better performance. Therefore, we can choose the mixed-data dataset to train the Finger ConvNet under the supervision signal softmax loss combined with mix loss.

4.3.3 Experiments on parameters α and β

According to the experiments in Section 4.3.1, we find that the learned Finger ConvNet under the joint supervision signal softmax loss combined with mix loss can achieve better performance than that under other supervision signals. As shown in (8), the mix loss $\mathcal{L}_{mix}$ includes the center loss and contrastive loss. Scalars α and β are used to balance the two loss functions. In this section, we train the Finger ConvNet under this joint supervision signal with different ratios of scalars α and β to find an appropriate ratio.

Figure 10 illustrates that the learned Finger ConvNet employing a ratio of $1:5$ for scalars α and β in the joint supervision signal softmax loss combined with mix loss can achieve better classification performance than other ratios. Finally, we obtain a learned CNN architecture to extract discriminative and compact hidden features from the last hidden layer for verification tasks.

4.4 Performance evaluation for verification tasks

In the fingerprint classification experiments, by observing the validation set accuracy rates, we choose the learned model under the supervision signal softmax loss combined with mix loss on the mixed-data dataset to extract fingerprint features. In this section, we evaluate the proposed method and its comparison with others for verification tasks.

4.4.1 Performance of the proposed method

After choosing the best learned Finger ConvNet model in Section 4.3 and inputting aligned fingerprint images, fingerprint features (320-dimensional feature vectors) can be extracted. Then, we concatenate two feature vectors from any two fingerprints (the same subject or different

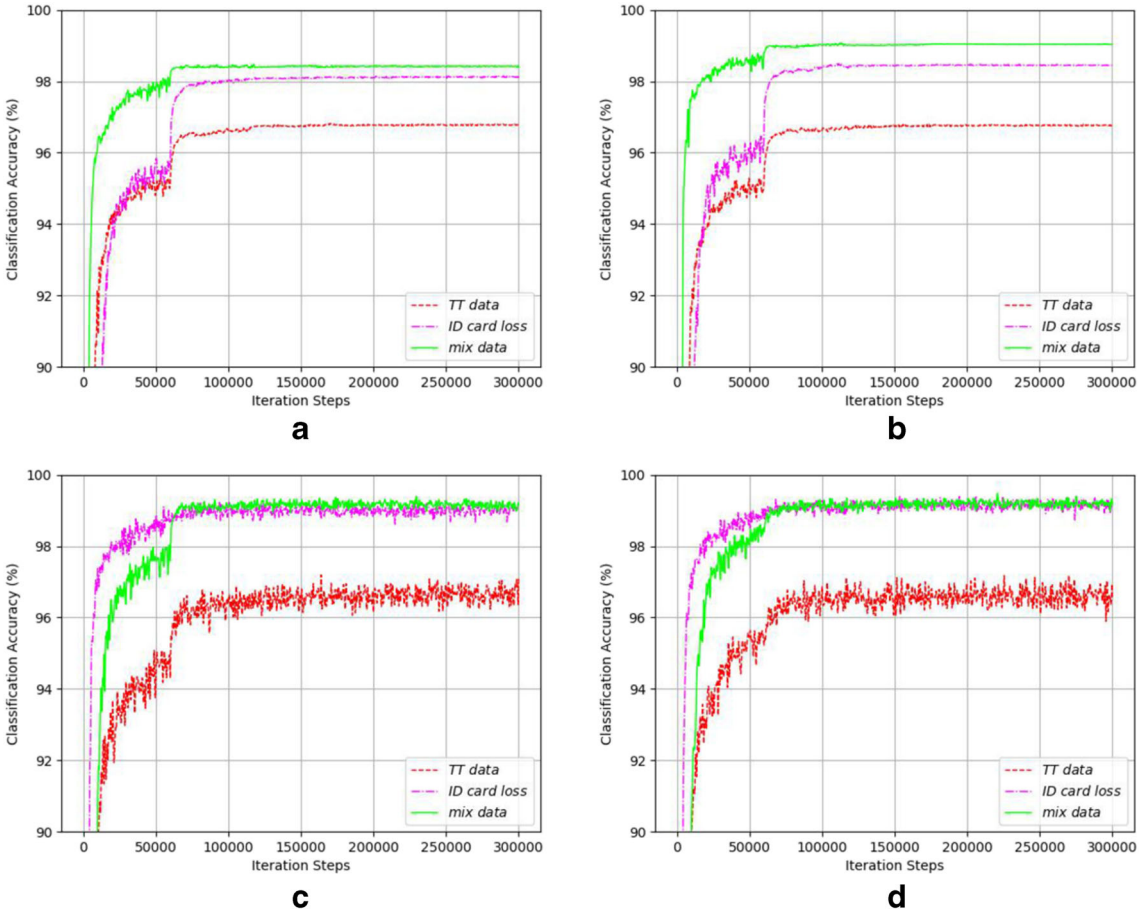

Fig. 9 The accuracy rate curves of the validation set of Finger ConvNet on different datasets. The red, purple and green markers show validation set accuracy rates of the Finger ConvNet, which is trained on the Ten-Finger Card fingerprints dataset, ID Card fingerprints dataset and mixed-data dataset, respectively, under supervision signal **a** softmax loss, **b** softmax loss combined with center loss, **c** softmax loss combined with contrastive loss and (d) softmax loss combined with mix loss

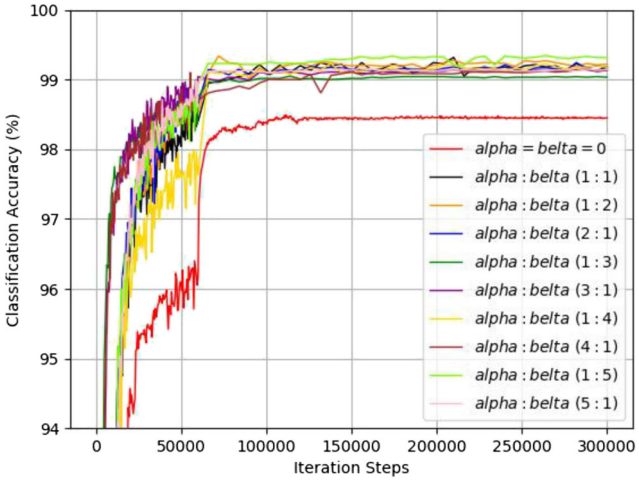

Fig. 10 The accuracy rate curves of the validation set of Finger ConvNet on different ratios for scalars α and β in the joint supervision signal softmax loss combined with mix loss. The red line shows the curve for $\alpha = \beta = 0$, namely, in the supervision signal softmax loss. Other lines show the curves of different ratios for scalars α and β

subjects) and input concatenated feature vectors to train a binary neural network classifier (shown in Fig. 6). For final verification tasks, we predict whether the two input fingerprints are from the same finger (label 0 for different fingers and label 1 for the same finger) through the learned binary classifier.

We evaluate the performance of our approach on the ID Card fingerprint dataset, Ten-Finger Card fingerprint dataset, NIST sd04 and NIST sd14. The false matching rate (FMR), false non-matching rate (FNMR), receiver operating characteristic (ROC) curve and equal-error rate (EER) are chosen as the indicators measured. In concrete terms, FMR represents the rate of different fingerprints that are considered to be the same by the template fingerprint matcher in a database. Each possible score has an associated FMR; the higher the score, the lower the FMR is. FNMR represents the rate of corresponding fingerprints that are erroneously considered different. EER represents the value where FMR and FNMR are equal, while ROC curves are

Table 4 The percentage results of the proposed method for verification tasks

Dataset	EER	FMR100	FMR1000	ZeroFMR
ID Card Data	0.722	0.678	2.450	29.719
Ten-Finger Card Data	1.152	1.276	4.752	33.726
NIST sd04	3.904	9.396	24.801	47.193
NIST sd14	2.011	3.269	11.789	39.863

the curves that plot the genuine matching rate (GMR = 1−FNMR) versus the FMR. The larger the corresponding area under the curve, the better the performance is.

The performance indices such as EER, FMR100, FMR1000 and ZeroFMR values on four different databases are presented in Table 4, where FMR100, FMR1000 and ZeroFMR mean the lowest achievable FNMR for FMR $\leq$ 1%, FMR $\leq$ 0.1% and FMR = 0%, respectively. Additionally, the ROC curves of the proposed algorithm over the four datasets are plotted in Fig. 11.

From Table 4, it can be seen that the experimental results of our approach on the ID Card fingerprint dataset achieve better performance than on the Ten-Finger Card fingerprint dataset, NIST sd04 and NIST sd14 dataset. It is mainly because the fingerprint image quality of the ID Card fingerprint dataset is better than that of the other three datasets, that is, ridges of the fingerprint images are more discriminative so that fingerprint features at three different levels (see Section 1) are clearer. The ROC curves also reveal the performance of our approach on four different databases. The corresponding area under the curve of the ID Card fingerprint dataset is larger than that of the others; therefore, we can also note that our approach on the ID Card fingerprint dataset achieves the best performance.

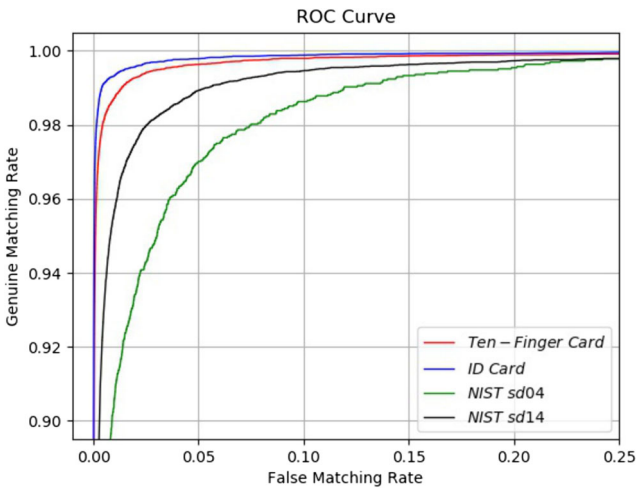

Fig. 11 The ROC curves of the proposed algorithm over the ID Card fingerprint dataset, Ten-Finger Card fingerprint dataset, NIST sd04 and NIST sd14

In particular, the average time for extracting a fingerprint feature is 3.9 milliseconds, and the average matching time for each fingerprint pair is 0.035 milliseconds on an NVIDIA GeForce GTX 1060 GPU, while the average times are 168 milliseconds and 0.083 milliseconds on an Intel Core i7-6700 CPU, respectively. Moreover, the process of feature extraction is performed offline. That is, the proposed method can achieve a very fast speed for online fingerprint matching even on a CPU.

4.4.2 Method comparison

In this section, we compare the recognition performance on FVC public datasets to demonstrate the effect of the proposed deep learning matching method more generally with conventional fingerprint matching algorithms: Bozorth3 in the NIST Biometric Image Software (NBIS),[4] MCC16n in [52], the method of He et al. [53], the method of Ghaddab et al. [9] and the method of Li et al. [11]. Similar to Section 4.4.1, we use the following verification measures: EER and FMR100. In addition, the average matching time (AMT) is also chosen as an additional performance evaluation indicator.

Table 5 reports the EER comparisons with conventional fingerprint matching methods over the twelve FVC datasets, and the best result for each database are stressed in boldface. From Table 5, we can note that our method outperforms other conventional fingerprint matching methods on EER performance over the 1st and 3rd databases of FVC2000 and over the 1st and 2nd databases of FVC2004. Our method can also achieve a competitive effect over the 2nd and 4th databases of FVC2000, the 1st and 3rd databases of FVC2002 and the 3rd database of FVC2004. This is mainly because the fingerprint features used in conventional fingerprint matching methods are artificially defined features (three levels of fingerprint features, see Section 1), such as fingerprint minutiae, which are detailed so that fingerprint features are discriminative. Therefore, conventional algorithms are sensitive to the minutiae quality as well as the fingerprint quality. In contrast, the features used in our method are not artificially defined: they are

[4]NIST Biometric Image Software (NBIS), http://www.nist.gov/itl/iad/ig/nbis.cfm.

Table 5 EER (%) comparisons with conventional fingerprint matching methods on FVC datasets

Database	Bozorth3	MCC16n [52]	He [53]	Li [11]	Ghaddab [9]	Ours
FVC2000 DB1a	7.481	3.292	1.791	1.67	1.572	**1.571**
FVC2000 DB2a	8.751	2.576	0.992	**0.81**	1.122	1.252
FVC2000 DB3a	18.750	4.133	3.543	4.64	4.228	**3.492**
FVC2000 DB4a	5.817	2.792	**1.647**	2.24	2.074	2.973
FVC2002 DB1a	15.286	**0.811**	1.963	1.57	0.955	1.793
FVC2002 DB2a	14.564	**0.611**	1.110	1.37	0.632	2.356
FVC2002 DB3a	20.062	4.187	4.312	**3.12**	3.569	4.311
FVC2002 DB4a	21.003	3.395	2.772	2.25	**1.305**	3.195
FVC2004 DB1a	17.374	7.039	9.335	5.08	5.125	**4.177**
FVC2004 DB2a	17.183	8.363	7.345	4.02	5.059	**2.495**
FVC2004 DB3a	6.265	5.292	8.529	**3.11**	3.645	3.498
FVC2004 DB4a	26.189	3.544	2.719	3.13	**2.148**	4.718
mean value	14.893	3.836	3.838	2.750	**2.619**	2.985
variance	38.972	4.657	7.939	1.647	2.423	**1.150**

Table 6 FMR100 (%) comparisons with conventional fingerprint matching methods on FVC datasets

Database	Bozorth3	MCC16n [52]	He [53]	Ghaddab [9]	Ours
FVC2000 DB1a	13.334	7.857	3.391	1.893	**2.504**
FVC2000 DB2a	18.712	6.714	1.492	**1.250**	1.707
FVC2000 DB3a	37.876	8.071	8.749	6.036	**5.760**
FVC2000 DB4a	14.012	8.657	5.051	**2.714**	8.883
FVC2002 DB1a	24.967	1.285	2.500	**1.000**	2.869
FVC2002 DB2a	22.645	0.714	1.250	**0.679**	4.101
FVC2002 DB3a	37.324	7.915	7.143	**5.000**	12.826
FVC2002 DB4a	52.152	5.285	3.429	**1.429**	9.106
FVC2004 DB1a	36.286	23.714	18.500	**8.393**	12.568
FVC2004 DB2a	35.089	18.571	13.393	6.536	**5.660**
FVC2004 DB3a	13.618	15.420	13.107	**4.786**	6.853
FVC2004 DB4a	60.790	13.103	4.214	**2.964**	15.366

Bold symbols are used to check the comparison results more clearly

Fig. 12 The example of incorrect matching cases from the FVC2004 DB1a database. The left image corresponds to the same finger with the middle and right fingerprint images. The left fingerprint image matches the middle fingerprint image with a high score (probability), while it does not match the right one

Table 7 AMT (ms) comparisons with conventional fingerprint matching methods on FVC datasets

Database	Bozorth3	MCC16n [52]	He [53]	Ghaddab [9]	Ours
FVC2000 DB1a	1.026	0.762	0.482	4.378	**0.082**
FVC2000 DB2a	1.719	0.979	0.541	4.341	**0.081**
FVC2000 DB3a	6.187	4.854	4.149	8.656	**0.081**
FVC2000 DB4a	3.145	0.436	0.434	1.403	**0.080**
FVC2002 DB1a	1.349	0.784	0.422	4.782	**0.081**
FVC2002 DB2a	1.233	0.901	0.551	8.514	**0.080**
FVC2002 DB3a	1.235	1.054	0.534	1.662	**0.085**
FVC2002 DB4a	1.268	0.745	0.397	3.026	**0.081**
FVC2004 DB1a	1.488	0.801	0.491	5.846	**0.081**
FVC2004 DB2a	1.534	1.275	0.680	4.269	**0.082**
FVC2004 DB3a	16.566	2.535	1.850	8.024	**0.082**
FVC2004 DB4a	1.312	0.840	0.461	4.184	**0.083**

Bold symbols are used to check the comparison results more clearly

learned by the proposed CNN architecture. Therefore, unlike other methods, the matching performance exhibits slight differences regardless of the dataset we employ to evaluate our method, even in the databases with fingerprint images that are of poor quality (e.g., FVC2000 DB3a). According to the mean value and variances of different methods over the FVC datasets, the variances of our method over different databases are smaller than that of

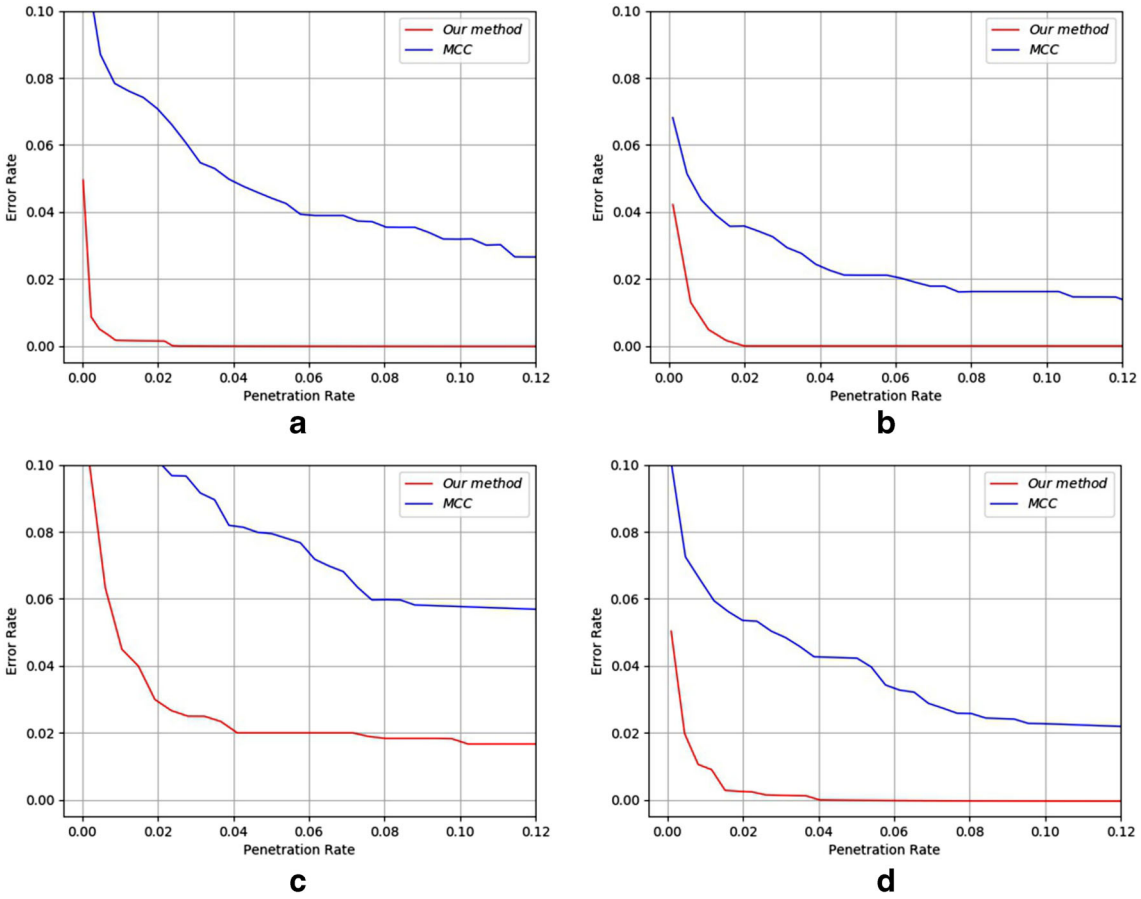

Fig. 13 Fingerprint indexing performance on FVC2000 for **a** DB1a, **b** DB2a, **c** DB3a and **d** DB4a. The blue and red markers show the screening performance in the fingerprint databases for MCC algorithm and our method, respectively

other methods. Thus, it demonstrates the robustness of our method, namely, our method is applicable to different types of fingerprint images.

As shown in Table 6, the FMR100 performance of our method is also competitive. Our method outperforms other conventional methods in terms of FMR100 over the 1st and 3rd databases of FVC2000 and the 2nd database of FVC2004. In other databases, our method gains a lower matching performance. However, upon evaluating the incorrect matching cases on these FVC datasets, we find that most of these cases appear due to the integrity of fingerprint images. Figure 12 shows one of the incorrect matching cases on the FVC2004 DB1a database. The left image corresponds to the same finger with the middle and right fingerprint images. In recognition tasks, the left fingerprint image matches the middle fingerprint image with a high score (probability), while it does not match the right one. From Fig. 12, we can see that the left and middle fingerprint images are relatively integral, but the right one misses part of the information. Therefore, we find that the integrity of fingerprint images can affect the matching performance of our method.

The process of feature extraction is accomplished offline and we compare the matching time based on the extracted feature, which is a fair approach to evaluate other conventional fingerprint matching methods. Then, we choose the average matching time (AMT) as a performance evaluation indicator for the matching speed comparison. Table 7 lists the results of AMT of our proposed method and the other conventional fingerprint matching methods on FVC datasets. From Table 7, it can be clearly seen that the proposed method achieves a very fast matching speed by comparing the speed performance of the conventional fingerprint matching algorithms. In particular, the matching speed is almost 5 times faster than the fastest one of the other algorithms over the twelve FVC datasets. Furthermore, since it is a fully feed-forward step and does not need to process some complex steps such as those performed in conventional fingerprint matching algorithms, the proposed method can achieve a fast and steady matching speed regardless of the dataset employed to conduct our experiments for fingerprint matching.

The proposed method does not contain a preprocessing operation and only computes network layer operations.

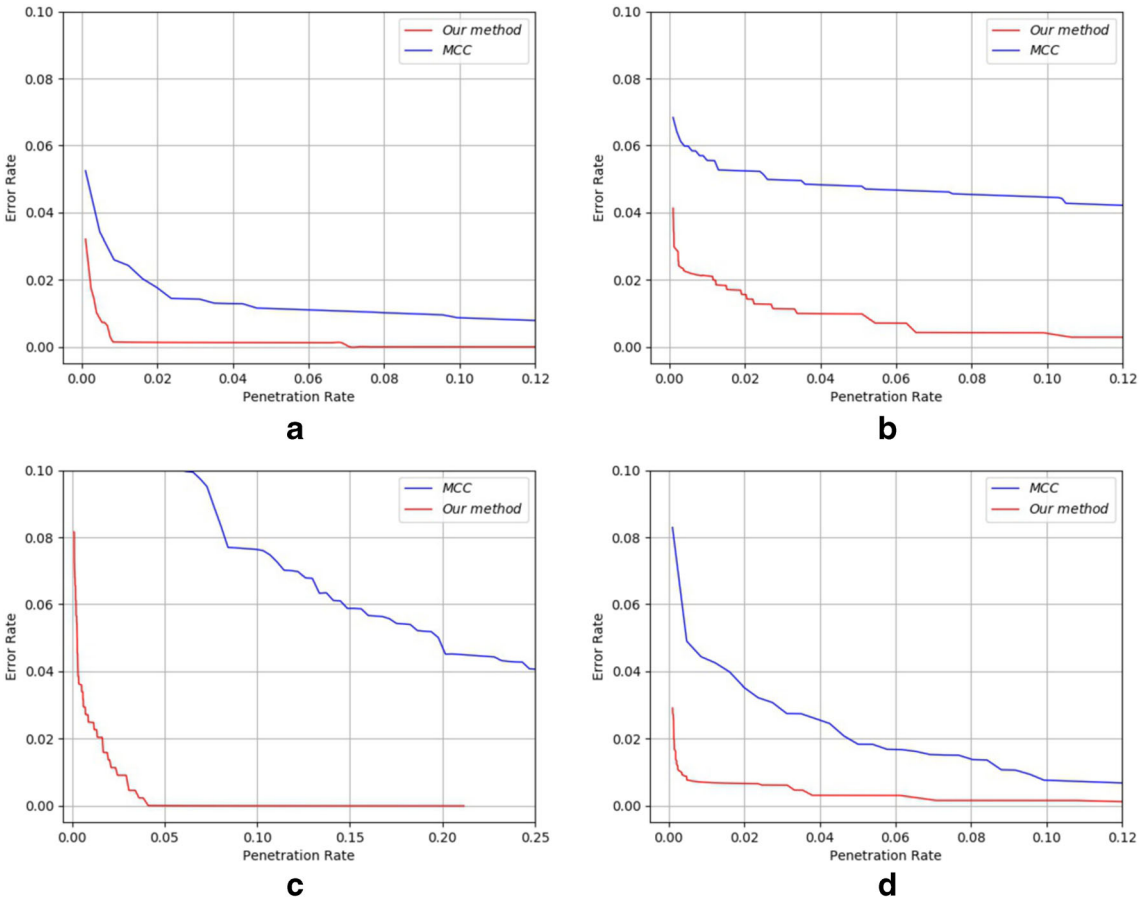

Fig. 14 Fingerprint indexing performance on FVC2002 for **a** DB1a, **b** DB2a, **c** DB3a and **d** DB4a. The blue and red markers show the screening performance in the fingerprint databases for MCC algorithm and our method, respectively

Therefore, by inputting a fingerprint pair into the trained network, we can determine whether it corresponds to the same finger or not, which is a very fast process with a matching speed that is almost 5 times faster than the fastest one of these conventional methods. This demonstrates that the proposed method has a strong computational performance, and especially when dealing with gigantic fingerprint databases, it can significantly improve the matching efficiency. Moreover, as shown in Tables 5 and 6, the proposed method outperforms other conventional algorithms and achieves competitive recognition performance in some databases, although it gains lower matching accuracy in the residual datasets. However, it is mainly because the fingerprint features used in conventional methods are artificially defined and detailed features, while the features used in the proposed method are not artificially defined and are learned by the proposed CNN architecture. Moreover, the integrity of fingerprint images can also affect the matching performance for our method. Though the proposed method does not outperform the conventional algorithms over all FVC databases, our method is more robust and the matching speed is much faster than other matching methods, so that it can be proven that the proposed method is efficient. Considering both the identification accuracy and the time cost, the proposed method shows attractiveness for the speedup of the fingerprint identification process.

4.5 Application case study: acceleration in identification tasks

In practical applications, fingerprint matching algorithms usually have to deal with gigantic fingerprint databases. Specifically, in fingerprint identification tasks, a query fingerprint identifies the matched template by operating in a 1:N matching process over such a large-scale fingerprint database, which usually increases significant time as well as the memory consumption. Furthermore, due to the high degree of fingerprint similarity in such a tremendous database, algorithms often cannot provide a reliable and precise result. Therefore, some acceleration methods are proposed to save computational resources and improve efficiency. For example, MCC (Minutia Cylinder Code) [34] is designed for fingerprint identification and retrieval,

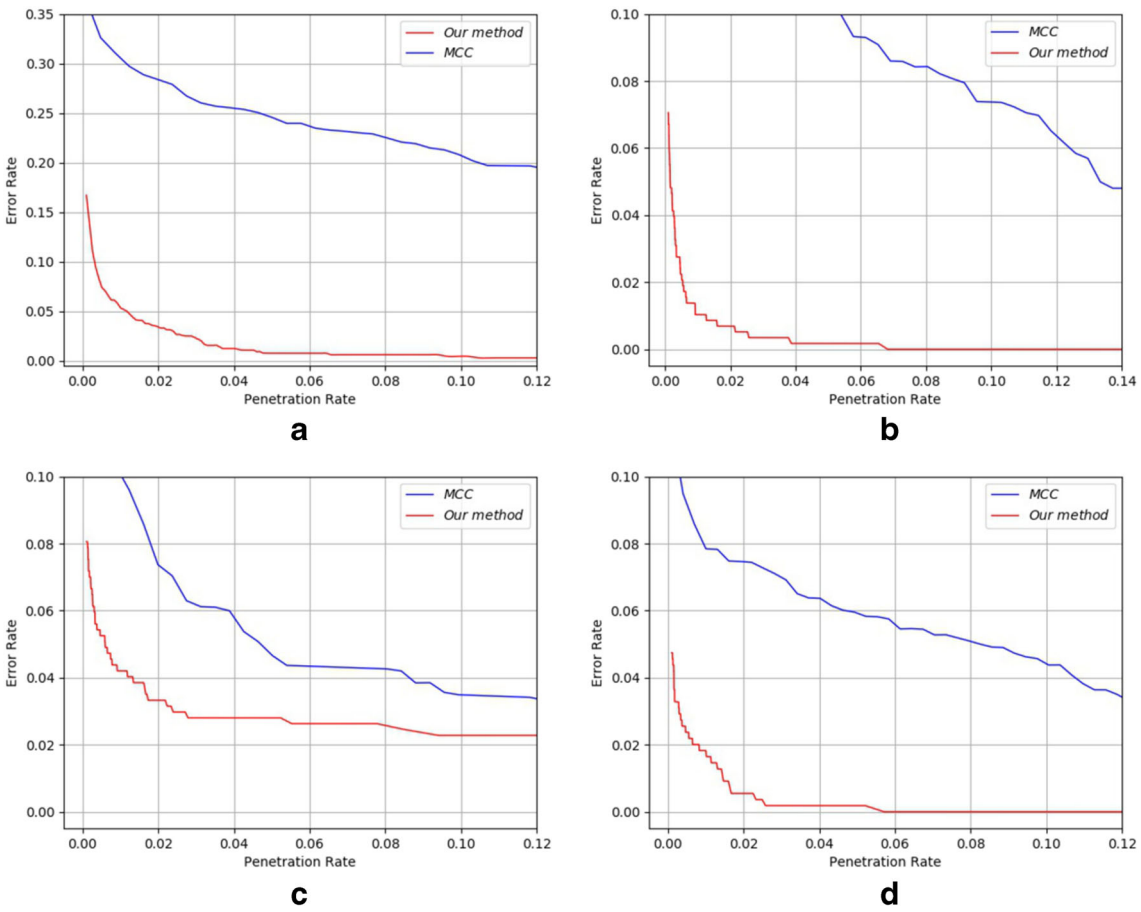

Fig. 15 Fingerprint indexing performance on FVC2004 for **a** DB1a, **b** DB2a, **c** DB3a and **d** DB4a. The blue and red markers show the screening performance in the fingerprint databases for MCC algorithm and our method, respectively

which is an operation that filters out a substantial amount of templates and only reserves very few possible matched templates.

Due to the competitive recognition performance and the fast matching speed of the proposed deep learning method in fingerprint matching, our method can support building a real-world application of acceleration in identification tasks. In this section, we compare the screening performance with the MCC algorithm on FVC public datasets according to the performance for the trade-off between the error rate (ER) and penetration rate (PR). Here, $ER = (Num_{err}/Num_d)$, where Num_{err} is the number of unfound searched fingerprints and Num_d is the number of all input searched fingerprints, while the penetration rate is the portion of the database that the algorithm has to search on the average.

We evaluate the proposed approach according to the screening performance in the fingerprint database and compare it with the MCC method. Here, the fingerprint database consists of 53,968 templates, from which we randomly choose 100 templates from each dataset of FVC2000, FVC2002 and FVC2004, and the residual data of FVC2000, FVC2002 and FVC2004 are used to search in this database to identify the matched templates. Figures 13, 14 and 15 show the trade-off between the penetration rate and error rate in FVC2000, FVC2002 and FVC2004 with all comparisons.

All of these figures clearly indicate that the proposed method outperforms the MCC method in fingerprint indexing performance and it can achieve a much higher precision than the MCC method. Importantly, our method can filter out a substantial amount of impossible templates and ensure high accuracy simultaneously when dealing with gigantic fingerprint databases. In other words, the proposed method can be used as an acceleration method to save appreciable computing time and memory consumption and improve the performance during identification process tasks.

5 Conclusions and future lines

A novel end-to-end method for fingerprint matching is proposed in this paper. Based on the framework of VGG-16, a CNN architecture (Finger ConvNet) is designed for fingerprint images, and we train the architecture to extract fingerprint features. According to these features, we can judge whether any two fingerprints correspond to the same finger or not by a learned binary neural network classifier. Furthermore, since our method is a fully feed-forward step and does not need to process complex preprocessing steps for verification tasks, our method can achieve a fast and steady speed and high precision on a GPU, even on a CPU.

In the experiment section, we evaluate the performance of the proposed method on our lab databases and two public datasets, NIST sd04 and NIST sd14. Then, we compare the performance on FVC public datasets with conventional fingerprint matching algorithms. Experimental results show that the proposed method is much faster than other conventional algorithms. Specifically, the matching speed of our method is almost 5 times faster than the fastest conventional algorithms. This demonstrates that the proposed method has a strong computational performance, and especially when dealing with gigantic fingerprint databases, it can significantly improve the matching efficiency. For the matching performance, the proposed method outperforms other conventional algorithms and achieves competitive recognition performance in some databases. In other databases, the proposed method gains a lower matching performance. The main reason is that the fingerprint features used in conventional fingerprint matching methods are artificially defined features, and we find that the integrity of fingerprint images can affect the matching performance for our method. Therefore, conventional algorithms are sensitive to the minutiae quality as well as the fingerprint quality. In contrast, the features used in our method are learned by the CNN architecture and are not artificially defined. Therefore, our method is robust regardless of the dataset employed to evaluate our method, even in the databases with images that are of poor quality. In addition, we use the proposed method as a fast matching method to reduce the searching space to accelerate identification tasks and compare the performance with the MCC method. Experimental results reveal that our method is more efficient for identification tasks. Furthermore, fingerprint matching algorithms have been fully developed in the past decades and there are very few new algorithms in recent years. Moreover, source codes of these new algorithms are not available, and we did not find other studies regarding the use of deep learning for fingerprint matching. Therefore, we do not compare the performance of fingerprint matching with the up-to-date algorithms.

In future work, we will continue to improve Finger ConvNet to achieve a more suitable architecture to extract discriminative fingerprint features and define a new feature type that combines three levels of fingerprint features to improve the matching performance. In addition, we also need to consider some reasonable strategies to achieve the fingerprint alignment for the proposed method and improve the matching precision, aiming to reach a commercial level.

Acknowledgment This research was funded by the State Key Program of National Natural Science Foundation of China under grant number 11731013 and 11331012, and by the National Natural Science Foundation of China under grant number 11571014.

References

1. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Handbook of fingerprint recognition. Springer, London
2. Hong L, Wan Y, Jain AK (1998) Fingerprint image enhancement: algorithm and performance evaluation. IEEE Trans Pattern Anal Mach Intell 20(8):777–789
3. Jain AK, Hong L, Bolle R (1997) On-line fingerprint verification. IEEE Trans Pattern Anal Mach Intell 19(4):302–314
4. Feng J, Jain AK (2011) Fingerprint reconstruction: from minutiae to phase. IEEE Trans Pattern Anal Mach Intell 33(2):209–223
5. Bazen AM, Verwaaijen GTB, Gerez SH, Veelenturf LPJ, Van der Zwaag BJ (2000) A correlation-based fingerprint verification system. In: Proceedings of the ProRISC2000 workshop on circuits, systems and signal processing, pp 205–213
6. Bazen AM, Gerez SH, Veelenturf L, Zwaag BV, Verwaaijen G (2000) A correlation-based fingerprint verification system. Stw Technology Foundation 31(5):652–655
7. Ross A, Reisman J, Jain AK (2002) Fingerprint matching using feature space correlation. International Workshop Copenhagen on Biometric Authentication 2359:48–57
8. Cavusoglu A, Gorgunoglu S (2007) A robust correlation based fingerprint matching algorithm for verification. J Appl Sci 7(21):3286–3291
9. Ghaddab MH, Jouini K, Korbaa O (2017) Fast and accurate fingerprint matching using expanded delaunay triangulation. In: Proceedings of the IEEE/ACS international conference on computer systems and applications, pp 751–758
10. Peralta D, Galar M, Triguero I, Miguel-Hurtado O, Benitez JM, Herrera F (2014) Minutiae filtering to improve both efficacy and efficiency of fingerprint matching algorithms. Eng Appl Artif Intell 32(32):37–53
11. Li Q, Nguyen VH, Liu J, Kim H (2017) Multi-feature score fusion for fingerprint recognition based on neighbor minutiae boost. IEIE Transactions on Smart Processing and Computing 6(6):387–400
12. Kumar R, Chandra P, Hanmandlu M (2016) A robust fingerprint matching system using orientation features. J Inf Process Syst 12(1):83–99
13. Guo JM, Liu YF, Chang JY, Lee JD (2014) Fingerprint classification based on decision tree from singular points and orientation field. Expert Syst Appl 41(2):752–764
14. Galar M, Derrac J, Peralta D, Triguero I, Paternain D et al (2015) A survey of fingerprint classification Part I: taxonomies on feature extraction methods and learning models. Knowl-Based Syst 81:76–97
15. Galar M, Derrac J, Peralta D, Triguero I, Paternain D et al (2015) A survey of fingerprint classification Part II: experimental analysis and ensemble proposal. Knowl-Based Syst 81:98–116
16. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the advances in neural information processing systems, vol 60, pp 1097–1105
17. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
18. Szegedy C, Liu W, Jia Y, Sermanet P et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
20. Luo X, Tian J, Wu Y (2000) A minutiae matching algorithm in fingerprint verification. In: Proceedings of the IEEE international conference on pattern recognition, vol 4, pp 833–836
21. Jiang X, Yau WY (2000) Fingerprint minutiae matching based on the local and global structures. In: Proceedings of the IEEE international conference on pattern recognition, vol 2, pp 1038–1041
22. Ouyang W, Zeng X, Wang X, Qiu S, Luo P, Tian Y et al (2017) Object detection with deformable part based convolutional neural networks. IEEE Trans Pattern Anal Mach Intell 39(7):1320–1334
23. Xue D, Wu L, Hong Z, Guo S, Gao L, Wu Z et al (2018) Deep learning-based personality recognition from text posts of online social networks. Appl Intell 48(11):4232–4246
24. Sun Y, Chen Y, Wang X, Tang X (2014) Deep learning face representation by joint identification-verification. In: Proceedings of the advances in neural information processing systems, vol 60, pp 1988–1996
25. Tripathi BK (2017) On the complex domain deep machine learning for face recognition. Appl Intell 47(3):1–15
26. Noda K, Yamaguchi Y, Nakadai K, Okuno HG, Ogata T (2015) Audio-visual speech recognition using deep learning. Appl Intell 42(4):722–737
27. Cho SB, Won HH (2007) Cancer classification using ensemble of neural networks with multiple significant gene subsets. Appl Intell 26(3):243–250
28. Acharya UR, Fujita H, Shu LO, Hagiwara Y, Tan JH, Adam M et al (2018) Deep convolutional neural network for the automated diagnosis of congestive heart failure using ecg signals. Appl Intell 49(1):16–27
29. Fujita H, Cimr D (2019) Computer Aided detection for fibrillations and flutters using deep convolutional neural network. Inf Sci 486:231–239
30. Wang R, Han C, Guo T (2016) A novel fingerprint classification method based on deep learning. In: Proceedings of the IEEE international conference on pattern recognition, pp 931–936
31. Nguyen DL, Cao K, Jain AK (2018) Robust minutiae extractor: integrating deep networks and fingerprint domain knowledge. In: Proceedings of the international conference on biometrics, pp 9–16
32. Liu Y, Zhou B, Han C, Guo T, Qin J (2018) A method for singular points detection based on faster-RCNN. Appl Sci 8(10):1853
33. Qin J, Tang S, Han C, Guo T (2017) Partial fingerprint matching via phase-only correlation and deep convolutional neural network. In: Proceedings of the international conference on neural information processing, pp 602–611
34. Cappelli R, Ferrara M, Maltoni D (2011) Fingerprint indexing based on minutia cylinder-code. IEEE Trans Pattern Anal Mach Intell 33(5):1051–1057
35. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
36. Clevert DA, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). arXiv:1511.07289
37. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. Comput Sci 3(4):212–223
38. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov RR (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
39. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the international conference on artificial intelligence and statistics, vol 15, pp 315–323
40. Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. In: Proceedings of the European conference on computer vision, pp 499–515
41. Chopra S, Hadsell R, Lecun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In:

Proceedings of the IEEE conference on computer vision and pattern recognition, vol 1, pp 539–546
42. Norouzi M, Fleet DJ, Salakhutdinov R (2012) Hamming distance metric learning. In: Proceedings of the advances in neural information processing systems, vol 2, pp 1061–1069
43. Jia Y, Shelhamer E, Donahue J, Karayev S et al (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM international conference on multimedia, pp 675–678
44. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2002) Fvc2000: Fingerprint verification competition. IEEE Trans Pattern Anal Mach Intell 24(3):402–412
45. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2002) FVC 2002: second fingerprint verification competition. In: Proceedings of the IEEE international conference on pattern recognition, pp 811–814
46. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2004) Fvc2004: third fingerprint verification competition. Lect Notes Comput Sci 3072(2):1–7
47. Watson CI, Wilson C (1992) Nist special database 4, Fingerprint database national institute of standards and technology
48. Watson CI (1992) Nist special database 14, Fingerprint database national institute of standards and technology
49. Cappelli R, Erol A, Maio D, Maltoni D (2000) Synthetic fingerprint-image generation. In: Proceedings of the IEEE international conference on pattern recognition, pp 471–474
50. Cappelli R, Maio D, Maltoni D (2002) Synthetic fingerprint-database generation. In: Proceedings of the IEEE international conference on pattern recognition, pp 744–747
51. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z et al (2016) TensorFlow: large-scale machine learning on heterogeneous systems. arXiv:1603.04467. Software available from tensorflow.org
52. Cappelli R, Ferrara M, Maltoni D (2010) Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. IEEE Trans Pattern Anal Mach Intell 32(12):2128–2141
53. He Y, Tian J, Li L, Chen H, Yang X (2006) Fingerprint matching based on global comprehensive similarity. IEEE Trans Pattern Anal Mach Intell 28(6):850–862

Yonghong Liu received a degree of B.S. from China University of Mining & Technology, Beijing in 2016. Currently, he is currently pursuing a Ph.D. degree in University of Chinese Academy of Sciences, China. His current research interests include image processing, machine learning, pattern recognition and computer vision.

Baicun Zhou received the B.S. degree in Mathematics and Applied Mathematics from University of Science and Technology of China in 2016. He is currently pursuing M.S. degree in Operational Research and Cybernetics from University of Chinese Academy of Sciences. His research interests include optimization, machine learning, computer vision and pattern recognition.

Congying Han received the P.H. degree in Applied Mathematics from University of Shanghai Jiaotong University in 2008. Now, she is a professor and doctorate tutor of University of Chinese Academy of Sciences. Her current research interests include optimization model and algorithm, machine learning, and pattern recognition.

Tiande Guo is a professor and doctorate tutor of University of Chinese Academy of Sciences. His current research interests include image processing, machine learning and optimization method & application. He has published some articles such as the journal of PAMII, Image Processing and so on.

Jin Qin received received the P.H. degree from University of Chinese Academy of Sciences, China in 2018. His current research interests include image processing, machine learning and pattern recognition.