

Learning discriminative and invariant representation for fingerprint retrieval

Dehua SONG^{*}, Ruilin LI, Fandong ZHANG & Jufu FENG^{*}

Key Laboratory of Machine Perception (Ministry of Education), Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Appendix A Related work

Appendix A.1 Fingerprint retrieval

Along with the increasing of fingerprint database, fingerprint retrieval has been a hot topic. In the last two decades, researchers proposed many fingerprint retrieval methods to improve the accuracy and efficiency of retrieval. The conventional features utilized in fingerprint retrieval can be roughly divided into three categories: global feature, local feature and matching scores:

- **Global feature:** These features describe the global patterns of ridge structure. Directional Field (DF), FingerCode and deep feature are usually employed to represent the global patterns. Lumini et al. [1] first proposed the retrieval method for fingerprint identification and used directional field as feature vector. On this basis, Jiang et al. [2] incorporated dominant ridge distance as a auxiliary feature to improve the retrieval performance. Liu et al. [3] employed a set of Polar Complex Moments (PCMs) for extraction of rotation invariant fingerprint feature. FingerCode was first employed for fingerprint retrieval in literature [4], which represented the texture attributes of core region by a series of Gabor filters. Leung et al. [5] modified FingerCode by using square sector instead of radial sector. Besides, Uysal et al. [6] attempted to represent the ridge patterns within a triangular area for elastic distortion. In 2017, deep feature [7, 8] was proposed to improve the accuracy of fingerprint retrieval.

- **Local feature:** Most methods attempt to represent minutiae local structure, for example, minutiae triplets, minutiae quadruplets and Minutia Cylinder-Code (MCC). Boer et al. [4] first employed minutiae triplets as feature for fingerprint retrieval. Then, Andres et al. [9] defined a triangle set based on extensions of delaunay triangulations to deal with missing and spurious minutiae. In 2017, authors of literature [10] further proposed expanded delaunay triangulation algorithm based on quality of images. In addition, Minutiae quadruplets [11] and minutiae pairs [12] were proposed to promote fingerprint indexing. Cappelli et al. [13] employed MCC as feature to improve retrieval performance. Then, Su et al. [14] added pose constraint to MCC method to reduce false matches.

- **Matching scores:** A few retrieval approaches [15] are based only on the matching scores between an input fingerprint and reference fingerprints instead of features.

Matching scores cannot narrow down the search space sufficiently because of weak discriminability. The retrieval methods based on local features have to compare each local feature pair, which limit the efficiency of retrieval. But, global features just need to compute the similarity between two feature vectors, which achieve better performance on efficiency than local features. Hence, this paper focuses on global feature. However, its hard for DF and FingerCode methods to deal with the plastic distortion of ridge structure. To solve this problem, this paper employs DCNN to learn a discriminative and distortion invariant representation for fingerprint retrieval.

Appendix A.2 Distortion addressing

The presence of distortion in fingerprints plays a significant role in the automatic fingerprint identification system. To deal with the problem of nonlinear distortion in fingerprint images, three types of methods have been investigated in literatures.

^{*} Corresponding author (email: songdehua@pku.edu.cn, fjf@cis.pku.edu.cn)

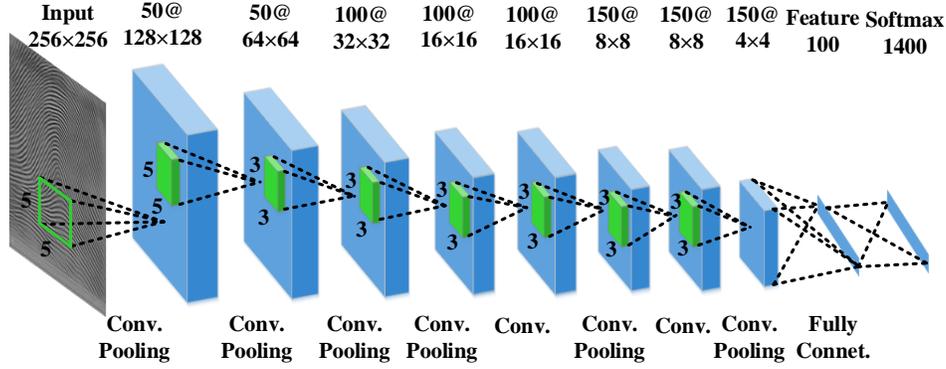


Figure B1 An illustration of the architecture of our DCNN. (Conv. is the abbreviation of convolution. Connet. is the abbreviation of connection.)

(1) Remove the distortion before feature extraction: Ross et al. [16] proposed a thin-plate spline (TPS) function to estimate the nonlinear distortion in fingerprint pairs based on ridge curve correspondence. The estimated average deformation was utilized to distort the template fingerprint prior to matching it with an input fingerprint. Si et al. [17] proposed a novel algorithm based on Support Vector Machine (SVM) classifier to detect the distortion. Then, they rectified the distortion by regression method.

(2) Design distortion invariant feature: Liang et al. [18] proposed a novel feature based on minutia neighborhood structure and low-order delaunay triangles, which was insensitive to fingerprint distortion. Then, extensions of delaunay triangles [9] were utilized to design distortion invariant feature.

(3) Consider distortion during similarity computing stage: Chen et al. [19] proposed normalized fuzzy similarity measure (NFSM) to deal with nonlinear distortion. Besides, to compute stable similarity, Cao et al. [20] proposed an ant colony optimization algorithm to establishment minutiae correspondences.

Due to the complexity of elastic distortion, it's hard to accurately rectify the distortion of fingerprint images. If we consider the distortion during similarity computing stage, it will seriously suppress the efficiency of retrieval. Hence, we devote oneself to learn a distortion invariant representation. Most researchers focus on invariant minutiae local structure which slightly suffers from fingerprint distortion. In consideration of retrieval efficiency, this paper investigates the distortion invariant of ridge structure.

Appendix B Feature representation

Appendix B.1 Architecture of our DCNN

The most challenging problem to apply DCNN to fingerprint retrieval is the scarcity of training data. In particular, the number of impressions per finger is rather limited (no more than 12). Hence, it's hard to utilize very deep convolutional neural networks with hundreds of layers. In this situation, the architecture of our DCNN is illustrated in Figure B1, which is adjusted from VGG network [21]. It contains ten learned layers: eight convolutional layers, a fully-connected layer and a softmax layer. There are six convolutional layers which are followed by a max pooling layer. The main function of max pooling layers is to reduce the resolution of the feature map and alleviate the sensitivity of feature to translations and distortions.

Differently from the typical DCNN, the size of the fully connected layer is designed as small as possible to ensure that the learned representation is low dimensional. In typical DCNN, the size of the fully connected layer is much larger than that of softmax layer. On the contrary, in our model, the size of the fully connected layer is much smaller than that of output layer. Hence, it ensures that the learned feature is compact and can improve the efficiency of retrieval system.

The learned feature extractor consists of convolutional layers, max pooling layers and the fully connected layer. The activation values of the fully connected layer are the learned feature illustrated in Figure B1, which are viewed as deep convolutional feature \mathbf{f}_c . Then, mean subtraction and l_2 -normalization are employed on the deep convolutional feature \mathbf{f}_c . The Normalized Deep Convolutional (NDC) feature \mathbf{f}_N is defined in equation B1.

$$\mathbf{f}_N = \frac{\mathbf{f}_c - \bar{\mathbf{f}}_c}{\|\mathbf{f}_c - \bar{\mathbf{f}}_c\|_2} \quad (\text{B1})$$

where $\bar{\mathbf{f}}_c$ denotes the mean of feature \mathbf{f}_c .

Appendix B.2 Training

To learn a discriminative and distortion invariant representation, we employ two standard techniques: data augmentation and metric loss.

Data augmentation A challenging problem to apply DCNN to fingerprint retrieval is the scarcity of training data. In particular, the number of samples per finger is rather limited. The DCNN illustrated in Figure B1 contains 1006250

parameters, but the total number of fingerprint images in all of four FVC databases is 16800. Furthermore, FVC databases (2000, 2002 and 2004) contain only eight instances per finger. Due to the scarcity of training data, the DCNN will suffer from severe overfitting. Hence, data augmentation technique which artificially enlarges the database using label-preserving transformations is employed to alleviate the overfitting of DCNN.

This paper employs four different forms of data augmentation. In order to ensure the training dataset contains translated and rotated fingerprint images, we randomly crop $w \times h$ patches in different locations from the enhanced images. The second form is to rotate the enhanced fingerprint images in different angles. Both cropping and rotation are rigid transformation. To ensure the distortion variance in training data, we propose two types of augmentation: slight shear transformation and distortion model method [22]. Slight shear transformation simulates the linear distortion and distortion model method simulates the nonlinear distortion. The distortion model is defined by a mapping $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ which can be viewed as an affine transformation (without scale change) which is progressively “braked” as it moves from external region c toward close-contact region a . Each point \mathbf{v} is mapped into $distortion(\mathbf{v})$ as equation B2 [22]. It can effectively strengthen the generalization ability of the DCNN when the training data is scarce. Besides, it is helpful to learn distortion invariant feature.

$$distortion(\mathbf{v}) = \mathbf{v} + \Delta(\mathbf{v}) \cdot brake(shapedist_a(\mathbf{v}), k) \quad (B2)$$

where $\Delta(\mathbf{v})$ specifies the affine transformation of the external region c ; $shapedist_a(\mathbf{v})$ indicates the position of \mathbf{v} with respect to the region a ; $brake()$ is a monotonically increasing function that rules the gradual transition from region a towards region c ; k is a skin plasticity coefficient.

Loss function To learn a discriminative and invariant representation, we joint softmax loss and cosine loss [23] as metric loss, which utilizes softmax loss to constrain the discrimination of inter-class and cosine center loss to constrain the invariance of intra-class. The whole loss function is defined in equation B3. Each finger is served as a class.

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C \quad (B3)$$

$$\mathcal{L}_S = - \sum_i^K \log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_j^J e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}} \quad (B4)$$

$$\mathcal{L}_C = - \sum_i^K \frac{\mathbf{x}_i \cdot \mathbf{c}_{y_i}}{\|\mathbf{x}_i\| \|\mathbf{c}_{y_i}\|} \quad (B5)$$

where \mathbf{W}_{y_i} and \mathbf{b} are the parameters of softmax layer, and \mathbf{x}_i denotes the i th deep convolutional feature, belonging to the y_i class. The size of mini-batch and number of classes is K and J , respectively. \mathbf{c}_{y_i} is the centroid of class y_i . λ is the coefficient of cosine center loss \mathcal{L}_C .

Optimization Stochastic Gradient Decent (SGD) is employed to train the DCNN by minimizing an objective function. Despite the power function is non-saturated, it still constrains the gradient of positive part. Figure ?? illustrates the gradient of five activation functions. It shows that the gradient of power is much lower than that of PReLU on most domain of definition. Hence, the power function converges much slower than PReLU. This issue is verified in section Appendix D.2 by experiment. To accelerate the converge speed of DCNN, we use DCNN whose activation function of all layers is PReLU to pre-train the model illustrated in Figure B1. After that, we fine-tune DCNN model with power function in the fully connected layer by SGD technique. Algorithm B1 shows the optimization algorithm for DCNN with power activation function.

Algorithm B1 Training the DCNN model with power activation function.

Input: Training set \mathbf{T} and validation set \mathbf{V} ;

Output: Parameters of DCNN \mathbf{W} ;

- 1: Assign PReLU to all convolutional layers and fully connected layer of DCNN.
 - 2: Initialize the weight with a zero-mean Gaussian distribution whose standard deviation is $\sqrt{\frac{2}{(1+a^2)n_l}}$ [24]. n_l denotes the number of connections of a response in layer l .
 - 3: Pre-train the DCNN model by SGD technique with \mathbf{T} and acquire the best parameters \mathbf{W}' in \mathbf{V} .
 - 4: Assign power function to the fully connected layer of DCNN and initialize it with \mathbf{W}' .
 - 5: Fine-tune the DCNN model by SGD technique with \mathbf{T} and acquire the ultimate parameters \mathbf{W} .
-

Appendix C Fingerprint retrieval system

The fingerprint retrieval system illustrated in Figure C1 is composed of two stages: an offline stage and an online stage. The model optimization and indexing features construction are carried out during the offline stage, and the fingerprint retrieving is carried out during the online stage.

Offline stage During the offline stage, we train the DCNN first with all the training data to acquire the feature extractor. Then, according to Figure C2, we extract and index features from all the fingerprints in gallery database. To catch more detail information of fingerprint, we utilize features extracted from multiple patches of one image to represent the fingerprint. For this purpose, we have to detect registration point which is defined by core point. For tented arch and loop, the core point is registration point. For whorl, the geometric center point of two core points is the registration point. In contrast,

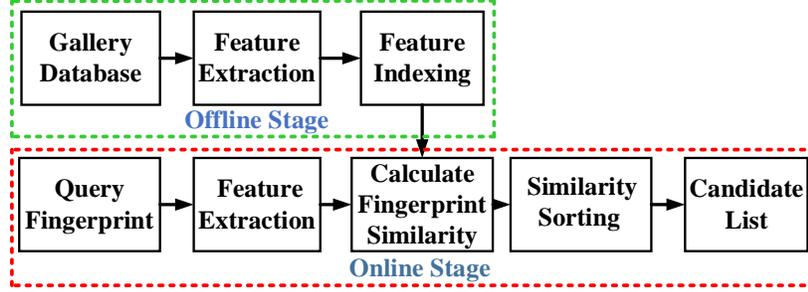


Figure C1 The flowchart of fingerprint retrieval system. (The procedure of feature extraction is illustrated in Figure C2)

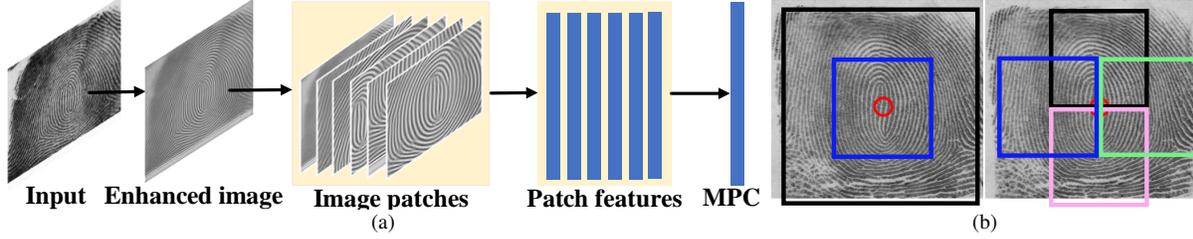


Figure C2 The flowchart of feature extraction for fingerprint retrieval. (a) The flowchart of feature extraction. (b) Patch cropping of fingerprint image. (○ denotes registration point. The largest patch will be down sampled by 2. Then the size of each patch is 256×256 .)

the registration point of arch is defined by the point of maximum curvature. The core point is detected by algorithm [25] based on gradient of double orientation field and Poincare. After detecting the registration point, we crop six patches from the image according to the Figure C2. DCNN is trained with the same architecture for each patch, respectively. We extract normalized deep convolutional feature f_N from each patch according to Figure B1. For retrieval efficiency, these normalized features are orderly concatenated with parameter η to construct a fixed-length feature vector f_M , named Multiple Patch Convolutional (MPC) feature, which is defined in equation C1.

$$f_M = [\eta_1 f_{N_1}^T, \eta_2 f_{N_2}^T, \dots, \eta_P f_{N_P}^T]^T \quad (C1)$$

where $\sum_{p=1}^P \eta_p^2 = 1$. P denotes the number of patches of one image.

Online stage During the online stage, the feature of query fingerprint image is extracted according to Figure C2. Then the similarity between query fingerprint feature and gallery fingerprint features is calculated in accordance with equation C2. In the end, the top N_C ranked fingerprints are selected as the candidates list by sorting the gallery fingerprints according to the similarity. Two retrieval strategies are employed in this paper: fixed order search [5] and incremental search. In fixed order search, given a target penetration rate, a candidates list whose length is proportional to the penetration rate is produced based on the similarity. In incremental search, fingerprints are visited according to the similarity between query fingerprint and gallery fingerprints. The search continues until a match is found.

$$Sim(f_{M_1}, f_{M_2}) = \langle f_{M_1}, f_{M_2} \rangle \quad (C2)$$

where f_{M_1} and f_{M_2} denote two MPC feature vectors of fingerprints. $\langle \rangle$ denotes scalar product.

Appendix D Performance evaluation

Appendix D.1 Details of experiments and databases

Three databases are utilized to train the DCNN: FVC2002, FVC2004 and FVC2006 database. Each database has four subsets which are collected by different scanners. In database FVC2002 and FVC2004, each set contains 800 fingerprint images taken from 100 fingers (eight impressions per finger). But, in FVC2006 database, each set contains 1800 fingerprint images taken from 150 fingers (twelve impressions per finger). To choose the optimal parameters of DCNN, we divide the training databases into training set and validation set. We randomly choose one fingerprint image from each finger to form the validation set and the remainder as the training set. Before data augmentation, the training set contains 12200 images taken from 1400 fingers, and the validation set contains 1400 fingerprint images taken from 1400 fingers. The learning rate of DCNN is 0.03. The coefficient λ of cosine loss is 0.01. The parameter $\eta_1 = \eta_2 = 0.58$, other $\eta_p = 0.15$. The dimension of final MPC feature f_M is 600.

To evaluate the performance of proposed fingerprint retrieval system, this paper carries out experiments on four benchmark databases which are used in most of published methods for fingerprint retrieval. They are the following databases:

- FVC2000 DB2a: The second set of FVC2000 database, which contains 800 fingerprint images taken from 100 fingers.
- FVC2000 DB3a: The third set of FVC2000 database, which contains 800 fingerprint images taken from 100 fingers.

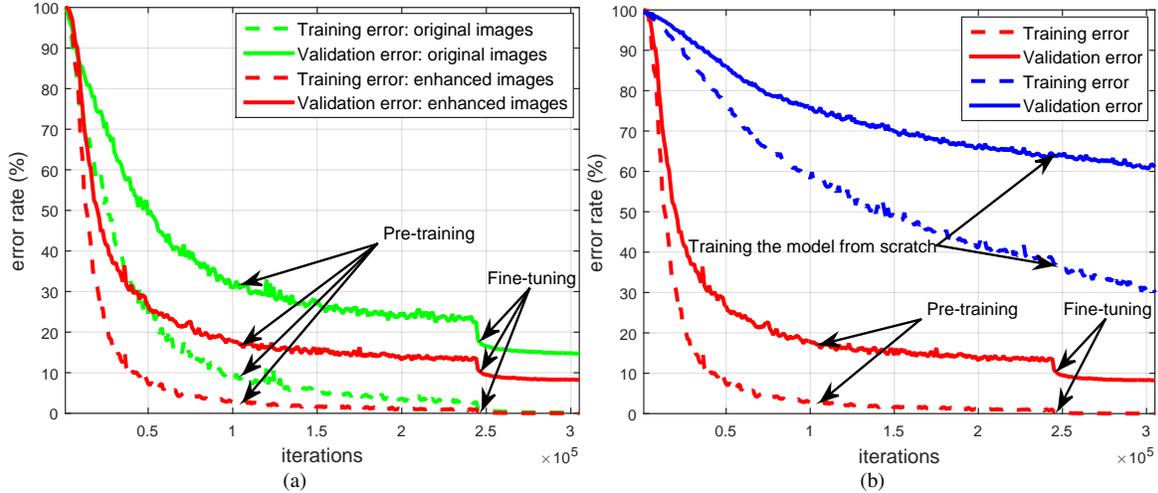


Figure D1 The training error of our DCNN with different strategies. (a) Error rate of DCNN trained by original images and enhanced images. (b) Error rate of DCNN trained by different initialization methods.

- NIST-4: The NIST special database 4 contains 2000 rolled fingerprint image pairs which are scanned from cards. The fingerprints are evenly distributed over each of five classes (Right Loop, Left Loop, Whorl, Arch, Tented Arch).
- Natural NIST-4 : We extract 2408 fingerprint images taken from 1204 fingers (two impressions per finger) from NIST-4 database following the natural proportion of Henry class distribution [1].

There are two standard indicators to evaluate the performance of fingerprint retrieval system in published literatures. The first indicator is the trade-off between error rate and penetration rate, which usually depends on the maximum number of candidates list. The error rate is defined as the percentage of query fingerprints whose true mate is not present in the candidate list. The penetration rate is the percentage of fingerprints in the gallery database that are chosen to be candidates. The second indicator of fingerprint retrieval system is the average penetration rate during incremental search, which utilizes an ideal matching algorithm to stop the search as soon as the true mate is retrieved. In experiments, the first samples are used as the gallery database templates to be indexed while the other samples serve as query fingerprints for retrieval, and cross validation is performed to test.

Appendix D.2 Evaluating the design of fingerprint retrieval system

Optimization evaluation Comparison experiments are carried out to evaluate the design of enhancement and optimization algorithm. Firstly, we evaluate whether the fingerprint enhancement can improve the performance of DCNN. The DCNN is trained with original images and enhanced images, respectively. The training and validation error rate versus iterations are illustrated in Figure D1. It shows that all the training set error rates of different training sets can drop to about zero, but the validation set error rate of enhanced images is much lower than that of original images. Hence, the fingerprint enhancement can effectively improve the performance of DCNN when the training data is scarce. This can be explained by the fact that the fingerprint enhancement effectively improve the ridge structures and filter out noises. It is helpful for DCNN to learn the common character of ridge structures.

Secondly, we evaluate whether the optimization algorithm can effectively accelerate the converge speed of DCNN. Comparison experiments are carried out with different training approaches. We trained the DCNN from scratch which is initialized with a zero-mean Gaussian distribution. In contrast, we trained the DCNN by the proposed optimization approach summarized in algorithm B1. The results of comparison experiment are illustrated in Figure D1. It reveals that converge speed of DCNN trained by the proposed optimization approach is much faster than that of DCNN trained from scratch. In conclusion, the proposed optimization approach can effectively accelerate the converge speed.

Activation function evaluation To evaluate the performance of power activation function, we first carry out experiments to choose an appropriate parameter for power function. Hence, we trained DCNN with different parameters of power function and evaluate the retrieval performance of different representations on Natural NIST-4 database. Table D1 summarizes the experimental results. It shows that parameter 7/9 achieves lower error rate than other parameters on most penetration rate. When parameter increases from 7/9, the error rate of retrieval begins to arise. Hence, all the following experiments employ the function defined in equation D1 as the activation function of the fully connected layer.

$$f(x) = x^{\frac{7}{9}} \quad (D1)$$

where x is the input of activation function.

Experiments are carried out to compare the performance of power activation function and PReLU and exhibit the drawback of PReLU on discriminative feature extraction. We trained the DCNNs with different activation functions on the fully connected layer and computed their retrieval error rate on Natural NIST-4 databases. The similarity distributions of

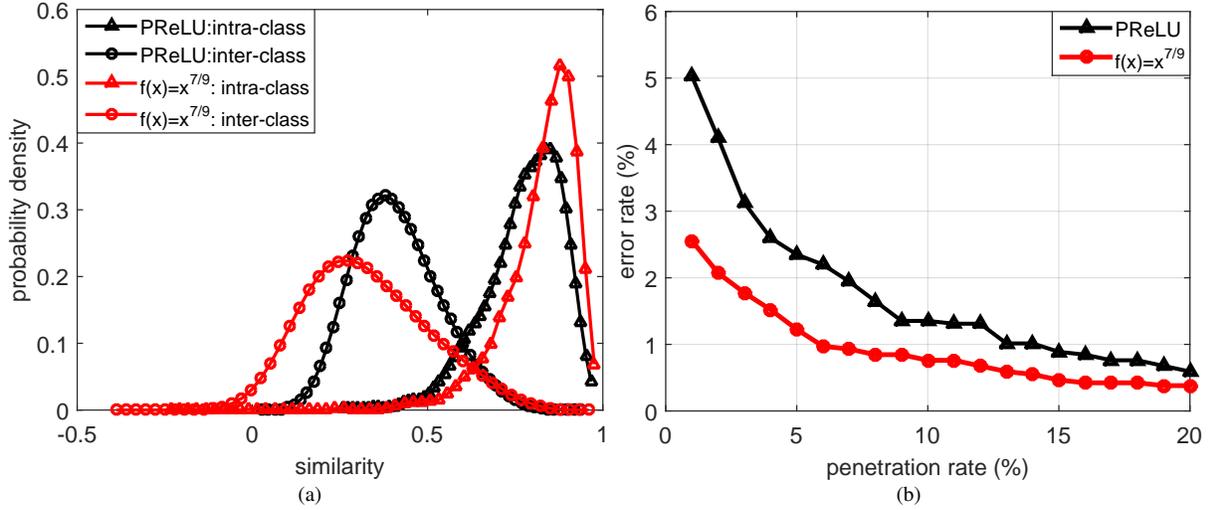


Figure D2 Experimental results of activation function evaluation on Natural NIST-4 database. (a) Probability density distribution of feature similarity from intra-class and inter-class. (b) Retrieval performance of different activation functions.

Table D1 Error rate versus penetration rate of power function with different parameters on Natural NIST-4 database. Best results are highlighted with bold font.

Parameter (p/q)	Penetration rate (%)					
	1%	2%	5%	10%	15%	20%
1/3	4.02%	3.57%	2.12%	1.14%	0.92%	0.63%
3/5	2.94%	2.43%	1.50%	0.91%	0.58%	0.48%
5/7	2.83%	2.37%	1.42%	0.87%	0.53%	0.44%
7/9	2.55%	2.07%	1.22%	0.76%	0.46%	0.38%
9/11	2.59%	2.08%	1.20%	0.75%	0.49%	0.38%
11/13	2.85%	2.54%	1.51%	0.86%	0.63%	0.59%

features from intra-class and inter-class over the Natural NIST-4 database are illustrated in Figure D2. It reveals that the power function effectively degrades the average similarity of inter-class (i.e., increases the angles between two features of different fingerprints). It's mainly because power function enables DCNN to distribute the centroids of fingerprint classes uniformly in the whole space, which contributes to learn a discriminative representation. From Figure ?? we can see that power function effectively enlarge the angles between two features of different fingerprints. Besides, Figure D2 shows that power function effectively increases the average similarity of intra-class (sharper distribution). It's mainly because power function makes DCNN learn more invariant representation to distortion, which is illustrated in Figure ?. Hence, power function degrades the intersection area of inter-class and intra-class in probability density distribution chart. In other words, it enhances the discriminability of representation learned by DCNN. The retrieval performance of the two features on Natural NIST-4 database is illustrated in Figure D2. It shows that power function achieves better performance than PReLU. In conclusion, power function can effectively improve the discriminability and invariance of learned representation.

Appendix D.3 Retrieval performance evaluation

Retrieval performance on benchmark databases Comprehensive experiments are conducted to evaluate the retrieval performance of proposed method on four databases: FVC2000 DB2a, FVC2000DB3a, NIST-4 and Natural NIST-4 database. The retrieval results are illustrated in Figure D3 and Figure D4. Make a general survey of the four performance curves, we can see that the proposed retrieval approach achieves lower error rate than other prominent methods. We didn't compare with method in literature [8], which utilized non-public database with large-scale fingerprint images to train DCNN. We only use public databases to train the DCNN. FVC2000 DB3a is a typical database in which images are of bad quality. Figure D3 reveals that the proposed method even achieves much better performance than other approaches. It's mainly because power activation function enables DCNN to learn more discriminative and invariant representation to distortion.

Table D2 reports the average penetration rate for the incremental search scenario on all the benchmark databases and compares it with the other prominent results. It reveals that the proposed approach achieves the state-of-the-art performance on four benchmark databases.

Efficiency evaluation We implement the DCNN by Python programming language with Pytorch to evaluate the efficiency of feature extraction. The average time of extracting all patch features from an image on Tesla K80 is 17.55

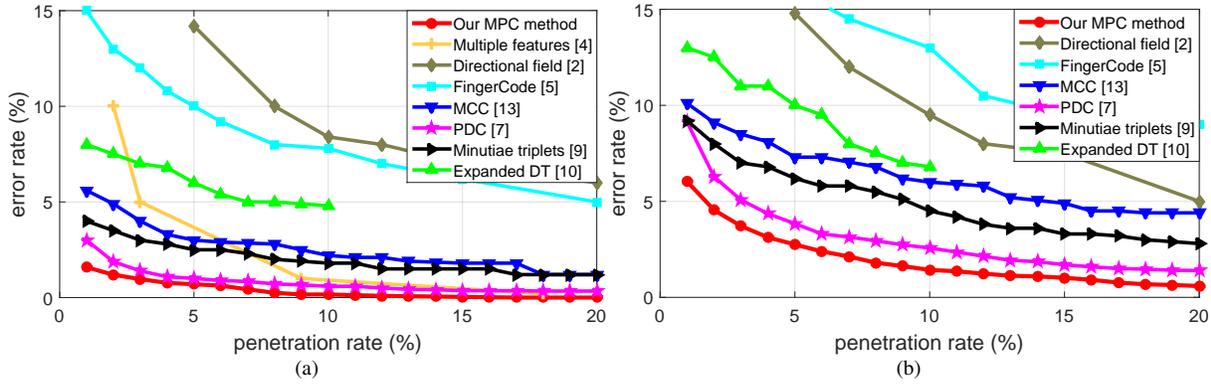


Figure D3 Retrieval performance on FVC databases. (MCC: Minutia Cylinder-Code, PDC: Pyramid Deep Convolutional, DT: Delaunay Triangulation) (a) Retrieval performance on FVC2000 DB2a database. (b) Retrieval performance on FVC2000 DB3a database.

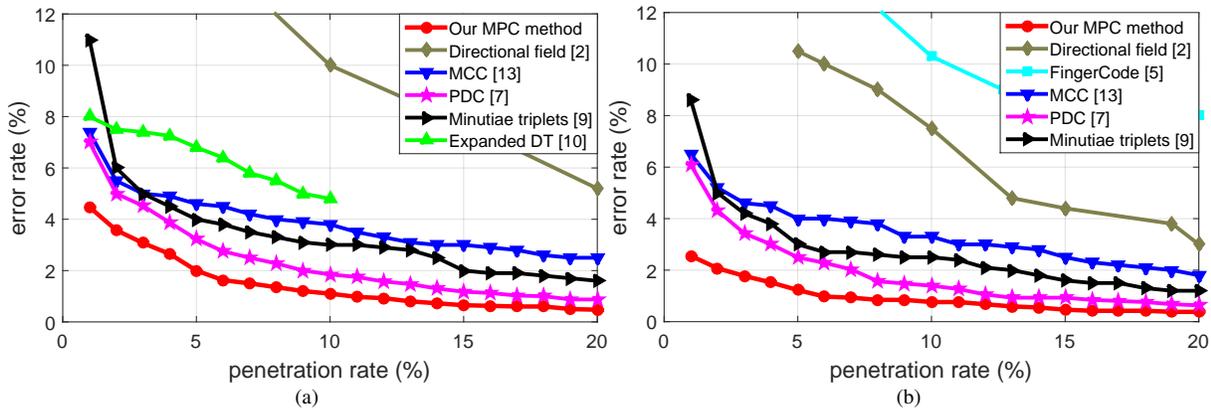


Figure D4 Retrieval performance on NIST databases. (MCC: Minutia Cylinder-Code, PDC: Pyramid Deep Convolutional, DT: Delaunay Triangulation) (a) Retrieval performance on NIST-4 database. (b) Retrieval performance on Natural NIST-4 database.

milliseconds. Compared with human-crafted feature extraction methods, the efficiency of feature extraction based on DCNN is available for fingerprint retrieval.

To evaluate the retrieval efficiency of proposed method, we implement the MCC-based method [13], PDC method [7] and our method by MATLAB programming language, and executed them with Intel Core i5 CPU at 2.4GHz. As a consequence, the proposed method spends 0.32 milliseconds to search a fingerprint against gallery database with 1204 images, but PDC method requires 0.52 milliseconds and MCC-based method requires 56.5 milliseconds. The efficiency of our method is about two orders of magnitude faster than MCC-based method. It mainly because our method just needs to compute the scalar product of two feature vectors, however MCC-based method has to compute the similarity of each MCC pair of two fingerprints. The efficiency of our method is also faster than PDC approach. Because the dimension of MPC feature is more compact than that of PDC feature (600 VS. 1500).

References

- 1 Lumini A, Maio D, Maltoni D. Continuous versus exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, 1997, 18:1027-1034.
- 2 Jiang X, Liu M, Kot A C. Fingerprint Retrieval for Identification. *IEEE Transactions on Information Forensics and Security*, 2006, 1:532-542.
- 3 Liu M, Yap P T. Invariant representation of orientation fields for fingerprint indexing. *Pattern Recognition*, 2012, 45:2532-2542.
- 4 Boer J D, Bazen A M, Gerez S H. Indexing fingerprint databases based on multiple features. *Proc.annu.workshop Circuits Systems Signal Processing*, 2001.
- 5 Leung K C, Leung C H. Improvement of Fingerprint Retrieval by a Statistical Classifier. *IEEE Transactions on Information Forensics and Security*, 2011, 6:59-69.
- 6 Uysal M, Gorgunoglu S. Ridge Pattern Representation for Fingerprint Indexing. *Elektronika Ir Elektrotechnika*, 2014,

Table D2 Average penetration rate for incremental search scenario: best results are highlighted with bold font.

Method	NIST		FVC2000	
	Natural DB4	DB4	DB2a	DB3a
Our approach	0.38%	0.51%	1.08%	1.49%
Pyramid Deep Convolutional (PDC) [7]	0.67%	0.79%	1.29%	2.02%
Minutia Cylinder-Code (MCC) [13]	1.32%	1.59%	1.72%	3.63%
Polar Complex Moments (PCMs) [3]	-	5.34%	-	-
Directional Field [2]	2.93%	-	-	-
Directional Field [1]	6.90%	-	-	-
Directional Field [4]	-	-	2.58%	-
Triplet [4]	-	-	7.27%	-
FingerCode [4]	-	-	2.40%	-
Combined [4]	-	-	1.34%	-

20.

- 7 Song D, Feng J. Fingerprint indexing based on pyramid deep convolutional feature. *IEEE International Joint Conference on Biometrics*, 2017, 200-207.
- 8 Cao K, Jain A K. Fingerprint indexing and matching: An integrated approach. *IEEE International Joint Conference on Biometrics*. IEEE, 2017, 437-445.
- 9 Gago-Alonso A, Hernandez-Palancar J, Rodriguez-Reina E, et al. Indexing and retrieving in fingerprint databases under structural distortions. *Expert Systems with Applications*, 2013, 40:2858-2871
- 10 Khodadoust J, Khodadoust A M. Fingerprint indexing based on expanded Delaunay triangulation. *Expert Systems with Applications*, 2017, 81: 251-267.
- 11 Iloanusi O N. Fusion of finger types for fingerprint indexing using minutiae quadruplets. *Pattern Recognition Letters*, 2014, 38:8-14.
- 12 Khodadoust J, Khodadoust A M. Fingerprint indexing based on minutiae pairs and convex core point. *Pattern Recognition*, 2017, 67: 110-126.
- 13 Cappelli R, Ferrara M, Maltoni D. Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, 33: 1051-1057.
- 14 Su Y, Feng J, Zhou J. Fingerprint indexing with pose constraint. *Pattern Recognition*, 2016, 54:1-13.
- 15 Gyaourova A, Ross A. A novel coding scheme for indexing fingerprint patterns. *Joint IAPR International Workshops on Statistical, Structural and Syntactic Pattern Recognition*, 2008, 755-764.
- 16 Ross A, Dass S C, Jain A K. Fingerprint warping using ridge curve correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, 28:19-30.
- 17 Si X, Feng J, Zhou J, et al. Detection and Rectification of Distorted Fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37:555-568.
- 18 Liang X, Bishnu A, Asano T. A Robust Fingerprint Indexing Scheme Using Minutia Neighborhood Structure and Low-Order Delaunay Triangles. *IEEE Transactions on Information Forensics and Security*, 2007, 2:721-733.
- 19 Chen X, Tian J, Yang X. A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure. *IEEE Transactions on Image Processing*, 2006, 15: 767-776.
- 20 Cao K, Yang X, Chen X, et al. A novel ant colony optimization algorithm for large-distorted fingerprint matching. *Pattern Recognition*, 2012, 45:151-161.
- 21 Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Computer Science*, 2014.
- 22 Maltoni D, Cappelli R. Advances in fingerprint modeling. *Image and Vision Computing*, 2009, 27:258-268.
- 23 Wang J, Li Y, Miao Z, et al. Learning deep discriminative features based on cosine loss function. *Electronics Letters*, 2017, 53:918-920.
- 24 He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*. 2015, 1026-1034.
- 25 Fan L L, Wang S G, Guo T D. Global and local information combined to detect singular points in fingerprint images. *Science China (Information Sciences)*, 2013, 56:62103-062103.