SCIENCE CHINA
Information Sciences

• **Supplementary File** •

FClassNet: A fingerprint classification network integrated with domain knowledge

Yao TANG, Ruilin LI, Yuhang LIU & Jufu FENG*

Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, China

Appendix A Methodology

Appendix A.1 Related work

Fingerprint classification is still a difficult task due to the high intra-class variability, low inter-class variability and the presence of complex noise. The development of fingerprint classification techniques is also the way to overcome more realistic and complex noise. Fingerprint classification algorithms usually contain two modules: feature extraction and classifier design.

Features used in classification can be categorized as the ridge line flow, orientation field, singular points, and Gabor filter responses. The ridge line flow is traced according to the orientation field and represents the global ridge flow [1]. The orientation field is the most important feature of a fingerprint which describes the overall ridge orientation. Since the orientation field contains sufficient information for classification, most fingerprint classification techniques are based on the orientation field [2–4]. Singular points contain core and delta points in an orientation field, which is usually used to determine fingerprint classes according to some human designed rules [5–7]. Some authors proposed specific enhancement algorithms to strengthen the enhanced value in singular points [8].

The orientation field and singular points are the most widely used features for fingerprint classification, and they both have shortages. Many researchers attempt to combine these features to boost the classification performance. Two neural networks are designed based on the orientation field and singular points respectively in [9]. A third network is then used to combine the results. Zhang et al. [6] exploit singular points and pseudoridge tracing based on the orientation field to get robust results. The coefficients of the orientation field and singularity features are combined as input to an SVM classifier in [7].

Classifiers used in fingerprint classification can be categorized as hand-crafted rules and learning based methods. Hand-crafted rules contain rule-based [6, 10], syntactic [11], structural [12] methods. These methods usually design some rules based on singular points and orientation fields. Such as the location and number of singular points and the orientation of the core point. Learning based methods contain statistical [13], SVM [7], neural network [14] and multiple classifiers [1, 5, 8, 15]. These methods take orientation, singular points and enhancement as input and learn features from data. Since there may be ambiguity to categorize some fingerprints into specific classes, some authors exploit fuzzy classification techniques to represent the fingerprint class as a probability value [4].

Recently, considering the great success of deep convolutional networks on natural images [16–21], plenty of deep learning based methods are proposed to classify fingerprints end-to-end. Stacked autoencoder is used to obtain the reconstruction of fingerprints and softmax regression is used for classification in [4]. However, the performance is limited due to the shallow network. The block average of the orientation field and segmentation are put into a 6 layer convolutional network for fingerprint classification in [22]. The orientation field is estimated by a gradient based method which limits the performance on low quality fingerprints. VGG network [17] which is trained on natural images is directly used to classify fingerprint images with fine tuning in [23]. AlexNet [16] and a shallow version are used in [24]. However, the VGG network and AlexNet are time-consuming and occupy large memories. Besides, these networks are usually regarded as black boxes which are difficult to further analyze and optimize.

There is rarely published work on latent fingerprint classification. We believe that one of the main reasons is that the presence of complex noise makes it hard to accurately recognize the fingerprint class, let alone to consider the algorithm's time complexity. Besides, some incomplete latent fingerprints do not contain sufficient information to determine the class. Khan et al. [25] classify fingerprints through 3 steps: enhancement, segmentation and classification. The first two steps use classic methods, including gradient based orientation field estimation and Gabor enhancement. The last step uses a

* Corresponding author (email: fjf@cis.pku.edu.cn)

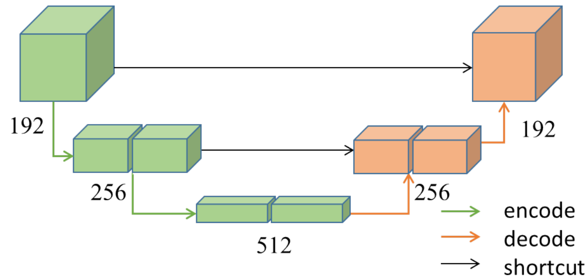


Figure A1 The architecture of UNet used in orientation field estimation, segmentation and singular points extraction. The number below the feature map denotes the number of channels.

convolutional network for classification. However, the gradient based orientation field estimation and segmentation are sensitive to noise and the enhancement will also be affected. In this paper, we attempt to classify fingerprints robustly and quickly by integrating the deep network and domain knowledge.

Appendix A.2 Orientation field and segmentation

Based on the backbone network, the design and training of orientation field estimation are described in the following subsections.

Appendix A.2.1 *Network architecture*

Based on the backbone network, two UNets [26] are used to produce the orientation field and segmentation from OF and QF. The architecture of UNet is illustrated in Figure A1, which explicitly integrates multi-scale information to acquire robust results. The UNet adopt encode and decode layers just like a fully convolutional network. Moreover, a shortcut is connected from the encode layer to decode layer to combine high-level semantics and low-level fine-grained information.

Appendix A.2.2 *Orientation representation*

There are two points to consider in the orientation representation of fingerprints: the periodicity of orientation and the equivalence of 0 and 180 degrees. The commonly used fingerprint orientation representation method is to represent an orientation value θ as $(cos(2\theta), sin(2\theta))$. In this paper, we represent the orientation value in the form of a probability vector. Compared with the sine-cosine representation, the probability vector contains more information (probability of each orientation). Specifically, we discretize the 180 degrees ranged orientation field to 90-dimensional vectors, and the value of the vector indicates the probability of the orientation. So each orientation value is expressed as $p = \{p(i)\}_{i=0}^{N-1}$, where $p(i)$ denotes the probability of the orientation to be $2 * i$. To get the precise orientation value, the weighted average response [27] can be used as,

$$\theta_{ave}(x,y) = \frac{1}{2} atan2(\bar{d}_{sin2}(x,y), \bar{d}_{cos2}(x,y)), \tag{A1}$$

where $\bar{d}_{-2}(x,y)$ denotes the weighted sum of sine and cosine of the double angle, which is usually used to calculate the mean angle against the periodicity of the orientation. It is defined as,

$$\bar{d}_{cos2}(x,y) = \frac{1}{N} \sum_{N} p(i) \cdot cos(2 * 2 * i), \tag{A2}$$

$$\bar{d}_{sin2}(x,y) = \frac{1}{N} \sum_{N} p(i) \cdot sin(2 * 2 * i), . \tag{A3}$$

Appendix A.2.3 *Label and loss function*

Since there are no manually marked labels of orientation field or segmentation, and manual labeling is time-consuming and labor-intensive, we use weakly supervised learning to train fingerprint backbone network. Considering that most data sets have manually marked minutiae points and matching relationships between latent and tenprint fingerprints, we generate the orientation field labels from corresponding tenprint fingerprints according to the matching relationship.

Orientation labels The orientation labels consists of global and local orientation labels. The global orientation labels are orientation fields for fingerprints. These labels are generated from matched and aligned tenprint fingerprints through the following 3 steps:

(1) Matching the paired tenprint and latent fingerprints by a minutiae-based matching algorithm and generating the matched minutiae pairs.

(2) Aligning the paired fingerprints according to the matched minutiae pairs.

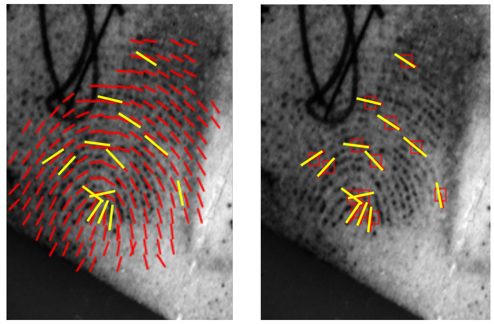


Figure A2 Orientation labels. Red lines denote global orientation labels and yellow lines denote local orientation labels.

(3) Calculating the orientation field on tenprint fingerprints by FingerNet [28] and used as both tenprint and latent fingerprints' global orientation labels.

Because of the existence of non-linear deformation, the global orientation field is not absolutely accurate. Since the minutiae orientation is manually marked, we define local orientation labels from minutiae. The orientation of minutiae and orientation fields have different ranges, such as 360 degrees and 180 degrees. The local orientation label from minutiae orientation is defined as $p_{ori} = mod(p_{mnt}, \pi)$. Figure A2 shows the orientation labels.

Segmentation labels The segmentation labels are generated by using the manually marked minutiae points. We use dilated and Gaussian smoothed convex hulls of the minutiae points as the weak segmentation labels of fingerprints.

Loss functions For orientation labels, the weighted cross entropy loss is used as,

$$L_{ori} = -\frac{1}{|ROI|}\sum_{ROI}(\beta y_i log(p_i) + (1-\beta)(1-y_i)log(1-p_i)), \tag{A4}$$

where β is used to balance the positive and negative weights, y_i is the ground truth value and p_i is the predicted value. $|ROI|$ is the foreground region which represents segmentation area and minutiae centred area in global and local orientation labels respectively.

Orientation coherence [27] reveals the characteristic of fingerprint orientation as a consistent field, which is an important regularization term. It is defined as,

$$\begin{aligned} \bar{d} &= [\bar{d}_{cos2}(x,y), \bar{d}_{sin2}(x,y)], \\ ||\bar{d}||_2 &= \sqrt{(\bar{d}_{cos2}(x,y))^2 + (\bar{d}_{sin2}(x,y))^2}, \\ Coh &= \frac{||\sum_{3*3}(\bar{d})||_2}{\sum_{3*3}(||\bar{d}||_2)}, \end{aligned} \tag{A5}$$

where $\sum_{3*3}$ denotes the sum in a 3*3 neighborhood, and $||\cdot||_2$ denotes the two-norm of a vector. The closer the Coh is to 1, the more consistent the orientation field.

For the segmentation loss function, we use intersection over union (IoU) loss function which is more consistent with the segmentation target. IoU is usually used to measure the overlap between the predicted region and the target region. It is defined as the intersection of two regions divided by the union of them. Here we use IoU loss function to constrain segmentation. The IoU loss function is defined as follows:

$$L_{IoU} = \frac{\sum_{i=1}^{N} y(i) * p(i)}{\sum_{i=1}^{N} y(i) + \sum_{i=1}^{N} p(i) - \sum_{i=1}^{N} y(i) * p(i)}, \tag{A6}$$

where y_i is the ground truth value and p_i is the predicted value.

Since the backbone network is constrained directly by the orientation field and segmentation, these features can be acquired through a simple network inference. Moreover, a simple convolutional layer with given weights is designed to smooth the orientation output. For the segmentation, we make a simple fixed threshold segmentation and add morphological open and close operations to remove burrs and fill holes.

Appendix A.3 Hybrid classification

To make full use of the orientation field, two branches of classification methods are designed.

Appendix A.3.1 *OF based classification*

Since the orientation field (OF) contains sufficient information for fingerprint classification, we take the orientation field and segmentation as input to generate a 4-dimensional feature vector, which represents the probability of the fingerprint being Arch, Left Loop, Right Loop and Whorl. To better deal with fingerprints of different input sizes, fully convolutional network [19] and global average pooling [29] are used. These techniques enable the network to receive fingerprints of any size, such as 512×512 and 800×768.

The class labels of fingerprints are used to supervise the network. To better supervise the features, softmax and center loss [30] are used to constrain the inter and intra classes. The loss function is defined as,

$$L = L_{Softmax} + \lambda L_{Center} \tag{A7}$$

$$L_{Softmax} = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \tag{A8}$$

$$L_{Center} = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|^2, \tag{A9}$$

where x_i denotes the ith output of the last layer, and y_i is the ground truth class. W and b are the weights and bias term. c_{y_i} is the class center of y_i. Following the advice of [30], the class center is updated in a mini-batch.

Noticing that some latent fingerprints are too small to contain sufficient information to determine the class, we remove these samples during training.

Appendix A.3.2 *SP based classification*

Singular points (SPs) are the most intuitive and important feature for fingerprint classification. Based on the number and location of singular points, we can easily classify fingerprints through some rules [10].

To get robust singular points, a UNet is designed for end-to-end singular points extraction. Inspired by the minutiae representation in FingerNet [28], singular points are represented by 3 feature maps. The feature maps represent the probability of each location to be assigned as core, delta and background. The network architecture is illustrated in Figure A1.

The singular point extraction problem is posed as a pixel-wise classification task. Each point is assigned to be the core, delta or background point. The loss function is defined as,

$$L_{sp} = -\frac{1}{|ROI|} \sum_{ROI} (\beta y_i log(p_i)), \tag{A10}$$

where β is the class weights, y_i is the ground truth class and p_i is the predicted class. $|ROI|$ is the foreground region.

After singular points extraction, fingerprint classification is performed by the SPs based rules [6].

Appendix A.3.3 *Hybrid classification*

A 4-dimensional probability vector is acquired from OF based method and singular points are acquired from the SP based method. As shown in Algorithm A1, a fusion algorithm is designed to integrate the two methods complementarily.

Algorithm A1 Fusion algorithm for hybrid fingerprint classification

Input: P: Probability vector, S: Singular points, R: Rules.
Output: C: Class.

```
1: if max(P) > 0.7 then
2:     C = argmax P
3: else
4:     if R(S) is not reject then
5:         C = R(S)
6:     end if
7: end if
8: Return C
```

Appendix B Experiments

Our proposed FClassNet is evaluated on different quality databases in terms of orientation field estimation and fingerprint classification.

Appendix B.1 Databases

Training set We establish an in-house fingerprint database for training. It contains about 8000 latent fingerprints and their corresponding tenprint fingerprints. Each fingerprint contains manually marked minutiae, singular points and classes. The tenprint fingerprints are 640×640 pixels in size. The latent fingerprints are 512×512 pixels in size. Since fully convolutional network [19] and global average pooling [29] are used in the network, fingerprints with different sizes can be processed well.

Table B1 Orientation estimation performance compared with other algorithms in terms of Root Mean Square Deviation in degrees.

Algorithm	All	Good	Bad	Ugly
Proposed	**11.78**	**9.57**	**12.50**	**13.34**
FingerNet [28]	13.13	10.98	14.19	14.47
ConvNet [34]	13.51	10.76	13.94	16.00
LocalDict [35]	13.76	10.87	14.12	16.40
GlobalDict [33]	18.44	14.40	19.18	21.88
FOMFE [36]	28.12	22.83	29.09	32.63
STFT [37]	32.51	27.27	34.10	36.36

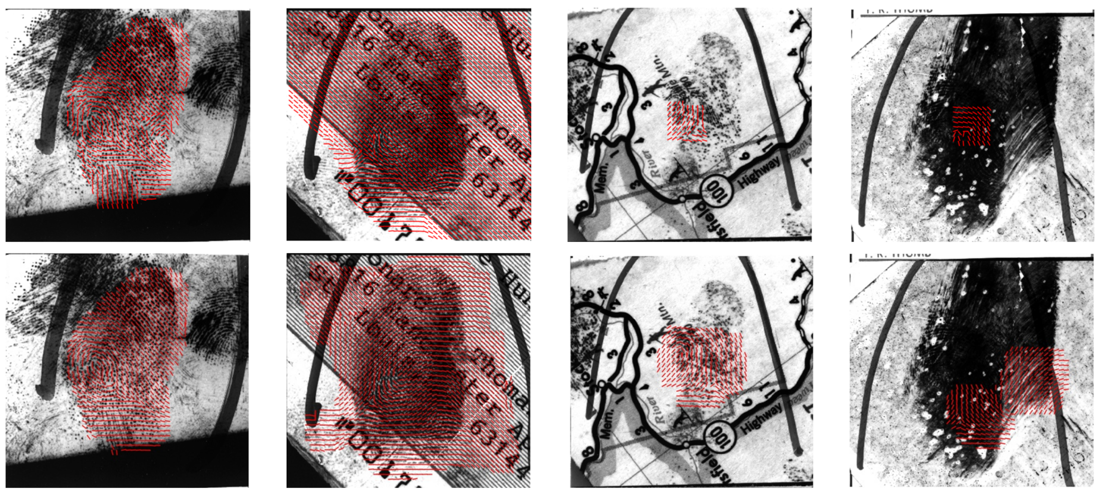


Figure B1 The segmented orientation estimation results on several hard cases of NIST SD27. The first row denotes FingerNet's orientation estimation results and the second row denotes FClassNet's orientation estimation results.

Test set We evaluate the proposed method on the public tenprint database NIST 4 [31], public latent fingerprint database NIST SD27 [32] and our in-house latent fingerprint database CISL1000. NIST 4 contains 4000 tenprint fingerprints which is 512×512 pixels in size. NIST SD27 contains 258 latent fingerprints of different quality. Each latent fingerprint is 800×768 pixels in size and assigned to be good, bad and ugly quality. CISL1000 contains 1000 latent fingerprints which is 512×512 pixels in size. All the fingerprints we used are 500 pixels per inch.

Appendix B.2 Orientation estimation and segmentation

The Root Mean Square Deviation (RMSD) is used as a statistical indicator to evaluate the orientation estimation performance with manually marked orientation field [33]. Table B1 compared the orientation estimation results with other algorithms on NIST SD27 in terms of RMSD. It can be seen that our method outperforms other methods. It takes about 40ms to extract orientation fields and segmentation on a 768×800 fingerprint.

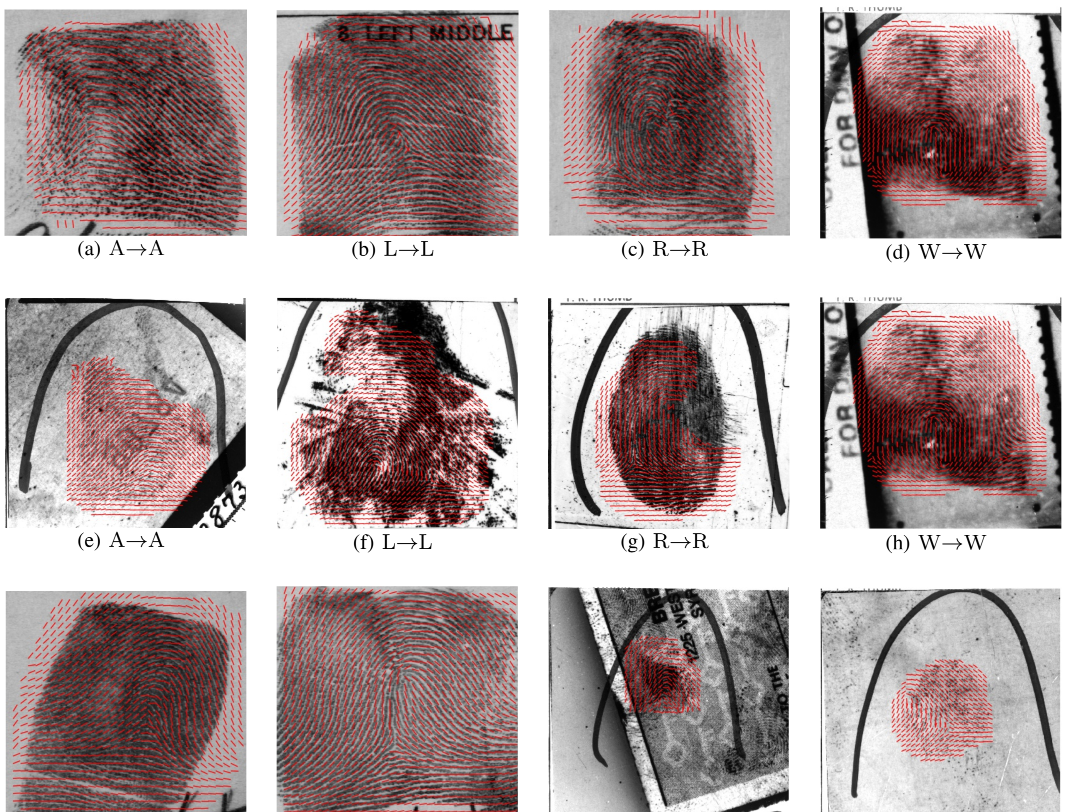
For the evaluation of orientation field estimation, on the one hand, we look at the statistical RMSD which reflects the overall performance of the algorithm. On the other hand, we also look at its performance on some hard cases to evaluate whether it has learned the pattern of the fingerprint orientation. Figure B1 shows the performance of the orientation estimation and segmentation module in some hard cases. It can be seen that the proposed orientation field estimation and segmentation algorithm performs better in some hard cases and really learns the directional texture information of fingerprints.

Appendix B.3 Fingerprint classification

Table B2 compares the fingerprint classification results with other algorithms on NIST 4 database [31], which is usually used to verify the classification results because of its uniform category distribution. The rejection rate is used to assign some low quality fingerprints into the "unknown" class. With the rejection of low quality fingerprints, the accuracy will increase a lot. The "whole DB" denotes that all the fingerprints are used for testing, while the 'half DB' denotes half of the fingerprints are used to fine-tuning the model and the remaining fingerprints are used for testing. "FClassNet" denotes our proposed method. "FClassNet-RE" rejects some ambiguous fingerprints whose confidence is smaller than 0.8, which a practical strategy to dealing with incomplete and ambiguous fingerprints. "FClassNet-OF" denotes the orientation field based classification results. "FClassNet-SP" denotes the singular points based classification results. "Gori+SP" denotes the gradient based orientation field estimation methods and singular points based classification. It can be seen that our proposed

Table B2 Fingerprint classification performance compared with other algorithms.

Algorithm	Test set	Accuracy	Rejection rate
FClassNet	whole DB	**95.1%**	**0%**
FClassNet-RE	whole DB	96.5%	5.1%
FClassNet-OF	whole DB	94.6%	0%
FClassNet-SP	whole DB	93.9%	4.5%
Gori+SP	whole DB	83.9%	6.9%
Shrein et al. [22]	half DB	94.5%	0%
Wang et al. [4]	half DB	91.4%	0%
Liu et al. [38]	whole DB	92.1%	0%
Zhang et al. [6]	whole DB	95.3%	11.8%
Yao et al. [39]	half DB	93.1%	1.8%
Jain et al. [40]	whole DB	91.2%	0%
Candela et al. [1]	half DB	88.6%	0%



(a) A→A (b) L→L (c) R→R (d) W→W
(e) A→A (f) L→L (g) R→R (h) W→W
(i) L→A (j) A→R (k) L→A (l) W→L

Figure B2 The classification results on several samples. The caption "X→Y" denotes the true class is X and the predicted class is Y. The first row illustrates tenprint fingerprints' results and the second row illustrates latent fingerprints' results. The third row illustrates some misclassified cases. Our predicted orientation field and segmentation are drawn on the image.

FClassNet outperforms other methods, and its internal modules also perform better than other algorithms. Table B3 shows the confusion matrix to give a more detailed classification result.

In practical application, the fingerprint classification needs a very high recall rate, that is, the truly matched fingerprint cannot be filtered out. Benefiting from that our output is a category probability vector rather than an exact category, we can combine the sub-class technique to filter fingerprints. The category with the second highest probability is defined as the sub-class. If the predicted class and sub-class contain the real class, the prediction is correct. In this case, we achieve 99.8% top-2 accuracy.

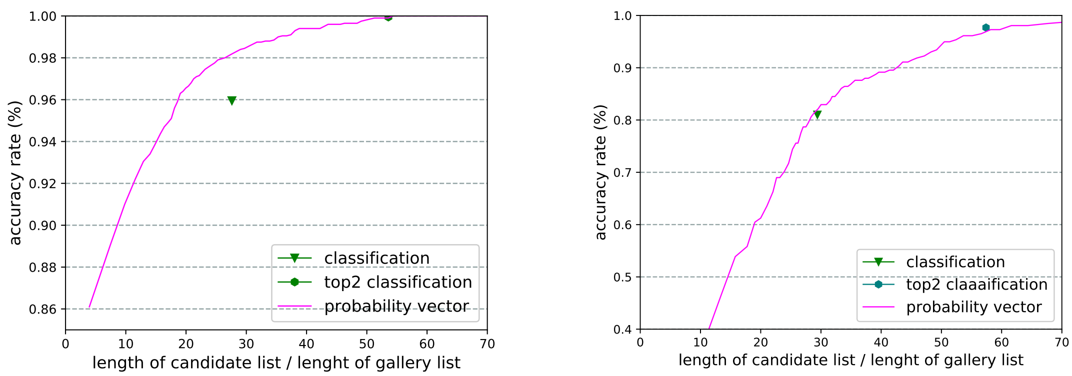
We also evaluate the classification performance on latent fingerprints. Table B4 shows the classification results on two latent databases. NIST SD27 get lower accuracy because it contains many extremely small area fingerprints. Figure B2

Table B3 The confusion matrix of classification results on NIST 4 database.

True class	Predicted class			
	A/T	L	R	W
A/T	1493	39	68	0
L	43	757	0	0
R	36	4	756	4
W	1	7	4	788

Table B4 Fingerprint classification performance on two latent databases.

Database	Total	Accuracy	TOP-2 accuracy
NIST SD27 [32]	258	79.8%	93.7%
CISL1000	1000	84.3%	95.9%



(a) (b)

Figure B3 The candidates filtering curves on (a) NIST 4 database and (b) NIST SD27 database. Retaining shorter candidate lists with higher accuracy is considered a better algorithm.

visualize some classification results. It can be seen that: (1) High quality fingerprints can be classified correctly. (2) Low quality fingerprints with sufficient area can be classified correctly. Our orientation field estimation module can compute accurate orientation field on low quality fingerprints and make it easier to classify latent fingerprints. (3) Misclassification occurs on small area fingerprints and ambiguous category fingerprints. It takes about 50ms for classification on a 768×800 fingerprint.

Appendix B.4 Candidates filtering

Since we acquire the probability vector from the last layer of FClassNet-OF, a novel indexing method based on the probability vector is proposed without rising more costs. The Jensen-Shannon divergence is used to measure the similarity between query's probability vector and the probability vector of fingerprints in the candidate list. The template in the gallery whose similarity between the query is bigger than a certain threshold will be filtered.

Figure B3 compares three methods of candidates filtering on NIST 4 database and NIST SD27 database: (1) The candidate list which has the same class with the query fingerprint. (2) The Jensen-Shannon divergence between the query's probability vector and probability vector of fingerprints in the candidate list is smaller than a certain threshold. (3) The candidate list whose top-2 classes are overlapped with the query fingerprint. From the candidate filtering results, we can see that (1) The probability vector gives flexible choices to balance the accuracy and candidate length. (2) On tenprint fingerprints, the probability vector performs better than fingerprint classes and about equal to top-2 classes. (3) On latent fingerprints, high accuracy is also acquired when filtering about half of the gallery list.

References

1 Candela, G.T., Grother, P.J., Watson, C.I., Wilkinson, R.A., Wilson, C.L.: Pcasys- a pattern-level classification automation system for fingerprints — nist. NIST Interagency/Internal Report (NISTIR) - 5647 (1995)

2 Neuhaus, M., Bunke, H.: A graph matching based approach to fingerprint classification using directional variance. Lecture Notes in Computer Science (2005) 191–200

3 Tan, X., Bhanu, B., Lin, Y.: Fingerprint classification based on learned features. systems man and cybernetics **35** (2005) 287–300

4 Wang, R., Han, C., Guo, T.: A novel fingerprint classification method based on deep learning. In: 2016 23rd International Conference on Pattern Recognition (ICPR). (2016) 931–936

5 Hong, J.H., Min, J.K., Cho, U.K., Cho, S.B.: Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive bayes classifiers. Pattern Recognition **41** (2008) 662–671

6 Zhang, Q., Yan, H.: Fingerprint classification based on extraction and analysis of singularities and pseudo ridges. Pattern Recognition **37** (2004) 2233–2243

7 Li, J., Yau, W.Y., Wang, H.: Combining singular points and orientation image information for fingerprint classification. Pattern Recognition **41** (2008) 353–366

8 Jain, A.K., Prabhakar, S., Hong, L.: A multichannel approach to fingerprint classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 348–359

9 Bowen, J.: The home office automatic fingerprint pattern classification project. In: IEE Colloquium on Neural Networks for Image Processing Applications, IET (1992) 1–1

10 Karu, K., Jain, A.K.: Fingerprint classification. Pattern recognition **29** (1996) 389–404

11 Chang, J.H., Fan, K.C.: A new model for fingerprint classification by ridge distribution sequences. Pattern Recognition **35** (2002) 1209–1223

12 Jung, H.W., Lee, J.H.: Fingerprint classification using the stochastic approach of ridge direction information. In: 2009 IEEE International Conference on Fuzzy Systems. (2009) 169–174

13 Le, T.H., Van, H.T.: Fingerprint reference point detection for image retrieval based on symmetry and variation. Pattern Recognition **45** (2012) 3360–3372

14 Nyongesa, H.O., Al-Khayatt, S., Mohamed, S.M., Mahmoud, M.: Fast robust fingerprint feature extraction and classification. Journal of Intelligent and Robotic Systems **40** (2004) 103–112

15 Cao, K., Pang, L., Liang, J., Tian, J.: Fingerprint classification by a hierarchical classifier. Pattern Recognition **46** (2013) 3186–3197

16 Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105

17 Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

18 He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

19 Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440

20 Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99

21 He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE (2017) 2980–2988

22 Shrein, J.M.: Fingerprint classification using convolutional neural networks and ridge orientation images. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). (2017) 1–8

23 Michelsanti, D., Guichi, Y., Ene, A.D., Stef, R., Nasrollahi, K., Moeslund, T.B.: Fast fingerprint classification with deep neural network. In: International Conference on Computer Vision Theory and ApplicationsInternational Conference on Computer Vision Theory and Applications, SCITEPRESS Digital Library (2018) 202–209

24 Peralta, D., Triguero, I., Garca, S., Saeys, Y., Bentez, J.M., Herrera, F.: On the use of convolutional neural networks for robust classification of multiple fingerprint captures. International Journal of Intelligent Systems **33** (2018) 213–230

25 Khan, A.I.: Lafin: A convolutional neural network-based technique for singular point extraction and classification of latent fingerprints. In: 2018 International Conference on Soft-computing and Network Security (ICSNS). (2018)

26 Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241

27 Kass, M., Witkin, A.: Analyzing oriented patterns. Computer vision, graphics, and image processing **37** (1987) 362–385

28 Tang, Y., Gao, F., Feng, J., Liu, Y.: Fingernet: An unified deep network for fingerprint minutiae extraction. In: Biometrics (IJCB), 2017 IEEE International Joint Conference on, IEEE (2017) 108–116

29 Lin, M., Chen, Q., Yan, S.: Network in network. international conference on learning representations (2014)

30 Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision. (2016) 499–515

31 Watson, C., Wilson, C.: Nist special database 4. Fingerprint Database, National Institute of Standards and Technology **17** (1992) 77

32 Garris, M.D., McCabe, R.M.: Nist special database 27: Fingerprint minutiae from latent and matching tenprint images. National Institute of Standards and Technology, Technical Report NISTIR **6534** (2000)

33 Feng, J., Zhou, J., Jain, A.K.: Orientation field estimation for latent fingerprint enhancement. IEEE transactions on pattern analysis and machine intelligence **35** (2013) 925–940

34 Cao, K., Jain, A.K.: Latent orientation field estimation via convolutional neural network. In: Biometrics (ICB), 2015 International Conference on, IEEE (2015) 349–356

35 Yang, X., Feng, J., Zhou, J.: Localized dictionaries based orientation field estimation for latent fingerprints. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014) 955–969

36 Wang, Y., Hu, J., Phillips, D.: A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application

to singular-point detection and fingerprint indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007) 573–585

37 Chikkerur, S., Cartwright, A.N., Govindaraju, V.: Fingerprint enhancement using stft analysis. Pattern recognition **40** (2007) 198–211

38 Liu, M.: Fingerprint classification based on singularities. In: 2009 Chinese Conference on Pattern Recognition. (2009) 1–5

39 Yao, Y., Frasconi, P., Pontil, M.: Fingerprint classification with combinations of support vector machines. In: International Conference on Audio-and Video-Based Biometric Person Authentication, Springer (2001) 253–258

40 Jain, A.K., Minut, S.: Hierarchical kernel fitting for fingerprint classification and alignment. In: Object recognition supported by user interaction for service robots. Volume 2. (2002) 469–473