

Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge

Dinh-Luan Nguyen, Kai Cao and Anil K. Jain
Michigan State University
East Lansing, Michigan, USA

nguye590@msu.edu, {kaicao, jain}@cse.msu.edu

Abstract

We propose a fully automatic minutiae extractor, called *MinutiaeNet*, based on deep neural networks with compact feature representation for fast comparison of minutiae sets. Specifically, first a network, called *CoarseNet*, estimates the minutiae score map and minutiae orientation based on convolutional neural network and fingerprint domain knowledge (enhanced image, orientation field, and segmentation map). Subsequently, another network, called *FineNet*, refines the candidate minutiae locations based on score map. We demonstrate the effectiveness of using the fingerprint domain knowledge together with the deep networks. Experimental results on both latent (NIST SD27) and plain (FVC 2004) public domain fingerprint datasets provide comprehensive empirical support for the merits of our method. Further, our method finds minutiae sets that are better in terms of precision and recall in comparison with state-of-the-art on these two datasets. Given the lack of annotated fingerprint datasets with minutiae ground truth, the proposed approach to robust minutiae detection will be useful to train network-based fingerprint matching algorithms as well as for evaluating fingerprint individuality at scale. *MinutiaeNet* is implemented in *Tensorflow*: <https://github.com/luannd/MinutiaeNet>

1. Introduction

Automatic fingerprint recognition is one of the most widely studied topic in biometrics over the past 50 years [12]. One of the main challenges in fingerprint recognition is to increase the recognition accuracy, especially for latent fingerprints. Fingerprint comparison is primarily based on minutiae set comparison [28, 11]. A number of hand-crafted approaches [10, 28] have been used to augment the minutiae with their attributes to improve the recognition accuracy. However, robust automatic fingerprint minutiae extraction, particularly for noisy fingerprint

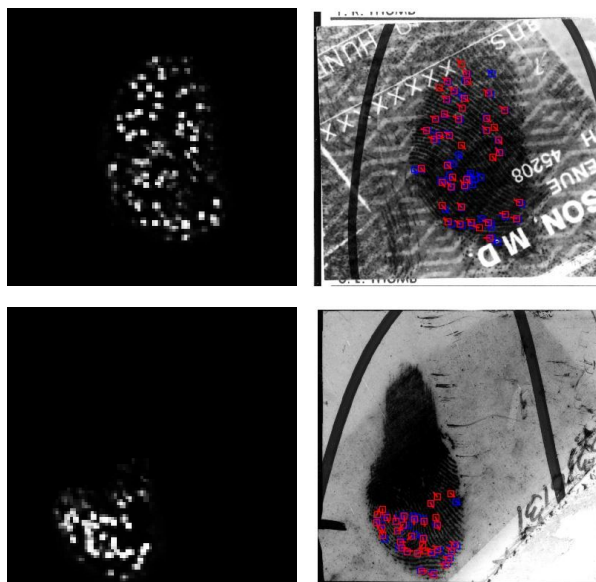


Figure 1. Minutiae detection by the proposed approach on two latent fingerprint images (#7 and #39) from the NIST SD27 dataset [6]. Left column: minutiae score maps obtained from the latent images shown in the right column. Right column: minutiae detected by the proposed framework (red) and ground truth minutiae (blue) overlaid on the latent image.

images, continues to be a bottleneck in fingerprint recognition systems.

With rapid developments and success of deep learning techniques in a variety of applications in computer vision and pattern recognition [8, 20], we are beginning to see network-based approaches being proposed for fingerprint recognition. Still, the prevailing methods of minutiae extraction primarily utilize fingerprint domain knowledge and handcrafted features. Typically, minutiae extraction and matching involves pre-processing stages such as ridge extraction and ridge thinning, followed by minutiae extraction [10, 4] and finally heuristics to define minutiae at-

Table 1. Published network-based approaches for automatic minutiae extraction.

Study	Method	Training Data	Testing Data	Comments	Performance Evaluation
Sankaran <i>et al.</i> [18]	Sparse autoencoders for classification	132, 560 plain fingerprint images; 129 images from NIST SD27	129 remaining latents from NIST SD27	Sliding window; manual segmentation of latent fingerprints	Patch-based and minutia-based, metric and matching performance ^(*)
Jiang <i>et al.</i> [13]	A combination of JudgeNet and LocateNet	200 live scan fingerprints	100 images from private database	Sliding window; hand-crafted dividing regions; no minutiae orientation information	Precision, recall, and F1 score
Tang <i>et al.</i> [21]	Fully convolutional neural network	4, 205 latent images from private database and 129 latents from NIST SD27	129 latents from NIST SD27	Hard thresholds to cut off candidate regions; plain network	Precision, recall, F1 score, and matching performance
Darlow <i>et al.</i> [3]	Convolutional network classifier	6, 336 images from FVC 2000, 2002, and 2004	1, 584 images from FVC 2000, 2002, and 2004	Sliding window; hard threshold for candidate regions (minutiae); separately estimated minutiae orientation	Equal error rate and matching performance
Tang <i>et al.</i> [22]	Unified network with domain knowledge	8, 000 images from private forensic latent database	Set A of FVC 2004 and NIST SD27	Plain network; depends largely on the quality of the enhancement and segmentation stages	Precision, recall, and matching performance
Proposed approach	Domain knowledge with Residual learning based CoarseNet and inception-resnet based FineNet	FVC 2002 with data augmentation (8, 000 images in total)	FVC 2004 (3, 200 images) and NIST SD27 (all 258 latents)	Residual network; automatic minutiae extractor utilizing domain knowledge; robust patch based minutiae classifier	Precision, recall, and F1 score under different location and orientation thresholds

(*) different matchers were used in different studies and none of them were state of the art, i.e. top performing latent or slap matchers identified in the NIST evaluations. For this reason, we *do not report* matching performance because otherwise it *would not be a fair comparison* with previous studies.

tributes. While such an approach works well for good quality fingerprint images, it provides inaccurate minutiae location and orientation for poor quality rolled/plain prints and, particularly for latent fingerprints. To overcome the noise in fingerprint images, Yoon *et al.* [27] used Gabor filtering to calculate the reliability of extracted minutiae. Although this approach can work better than [10], it also resulted in poor results with highly noisy images. Because these prevailing approaches are based on handcrafted methods or heuristics, they are only able to extract basic (or low level) features¹ of images. We believe learning based approaches using deep networks will have better ability to extract high level features² from low quality fingerprint images.

In this paper, we present a novel framework that exploits useful domain knowledge coded in the deep neural networks to overcome limitations of existing approaches to minutiae extraction. Figure 1 visualizes results of the proposed framework on two latent fingerprints from the NIST SD27 dataset.

Specifically, our proposed approach comprises of two networks, called CoarseNet and FineNet:

- **CoarseNet** is a residual learning [8] based convolutional neural network that takes a fingerprint image as initial input, and the corresponding enhanced image, segmentation map, and orientation field (computed by the early stages of CoarseNet) as secondary input to generate the minutiae score map. The minutiae orientation is also estimated by comparing with the fingerprint orientation.

- **FineNet** is a robust inception-resnet [20] based minutiae classifier. It processes each candidate patch, a square region whose center is the candidate minutiae point, to refine the minutiae score map and approximate minutiae orientation by regression. Final minutiae are the classification results.

Deep learning approach has been used by other researchers for minutiae extraction (see Table 1). But, our approach differs from published methods in the way we encode fingerprint domain knowledge in deep learn-

ing. Sankaran *et al.* [18] classified the minutiae and non-minutiae patches by using sparse autoencoders. Jiang *et al.* [13] introduced a combination of two networks: JudgeNet for classifying minutiae patches, and LocateNet for locating precise minutiae location. While Jiang *et al.* use neural networks, their approach is very time-consuming due to use of sliding window to extract minutiae candidates. Another limitation of this approach is that it does not provide minutiae orientation information.

Tang *et al.* [21] utilized the idea of object detection to detect candidate minutiae patches, but it suffers from two major weaknesses: (i) hard threshold to delete the candidate patches, and (ii) the same network is used for both candidate generation and classification. By using sliding windows, Darlow *et al.* [3] fed each pixel of the input fingerprint to a convolutional neural network, called MENet, to classify whether it corresponds to a minutia or not. It also suffers from time-consuming sliding windows as in [13], and separate modules for minutiae location and orientation estimates. Tang *et al.* [22] proposed FingerNet that maps traditional minutiae extraction pipeline including orientation estimation, segmentation, enhancement, and extraction to a network with fixed weights. Although this approach is promising because it combines domain knowledge and deep network, it still uses plain³ network architecture and hard threshold in non-maximum suppression⁴. Finally, the accuracy of FingerNet depends largely on the quality of the enhanced and segmentation stage while ignoring texture information in the ridge pattern.

In summary, the published approaches suffer from using sliding windows to process each pixel in input images, setting hard threshold in post-processing step, and using plain convolutional neural network to classify candidate regions. Furthermore, the evaluation process in these studies is not consistent in terms of defining “correct” minutiae.

The contributions of our approach are as follows:

- A network-based *automatic minutiae extractor* utilizing domain knowledge is proposed to provide reli-

¹Features such as edges, corners, etc.

²Abstract/semantic features retrieved from deep layers.

³A series of stacked layers.

⁴A post-processing algorithm that merges all detections belonging to the same object.

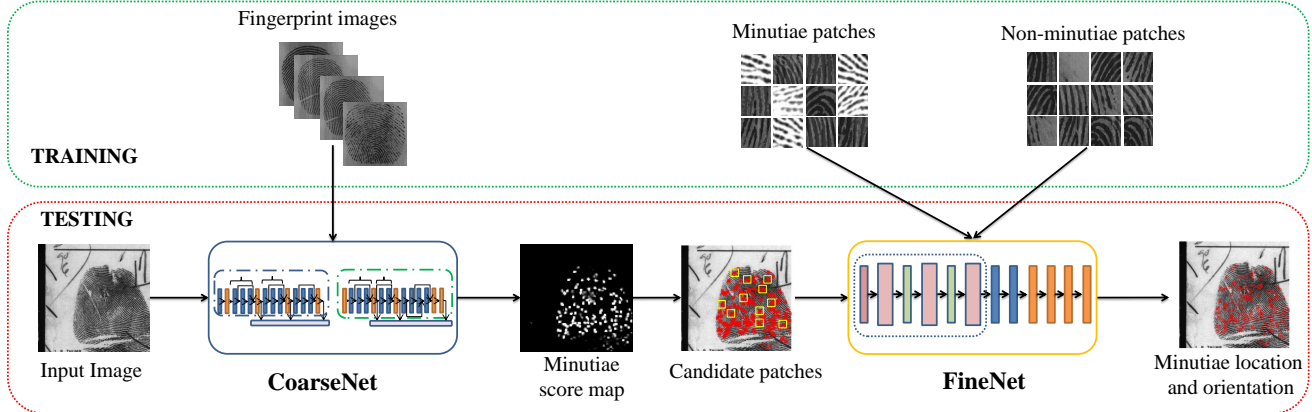


Figure 2. Proposed automatic minutiae extraction architecture. While CoarseNet takes full fingerprint image as input, FineNet processes minutiae proposed patches output by CoarseNet.

able minutiae location and orientation without a hard threshold or fine tuning.

- A *robust patch based minutiae classifier* that significantly boosts the precision and recall of candidate patches. This can be used as a robust minutiae extractor with compact embedding of minutiae features.
- A *non-maximum suppression* is proposed to get precise locations for candidate patches. Experimental evaluations on FVC 2004 [15] and NIST SD27 [6] show that the proposed approach is superior to published approaches in terms of precision, recall, and F1 score values.

2. Proposed framework

Our minutiae extraction framework has two modules: (i) residual learning based convolutional neural network, called CoarseNet that generates candidate patches containing minutiae from input fingerprint image; (ii) inception-resnet based network architecture, called FineNet which is a strong minutiae classifier that classifies the candidate patches output by CoarseNet. These two networks also provide minutiae location and orientation information as outputs. Figure 2 describes the complete network architecture for automatic minutiae location and orientation for an input fingerprint image. Section 2.1 presents the architecture of CoarseNet. In Section 2.2, we introduce FineNet with details on training to make it a strong classifier.

2.1. CoarseNet for minutiae extraction

We adopt the idea of combining domain knowledge and deep representation of neural networks in [22] to boost the minutiae detection accuracy. In essence, we utilize the automatically extracted segmentation map, enhanced image, and orientation map as complementary information to the input fingerprint image. The goal of CoarseNet is not to

produce the segmentation map or enhanced image or orientation map. They are just the byproducts of the network. However, these byproducts as fingerprint domain knowledge must be reliable to get robust minutiae score map. Because Tang *et al.* [22] proposed an end-to-end unified network that maps handcrafted features to network based architecture, we use this as a baseline for our CoarseNet.

2.1.1 Segmentation and orientation feature sharing

Adding more layers in the deep network with the hope of increasing accuracy might lead to the exploding or vanishing gradients problem. From the success of residual learning [8], we use residual instead of just plain stacked convolutional layers in our network to make it more powerful. Figure 3 shows the detailed architecture of the network.

Unlike existing works using plain convolutional neural network [21, 22] or sliding window [18, 3] to process each patch with fixed size and stride, we use a deeper residual learning based network with more pooling layers to scale down the region patch. Specifically, we get the output after the 2nd, 3rd, and 4th pooling layer to feed to an ASPP network [2] with corresponding rates for multiscale segmentation. This ensures the output has the same size as input without a loss of information when upsampling the score map.

By using four pooling layers, each pixel in the j^{th} feature map, called level j , corresponds to a region $2^j \times 2^j$ in the original input. Result layers at level 4 and 3 will be tested as coarse estimates while the level 2 serves as fine estimation.

Segmentation map. Image segmentation and fingerprint orientation estimation share the same convolutional layers. Thus, by applying multi-level approach mentioned above, we get probability maps of each level-corresponding region in input image. For instance, to get finer-detailed segmentation for each region level j_l , we continue to process probability map of region level $j_l/2$.

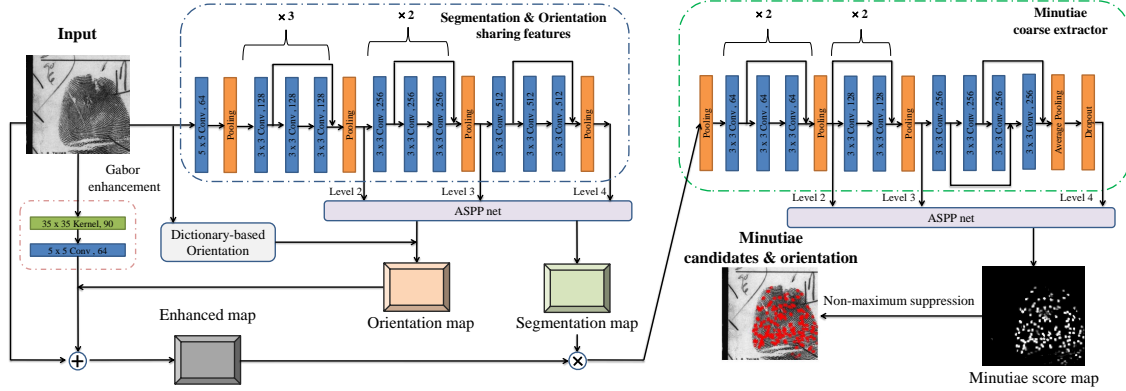


Figure 3. CoarseNet architecture.

Orientation map. To get complete minutiae information from context, we adopt the fusion idea of Cao *et al.* [1]. We fuse the results of Dictionary-based method [26] with our orientation results from CoarseNet. Because [26] uses a hand-crafted approach, we set the fusion weight ratio of its output with our network-based approach as 1:3.

2.1.2 Candidate generation

The input fingerprint image might contain large amounts of noise. So, without using domain knowledge we may not be able to identify prominent fingerprint features. The domain knowledge comprises of four things: raw input image, enhanced image, orientation map, and segmentation map. In the Gabor image enhancement module, we take the average of filtered image and the orientation map for ridge flow estimation. To emphasize texture information in fingerprints, we stack the original input image with the output of enhancement module to obtain the final enhancement map. To remove spurious noises, we apply segmentation map on the enhancement map and use it as input to coarse minutiae extractor module.

To obtain the precise location of minutiae, each level of residual net is fused to get the final minutiae score map with size $h/16 \times w/16$, where h and w are the height and width of the input image. Figure 4 shows the details of processing

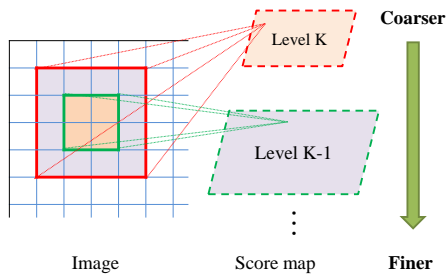


Figure 4. Candidate patch processing map. Lower the level of score map, more detail it provides.

score map. To reduce the processing time, we use score map at level 4 as a coarse location. To get precise location, lower level score maps are used.

2.1.3 Non-maximum suppression

Using non-maximum suppression to reduce the number of candidates is common in object detection [7, 17]. Some of the candidate regions are deleted to get a reliable minutiae score map by setting a hard threshold [3] or using heuristics [21, 22]. However, a hard threshold can also suppress valid minutiae locations. A commonly used heuristics is to sort the candidate scores in ascending order. The L^2 distance between pairwise candidates is calculated with hard thresholds for distance and orientation. By iteratively comparing each candidate with the rest in the candidate list, only the candidate with higher score and score above the thresholds is kept. However, this approach fails when two minutiae are near each other and the inter-minutiae distance is below the hard thresholds.

Since each score in the minutiae map corresponds to a specific region in the input image, we propose to use the intersection over union strategy. Specifically, after sorting the scores of the candidate list, we keep high score candidates while ignoring the lower scoring candidates with at least 50% overlap with the candidates already selected.

2.1.4 Training data for CoarseNet

Given the lack of datasets with ground truth, we use the approach in Tang *et al.* [22] to generate weak labels for training the segmentation and orientation module. The coarse minutiae extractor module uses minutiae location and minutiae orientation ground truth provided in the two datasets. We also use data augmentation techniques as mentioned in Section 3.

2.2. FineNet

Extracting minutiae based on candidate patches is not adequate. Although CoarseNet is reliable, it still fails to de-

detect true minutiae or detects spurious minutiae. This can lead to poor performance in fingerprint matching. This motivates our use of FineNet - a minutiae classifier from generated candidate patches. FineNet takes candidates from the output of CoarseNet as input to decide whether the region 10×10 in the center of the corresponding patch has a valid minutia or not.

2.2.1 FineNet architecture

Figure 5 describes the architecture of FineNet. As mentioned in Section 2.1.1, we use the Inception-Resnet v1 architecture as a core network in FineNet.

For FineNet training, we extract an equal number of $t_1 \times t_1$ sized minutiae and non-minutiae patches with $t_1 = 45$. FineNet determines the whether the 10×10 pixel region in the center of each patch contains a valid minutia or not. The candidate patches are resized into $t_2 \times t_2$ pixels that feed to FineNet. Based on the observation that the original input image size (without rescaling) is not large in comparison with images for object classification, too much up scaling the image can cause blurring the tiny details, and too small an input image size is not sufficient for network with complex architecture, we choose $t_2 = 160$ pixels.

Training data for FineNet is extracted from the input gray scale images where minutiae data are based on the ground truth minutiae location and non-minutiae ones are from ran-

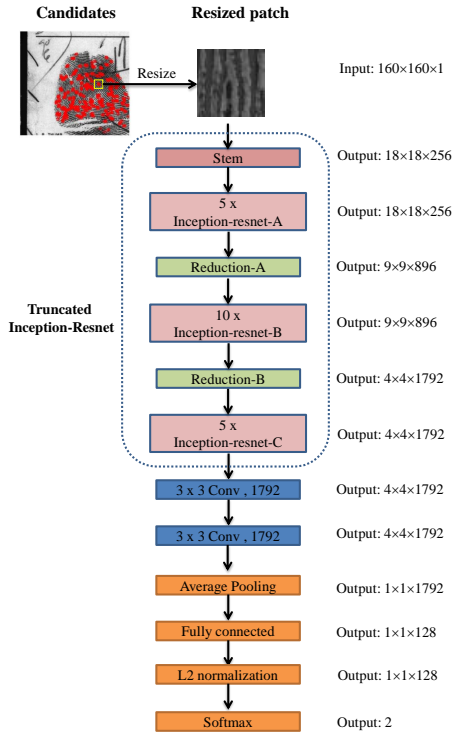


Figure 5. FineNet architecture. For details of the Inception-Resnet v1 architecture block, we refer the readers to [20].

dom sampling with the center regions do not contain partial or fully minutiae location. To make the network more robust, we use some small techniques like Batch Normalization [9], rotation, flipping, scaled augmentation [19], and bounding blurring [13] as pre-processing stage.

2.2.2 Losses and implementation details

Intra-Inter class losses. Because input captured fingerprint image is not always in the ideal condition, it might be easily affected by distortion, finger movement, or quality (wet/dry) of finger. Thus, the variation of minutiae shapes and surrounding ridges can affect the accuracy of the classifier. To handle this situation and make the FineNet more robust with intra-class variations, we use Center Loss [25] as a complementary of softmax loss and minutiae orientation loss. While softmax loss tends to push away features of different classes, center loss tends to pull features in the same class closer. Let \mathcal{L} , \mathcal{L}_c , \mathcal{L}_s , \mathcal{L}_o be the total loss, center loss, softmax loss, and orientation loss, the total loss for training is calculated as follows:

$$\mathcal{L} = \alpha \mathcal{L}_c + (1 - \alpha) \mathcal{L}_s + \beta \mathcal{L}_o \quad (1)$$

where we set $\alpha = 0.5$ to balance between intra-class (center) and inter-class (softmax) loss and $\beta = 2$ to emphasize the importance of precision of minutiae orientation.

Parameter settings. As mentioned in Section 2.2.1, fingerprint patches are input to FineNet where the patch size is 160×160 . To ensure our network can handle distortion in input image, we apply scaled augmentation [19], random cropping, and brightness adjustment. Horizontal and vertical flip with pixel mean subtraction are also adopted. We randomly initialize all variables by using a Gaussian distribution $N(0, 0.01)$. Batch size is set to 100. We use schedule learning rate after particular iterations. Specifically, we set it as 0.01 in the beginning and reduce 10 times after $50K$ iterations. To prevent vanishing gradient problem, we set the maximum epoch to $200K$. We use 0.0004 as the value for momentum and weight decay is 0.9.

3. Experimental results

We evaluate our method on two different datasets with different characteristics under different settings of parameters \mathcal{D} and \mathcal{O} (see Eq. (2)). We also visualize examples of score maps with correct and incorrect minutiae extractions in Figure 7. All experiments were implemented in TensorFlow and ran on Nvidia GTX GeForce.

3.1. Datasets

We use FVC 2002 dataset [16] with data augmentation consisting of 3, 200 plain fingerprint images for training. To compensate for the lack of a large scale dataset for training, we distort the input images in x and y coordinates in the spirit of hard-training with non-ideal input fingerprint images. Furthermore, we also apply additive random noise to

Table 2. Comparison of different methods for minutiae extraction on FVC 2004 and NIST SD27 datasets. Note that [18, 21] reported their results only on subsets of FVC 2004 and NIST SD27 as mentioned in Table 1. “-” means the authors neither provided these results in their paper nor made their code available. \mathcal{D} and \mathcal{O} are parameters defined in Eq. (2).

Dataset	Methods	Setting 1 ($\mathcal{D} = 8, \mathcal{O} = 10$)			Setting 2 ($\mathcal{D} = 12, \mathcal{O} = 20$)			Setting 3 ($\mathcal{D} = 16, \mathcal{O} = 30$)		
		Precision	Recall	F1 score	Precision	Recall	F1 score	Precision	Recall	F1 score
NIST SD27	MINDTCT [24]	8.3%	14.7%	0.106	10.0%	16.4%	0.124	11.2%	18.9%	0.141
	VeriFinger [23]	3.6%	40.1%	0.066	5.3%	47.9%	0.095	7.6%	58.3%	0.134
	Gao <i>et al.</i> [5]	-	-	-	-	-	-	23.5%	8.7%	0.127
	Sankaran <i>et al.</i> [18]	-	-	-	-	-	-	26.4%	63.1%	0.372
	Tang <i>et al.</i> [21]	-	-	-	-	-	-	53.0%	53.4%	0.532
	FingerNet [22]	53.2%	49.5%	0.513	58.0%	58.1%	0.58	63.0%	63.2%	0.631
	Proposed method	69.2%	67.7%	0.684	70.5%	72.3%	0.714	71.2%	75.7%	0.734
FVC 2004	MINDTCT [24]	30.8%	64.3%	0.416	37.7%	72.1%	0.495	42.1%	79.8%	0.551
	VeriFinger [23]	39.8%	69.2%	0.505	45.6%	77.5%	0.574	51.8%	81.9%	0.635
	Gao <i>et al.</i> [5]	-	-	-	-	-	-	48.8%	82.7%	0.614
	FingerNet [22]	68.7%	62.1%	0.643	72.9%	70.4%	0.716	76.0%	80.0%	0.779
	Proposed method	79.0%	80.1%	0.795	83.6%	83.9%	0.837	85.9%	84.8%	0.853

the input images. Thus, for training CoarseNet, we have an augmented dataset of 8,000 images. To obtain data for training FineNet, we extract 45×45 pixel patches from these 8K training images for CoarseNet whose center is a ground truth minutia point. For non-minutiae patches, we randomly extract patches with the criteria that the 10×10 center of each patch does not contain any minutia. Thus, we collect around 100K minutia and non-minutia patches for training FineNet.

As mentioned in Section 2.1.4, we use FingerNet [22] to generate labels for domain knowledge groundtruth. Besides, we manually correct segmentation groundtruth results from FingerNet to ensure better learning for CoarseNet.

3.2. Evaluation

To demonstrate the robustness of our framework, we compare our results with published approaches on FVC 2004 [15]⁵ and NIST SD27 [6] datasets under different criteria of distance and orientation thresholds. Let the tuples (l_p, o_p) and (l_{gt}, o_{gt}) be the location coordinates and orientation values of predicted and ground truth minutia. The predicted minutia is called true if it satisfies the following constrains:

$$\begin{cases} \|l_p - l_{gt}\|_2 \leq \mathcal{D} \\ \|o_p - o_{gt}\|_1 \leq \mathcal{O} \end{cases} \quad (2)$$

where \mathcal{D} and \mathcal{O} are the thresholds in pixels and degrees, respectively. Specifically, we set the range of distances between detected and ground truth minutiae from 8 to 16 pixels (in location) and 10 to 30 degree (in orientation) with default threshold value (0.5). We choose these settings to demonstrate the robust and precise results from the proposed approach while published works degrade rather quickly.

Table 2 shows the precision and recall comparisons of different approaches to minutiae extraction.

⁵We obtained the groundtruth from [14]

Table 3. The importance of non-maximum suppression in our framework. NMS and NMS^* denote the non-maximum approaches of FingerNet and the proposed approach, respectively.

Configuration	Precision	Recall	F1 score
FingerNet + NMS	63.0%	63.2%	0.631
FingerNet + NMS^*	65.2%	65.4%	0.653
Proposed method + NMS	69.4%	73.5%	0.714
Proposed method + NMS^*	71.2%	75.7%	0.734

MINDTCT [24] is the open source NIST Biometric Image Software. VeriFinger [23] is a commercial SDK for minutiae extraction and matching. Since Gao *et al.* [5] did not release their code in public domain, we report their results on NIST SD27 and FVC 2004 database from [22]. Darlow *et al.* [3] use only a subset of the FVC dataset for training and the rest for testing, we do not include in our evaluation.

Table 2 shows that the proposed method outperforms state-of-the-art techniques under all settings of parameters (thresholds) \mathcal{D} and \mathcal{O} for both FVC 2004 and NIST SD27. Our results also reveal that by using only rolled/plain fingerprint images for training, our framework can work pretty well for detecting minutiae in latents.

Table 3 shows a comparison between using and not using our proposed non-maximum suppression method on the NIST SD27 dataset with setting 3 in Table 2. Because non-maximum suppression is a post processing step, it helps improve precision, recall and F1 values.

To make a complete comparison (at all the operating points) with published methods, we present the precision-recall curves in Figure 6. The proposed approach surpasses all published works on both FVC 2004 and NIST SD27 datasets.

Figure 7 shows the minutiae extraction results on both FVC 2004 and NIST SD27 datasets with different quality images. Our framework works well in difficult situations such as noisy background or dry fingerprints. However, there are some cases where the proposed framework

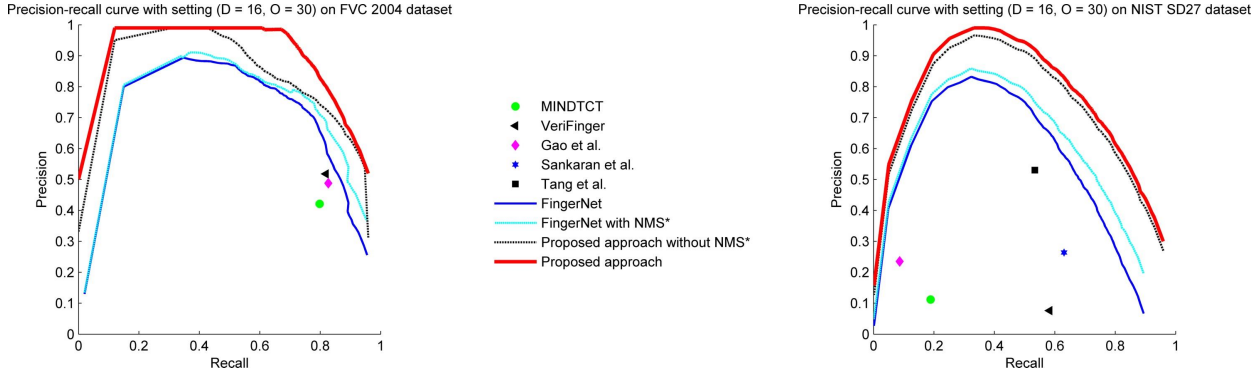


Figure 6. Precision-Recall curves on FVC 2004 (left) and NIST SD27 (right) datasets with published approaches in Setting 3.

either misses the true minutiae or extracts spurious minutiae. For the FVC 2004 dataset and rolled fingerprints from NIST SD27 dataset, we obtain results that are close to the ground truth minutiae. However, some minutiae points are wrongly detected (image a) because of the discontinuity of ridges or missed detections (image c) because the location of minutiae is near the fingerprint edge. For the latent fingerprints from NIST SD27 dataset, besides the correctly extracted minutiae, the proposed method is sensitive to severe background noise (image e) and poor latent fingerprint quality (image g). The run time per image is around 1.5 seconds for NIST SD27 and 1.2 seconds for FVC 2004 on Nvidia GTX GeForce.

4. Conclusions

We have presented two network architectures for automatic and robust fingerprint minutiae extraction that fuse fingerprint domain knowledge and deep network representation:

- **CoarseNet**: an automatic robust minutiae extractor that provides candidate minutiae location and orientation without a hard threshold or fine tuning.

- **FineNet**: a strong patch based classifier that accelerates the reliability of candidates from CoarseNet to get final results.

A non-maximum suppression is proposed as a post processing step to boost the performance of the whole framework. We also reveal the impact of residual learning on minutiae extraction in the latent fingerprint dataset despite using only plain fingerprint images for training. Our experimental results show that the proposed framework is robust and achieves superior performance in terms of precision, recall and F1 values over published state-of-the-art on both benchmark datasets, namely FVC 2004 and NIST SD27.

The proposed framework can be further improved by (i) using larger training set for network training that includes latent images, (ii) constructing context descriptor to exploit the region surrounding minutiae, (iii) improving processing time, and (iv) unifying minutiae extractor into an end-to-end

fingerprint matching framework.

References

- [1] K. Cao and A. K. Jain. Automated latent fingerprint recognition. *arXiv preprint arXiv:1704.01925*, 2017. 4
- [2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. PAMI*, 2017. 3
- [3] L. Darlow and B. Rosman. Fingerprint minutiae extraction using deep learning. In *Proc. IEEE IJCB*, 2017. 2, 3, 4, 6
- [4] J. Feng. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 41(1):342–352, 2008. 1
- [5] X. Gao, X. Chen, J. Cao, Z. Deng, C. Liu, and J. Feng. A novel method of fingerprint minutiae extraction based on Gabor phase. In *Proc. 17th IEEE ICIP*, pages 3077–3080, 2010. 6
- [6] M. D. Garris and R. M. McCabe. NIST special database 27: Fingerprint minutiae from latent and matching tenprint images. *NIST Technical Report NISTIR*, 6534, 2000. 1, 3, 6
- [7] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *Proc. IEEE CVPR*, pages 437–446, 2015. 4
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016. 1, 2, 3
- [9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, pages 448–456, 2015. 5
- [10] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Trans. PAMI*, 19(4):302–314, 1997. 1, 2
- [11] A. K. Jain, Y. Chen, and M. Demirkus. Pores and ridges: High-resolution fingerprint matching using level 3 features. *IEEE Trans. PAMI*, 29(1):15–27, 2007. 1
- [12] A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 79:80–105, 2016. 1
- [13] L. Jiang, T. Zhao, C. Bai, A. Yong, and M. Wu. A direct fingerprint minutiae extraction approach based on convolutional neural networks. In *Proc. IEEE IJCNN*, pages 571–578, 2016. 2, 5
- [14] M. Kayaoglu, B. Topcu, and U. Uludag. Standard fingerprint databases: Manual minutiae labeling and matcher performance analyses. *arXiv preprint arXiv:1305.1443*, 2013. 6
- [15] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. FVC2004: Third fingerprint verification competition. In *Biometric Authentication*, pages 31–35. Springer, 2004. 3, 6

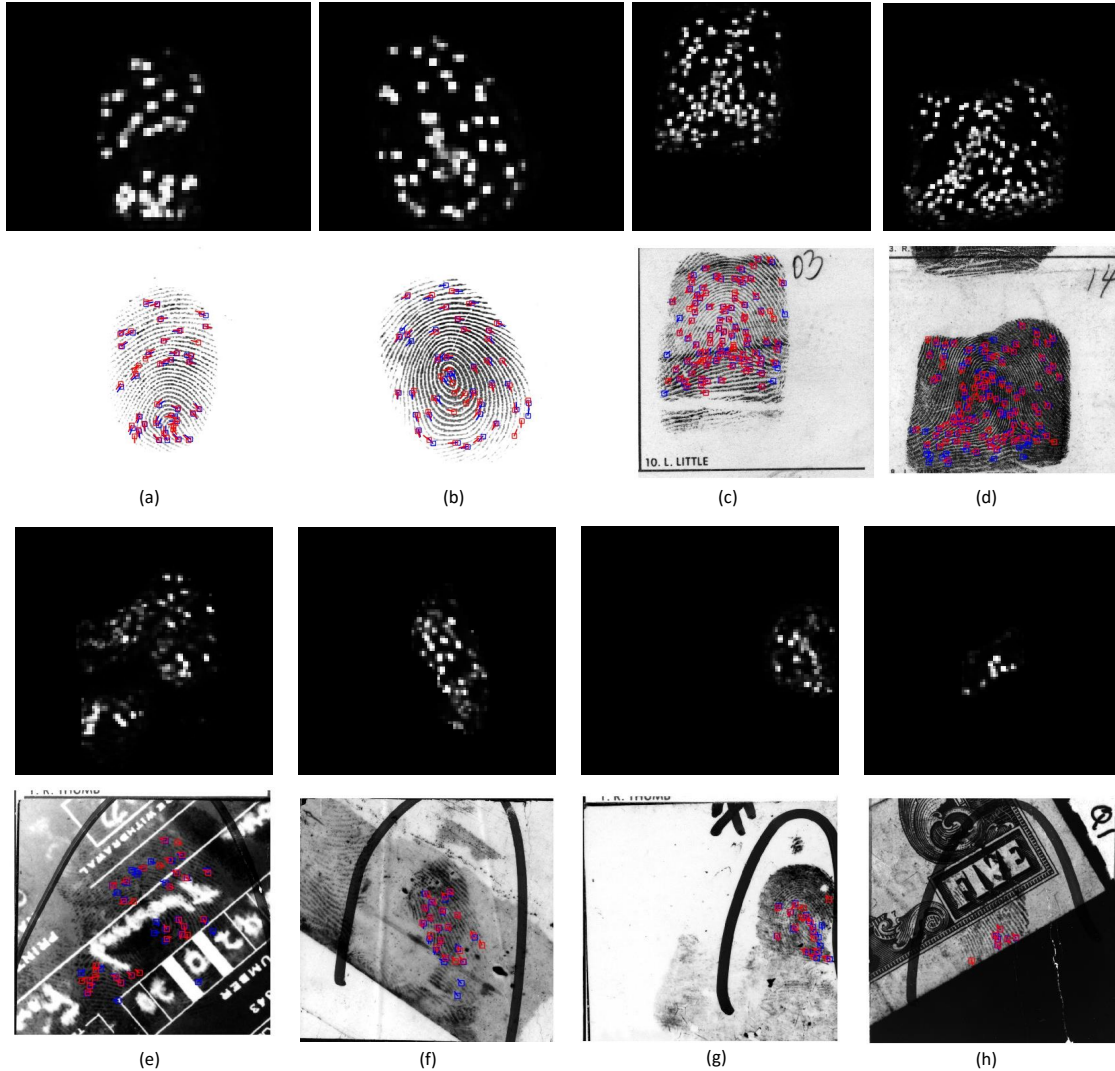


Figure 7. Visualizing minutiae extraction results. From top to bottom in each column: score maps, and minutiae extraction results overlaid on fingerprint images. (a)-(b): two plain images from FVC 2004; (c)-(d): rolled (reference) fingerprints from NIST SD27; (e)-(h): latent fingerprint images from NIST SD27.

- [16] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2002: Second fingerprint verification competition. In *Proc. 16th ICPR*, volume 3, pages 811–814, 2002. 5
- [17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. PAMI*, 39(6):1137–1149, 2017. 4
- [18] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh. On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders. In *Proc. IEEE IJCB*, pages 1–7, 2014. 2, 3, 6
- [19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5
- [20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. AAAI*, pages 4278–4284, 2017. 1, 2, 5
- [21] Y. Tang, F. Gao, and J. Feng. Latent fingerprint minutia extraction using fully convolutional network. In *Proc. IEEE IJCB*, 2017. 2, 3, 4, 6
- [22] Y. Tang, F. Gao, J. Feng, and Y. Liu. Fingernet: An unified deep network for fingerprint minutiae extraction. In *Proc. IEEE IJCB*, 2017. 2, 3, 4, 6
- [23] Verifinger. Neuro-technology, 2010. 6
- [24] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, and K. Ko. User’s guide to NIST biometric image software (NBIS). *NIST Interagency/Internal Report 7392*, 2007. 6
- [25] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. ECCV*, pages 499–515. Springer, 2016. 5
- [26] X. Yang, J. Feng, and J. Zhou. Localized dictionaries based orientation field estimation for latent fingerprints. *IEEE Trans. PAMI*, 36(5):955–969, 2014. 4
- [27] S. Yoon, J. Feng, and A. K. Jain. Latent fingerprint enhancement via robust orientation field estimation. In *Proc. IEEE IJCB*, pages 1–8, 2011. 2
- [28] F. Zhao and X. Tang. Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. *Pattern Recognition*, 40(4):1270–1281, 2007. 1