This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

Automated Latent Fingerprint Recognition

Kai Cao and Anil K. Jain, *Fellow, IEEE*

Abstract—Latent fingerprints are one of the most important and widely used evidence in law enforcement and forensic agencies worldwide. Yet, NIST evaluations show that the performance of state-of-the-art latent recognition systems is far from satisfactory. An automated latent fingerprint recognition system with high accuracy is essential to compare latents found at crime scenes to a large collection of reference prints to generate a candidate list of possible mates. In this paper, we propose an automated latent fingerprint recognition algorithm that utilizes Convolutional Neural Networks (ConvNets) for ridge flow estimation and minutiae descriptor extraction, and extract complementary templates (two minutiae templates and one texture template) to represent the latent. The comparison scores between the latent and a reference print based on the three templates are fused to retrieve a short candidate list from the reference database. Experimental results show that the rank-1 identification accuracies (query latent is matched with its true mate in the reference database) are 64.7% for the NIST SD27 and 75.3% for the WVU latent databases, against a reference database of 100K rolled prints. These results are the best among published papers on latent recognition and competitive with the performance (66.7% and 70.8% rank-1 accuracies on NIST SD27 and WVU DB, respectively) of a leading COTS latent Automated Fingerprint Identification System (AFIS). By score-level (rank-level) fusion of our system with the commercial off-the-shelf (COTS) latent AFIS, the overall rank-1 identification performance can be improved from 64.7% and 75.3% to 73.3% (74.4%) and 76.6% (78.4%) on NIST SD27 and WVU latent databases, respectively.

Index Terms—Latent fingerprints, reference prints, automated latent recognition, minutiae descriptor, convolutional neural networks, texture template.

1 Introduction

Ever since latent fingerprints (latents or marks[1]) were first introduced as evidence to convict a suspect in Argentina in 1893, they have become one of the most important and widely used sources of evidence in law enforcement and forensic agencies worldwide [4]. Latent fingerprint recognition requires recognizing the mate of a latent print evidence in a database of reference prints (rolled or slap fingerprints). See Figs. 1 and 2. A majority (60%) of crime laboratories in the United States reported analyzing latent fingerprints recovered from crime scenes, and a total of 271,000 latent prints were processed by public forensic crime laboratories in 2009 alone[2]. During January 2017, FBI's Integrated Automated Fingerprint Identification System (IAFIS), which maintains the largest criminal fingerprint database in the world, conducted 17,758 latent "feature" searches (latent features were manually marked by latent examiners), and an additional 4,160 latent "image" searches [5] (latent features were automatically extracted by IAFIS).

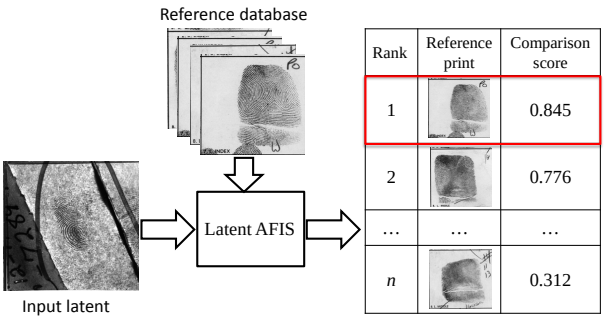


Fig. 1: Automated latent recognition framework. A latent image is input to a latent AFIS, and the top n candidates with their comparison scores are presented to a latent expert. The number of candidates, n, examined is typically less than 20. The true mate in this example is outlined in red.

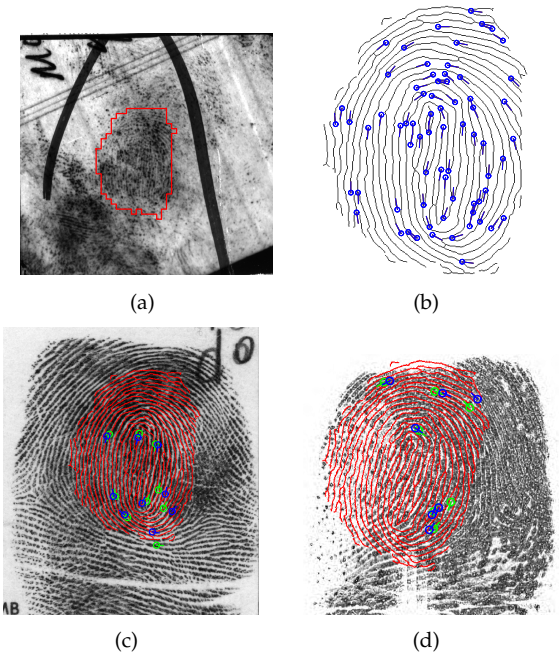
Compared to rolled and slap prints (or reference prints), which are acquired under supervision, latent prints are lifted after being unintentionally deposited by a subject, e.g., at crime scenes, typically resulting in poor quality in terms of ridge clarity and large background noise. Unlike reference prints, the action of depositing finger mark on a surface is not repeatable if latent prints are found to be of poor quality. National Institute of Standards & Technology (NIST) periodically conducts technology evaluations of fingerprint recognition algorithms, both for rolled (or slap) and latent prints. In NIST's most recent evaluation of rolled and slap prints, FpVTE 2012, the best performing Automated Fingerprint Identification

• *Kai Cao and A.K. Jain are with the Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 U.S.A.*
E-mail: {kaicao,jain}@cse.msu.edu

1. Latent and mark both refer to a partial and smudgy friction ridge impression from an unknown source. The term latent is preferred in North America while mark is preferred outside North America [1]. We adopt the term latent here to be consistent with our previous work [2], [3].

2. Bureau of Justice Statistics, Census of Publicly Funded Forensic Crime Laboratories, 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b) (c) (d)

Fig. 2: Illustration of latent to reference (rolled) comparison. (a) Input latent with ROI outlined in red, (b) automatically extracted minutiae in (a) shown on the latent skeleton, (c) alignment and minutiae correspondences between the latent and its true mate (rank-1 retrieval) and (d) alignment and minutiae correspondences between the latent and the rank-2 retrieved rolled print. Blue circles denote latent minutiae and green circles denote rolled minutiae.

System (AFIS) achieved a false negative identification rate (FNIR) of 1.9% for single index fingers, at a false positive identification rate (FPIR) of 0.1% using 30,000 search subjects (10,000 subjects with mates and 20,000 subjects with no mates) [6]. For latent prints, the most recent evaluation is the NIST ELFT-EFS where the best performing automated latent recognition system could only achieve a rank-1 identification rate of 67.2% in searching 1,114 latents against a background containing 100,000 reference prints [7]. The rank-1 identification rate of the best performing latent AFIS was improved from 67.2% to 70.2%[3] [7] when feature markup by a latent expert was also input, in addition to the latent images, to the AFIS. This gap between reference fingerprint recognition and latent fingerprint recognition capabilities is primarily due to the poor quality of friction ridges in latent prints. This underscores the need for developing automated latent recognition with high accuracy[4].

1.1 Current Practice

The standard procedure for latent recognition, as practiced in forensics agencies, involves four phases: *Analysis*, *Comparison*, *Evaluation*, and *Verification* (ACE-V) [8]. A number of studies have highlighted limitations of the ACE-V methodology.

1) *Repeatability/reproducibility of feature markup.* Ulery et al. [9] and Arora et al. [10] observed a large variation among the feature markups on the same latent provided by different examiners which affects the latent recognition accuracy [7]. The median value of markup reproducibility was found to be only 46% [9].
2) *Repeatability/reproducibility of decision.* Examiner repeatability of comparison decisions was found to be 90.0% for mated pairs, and only 85.9% for non-mated pairs [11]. These values were even lower for comparisons assessed by the examiners as "difficult" (i.e., low quality latents).
3) *Throughput.* Manual markup requires significant effort (∼15 min/latent[5]) by latent examiners.
4) *Bias.* Since the second examiner in the verification phase is only assessing the comparison decision made by the first examiner, it creates the potential for confirmation bias (see page 90 in [12]).

Given the current state of latent processing that relies heavily on forensic examiners, an automated latent recognition algorithm is urgently needed to give an accurate, reliable and efficient (i.e., a short) candidate list for ever growing case workload. An automated latent recognition system will also assist in developing quantitative validity and reliability measures[6] for latent fingerprint evidence as highlighted in the 2016 PCAST [12] and the 2009 NRC [13] reports.

Over the past few years, deep networks, in particular, convolutional neural networks (ConvNets) have become the dominant approach for addressing problems involving noisy, occluded and partial patterns and large numbers of classes. This is supported by state-of-the-art performance of deep networks in large-scale image recognition [14], unconstrained face recognition [15] and speech recognition in cluttered background [16], where traditional representation and matching approaches fail. So, it is natural to consider ConvNets for latent fingerprint recognition. However, only a few published studies have applied ConvNets to latent fingerprint recognition, and even these studies are limited to individual modules, such as ridge flow estimation [17] and minutiae extraction [18] [19] of latent AFIS. To our knowledge, there is no pub-

3. The best result using both markups and images is 71.4% rank-1 accuracy.

4. In forensics and law enforcement, automated latent recognition is also referred as *lights-out recognition* where the objective is to minimize the role of latent examiners in latent recognition.

5. https://www.noexperiencenecessarybook.com/eVbzD/microsoft-powerpoint-nist-fingerprint-testing-standards-v2-02282013-pptx.html

6. AFIS available from vendors neither provide the latent features they extract nor the true comparison scores between a latent and a reference print.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

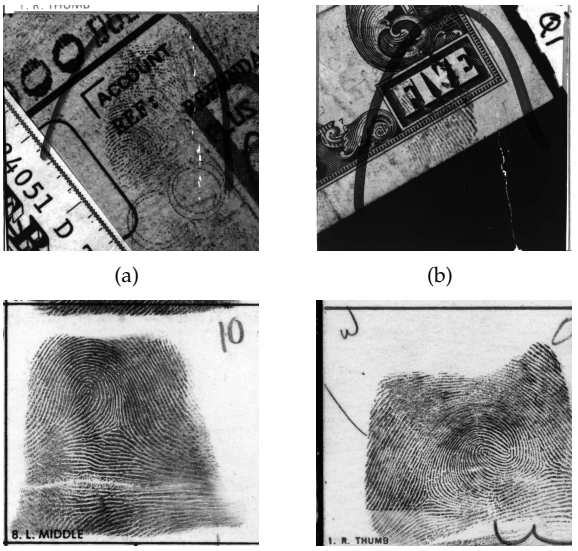


Fig. 3: Example latents (first row) from NIST SD27 whose true rolled mates (second row) could not be retrieved at rank-1 by a leading COTS latent AFIS. This can be attributed to large background noise and poor quality ridge structure in (a), and small friction ridge area in (b).

lished study on designing a complete latent AFIS based on ConvNets.

1.2 Contributions

In this paper, we design and build an automated latent recognition system[7] and evaluate its performance against a leading latent AFIS. Meagher and Dvornychenko [20] define seven tiers of possible latent print "lights out" scenarios. They go on to say that "for technical reasons, only Tiers 1 and 2 are implementable now or in the near term. Tiers 3 through 7 reflect our concept of an incremental approach to full lights-out capability." Our automated latent recognition system follows under Tier 2 of their definition where latent print experts submit latent searches and then receive the AFIS candidate list. All preprocessing, except region of interest (ROI), minutiae extraction, template generation and search has been automated. See Fig. 4.

The main contributions of this paper are as follows:

1) Input latent is represented by three different templates, each providing complementary information. Two of the templates are minutiae-based, whereas the third template is texture-based. The minutiae in the minutiae-based templates are extracted, respectively, based on (i) ridge flow learned from a ConvNet, and (ii) dictionary of ridge structure elements.
2) Multi-scale and multi-location windows in the neighborhood of minutiae are used to learn minutiae descriptors. To develop salient minutiae descriptors, we train 14 different ConvNets, where each descriptor ConvNet is trained on a specific patch size at a specific location around the minutiae. A systematic feature selection (sequential forward selection) showed that only 3 out of the 14 ConvNets are adequate to maintain rank-1 recognition accuracy at significant computational savings.
3) Second order (minutiae pairs) and third order (minutiae triplets) graph-based minutiae correspondence algorithms are developed to minimize false minutiae correspondences in latent to its non-mate comparisons.
4) A prototype of our latent recognition algorithm was evaluated on two different benchmark databases: NIST SD27 (258 latents) [21] and WVU latent DB (449 latents) [22] against a reference database of 100,000 rolled prints. The rank-1 retrieval for these two databases are: 64.7% for NIST SD27 and 75.3% for WVU latent DB. These results with automated preprocessing, feature extraction, and comparison are superior to published results on these two databases.
5) Score-level (rank-level) fusion of our algorithm with a leading COTS latent AFIS, improves the rank-1 accuracies to 73.3% (74.4%) for NIST SD27 and to 76.6% (78.4%) for WVU latent DB. This demonstrates that our approach to automated latent recognition based on ConvNets is complementary to that used in the COTS latent AFIS.

2 Related Literature

Given a latent image, the main modules of a latent AFIS include preprocessing (ROI segmentation, ridge flow estimation and ridge enhancement), feature (minutiae and texture) extraction and comparison. Fig. 3 shows challenges in latent processing: background noise, low contrast of friction ridge structure, and small friction ridge area. In the following, we briefly review major published algorithms pertaining to different modules. For a detailed review, see [23].

(a) *ROI segmentation*. Published algorithms [24], [25], [26], [27], [3] do not work well on poor quality latents. Further, it is a common practice in forensics for an examiner to mark the ROI, also known as cropping (see Fig 4), especially when there are overlapping latent impressions. We assume that ROI for the query latent has been marked.

(b) *Ridge flow estimation*. Two approaches for computing ridge flow have shown promise: (i) dictionary based learning, [28], [3] and (ii) ConvNet based learning [17]. The ridge flow estimates from ConvNet generally perform better than dictionary based methods

7. The SDK will be available to interested readers once the paper is accepted.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

when evaluated against manually marked ridge flow [17].

(c) *Latent enhancement*. Gabor filtering is the most popular and effective approach, [28], [3], [17]. Other published approaches include multi-scale ridge dictionary using a set of Gabor elementary functions [29], and a ridge dictionary with variable ridge and valley spacings [30].

(d) *Feature extraction*. A latent minutiae extractor using stacked denoising sparse autoencoder was proposed in [18], but it showed poor performance on NIST SD27. While Cao *et al.* [30] extracted minutiae, ridge clarity, singular point, and ridge orientation for automated latent value assessment, they did not integrate it with a latent matcher. Tang *et al.* [19] developed a fully convolutional network for minutiae extraction, but it performed poorly compared to manually marked minutiae.

(e) *Latent comparison*. In the absence of a robust latent minutiae extractor, published latent comparison algorithms [2], [31], [32], [33] rely on manually marked minutiae.

In summary, to our knowledge, no automated latent recognition algorithm has been published in the literature. While ConvNets have been used for individual modules of a latent AFIS, their performance has not been evaluated in an end-to-end system. Even the number of available COTS latent AFIS is limited. In the 2012 NIST ELFT-EFS #2 evaluation, there were only six participants; the top three performers had significantly superior performance compared to the other three. The flowchart of the proposed latent recognition framework is shown in Fig. 5.

3 Preprocessing and Feature Extraction

Latent feature extraction is presented in section 3.1, where latent preprocessing is embedded into minutiae set extraction, and reference print feature extraction is provided in section 3.2.

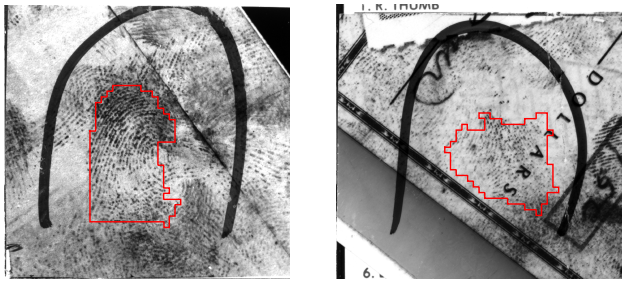


Fig. 4: Latent fingerprints at a crime scene often contain multiple latent impressions, either of different individuals or multiple fingers of the same person. For this reason, a region of interest (ROI), also called cropping, outlined in red, is typically marked by examiners to highlight the friction ridge region of interest.

3.1 Latent Feature Extraction

Minutiae are arguably the most important features in fingerprint recognition. Two minutiae templates and one texture template are extracted for each latent (see Fig. 5). While the two minutiae templates use the same framework (Fig. 6), they are based on different ridge flow estimation methods (ConvNet-based and Dictionary-based) and ridge enhancement methods (Dictionary-based and Gabor filtering-based). A minutiae template consists of ridge flow, a minutiae set (minutiae locations and orientations), and minutiae descriptors extracted by ConvNets using local latent patches.

3.1.1 Minutiae Set 1

The first minutiae set is extracted from the approach in [30], which consists of the following steps: 1) ridge flow estimation using ConvNet, 2) ridge and valley contrast enhancement, 3) ridge enhancement by a ridge structure dictionary with variable ridge and valley spacing, 4) ridge binarization and thinning, and 5) minutiae detection [34] in the skeleton image.

3.1.2 Minutiae Set 2

A coarse to fine dictionary is adopted to estimate ridge flow and ridge spacing [3]. Gabor filtering tuned using the estimated ridge flow and ridge spacing is used to enhance the ridge structure. Minutiae are then extracted from the enhanced latent to obtain minutiae set 2. A comparison in Fig. 5 shows the complementary nature of minutiae sets 1 and 2.

3.1.3 Texture Template

A texture template is introduced to account for situations where the latent is of such a small area that it does not contain sufficient number of minutiae (for reliable comparison to reference prints) or the latent is of very poor quality so the minutiae extraction is not reliable. In a texture template, we represent each non-overlapping local block ($s_b \times s_b$ pixels) in the latent by a pair of *virtual minutiae*. Let (x, y) and α be the location and orientation of the center of a block. Then the virtual minutiae pair is located at (x, y, α) and $(x, y, \alpha + \pi)$. Note that the virtual minutiae do not correspond to ridge endings and bifurcations and the virtual minutiae close to the border are removed. The same minutia descriptor algorithm (section 3.1.4) used for the true minutiae sets is also used for virtual minutiae. The block size is set to 16×16 to balance template efficacy and computational efficiency.

3.1.4 Minutiae Descriptor

A minutia descriptor contains attributes of the minutia based on the image characteristics in its neighborhood. Salient descriptors are needed to eliminate false minutiae correspondences between a latent and reference prints. Instead of specifying the descriptor

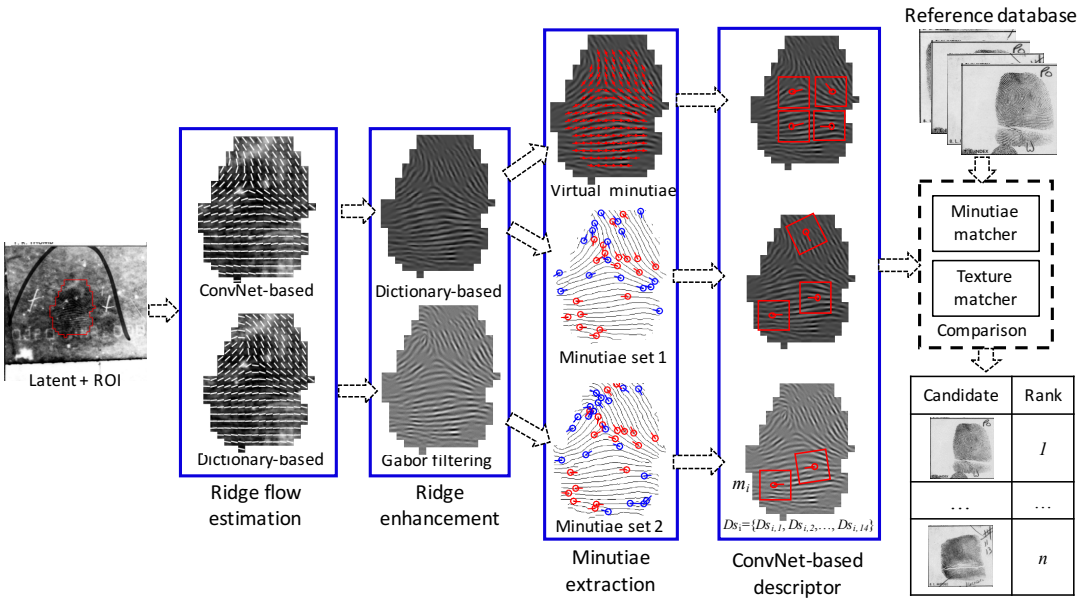


Fig. 5: Flowchart of the proposed latent recognition approach. The common minutiae in two true minutiae sets are shown in red.

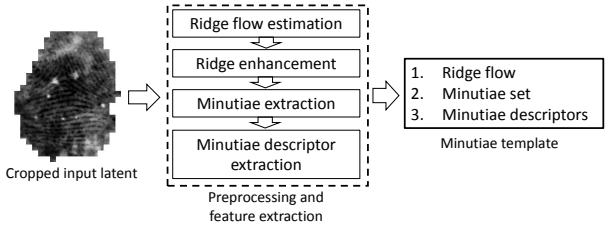


Fig. 6: Minutiae template generation. The same procedure is used for both minutiae template 1 and minutiae template 2.

in an ad hoc manner [2], we train ConvNets to learn minutiae descriptor from local fingerprint patches. As demonstrated in face recognition, for example, [36], training a set of ConvNets using multiple image patches at different scales and regions can significantly boost the recognition performance. In this paper, we adopt a multi-scale approach, where fingerprint patches of different sizes and at different locations (a total of 14 patches) are defined as shown in Fig. 8. Multiple instances of patches extracted for the same minutia are used to train 14 different ConvNets[8]. The flowchart of minutiae descriptor extraction for one of the 14 ConvNets is illustrated in Fig. 7. The details are as follows.

1) *Training patch selection*. Multiple patches around the same minutiae extracted from different fingerprint impressions of the same finger are needed. For this purpose, we utilize MSP longitudinal fingerprint database[9] [35], which contains 1,311 subjects with at least 10 rolled impressions, collected over at least 5 years, with a total of 165,880 fingerprints. Only those minutiae in these prints which can be extracted in eight or more impressions of the same finger are retained for training. This ensures that we are only using reliable minutiae. Local fingerprint patches around these selected minutiae are extracted to train the ConvNets.
2) *Training*. We adopt the same ConvNet architecture proposed in [17] for all 14 patch types. Smaller patches are resized to 160×160 pixels using bilinear interpolation to ensure that we can use the same ConvNet [17] with 160×160 images as input. Random shifts (-5 to 5 pixels) and rotations (-5° to 5°) of the patches are used to augment the training set.
3) *Latent minutiae descriptor extraction*. For each ConvNet, its 128-dimensional output of the last fully connected layer is considered as a feature vector. A minutia descriptor could be a concatenation of a subset of the 14 feature vectors output by the 14 ConvNets.

3.2 Reference Print Feature Extraction

Reference prints are typically of higher quality compared to latents, so it is easier to get reliable minutiae from them. For this reason, we extract only one minutiae template, but we still extract the texture template. The reference print minutiae are extracted by a COTS tenprint AFIS rather than the proposed minutiae extractor for latents. The ridge flow is extracted by Short

8. The toolbox MatConvNet [37] is used to implement the ConvNet architecture. Offline training of the ConvNet is conducted on a Linux server with Tesla K20 GPUs.

9. No longitudinal latent database is available for training descriptor ConvNet.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

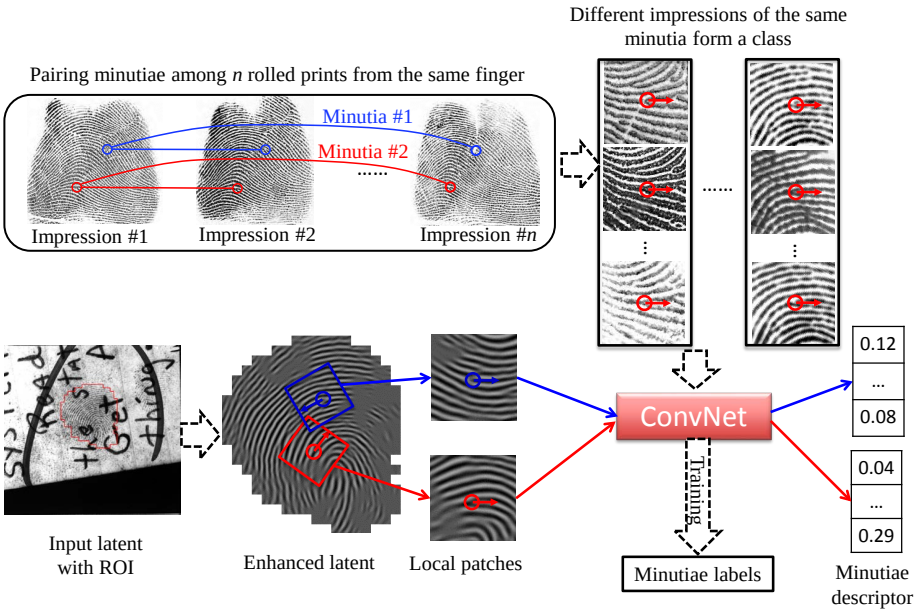


Fig. 7: Minutiae descriptor extraction via ConvNet. The dotted arrows show the offline training process, while solid arrows show the online process for minutiae descriptor extraction. A total of 800K fingerprint patches from 50K minutiae, extracted from the MSP longitudinal fingerprint database [35], were used for training the ConvNet. The patch size shown here is 80×80 pixels.

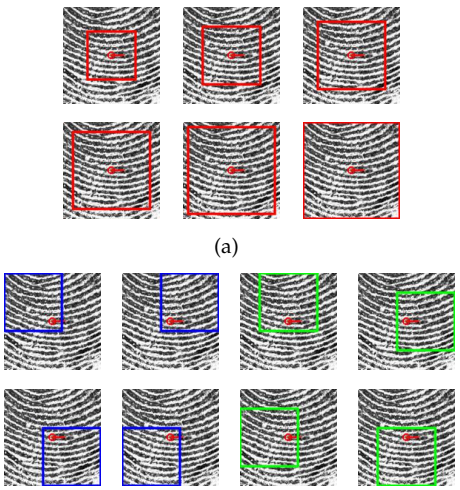


Fig. 8: Fourteen types of fingerprint patches, with different size and location, centered at a minutia (shown in red). Patches at (a) 6 different scales and (b) in 8 different locations around minutia: top left, top right, bottom right and bottom left, top, right, left and bottom. The fingerprint patches shown here are of size 160×160 pixels. The window sizes (scale) in (a) are 80×80, 96×96, 112×112, 128×128, 144×144, and 160×160 pixels. The windows in (b) are all of size 96×96 pixels.

Time Fourier Transform (STFT) [38]. A reference print minutiae template, similar to latents, includes (i) ridge flow, (ii) minutiae set and (iii) minutiae descriptors (section 3.1.4).

The texture template for reference print is extracted in a manner similar to latents (section 3.1.3). For computational efficiency, each nonoverlapping block of $s_b \times s_b$ pixels is considered to define a single virtual minutia. On average, there are 1,018 virtual minutiae in a reference print. The texture template consists of a virtual minutiae set, and their descriptors (section 3.1.4). Since the latent texture template considers two virtual minutiae, we expect that at least one of them will be in correspondence with the reference print virtual minutia in the true mate.

4 Latent to Rolled Comparison

Two latent-to-reference print comparison algorithms are designed: (i) a minutiae template comparison algorithm and (ii) a texture template comparison algorithm.

4.1 Minutiae Template Comparison

Let $M^l = \{m_i^l = (x_i^l, y_i^l, \alpha_i^l)\}_{i=1}^{n_l}$ denote the latent minutiae set with n_l minutiae, where (x_i^l, y_i^l) and α_i^l are the location and orientation of the i^{th} minutia, respectively. Let $M^r = \{m_j^r = (x_j^r, y_j^r, \alpha_j^r)\}_{j=1}^{n_r}$ denote a reference print minutiae set with n_r minutiae, where (x_j^r, y_j^r) and α_j^r are the location and orientation of the j^{th} rolled minutia, respectively. The minutiae template comparison algorithm seeks to establish the minutiae correspondences between M^l and M^r. We impose the constraint that no minutia in one set should match more than one minutia in the other set. The problem of minutiae correspondence can be formulated as an

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

optimization problem to find the assignment $X \in \mathbb{S}$, where:

$$\mathbb{S} = \{X \in \{0,1\}^{n_l \times n_r}, \forall i, \sum_i X_{i,j} \leqslant 1, \forall j, \sum_j X_{i,j} \leqslant 1\},$$

$X_{i,j} = 1$ if m_i^l and m_j^r are in correspondence and $X_{i,j} = 0$, otherwise.

In the second-order graph based minutiae correspondence algorithm [39], the objective function S_2 is defined as:

$$S_2(X) = \sum_{i_1,i_2,j_1,j_2} H^2_{i_1,i_2,j_1,j_2} X_{i_1,i_2} X_{j_1,j_2}, \quad (1)$$

where $H^2 \in \mathbb{R}^{n_l \times n_r \times n_l \times n_r}$ is a 4-dimensional tensor and $H^2_{i_1,i_2,j_1,j_2}$ measures the compatibility between latent minutiae pair $(m^l_{i_1}, m^l_{j_1})$ and rolled minutiae pair $(m^r_{i_2}, m^r_{j_2})$.

One limitation of the second-order graph matching (or pairwise minutiae correspondence) is that it is possible that two different minutiae configurations may have similar minutiae pairs. To circumvent this, higher order graph matching, has been proposed to reduce the number of false correspondences [40]. Here, we consider the third-order graph matching (minutiae triplets) whose objective function is given as:

$$S_3(X) = \sum_{i_1,j_1,k_1,i_2,j_2,k_2} H^3_{i_1,i_2,j_1,j_2,k_1,k_2} X_{i_1,i_2} X_{j_1,j_2} X_{k_1,k_2}, \quad (2)$$

where $H^3 \in \mathbb{R}^{n_l \times n_r \times n_l \times n_r \times n_l \times n_r}$ is a 6-dimensional tensor and $H^3_{i_1,i_2,j_1,j_2,k_1,k_2}$ measures the compatibility between latent minutiae triplet $(m^l_{i_1}, m^l_{j_1}, m^l_{k_1})$ and reference print minutiae triplet $(m^r_{i_2}, m^r_{j_2}, m^r_{k_2})$. Since H^3 is of size $(n_l \cdot n_r)^3$ and H^2 is of size $(n_l \cdot n_r)^2$, this approach is more computationally demanding than the second-order graph matching.

4.1.1 Proposed Minutiae Correspondence Algorithm

Minutiae descriptors allow us to consider only a small subset of minutiae correspondences among the $n_l \times n_r$ possible correspondences. For computational efficiency, only the top N ($N = 120$) minutiae correspondences are selected based on their descriptor similarities. Since the second-order graph matching is able to remove most of the false correspondences, we first use the second-order graph matching, followed by the third-order graph matching for minutiae correspondence. **Algorithm** 1 shows the main steps of the proposed minutiae correspondence algorithm.

In the following, we first present how to construct H^2 and H^3 and then give details of the minutiae correspondence algorithm.

4.1.2 Construction of H^2 and H^3

The term $H^2_{i_1,i_2,j_1,j_2}$ in Eq. (1) measures the compatibility between a minutiae pair $(m^l_{i_1}, m^l_{j_1})$ of the latent and a minutiae pair $(m^r_{i_2}, m^r_{j_2})$ of the reference print. A 4-dimensional feature vector is computed to characterize each minutiae pair. Let (d_{i_1,j_1}, θ_{i_1}, θ_{j_1}, θ_{i_1,j_1}) and (d_{i_2,j_2}, θ_{i_2}, θ_{j_2}, θ_{i_2,j_2}) denote two feature vectors for a minutiae pair from a latent and a reference print, respectively. Fig. 9 (a) illustrates the feature vector. $H^2_{i_1,i_2,j_1,j_2}$ is computed as:

$$H^2_{i_1,i_2,j_1,j_2} = \Pi_{p=1}^4 Z(d_p, \mu_p, \tau_p, t_p), \quad (3)$$

where

$$d_1 = |d_{i_1,j_1} - d_{i_2,j_2}|, \quad (4)$$

$$d_2 = \min(|\theta_{i_1} - \theta_{i_2}|, 2\pi - |\theta_{i_1} - \theta_{i_2}|), \quad (5)$$

$$d_3 = \min(|\theta_{j_1} - \theta_{j_2}|, 2\pi - |\theta_{j_1} - \theta_{j_2}|), \quad (6)$$

$$d_4 = \min(|\theta_{i_1,j_1} - \theta_{i_2,j_2}|, 2\pi - |\theta_{i_1,j_1} - \theta_{i_2,j_2}|), \quad (7)$$

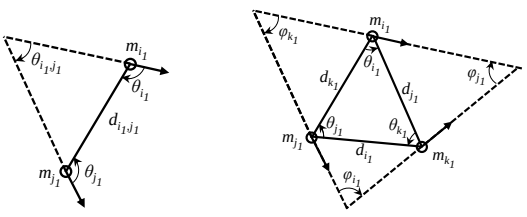
Z is a truncated sigmoid function, which is defined as:

$$Z(v, \mu_p, \tau_p, t_p) = \begin{cases} \frac{1}{1+e^{-\tau_p(v-\mu_p)}}, & \text{if } v \leq t_p, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

and μ_p, τ_p and t_p are parameters of function Z.

Algorithm 1 Minutiae correspondence algorithm

1: **Input:** Latent minutiae template with n_l minutiae and reference print minutiae template with n_r minutiae
2: **Output:** Minutiae correspondences
3: Compute the $n_l \times n_r$ minutiae similarity matrix using Eq. (10)
4: Select the top N minutiae correspondences based on the above minutiae similarity matrix
5: Construct H^2 based on these N minutiae pairs
6: Remove false minutiae correspondences using **Algorithms** 2 and 4
7: Construct H^3 for the remaining minutiae pairs
8: Remove false minutiae correspondences using **Algorithms** 3 and 4
9: Output final minutiae correspondences.



(a) (b)

Fig. 9: Illustration of feature representation of (a) a minutiae pair (m_{i_1}, m_{j_1}) and (b) a minutiae triplet $(m_{i_1}, m_{j_1}, m_{k_1})$, where the solid arrows denote minutiae orientations.

The term $H^3_{i_1,i_2,j_1,j_2,k_1,k_2}$ in Eq. (2) measures the compatibility between a minutiae triplet $(m^l_{i_1}, m^l_{j_1}, m^l_{k_1})$ of the latent and a minutiae triplet $(m^r_{i_2}, m^r_{j_2}, m^r_{k_2})$ of the reference print. A 9-dimensional vector is computed to characterize each minutiae triplet, as illustrated in Fig. 9 (b). Let $(d_{i1}, d_{j1}, d_{k1}, \theta_{i1}, \theta_{j1}, \theta_{k1}, \varphi_{i1}, \varphi_{j1}, \varphi_{k1})$ and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 2 Power iteration for the second-order eigenvalue problem

Input: Matrix H^2
Output: Y, principal eigenvector of H^2
Initialize Y with small random positive numbers
while no convergence **do**
    $Y \leftarrow HY$
    $Y \leftarrow \frac{1}{||Y||_2} Y$

Algorithm 3 Power iteration for the third-order eigenvalue problem

Input: Matrix H^2
Output: Y, principal eigenvector of H^2
Initialize Y with small random positive numbers
while no convergence **do**
    for i **do**
        $Y_i \leftarrow \sum_{j,k} H^3_{i,j,k} Y_j Y_k$
    $Y \leftarrow \frac{1}{||Y||_2} Y$

$(d_{i2}, d_{j2}, d_{k2}, \theta_{i2}, \theta_{j2}, \theta_{k2}, \varphi_{i2}, \varphi_{j2}, \varphi_{k2})$ denote two feature vectors corresponding to the two minutiae triplets from the latent and the reference print, respectively. Then $H^3_{i_1,j_1,i_2,j_2,k_1,k_2}$ is computed as:

$$H^3_{i_1,j_1,i_2,j_2,k_1,k_2} = \Pi_{p=i,j,k} \Pi^3_{q=1} Z(d_{pq}, \mu_{pq}, \tau_{pq}, t_{pq}), \quad (9)$$

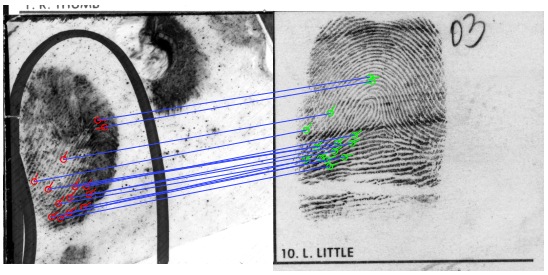
where

$$\begin{aligned} d_{p1} &= |d_{p1} - d_{p2}|, \\ d_{p2} &= \min(|\theta_{p1} - \theta_{p2}|, 2\pi - |\theta_{p1} - \theta_{p2}|), \\ d_{p3} &= \min(|\phi_{p1} - \phi_{p2}|, 2\pi - |\phi_{p1} - \phi_{p2}|), \\ p &= i, j, k. \end{aligned}$$

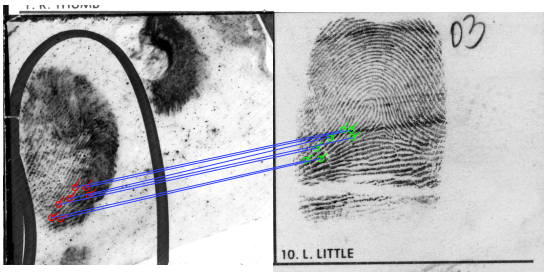
There are two kinds of distances used in computing H^2 and H^3, i.e. Euclidean distance (e.g., Eq. (4)) between minutiae locations, and directional distance (e.g., Eqs. (5), (6) and (7)) between minutiae angles. For the Euclidean distance, μ, τ and t are set as 15, $-\frac{1}{5}$ and 40, respectively. For the directional distance, μ, τ and t are set as $\frac{1}{12}$, -15 and $\pi/4$, respectively. These tolerance values were determined empirically.

Algorithm 4 Discretization to ensure a one-to-one matching

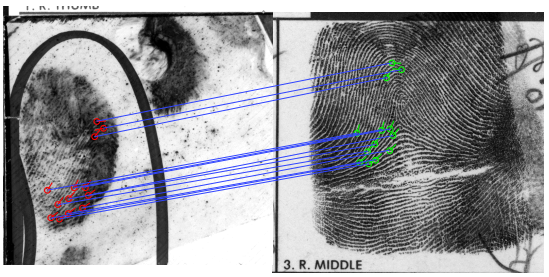
Input: Eigenvector Y output by **Algorithms** 3 or 2
Output: Minutiae correspondences C
Initialize threshold T
Initialize minutiae pair $C = \{\}$
Set $flag_l(p) = 0$, $p = 1, 2, ..., n_l$
set $flag_r(q) = 0$, $q = 1, 2, ..., n_r$
while $\max(Y) > T$ **do**
    $i = \arg\max(Y)$
    Y(i) = 0
    if $flag_l(i_1) == 1$ or $flag_r(i_2) == 1$ **then**
        continue
    else
        C.append(i_1,i_2)
        $flag_l(i_1) = 1$
        $flag_r(i_2) = 1$



(a)



(b)



(c)

Fig. 10: Comparison of minutiae correspondences. (a) 14 minutiae pairs found in correspondence between the latent and a non-mate [31], (b) 7 minutiae pairs found in correspondence for the same comparison as in (a) by the proposed method and (c) 13 minutiae pairs found in correspondences between the latent and its true mate by the proposed method. Note that we use manually marked minutiae and MCC descriptor [41] for a fair comparison with [31].

4.1.3 Proposed Minutiae Correspondence

Suppose $Des^l_{i_1} = \{Des^l_{i_1}(p)\}_{p \in P}$ and $Des^r_{i_2} = \{Des^r_{i_2}(p)\}_{p \in P}$ are two sets of minutia descriptors of the i_1th latent minutia and the i_2th reference print minutia, respectively, where P is a subset of the 14 ConvNets. The descriptor similarity $DesSim(i_1, i_2)$ between $Des^l_{i_1}$ and $Des^r_{i_2}$ is computed based on cosine

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

distance as follows:

$$DesSim(i_1, i_2) = \frac{1}{\sum_{p \in P} 1} \sum_{p \in P} \frac{(Des^l_{i_1}(p))^T \cdot Des^r_{i_2}(p)}{||Des^l_{i_1}(p)|| \cdot ||Des^r_{i_2}(p)||}. \quad (10)$$

As in section 4.1.2, the top N minutiae correspondences with the highest similarity values in (10) are selected. Suppose $\{(i_1, i_2)\}_{i=1}^N$ are the N selected minutiae pairs, and Y is an N-dimensional correspondence vector, where the i^{th} element (Y_i) indicates whether i_1 is assigned to i_2 ($Y_i = 1$) or not ($Y_i = 0$). The objective function in (1) can be simplified as

$$S_2(Y) = \sum_{i,j} H^2_{i,j} Y_i Y_j, \quad (11)$$

where $i = (i_1, i_2)$ and $j = (j_1, j_2)$ are two selected minutiae correspondences, and $H^2_{i,j}$ is equivalent to $H^2_{i_1,j_1,i_2,j_2}$. Objective function (2) can be similarly rewritten as:

$$S_3(Y) = \sum_{i,j,k} H^3_{i,j,k} Y_i Y_j Y_k, \quad (12)$$

where $i = (i_1, i_2)$, $j = (j_1, j_2)$ and $k = (k_1, k_2)$ are three selected minutiae correspondences, and $H^3_{i,j,k}$ is equivalent to $H^3_{i_1,i_2,j_1,j_2,k_1,k_2}$.

The second-order graph matching problem (11) is a quadratic assignment problem, with no known polynomial time algorithm for solving it. This also holds for the third-order graph matching problem (12). A strategy of power iteration, followed by discretization [40] is a simple but efficient approach to obtain approximate solution for (11) and (12). The power iteration methods for (11) and (12) are shown in **Algorithms** 2 and 3, respectively. **Algorithm** 4 is the discretization step to ensure a one-to-one matching.

Figs. 10 (a) and (b) compare the proposed minutiae correspondence algorithm with the method of [31] on an impostor comparison (latent to a non-mate comparison). Fig. 10 (c) shows an example of minutiae correspondences for a genuine match between a latent and its rolled mate.

4.1.4 Minutiae Template Similarity

The similarity between a latent minutiae template and a reference minutiae template consists of two parts: (i) minutiae similarity, i.e., similarity of descriptors of matched minutiae correspondences, and (ii) ridge flow similarity. Suppose $\{(m^l_{i_1} = (x^l_{i_1}, y^l_{i_1}, \alpha^l_{i_1}), m^r_{i_2} = (x^r_{i_2}, y^r_{i_2}, \alpha^r_{i_2}))\}_{i=1}^n$ are the n matched minutiae correspondences between the latent and the reference print by **Algorithm** 1. The minutiae similarity S_M is defined as:

$$S_M = \sum_{i=1}^{n} DesSim(i_1, i_2), \quad (13)$$

where $DesSim(i_1, i_2)$ is the descriptor similarity between $Des^l_{i_1}$ and $Des^r_{i_2}$ in Eq. (10). The ridge flow similarity is computed by first aligning the two ridge flow maps using the minutiae correspondences and then computing the orientation similarity of overlapping blocks. The rotation $\Delta\alpha$, and translation $(\Delta x, \Delta y)$ is computed as:

$$\Delta\alpha = \arctan(\sum_{i=1}^{n} \sin(\Delta\alpha_i), \sum_{i=1}^{n} \cos(\Delta\alpha_i)), \quad (14)$$

$$\Delta x = \frac{1}{n} \sum_{i=1}^{n} (x^r_{i_2} - x^l_{i_1} \cos(\Delta\alpha) + y^l_{i_1} \sin(\Delta\alpha)), \quad (15)$$

$$\Delta y = \frac{1}{n} \sum_{i=1}^{n} (y^r_{i_2} - y^l_{i_1} \cos(\Delta\alpha) - x^l_{i_1} \sin(\Delta\alpha)), \quad (16)$$

where $\Delta\alpha_i = (\alpha^r_{i_2} - \alpha^l_{i_1})$. The values of $\Delta\alpha$ and $(\Delta x, \Delta y)$ are used for ridge flow alignment. Let $\{O_{k,1}\}_{k=1}^K$ and $\{O_{k,2}\}_{k=1}^K$ denote the orientations in the overlapping K blocks for the latent and the reference print, respectively. The ridge flow similarity S_O is given by

$$S_O = \frac{1}{K} |\sum_{k=1}^{K} e^{(2\sqrt{-1}(O_{k,1} - O_{k,2}))}|. \quad (17)$$

The minutiae template similarity S_{MT} is computed as the product of the minutiae similarity and ridge flow similarity,

$$S_{MT} = S_M \cdot S_O. \quad (18)$$

4.2 Texture Template Similarity

The same minutiae comparison algorithm proposed in section 4.1 can be used for virtual minutiae comparison in texture template. However, there are two main differences: (i) top $N = 200$ virtual minutiae correspondences, rather than 200 for real minutiae, are selected based on descriptor similarity, and (ii) the texture template similarity S_{TT} only consists of the sum of the similarities of matched virtual minutiae correspondences in Eq. (13).

4.3 Similarity Score Fusion

Two minutiae templates and one texture template are extracted for each latent, but only one minutiae template and one texture template are extracted for each reference print. Two minutiae template similarity scores ($S_{MT,1}$ and $S_{MT,2}$) are generated by comparing the two latent minutiae templates against the single reference minutiae template. The texture similarity score (S_{TT}) is generated by comparing the latent and reference print texture templates. The final similarity score S between the latent and the reference print is computed as the weighted sum of $S_{MT,1}$, $S_{MT,2}$ and S_{TT} as below:

$$S = \lambda_1 S_{MT,1} + \lambda_2 S_{MT,2} + \lambda_3 S_{TT}, \quad (19)$$

where λ_1, λ_2 and λ_3 are the weights. We empirically determine the values of λ_1, λ_2 and λ_3 to be 1, 1 and 2, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

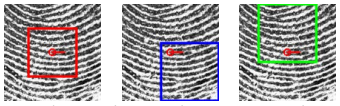


Fig. 11: Three selected patch types. The window size of the leftmost is 80×80 pixels. The other two windows are both of size 96×96 pixels.

5 Experimental Results

There is a dearth of latent fingerprint databases available to academic researchers. In this paper, we use two latent databases, NIST SD27 [21] and the West Virginia University latent database[10] (WVU DB) [22] available to us, to evaluate the proposed latent recognition algorithm. The NIST SD27 contains 258 latent fingerprints with their mated reference prints. The WVU DB contains 449 latents with their mated reference prints. Note that the NIST SD27 latent database is a collection of latents from the casework of forensics agencies, whereas WVU DB was collected in a laboratory setting, primarily by students, at West Virginia University. As such, the characteristics of these two databases are quite different in terms of background noise, ridge clarity, and the number of minutiae. The ridges in some of the latent images in WVU DB are broken apparently because of dry fingers. See Fig. 17 for a comparison of the images in the two databases.

In addition to the mated reference prints available in these databases, we use additional reference prints, from NIST SD14 [42] and a forensic agency, to enlarge the reference database to 100,000 for experiments reported here. The larger reference database allows for a challenging latent recognition problem. We follow the protocol used in NIST ELFT-EFS [43] [7] to evaluate the recognition performance of our system.

5.1 Selection of ConvNets for Minutiae Descriptor

Use of all 14 ConvNets, i.e., 14 patch types in Fig. 8, for minutiae descriptor may not be necessary to achieve the optimal recognition performance. We explore feature selection techniques to determine a subset of these 14 descriptors that will maintain the latent recognition accuracy. A sequential forward selection (SFS) [44] of the 14 patch types, using rank-1 accuracy as the criterion on the NIST SD27 database, revealed that 3 out of 14 patch types (Fig. 11) are adequate without a significant loss in accuracy (75.6% v. 74.4%) yet giving us a significant speed up. In the following experiments, we use only these 3 patch types.

5.2 Performance of Individual Latent Templates

Our objective for designing three different templates is to extract complementary information from latents. Fig. 12 (a) and Fig. 13 (a) compare the Cumulative Match Characteristic (CMC) curves of the three individual templates, namely, minutiae template 1, minutiae template 2 and texture template, on NIST SD27 and WVU DB, respectively. The minutiae template 1 performs significantly better than the minutiae template 2 on both latent databases. The main reason is that the ridge flow used for generating minutiae set 1, based on ConvNet, is more robust than minutiae set 2 extractor, based on ridge flow dictionary. Note that the performance of texture template, which does not utilize any of the true minutiae in latents, is close to the performance of minutiae template 2 on both NIST SD27 and WVU DB. This can be attributed to the virtual minutiae representation in the texture template and corresponding descriptors extracted by ConvNets. Fig. 14 shows an example latent whose true mate can be retrieved at rank 1 using minutiae template 1 but not minutiae template 2. The main reason is that the extracted ridge flow for this latent is better around the lower core point for minutiae template 1 than minutia template 2. The true mate of the latent shown in Fig. 15 (a) can be retrieved at rank 1 using minutiae template 2 but not minutiae template 1 even though their skeletons look similar. Fig. 16 shows two latent examples which lack reliable minutiae but the texture template is able to find their true mates at rank 1.

TABLE 1: Rank-1 (rank-20) identification rates on latents from NIST SD27 of different quality levels.

Quality	Minutiae template 1	Minutiae template 2	Texture template
Good	73.7% (84.1%)	71.6% (79.6%)	65.9% (83.0%)
Bad	52.9% (68.2%)	47.1% (61.2%)	51.7% (63.5%)
Ugly	27.1% (45.9%)	28.2% (37.7%)	31.8% (47.1%)

The identification accuracies on different quality latents in NIST SD27 are shown in Table 1. Note the superior (comparable) performance of the texture (virtual minutiae) template compared to the two minutiae templates on ugly (bad) latents. We also evaluate fusion of different subsets of the three templates. The fusion of any two templates using the weights in Eq. (19) performs better than any single template, and the performance can be further improved by fusing all three templates. This demonstrates that the three templates proposed here contain complementary information for latent recognition. Most significantly, the texture template, in conjunction with the two minutiae templates boosts the overall recognition performance (from 58.5% to 64.7% rank-1 accuracy on NIST SD27 and from 70.6% to 75.3% on WVU DB).

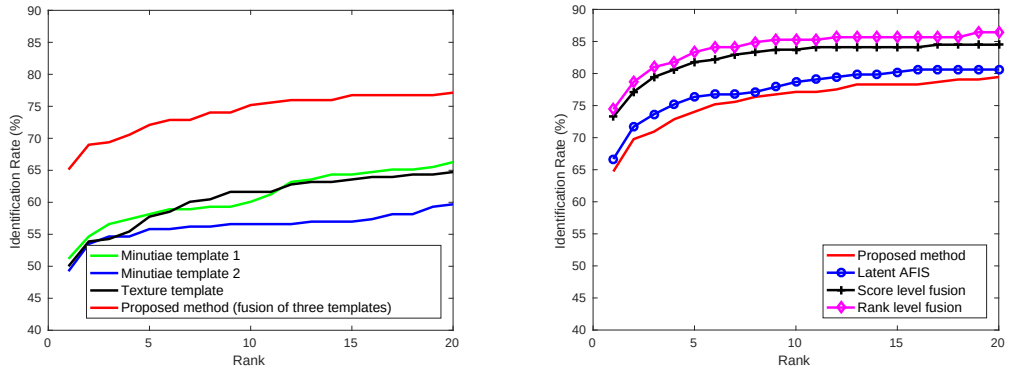
5.3 Benchmarking against COTS Latent AFIS

We benchmark the proposed latent recognition algorithm against one of the best COTS latent AFIS[11]

10. To request WVU latent fingerprint database, contact Dr. Jeremy Dawson (Email: Jeremy.Dawson@mail.wvu.edu)

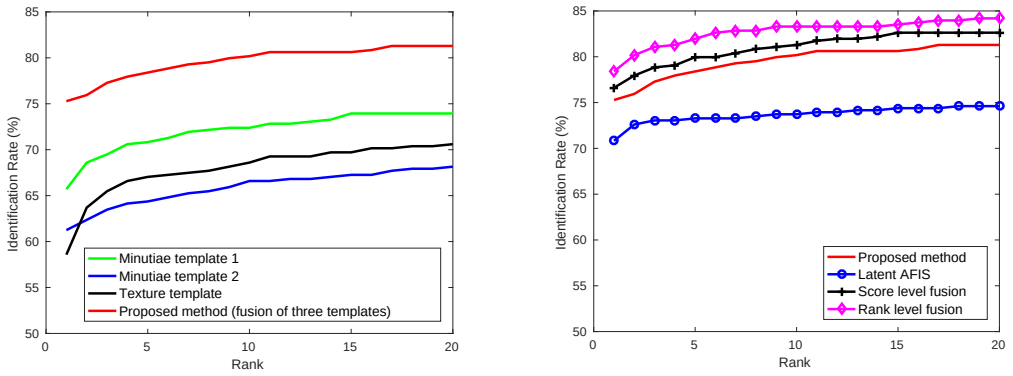
11. The latent AFIS used here is one of top-three performers in the NIST ELFT-EFS evaluations [43] [7]. Because of our non-disclosure agreement with the vendor, we cannot disclose the name.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b)

Fig. 12: Cumulative Match Characteristic (CMC) curves for NIST SD27 of (a) individual templates (minutia template 1, minutia template 2 and texture template) and their fusion, and (b) comparison of the proposed method with a COTS latent AFIS and score-level and rank-level fusion of the proposed method and COTS latent AFIS.



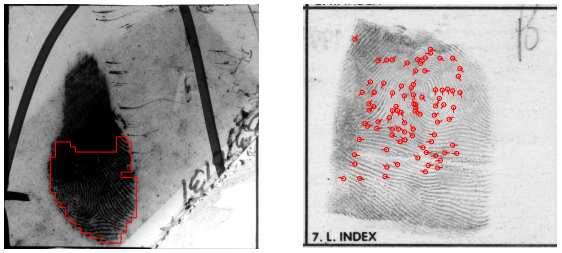
(a) (b)

Fig. 13: Cumulative Match Characteristic (CMC) curves for WVU DB of (a) individual templates (minutia template 1, minutia template 2 and texture template) and their fusion, and (b) comparison of the proposed method with a COTS latent AFIS and score-level and rank-level fusion of the proposed method and COTS latent AFIS.

as determined in NIST evaluations. The input to the latent AFIS are cropped latents using the same ROI as input to the proposed algorithm. While the COTS latent AFIS performs slightly better than the proposed algorithm (Rank-1 accuracy of 66.7% for COTS latent AFIS vs. 64.7% for the proposed algorithm) on NIST SD27, the proposed method outperforms the COTS latent AFIS on WVU DB (Rank-1 accuracy of 75.3% vs. 70.8%). See Figs. 12 (b) and 13 (b). The overall recognition performance can be further improved by a fusion of the proposed algorithms and COTS latent AFIS. Two fusion strategies, namely score-level fusion (with equal weights) and rank-level fusion (two top 100 candidates lists are fused using Borda count [45]) were implemented. Score level fusion of the COTS and the proposed algorithm results in significantly higher rank-1 accuracies, i.e., 73.3% on NIST SD27 and 76.6% on WVU DB. For NIST SD27 with a total of 258 latents, the score-level fusion leads to an additional 17 latents whose mates are now retrieved at rank-1 compared to the COTS latent AFIS alone. Rank-level fusion results in even better performance (Rank-1 accuracies of 74.4% on NIST SD27 and 78.4% on WVU DB).

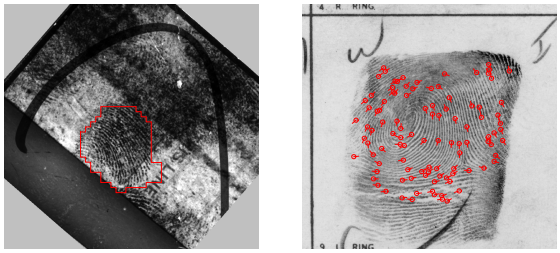
Note that rank-level fusion is preferred over score-level fusion when, for proprietary reasons, a vendor may not be willing to reveal the comparison scores. The CMC curves are shown in Figs. 12 and 13. Fig. 17 shows example latents whose true mates can be correctly retrieved at rank-1 by the proposed method, but the COTS latent AFIS was not successful. Although the two example latents from WVU DB (Figs. 17 (c) and (d)) have large friction ridge area, the latent AFIS outputs comparison scores of 0 between the latents and their mates. Apparently, the latent AFIS could not extract sufficient number of reliable minutiae in the latents where the ridges are broken. The proposed algorithm with its use of two different ridge flow estimation algorithms and dictionary-based and Gabor filtering-based enhancement, is able to obtain high quality ridge structures and sufficient number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) (b) (c) (d)

Fig. 14: A latent whose true mate was retrieved at rank-1 by minutiae template 1 but not by minutiae template 2 (rank-2,457). (a) Input latent with its ROI (G044 from NIST SD27), (b) mated reference print of (a) with overlaid minutiae, (c) minutiae set 1 of (a) overlaid on latent skeleton, and (d) minutiae set 2 of (a) overlaid on latent skeleton.



(a) (b) (c) (d)

Fig. 15: A latent whose true mate was retrieved at rank-1 by minutiae template 2 but not by minutiae template 1 (rank-2). (a) Input latent with its ROI (U277 from NIST SD27), (b) mated reference print with overlaid minutiae, (c) minutiae set of (a) 1 overlaid on latent skeleton, and (d) minutiae set 2 of (a) overlaid on latent skeleton.

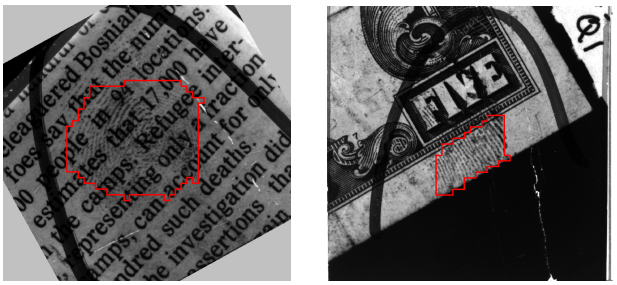
minutiae.

To compare the proposed ConvNet-based minutiae descriptor with MCC descriptor [41] which is a popular minutiae descriptor for reference prints, we replace the ConvNet-based descriptor in latent minutiae template 1 and reference print minutiae template by MCC descriptor. The rank-1 accuracies on NIST SD27 and WVU DB by comparing modified minutiae template 1 of latents against modified minutiae templates of 100K reference prints are only 21.3% and 35.2%, respectively. These accuracies are far lower than the accuracies of the proposed minutiae template 1 with learned descriptors based on ConvNet (rank-1 accuracies of 51.2% and 65.7% on NIST SD27 and WVU DB, respectively).

We also compare the proposed latent recognition algorithm with Paulino et al.'s algorithm [31], which uses manually marked minutiae and MCC descriptor. The rank-1 identification rates of the proposed method are about 20% and 32% higher than those reported in Paulino et al. [31] on NIST SD27 and WVU DB, respectively.

5.4 Computation Time

The algorithm was implemented in MATLAB and runs on a server with 12 cores @ 2.50GHz, 256 GB RAM and Linux operating system. Using 24 threads (MATLAB function: *parpool*), the average template extraction time (all three templates with three ConvNets for minutiae descriptor) per latent is 2,950 ms. Specifically, the average computation times for extraction



(a) (b)

Fig. 16: Example latents whose true mates were found at rank-1 by texture template but not by the minutiae templates. Reliable minutiae from these two latents could not be extracted due to (a) poor quality (U276 from NIST SD27) and (b) small friction ridge area (U292 from NIST SD27).

of minutiae set 1 and minutiae set 2 are 820 ms and 624 ms, respectively, and the average times for descriptor extraction of minutiae template 1, minutiae template 2 and texture template are 129 ms, 126 ms and 1,245 ms, respectively. The average computation times for minutiae template and texture template of rolled prints are 408 ms and 2,240 ms, respectively. Note that the template extraction for the rolled prints can be precomputed off-line. The average matching times for minutiae template 1, minutiae template 2 and texture template are 1.13 ms, 0.98 ms and 10.96 ms, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

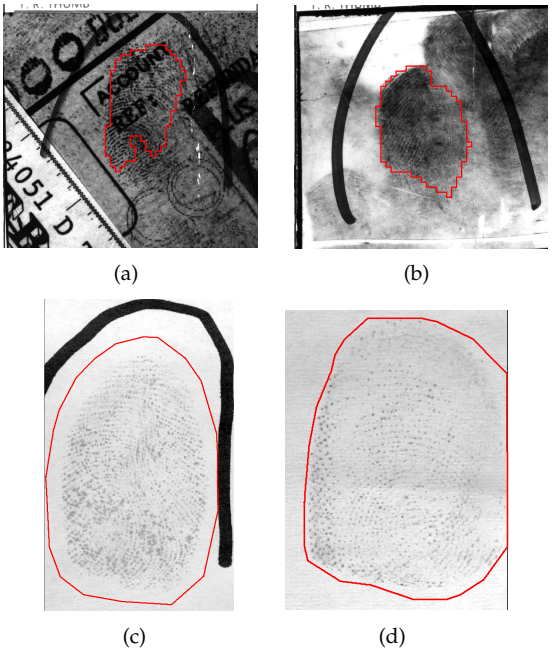


Fig. 17: Example of latent images which are correctly identified at rank-1 by the proposed method but not by a leading COTS latent AFIS. The retrieval rank of the true mate of (a) by the latent AFIS is 931, but for latents in (b), (c) and (d), their true mates could not be found because the comparison score was zero. Latents in (a) and (b) are from NIST SD27 whereas latents in (c) and (d) are from WVU DB.

6 Conclusions and Future Work

Latent fingerprints constitute one of the most important and widely used sources of forensic evidence in forensic investigations. Despite this, efforts to design and build accurate, robust, and fully automated latent fingerprint recognition systems have been limited. Only a handful of commercial companies are able to provide large-scale latent SDKs, but even they require significant time and effort of latent examiners in finding the true mate or a "hit" of a query latent. To our knowledge, open source literature does not contain any automated latent recognition method. The latent recognition problem is difficult due to poor ridge quality, severe background noise, small friction ridge area, and image distortion encountered in latent images.

We present an automated latent fingerprint recognition algorithm and benchmark its performance against a leading COTS latent AFIS. The contributions of this paper are as follows:

1) Three latent templates, namely, two minutiae templates and one texture template, are utilized. These templates extract complementary information from latents;
2) A total of 14 patch types are investigated for minutiae descriptors that are learned via a ConvNet. A systematic feature selection method shows that only 3 out of 14 patch types are needed to maintain the overall recognition accuracy at a significant savings in computation.
3) Second-order and third-order graph based minutiae correspondence algorithms are proposed for establishing minutiae correspondences.
4) Experimental results show that the proposed method performs significantly better than published algorithms on two benchmark databases (NIST SD27 and WVU latent DB) against 100K rolled prints. Further, our algorithm is competitive and complementary to a leading COTS latent AFIS. Indeed, a fusion of the proposed method and COTS latent AFIS leads to a boost in rank-1 recognition accuracy for both the benchmark latent databases.

Our algorithm for latent recognition can be further improved as follows.

1) ConvNet architectures, e.g., GoogeLeNet [46], should be considered to improve the recognition effectiveness.
2) Exploring the use of additional latent features, such as ridge count and singular points, to further boost the recognition performance.
3) Filtering strategies through a cascaded network of recognition engines should be studied to improve the system scalability for recognition against large scale reference set.
4) Acquiring a large collection of latents to train the ConvNet.
5) Improving the speed of feature extraction and comparison.

References

[1] B. T. Ulery, R. A. Hicklin, G. I. Kiebuzinski, M. A. Roberts, and J. Buscaglia, "Understanding the Sufficiency of Information for Latent Fingerprint Value Determinations," *Forensic Science International*, no. 1, pp. 99–106, 2013.
[2] A. K. Jain and J. Feng, "Latent fingerprint matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 88–100, 2011.
[3] K. Cao, E. Liu, and A. K. Jain, "Segmentation and enhancement of latent fingerprints: A coarse to fine ridge structure dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1847–1859, 2014.
[4] M. Hawthorne, *Fingerprints: Analysis and Understanding*. CRC Press, 2008.
[5] "NGI monthly face sheet," https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet/view.
[6] C. Watson, G. Fiumara, E. Tabassi, S. L. Cheng, P. Flanagan, and W. Salamon, "Fingerprint vendor technology evaluation: Evaluation of fingerprint matching algorithms," *NISTIR 8034*, 2012.
[7] M. D. Indovina, V. Dvornychenko, R. A. Hicklin, and G. I. Kiebuzinski, "Evaluation of latent fingerprint technologies: Extended feature sets (evaluation 2)," *Technical Report NISTIR 7859, NIST*, 2012.
[8] D. Ashbaugh, *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology*. CRC Press, 1999.
[9] B. T. Ulery, R. A. Hicklin, M. A. Roberts, and J. Buscaglia, "Interexaminer variation of minutia markup on latent fingerprints," *Forensic Science International*, vol. 264, pp. 89–99, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2818162, IEEE Transactions on Pattern Analysis and Machine Intelligence

[10] S. S. Arora, K. Cao, A. K. Jain, and G. Michaud, "Crowd powered latent fingerprint identification: Fusing afis with examiner markups," in *International Conference on Biometrics*, 2015, pp. 363–370.

[11] B. T. Ulery, R. A. Hicklin, J. Buscaglia, and M. A. Roberts, "Repeatability and reproducibility of decisions by latent fingerprint examiners," *PloS One*, vol. 7, no. 3, p. e32800, 2012.

[12] President's Council of Advisors on Science and Technology, "Forensic science in criminal courts: Ensuring scientific validity of feature-comparison methods."

[13] Committee on Identifying the Needs of the Forensic Sciences Community, National Research Council, "Strengthening forensic science in the united states: A path forward," https://www.ncjrs.gov/pdffiles1/nij/grants/228091.pdf.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[17] K. Cao and A. K. Jain, "Latent orientation field estimation via convolutional neural network," in *International Conference on Biometrics*, 2015, pp. 349–356.

[18] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh, "On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders," in *IEEE International Joint Conference on Biometrics*, 2014.

[19] Y. Tang, F. Gao, and J. Feng, "Latent fingerprint minutia extraction using fully convolutional network," *arXiv*, 2016.

[20] S. Meagher and V. Dvornychenko, "Defining AFIS latent print "lights-out"," *NISTIR 7811*.

[21] "NIST Special Database 27," http://www.nist.gov/srd/nistsd27.cfm.

[22] "Integrated pattern recognition and biometrics lab, West Virginia University," http://www.csee.wvu.edu/ ross/i-probe/.

[23] A. Sankaran, M. Vatsa, and R. Singh, "Latent fingerprint matching: A survey," *IEEE Access*, vol. 2, pp. 982–1004, 2014.

[24] S. Karimi-Ashtiani and C.-C. Kuo, "A robust technique for latent fingerprint image segmentation and enhancement," in *IEEE International Conference on Image Processing*, 2008, pp. 1492 –1495.

[25] N. J. Short, M. S. Hsiao, A. L. Abbott, and E. A. Fox, "Latent fingerprint segmentation using ridge template correlation," in *4th International Conference on Imaging for Crime Detection and Prevention*, 2011, pp. 1–6.

[26] J. Zhang, R. Lai, and C.-C. Kuo, "Adaptive directional total-variation model for latent fingerprint segmentation," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1261–1273, 2013.

[27] H. Choi, M. Boaventura, I. A. G. Boaventura, and A. K. Jain, "Automatic segmentation of latent fingerprints," in *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems*, 2012.

[28] X. Yang, J. Feng, and J. Zhou, "Localized dictionaries based orientation field estimation for latent fingerprints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 955–969, 2014.

[29] M. Liu, X. Chen, and X. Wang, "Latent fingerprint enhancement via multi-scale patch based sparse representation," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 6–15, 2015.

[30] K. Cao, T. Chugh, J. Zhou, E. Tabassi, and A. K. Jain, "Automatic latent value determination," in *International Conference on Biometrics*, 2016.

[31] A. A. Paulino, J. Feng, and A. K. Jain, "Latent fingerprint matching using descriptor-based hough transform," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 31–45, 2013.

[32] R. P. Krish, J. Fierrez, D. Ramos, J. Ortega-Garcia, and J. Bigun, "Pre-registration for improved latent fingerprint identification," in *22nd International Conference on Pattern Recognition*, Aug 2014, pp. 696–701.

[33] X. Si, J. Feng, B. Yuan, and J. Zhou, "Dense registration of fingerprints," *Pattern Recognition*, vol. 63, pp. 87 – 101, 2017.

[34] N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, no. 11, pp. 1657–1672, 1995.

[35] S. Yoon and A. K. Jain, "Longitudinal study of fingerprint recognition," *Proceedings of the National Academy of Sciences*, vol. 112, no. 28, pp. 8555–8560, 2015.

[36] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1891–1898.

[37] "MatConvNet," http://www.vlfeat.org/matconvnet/.

[38] S. Chikkerur, A. N. Cartwright, and V. Govindaraju, "Fingerprint enhancement using STFT analysis," *Pattern Recognition*, vol. 40, no. 1, pp. 198–211, 2007.

[39] X. Fu, C. Liu, J. Bian, J. Feng, H. Wang, and Z. Mao, "Extended clique models: A new matching strategy for fingerprint recognition," in *International Conference on Biometrics*, 2013.

[40] O. Duchenne, F. Bach, I. S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2383–2395, Dec 2011.

[41] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2128–2141, 2010.

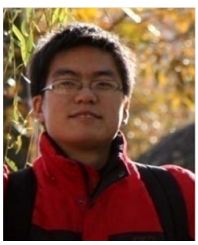
[42] "NIST Special Database 14," http://www.nist.gov/srd/nistsd14.cfm.

[43] M. D. Indovina, R. A. Hicklin, and G. I. Kiebuzinski, "Evaluation of latent fingerprint technologies: Extended feature sets (evaluation 1)," *Technical Report NISTIR 7775, NIST*, 2011.

[44] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[45] A. A. Ross, K. Nandakumar, and A. Jain, *Handbook of Multibiometrics*. Springer US, 2006.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, 2014.



Kai Cao received the Ph.D. degree from the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently a Post Doctoral Fellow in the Department of Computer Science & Engineering, Michigan State University. He was affiliated with Xidian University as an Associate Professor. His research interests include biometric recognition, image processing and machine learning.

Anil K. Jain is a University distinguished professor in the Department of Computer Science and Engineering at Michigan State University. His research interests include pattern recognition and biometric authentication. He served as the editor-in-chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence and was a member of the United States Defense Science Board. He has received Fulbright, Guggenheim, Alexander von Humboldt, and IAPR King Sun Fu awards. He is a member of the National Academy of Engineering and foreign fellow of the Indian National Academy of Engineering.