Joint Estimation of Pose and Singular Points of Fingerprints

Qihao Yin, Jianjiang Feng, *Member, IEEE*, Jiwen Lu, *Senior Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*— Fingerprint pose estimation is a challenging problem since the pose is not defined by salient anatomical features and fingerprint images usually suffer from noise and small area. In this article, we proposed a method for joint estimation of pose and singular points of fingerprints, with the expectation that the pose and singular points can improve each other. By virtue of that singular points can be located accurately, we hope to improve the accuracy of pose estimation. Meanwhile, the robustness of pose estimation can improve the anti-noise performance of singular point detection. To achieve this, we propose a multi-task deep neural network, which contains a feature extraction body and two estimation heads for singular point and pose respectively. The proposed network can deal with various types of fingerprints, including plain, rolled and latent fingerprints. Experiments on four databases (NIST SD4, SD14, SD27 and FVC2004 DB1A) show that (1) the estimated poses and detected singular points are close to manual annotations despite of different image qualities; (2) the estimated poses for mated fingerprint pairs are consistent; and (3) the proposed pose estimation method outperforms state-of-the-art methods while utilized as pose constraint for a fingerprint indexing algorithm.**

***Index Terms*— Fingerprint recognition, pose estimation, singular points detection, deep neural networks, multi-task learning.**

I. Introduction

AUTOMATIC fingerprint recognition systems are based on extraction and matching of various features in fingerprints [1]. These features are usually grouped into three levels from global to local. At the global level, namely level-1, the features are used to describe how the ridges run generally, like orientation fields and singular points. At the local level, namely level-2, the features can describe the fingerprint more precisely, such as ridge skeletons and minutiae. Image with a quite high resolution is needed (at least 500 ppi) for such features. At the finest level, namely level-3, features that describe the fingerprint even more precisely can be extracted, such as ridge contours and pores, and we need a image with an even higher resolution (e.g. 1000 ppi) to extract such features. In general, these features of different levels are extracted in a hierarchical order, from global to local [2].

Compared to the common features above, the pose of fingerprints is a relatively under-researched topic [3]. The pose of a fingerprint is given by the position of fingerprint center (x,y) and the fingerprint direction θ (see examples in Fig. 1). It defines the internal coordinate system to describe where the whole fingerprint locate at a very global level, thus it can be considered to be a level-0 feature of a fingerprint. One of the oblivious usages of fingerprint pose is to improve the speed and accuracy of fingerprint matching, which is very important for large-scale fingerprint identification systems [4]–[6]. Another usage of fingerprint pose which have received less attention is that after alignment by the pose, statistical characteristics based on different positions of fingerprint can be used to improve the anti-noise performance of extracting features at finer levels [3].

In fact, for most other biometrics like face [7], palm [8], iris [9] and gait [10], pose estimation is a key step in recognition. Most pose estimation methods for these biometrics are based on landmarks, such as eyes, nose tips or joints. However, unlike other biometrics, fingerprints lack anatomical landmarks. Fingerprint direction can be defined to be normal to the knuckle (as shown in Fig. 1), which is usually not available in fingerprint images however. Therefore almost all fingerprint matching algorithms align two fingerprints relatively during the matching process, called pre-registration, rather than align all fingerprints to a standard coordinate system, which is a common practice in face recognition. Also, most feature extraction algorithms do not utilize the statistical characteristics in different regions of fingerprints.

Yang *et al.* proposed a definition of fingerprint pose [3], which utilized several information in an integrated way, including knuckles, the orientation fields near knuckles and singular points. Based on this definition, several researchers proposed methods for fingerprint pose estimation [3], [6], [11] and utilized the estimated pose for orientation field estimation and fingerprint indexing. However, these methods fail to consider singular points explicitly, which is the base in the definition of pose. As a result, poses estimated by these methods

Manuscript received April 4, 2020; revised September 17, 2020; accepted October 24, 2020. Date of publication November 9, 2020; date of current version December 11, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61976121, Grant 61622207, and Grant 61527808; and in part by the Shenzhen Fundamental Research Fund (subject arrangement) under Grant JCYJ20170412170438636. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julian Fierrez. *(Corresponding author: Jianjiang Feng.)*

Qihao Yin, Jiwen Lu, and Jie Zhou are with the Beijing National Research Center for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: yinqh14@mails.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

Jianjiang Feng is with the Beijing National Research Center for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China, and also with the Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China (e-mail: jfeng@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIFS.2020.3036803

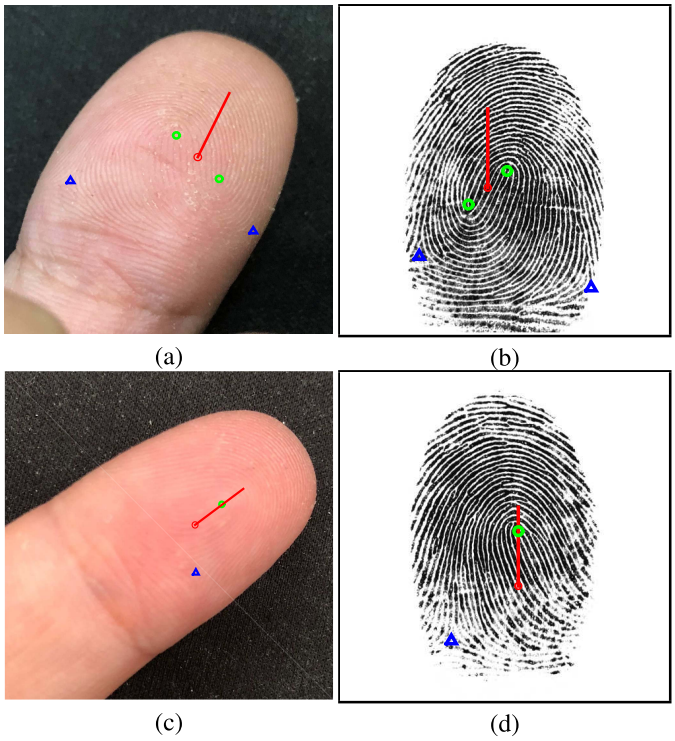

Fig. 1. Two examples of pose and singular points of fingerprints. (a) and (c): photograph of finger obtained by a smartphone camera; (b) and (d): corresponding fingerprint image obtained by an optical fingerprint sensor. The pose and singular points are manually labeled. The red circle indicates the center of fingerprint, the red line points to the direction, the green circles indicate core points and the blue triangles indicate delta points. Note that photograph and impression of a fingerprint are flipped.

are not precise even for some good quality fingerprints and may be inconsistent among different images of the same finger.

There is a strong correlation between fingerprint pose and singular points. For example, the definition of fingerprint pose in [3] is based on singular points, except for plain arch fingerprints. In addition, when fingerprint pose is known, the distribution of singular points is regular [12]. Also, the strengths and weaknesses of singular points and pose estimation can be complementary. Conventional singular point detection methods are very sensitive to local noise and mutilation, while the pose of a fingerprint can be estimated by the global information. On the other hand, the pose, which is hard to estimate precisely using only global information, can be inferred more accurately by the known singular points.

In this article, considering the relationship between singular points and pose of a fingerprint, we propose a network to solve singular point detection and pose estimation at the same time. To the best of our knowledge, the proposed network architecture is the first method that can deal with various types of fingerprint impressions, including plain, rolled and latent fingerprints. In particular, thanks to the fully-connected (FC) layers, the proposed network can deal with partial fingerprints whose centers are invisible in the image. Experimental results show that the proposed method outperforms state-of-the-art methods in four different databases, covering rolled fingerprints (NIST SD4 and SD14), plain fingerprints (FVC2004 DB1A), and latent fingerprints (NIST SD27).[1] We evaluate our results in three aspects: (1) the deviation from ground truth; (2) the the deviation of mated minutiae pairs after registered to the same coordinate system by estimated pose; and (3) the fingerprint indexing results by the same algorithm with different pose constraint. All the experiments show the improvements of the proposed method. More detailed explanation will be discussed in the experiment part. All the estimated poses used for testing the algorithm are uploaded along with the evaluation codes. Researchers can get the codes and the results as a benchmark for fingerprint pose estimation at https://github.com/luffyyqh/FPSNet.

The rest of this article is organized as follows: Section II reviews related works including singular point detection, pose estimation methods and deep learning based extraction methods for other fingerprint features. The proposed method is then illustrated in Section III, followed by implementation details explained in Section IV. Section V shows qualitative and quantitative experiment results. And finally, we summarize this article and come to the conclusion in Section VI.

II. Related Works

A. Pose Estimation

A main application of fingerprint pose is to serve as a common coordinate system to facilitate fingerprint matching. Before [3], researchers proposed several definitions of pose by taking other points as center and corresponding detection methods. We first review these methods in the following part, and then review the pose estimation methods based on the definition of [3]. Finally we review representative relative registration approaches.

1) Point Detection Based Approaches: The first category of pose estimation approaches is based on detecting certain special points which are universal in fingerprints of various patter types. Singular points are most commonly used in these methods. For example, Bazen and Gerez [13], Nilsson and Bigun [14] use the upper core point as reference point. Also, some other types of points are proposed for reference. Liu *et al.* [15] locate the reference point by multi-scale analysis of the orientation consistency. Focal point, defined as the crossing point of normal lines to ridges, is used in [16]. When dealing with rolled fingerprints with large area, the barycenter of the foreground region is also used by some researchers [12]. These approaches require good image quality to detect reference point precisely, thus can not deal with latent fingerprints. Yoon *et al.* [17] detect the point with maximum curvature of convex ridge structure, where the orientation fields are reconstructed by minutiae. Deerada *et al.* [18] proposed a systematic feedback approach to remedy the corrupted information and can estimate a reliable pose of latent fingerprints. These methods can reduce the impact of noise, but they still rely on the features around the fingerprint center, thus they cannot deal with fingerprints with missing center region properly.

[1] NIST Special Databases 4, 14 and 27 were previously publicly available.

2) Object Detection Based Approaches: These methods take the whole fingerprints into consideration and try to learn where the center locate at and to what direction the fingerprints point to. Su *et al.* [6] trained a set of SVM classifiers for different directions. When a sliding window slides across the whole fingerprint image, the histogram of ridge orientations are extracted as features and the classifiers attempt to tell whether it is a fingerprint of corresponding direction and where the center is. Ouyang *et al.* [11] is the first to apply deep learning to the topic of fingerprint pose estimation. They treat the pose estimation as a task of object detection, where fingerprints with different directions are treated as different objects, and train a faster R-CNN [19] to solve it. Similar with Ouyang *et al.* [11], Schuch *et al.* [20] trained a CNN to estimate a fingerprint's angle in an unsupervised way. These methods consider the whole fingerprint image and then extract features, thus are more robust than point detection based methods. However, they still cannot solve the problem of missing area, which is common in latent fingerprints. Yang *et al.* [3] proposed a pose estimation method based on Hough transform. In the off-line learning step, a set of prototype orientation patch dictionary along with the spatial distribution is learnt. In the on-line step, the initial orientation field of the query fingerprint is cut into patches and vote for the location of center based on the learnt dictionary. The direction is estimated by rotating the query fingerprint and finding the maximum voting score. Cao and Jain [5] first adopt the method in [17] to detect the fingerprint center and then searching the optimal direction by minimizing the difference between query fingerprint orientation field and one of the orientation field dictionary elements. Such idea is also adopted by Yin *et al.* [21], who also used orientation field for pose estimation. The initial orientation field of the query fingerprint is used to find the nearest reference one in a pre-aligned database by exhaustive search, and the corresponding transformation is recorded to get the pose of the query fingerprint. These approaches can overcome the limits of detection based methods since even the center region is missing, the other regions of fingerprint can make up for it and provide some information. However, the features are handcrafted, like orientation fields and spatial distribution. The descriptive ability can be improved by deep learning method, which is exactly what we propose in this article. The proposed method can also be classified into this category.

3) Relative Registration Approaches: Relative registration is aimed at computing the spatial transformation between a pair of fingerprints. As a key step in fingerprint matching, many relative registration algorithms have been proposed, such as orientation field based registration [22], minutiae-based registration [23], and ridge-based registration [24]. Pose estimation belongs to absolute registration, which operates on a single fingerprint rather than a pair of fingerprints.

B. Singular Point Detection

Singular points are Level-1 features of fingerprints where the orientation fields change rapidly, they are invariant to translation, rotation, enlargement, and shrinking so that can be used in fingerprint pattern classification and alignment for matching. Here we simply introduce some famous conventional methods of singular points detection.

1) Rules Based Approaches: The rules based methods focus on the orientation fields around a point to judge whether it is a singular point or not. The most common method is the Poincare index [25]. It takes the values π, $-\pi$ and 0 for cores, deltas and ordinary points respectively. Bazen and Gerez [13] made use of 2-dimensional filters based on Poincare index for singular points detection. Zhou *et al.* [26] proposed a novel DORIC feature to remove the fake singular points detected by Poincare index. Nilson and Bigun [14] adopted complex filtering directly to the orientation fields in multiple resolutions to detect singular points. Tiwari and Gupta [27] located the singular points by devising the concept of meandering energy potential on the skeleton image of fingerprint. Khan [28] proposed a CNN model to classify fingerprint into five different classes, and then fed fingerprint patches to a mini classifier to extract the singular points.

2) Model Based Approaches: Another category of singular point detection methods are model based. Instead of using local information around a certain point, these methods build different models and try to calculate the parameters, which can further locate where the singular points are, such as Zero-pole model [29] or FOMFE [30].

These methods also rely on handcrafted features, which are sensitive to noise. Fortunately, we find some conventional methods can be easily adapted to deep learning method, for example, the complex filtering in [14] can be seen as a set of convolution layers. Thus we propose an deep neural network to extract singular points in this article.

C. Feature Extraction by Deep Neural Networks

In the recent years, deep learning has shown its great ability in many area of computer vision. Thus, an increasing number of researchers have tried to use deep neural networks to fingerprint recognition, especially in the feature extraction step to solve the problems where traditional algorithms failed. Cao *et al.* [31] proposed an orientation field estimation method, following the idea of dictionary based approach [32] but replacing the look-up step by a convolution neural network (CNN). Schuch *et al.* [33], [34] treated the problem of orientation field estimation as a classify task and adopted a fully convolution network (FCN) and deep expectation for it. Tang *et al.* [35] proposed a unified network called FingerNet for minutiae extraction, along with orientation field estimation, segmentation and fingerprint enhancement. Nguyen *et al.* [36] improved FingerNet and proposed MinutiaeNet by moderating the backbone of the network and design a new network to further judging the quality of extracted minutiae. These deep learning based methods indicate great potential of utilizing deep neural networks (DNNs) to the fingerprints, which is an important motivation of the proposed method.

III. Proposed Method

A. Definition of Fingerprint's Pose

Unlike minutiae and singular point, fingerprint pose is not a local feature, whose definition is not oblivious. In this

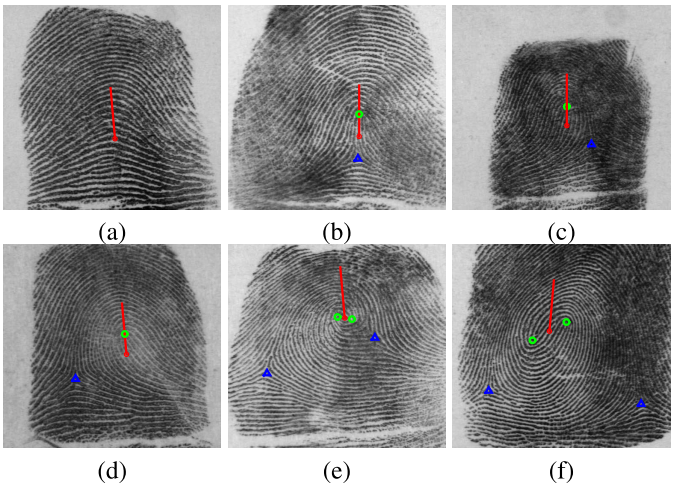

Fig. 2. Examples of the definition of fingerprint's pose. The red circle represents the defined center and the red line points at the defined direction. the green circle and blue triangle represents the core and delta points. (a)-(f) show six kinds of fingerprints with different number of singular points, separately. (a): a plain arch fingerprint with no singular points; (b): a tented arch fingerprint with one core and one delta; (c): a left loop fingerprint with one core and one delta; (d): a right loop fingerprint with one core and one delta; (e): a whorl fingerprint with two cores and two deltas; (f): a twin loop fingerprint with two cores and two deltas. The fingerprints are from NIST SD4.

work, to make comparison between other state-of-the-art methods more convincing, we follow the definition proposed by Yang *et al.* [3], which is also adopted in [6], [11]. The fingerprint's direction is perpendicular to the knuckle or the ridges located at the bottom area of the fingerprint, while the fingerprint's center depends on singular points available (Fig. 2):

1) For fingerprints with no singular points, namely plain arch fingerprints, we first locate the uppermost ridge which is parallel with the knuckle, and then draw a perpendicular to it crossing the point with maximum curvature. The center of the fingerprint is defined as the midpoint of the point with maximum curvature and the foot of the perpendicular (Fig. 2(a)).
2) For fingerprints with one core and one delta points, such as tented arch, left loop and right loop fingerprints, we first draw a line crossing the core point and parallel with the direction of the fingerprint, and then draw a perpendicular to it crossing the delta point. The center of the fingerprint is defined as the midpoint of the core point and the foot of the perpendicular (Fig. 2(b)-(d)).
3) For fingerprints with two core points, such as whorl and twin loop fingerprints, the center is defined as the midpoint of the two core points (Fig. 2(e)-(f)).

Although the ideal definition of fingerprint pose should be based on the 3D shape of the finger, which is not available, we have to rely on the fingerprint image. Thanks to the strong correlation between fingerprint ridge pattern and the finger shape [37], this definition is reasonable and have a high consistency of manual annotation.

B. Network Architecture

The basic idea of pose estimation is inspired by the Hough-Voting method [3]. Suppose that we have extracted the orientation field from the query fingerprint, which has been studied by several researchers, the orientation patches from every region will vote for the fingerprint center. In other words, each pixel in the feature map should have a connection to the output. Thus a fully-connected (FC) module is adopted.

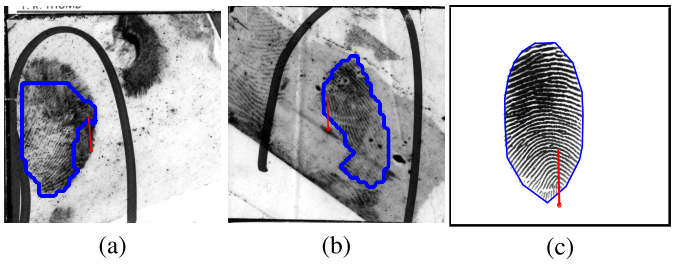

Fig. 3. Examples of missing center in a fingerprint. The red dot is manually labeled center and the red line point at the direction (ground truth). The blue line indicates the contour of the fingerprint. (a) and (b) are latent fingerprints from NIST SD27 while (c) is a plain fingerprint from FVC2004 DB1A. Note that the center is out of the area of the fingerprint, thus it is difficult to learn from the neighbour to judge whether a pixel is the center or not.

There is another reason for us to choose FC layers instead of fully convolution networks (FCNs), which is more frequently-used in computer vision and easier to train. Due to lack of regions of latent fingerprints, the center may be missing in the fingerprint image. Subject to limited receptive fields of convolution layers, FCNs cannot deal with such cases well. Fig. 3 shows some examples.

It is much easier to deal with singular points. Nilson and Bigun's method [14] has shown the prospect that singular points can be calculated by convolution of the orientation fields with filters, which can be implemented as a convolution module into the network.

Based on the analysis above, we propose a deep neural network for joint pose estimation and singular point detection. The network consists of four main parts: the feature extraction body part aims to extract features at different locations of a fingerprint, the singular point head calculates the probability of cores and deltas, the attention module selects where the features are meaningful for pose estimation, and the pose estimation head simulates the voting procedure by fully connected layers. The architecture of the whole network is illustrated in Fig. 4.

1) Body of Proposed Network: There are several existing networks designed for fingerprint orientation field estimation and some have achieved excellent results. In this work, we choose the orientation estimation part of FingerNet [35] as the body of our proposed network. Specifically, the feature extraction body has 3 conv-pooling blocks, each block contains a few convolutional blocks followed by a max-pooling layer and each convolutional block consists of a convolution layer, a BatchNorm layer [38] and a PReLU layer [39]. After that, ASPP-Net [40] is adopted for multi-scale solution. And at last, a N-dimension feature map is regressed as the final estimation.

2) Singular Point Detection Head: This part aims to calculate the probability of singular points from the features extracted by the body part. It contains two heads, one for cores and the other for deltas. Each head has 2 convolutional blocks and a 1×1 convolution layer, followed by a sigmoid

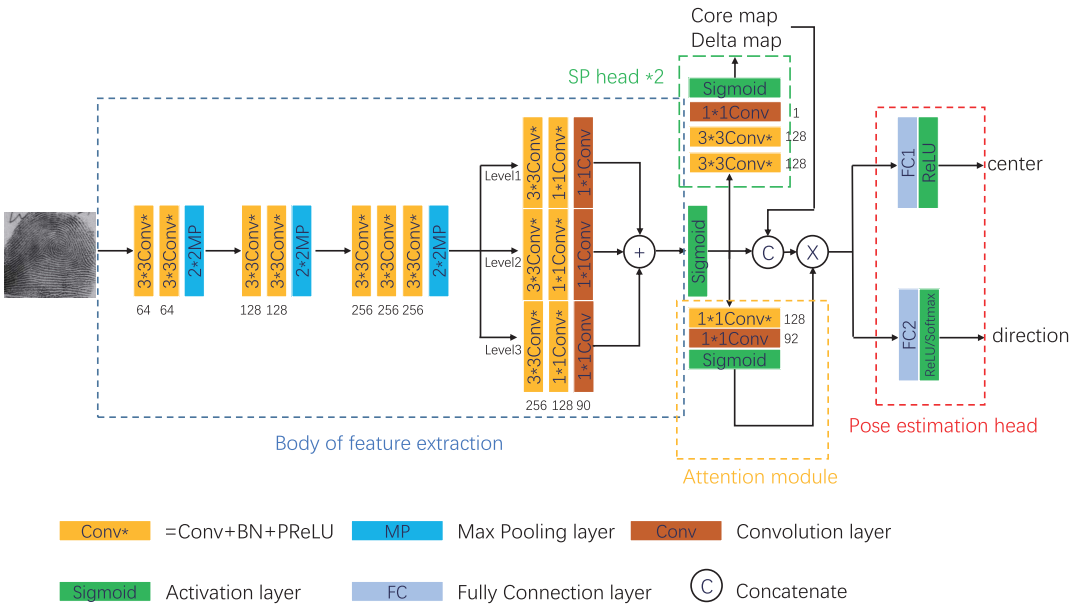

Fig. 4. The flowchart of our proposed network. There are four main modules: the body of feature extraction, two SP heads, the attention module and the pose estimation head. Blocks with different colors refer to different layers, the numbers under convolution layers indicate the number of channels.

layer. Each head outputs a heatmap, whose resolution is 1/8 of the fingerprint image and is the same as the feature map from the body. The values in the heatmap range from 0 to 1, representing the probability that there is a singular point in the corresponding 8 × 8 block in the fingerprint image. Also, the heatmaps are combined with the feature map from the body part and be sent to the final pose estimation head.

3) Attention Model: Before voting, an additional attention model part [41] is adopted to give foreground pixels more weight while voting. Although FingerNet also has a segmentation part that can be utilized for the same goal, the performance of segmentation by FingerNet is not good enough for latent fingerprints, and that may lead to wrong pose estimation. The attention model consists of a convolutional block described above, followed by a conv layer and an activation function. The output of attention model is used as a mask and multiplied with the feature map given by the body.

4) Pose Estimation Head: Two fully-connected (FC) layers are adopted to estimate the location of the center and the direction separately. The native idea is to treat the problem as a classification task. The output should be a heatmap $A_c(x, y)$ for finger center and a heatvector $A_d(\theta)$ for finger direction. Unfortunately, it is difficult to utilize the fully-connection directly to get a heatmap indicating center of the input fingerprint due to very high dimensionality. Suppose the image size of the fingerprint is 480 × 512, the size of the feature map given by the body will be 60 × 64 × 90. In case that the size of heatmap is 8 times smaller than the input image, the size will also be 60 × 64. Coupled with 13 directions, the number of parameters in the FC layer will be over 100 billion, which is impossible to train. There must be a way to reduce the dimension.

Given the heatmap and the heatvector, we can calculate the final estimation by an expectation way:

$$(\hat{x}, \hat{y}) = \sum_{x,y} (x * A_c(x, y), y * A_c(x, y)) \tag{1}$$

$$\hat{\theta} = \sum_{\theta} \theta * A_d(\theta). \tag{2}$$

where $A_c(x, y)$ indicates the heatmap of center and $A_d(\theta)$ indicates the heatvector. (x, y) are the coordinates across the heatmap while θ is the angle for each binned value in the heatvector. $(\hat{x}, \hat{y})$ and $\hat{\theta}$ are calculated center's coordinates and the angle of the fingerprint by expectation, respectively.

Note that all the computations are linear, we can merge the parameters in FC layers to reduce dimensions. As a result, we define the task of pose estimation as a regression task. The output layer for center estimation has 2 neurons, representing the coordinate of the center, while the output layer for direction estimation has 1 neuron.

IV. Implementation Details

A. Data Preparation

Since FC layers are utilized in the proposed network, the size of input image is fixed. However the image size of different database differs. The images size in NIST SD4 is fixed at 480 × 512, while the image size in NIST SD14 and NIST SD27 is fixed at 768 × 800 and the image size in FVC2004 DB1A is fixed at 480 × 640. Thus we select and manually label 2000 fingerprints in NIST SD4 for training and design a simple cropping strategy for fingerprints from other databases:

1) Gaussian Blur with a quite large kernel (empirically 19 × 19) is carried out to the fingerprint image.
2) The blurred image is then processed with open operation (an operator in grayscale morphology, the kernel size is 5 × 5) to fill holes.

TABLE I
THE DATABASES USED IN THE EXPERIMENTS

Database	Contents	Fingerprint Type	Usage
NIST SD4	2000 file fingerprints 2000 search fingerprints	Rolled	Training Testing
NIST SD14	Last 2700 pairs of fingerprints	Rolled	Testing
NIST SD27	All 258 pairs of fingerprints	Rolled and Latent	Testing
Hisign Latent Database	200 latent fingerprints	Latent	Training
FVC2004 DB1A	All 800 fingerprints	Plain	Testing

3) The centroid of the processed image is calculated and the original image is cropped to the fixed size (480 × 512) centered on the calculated centroid.

For latent fingerprints, we use the latents in Hisign Latent Database for training. Since some latent fingerprints may miss their singular points due to noise background or being partial, we first estimate the pose of mated rolled fingerprints, and transform the pose to latent fingerprints by Thin Plate Spline (TPS) [42] with manually marked mated minutiae as controlling points. Manual adjustment is followed for a more accurate ground truth. 100 latents in total are manually marked in this way. Note that although the image size differs between Hisign Database (512 × 512) and NIST SD27 (768 × 800), the same cropping strategy cannot be used on latents in NIST SD27 because of the poor quality. We cropped the latents in NIST SD27 to 512 × 512 manually to ensure the fingerprint is contained in the cropped image.

Data augmentation is also used while training. We randomly shift and rotate the training images at each epoch of training. The image is shifted in range of 1/4 length and width of image and rotated in range of -30 to 30 degrees.

B. Training

Thanks to that the body of proposed network is pretrained by Tang *et al.* [35] and the pretrained model is open source, the training procedure is greatly simplified. We directly train the whole network in the end-to-end way.

L2-Loss is chosen as the loss function of regression output for pose estimation and Cross Entropy Loss is chosen for the singular point detection heads.

$$L_{all} = L_{l2}(\hat{pose}, pose) + \lambda * (L_{CE}(\hat{h_c}, h_c) + L_{CE}(\hat{h_d}, h_d)) \tag{3}$$

where $\hat{pose}$ is the output of pose estimation head, $\hat{h_c}$ and $\hat{h_d}$ are heatmaps of cores and deltas output by SP estimation heads, while the others represent ground truths.

Adam optimizer [43] is adopted for training. The hyper paremetres are set to: $L_r = 1e-4$, $\beta = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, empirically.

C. Prior Knowledge for Singular Point Detection

Though both singular point and minutiae detection can be seen as key point detection, there is still some difference between them. One important feature of singular points is that one fingerprint have only up to two pairs of singular points. Thus, instead of setting threshold for point detection, we can improve the results by adding a simple prior knowledge in the following way. Two pixels with the highest probability are selected from the heatmaps of cores and deltas, the probability is c_1, c_2, d_1, d_2 respectively. The threshold is set to T. We calculate the total probability that the query fingerprint has 0, 1 and 2 pairs of singular points:

$$\begin{aligned} P_0 &= \sum_i (T - c_i) + (T - d_i) \\ &= 4T - c_1 - c_2 - d_1 - d_2 \\ P_1 &= (c_1 - T) + (d_1 - T) + (T - c_2) + (T - d_2) \\ &= c_1 - c_2 + d_1 - d_2 \\ P_2 &= \sum_i (c_i - T) + (d_i - T) \\ &= c_1 + c_2 + d_1 + d_2 - 4T \end{aligned} \tag{4}$$

The highest probability P is selected as the final result. In other words, no singular points are detected if P_0 is biggest, one core and one delta with the highest probability are detected if P_1 is biggest, and both pixels are detected if P_2 is biggest. Also, for latent fingerprints, due to lack of regions, such prior knowledge is not valid. The traditional threshold strategy is used.

V. EXPERIMENTS

In order to evaluate the proposed method quantitatively, we carry out several experiments to compare our result with state-of-the-art ones in different aspects. Firstly, we compare the deviation between estimated pose and the ground truth. Secondly, we align the mated fingerprints to the same coordinate system by estimated pose and compare the deviation between mated minutiae. And thirdly, we utilize a fingerprint indexing algorithm with pose constraint [6] to evaluate whether an estimated pose is precise or not. We also evaluate the singular point detection results by comparing with the commercial SDK VeriFinger, and then analyze the relationship between pose estimation and singular point detection. The databases used in the experiments are shown in Table I. Some examples are shown in Fig. 5.

A. Deviation From Ground Truth

For rolled fingerprints, we compare our method with two state-of-the-art methods proposed by Su *et al.* [6] and Ouyang *et al.* [11] First, we compare the pose estimation results qualitatively. Fig. 6 shows some examples on different fingerprints. As observed, the pose estimated by the proposed method is nearer to the ground truth, the direction is also more precise.

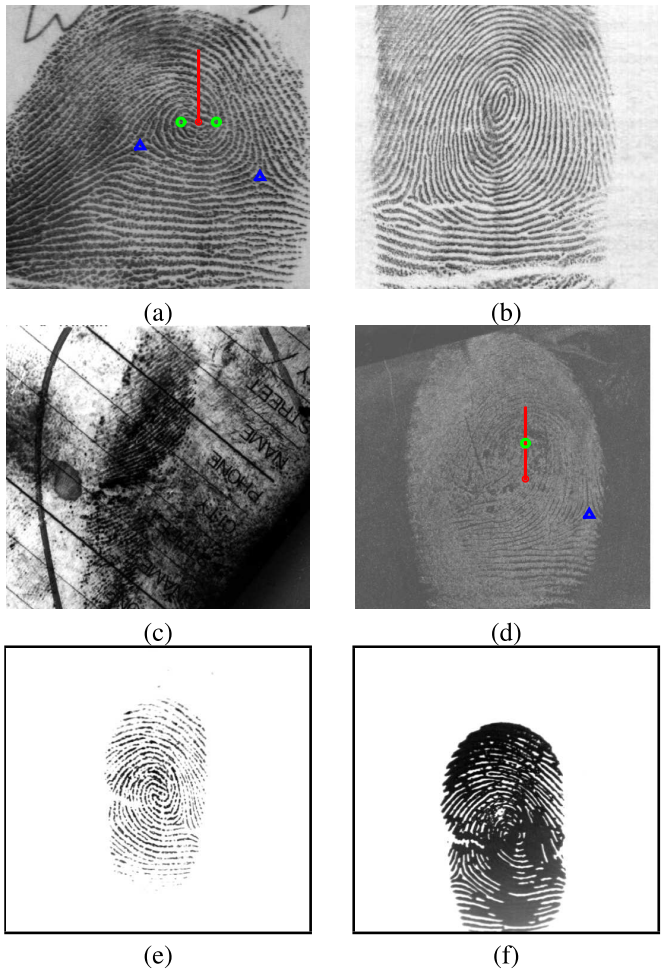

Fig. 5. Fingerprints from different databases: (a) NIST SD4; (b) NIST SD14; (c) NIST SD 27; (d) Hisign Latent Database; (e)-(f) Different impressions of the same fingerprint in FVC2004 DB1A. Fingerprints used for training are shown along with the ground truths. The red dot indicates the center while the line point at the direction. The green circles indicate cores and the blue triangles indicate deltas.

For plain fingerprints, Ouyang's method seems to fail on fingerprints in FVC2004 as they mentioned in [11], so we only compare our methods with Su's method. Fig. 7 shows some examples. The advantage of the proposed method is still obvious.

To make the comparison more convincing, we manually labeled the ground truth pose of last 50 pairs of fingerprints in NIST SD14 and calculate the deviation of the estimated pose. The results are shown in Table II. The improvement of estimated center is obvious, the mean deviation drops to 15.54 pixels, almost half the state-of-the-art method. Though the improvement of estimated direction is not as obvious as the estimated center, the mean deviation still drops over 1 degree.

B. Evaluation by Mated Minutiae Pairs

For a pair of mated fingerprints, the mated minutiae should be close to each other after registration to the standard coordinate system. Thus, we can use the mean deviation of mated minutiae pairs to evaluate the performance of different pose estimation methods.

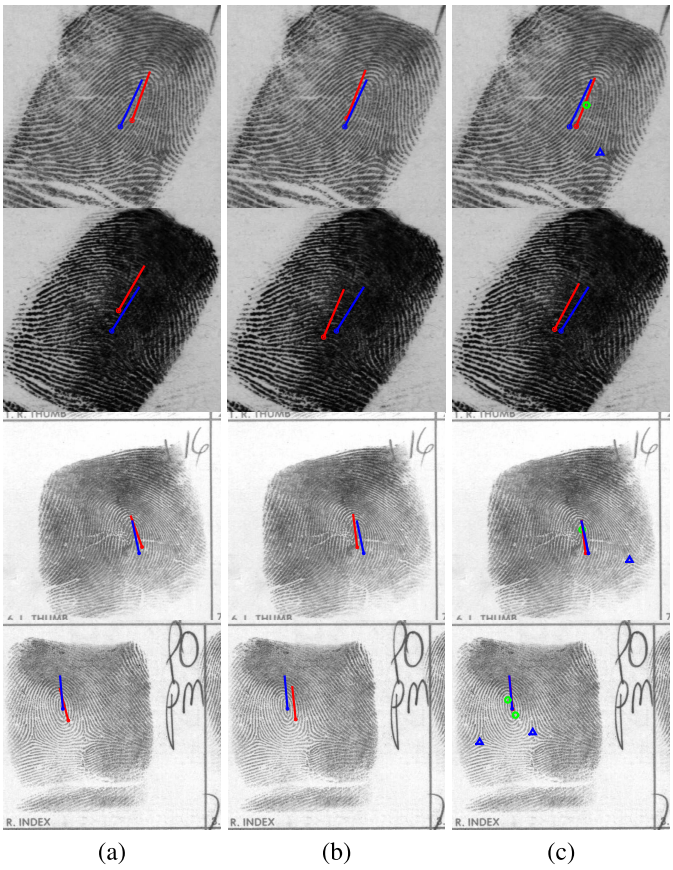

Fig. 6. The results of three methods for pose estimation on rolled fingerprints: (a) Su *et al.* [6]; (b) Ouyang *et al.* [11] and (c) proposed method; The top two rows are from NIST SD4 and the bottom two rows are form NIST SD14. The red circle indicates the estimated center while the red line points at the estimated direction. We also show the singular points detected by the proposed method in column (c), where green circles indicate cores and blue delta indicate deltas. To make the comparison more oblivious, we also manually label the ground truth in blue.

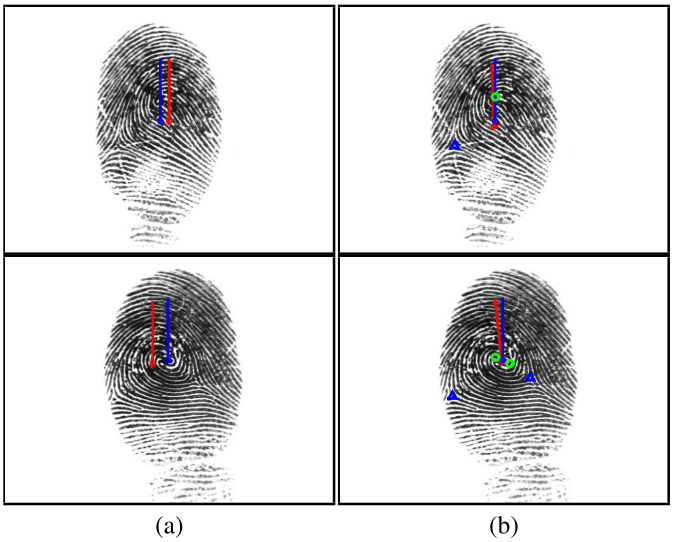

Fig. 7. The results of two methods for pose estimation on plain fingerprints: (a) Su *et al.* [6]; (b) proposed method; The fingerprints are from FVC2004 DB1A. The red circle indicates the estimated center while the red line points at the estimated direction. Here we also show the singular points detected by the proposed method in colomn (b) and the manually labeled pose for a better comparison.

Similar to the steps described in [11], we first extract minutiae from the mated fingerprints by the commercial software

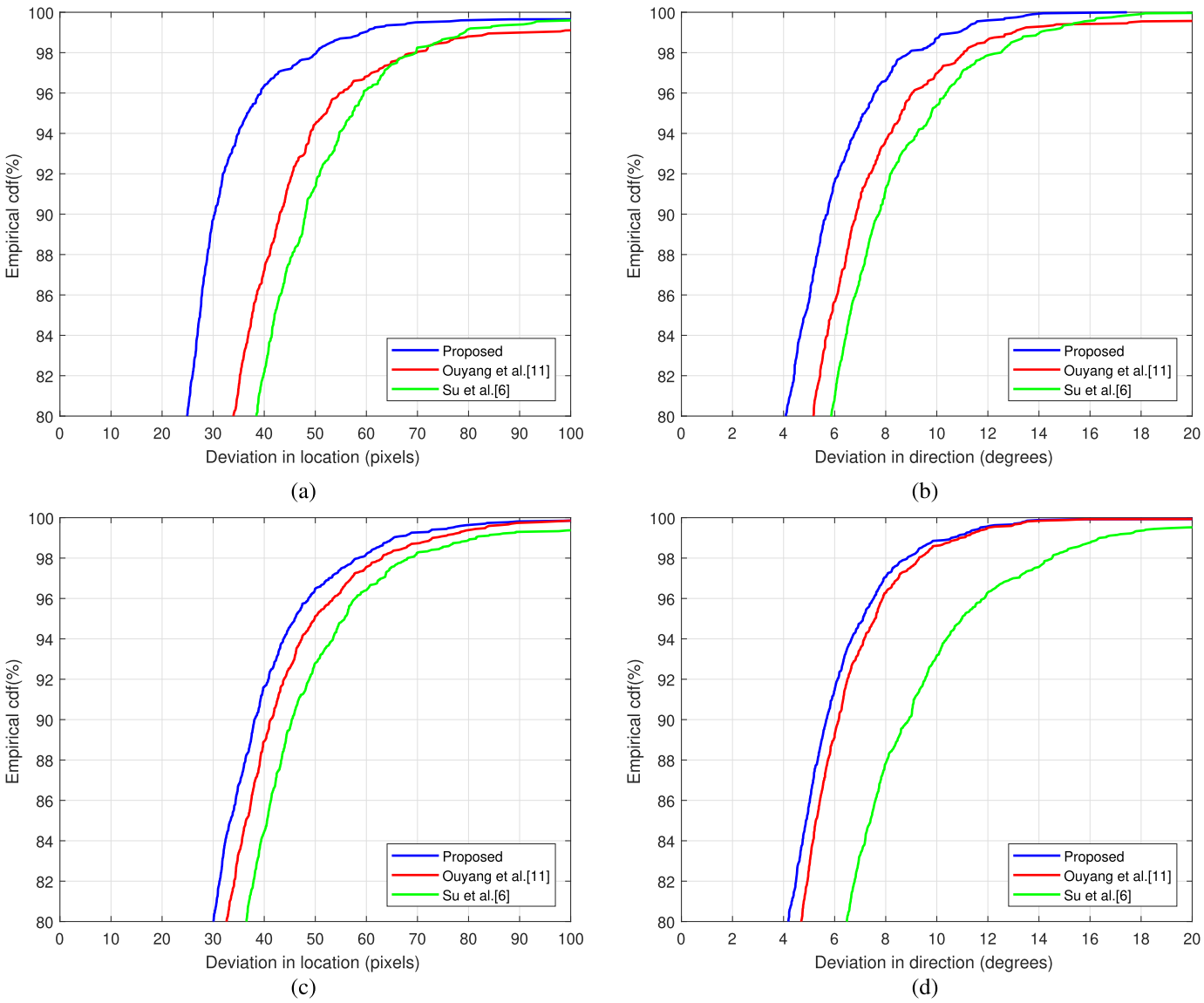

Fig. 8. The empirical cumulative distribution curves of mean location and direction deviation between mated minutiae pairs after registration by different pose estimation algorithms.. (a) and (b): the location and direction deviation of the 2000 pairs in NIST SD4; (c) and (d): the location and direction deviation of the last 2700 pairs in NIST SD14.

TABLE II

THE MEAN DEVIATION BETWEEN ESTIMATED POSE AND GROUND TRUTH OF LAST 100 FINGERPRINTS IN NIST SD14

Method	Deviation of location (pixels)	Deviation of direction (degrees)
Su et al. [6]	31.53	6.65
Ouyang et al. [11]	29.62	2.65
Proposed	15.54	1.45

VeriFinger, then the matching minutiae pairs are detected by the MCC method [44]. Second, the poses are estimated by the pose estimation method and the mated fingerprints are registered to the same coordinate system. Finally, the location and direction deviations of each minutiae pairs are calculated, and the mean deviation can be used to evaluate the performance of pose estimation.

Fig. 8 shows the results on NIST SD4 and NIST SD14, respectively. As we can observe from the curves, the proposed method outperforms the existing two pose estimation methods. Of the 2000 pairs of fingerprints in NIST SD4, about 97% pairs have the location deviation less than 45 pixels and the direction deviation less than 8 degrees. It is of great significance to fingerprint indexing for that the mean width of a ridge is about 9 pixels, after registration with the pose estimated by our method, the mated minutiae are close enough. Based on this, we can easily delete those impostors in order to get a better indexing performance, which is to be discussed in the following section. Compared to the two state-of-the-art methods, only 92% and 88% pairs achieve the same level. The experiments on NIST SD14 show the similar results, with a little drop on the superiority of our proposed method.

Here we also evaluate the proposed method on latent fingerprints in NIST SD27. Different from rolled fingerprints, latent fingerprints are usually noisy and lack of regions. Thus the two pose estimation methods [6] and [11] both fail on

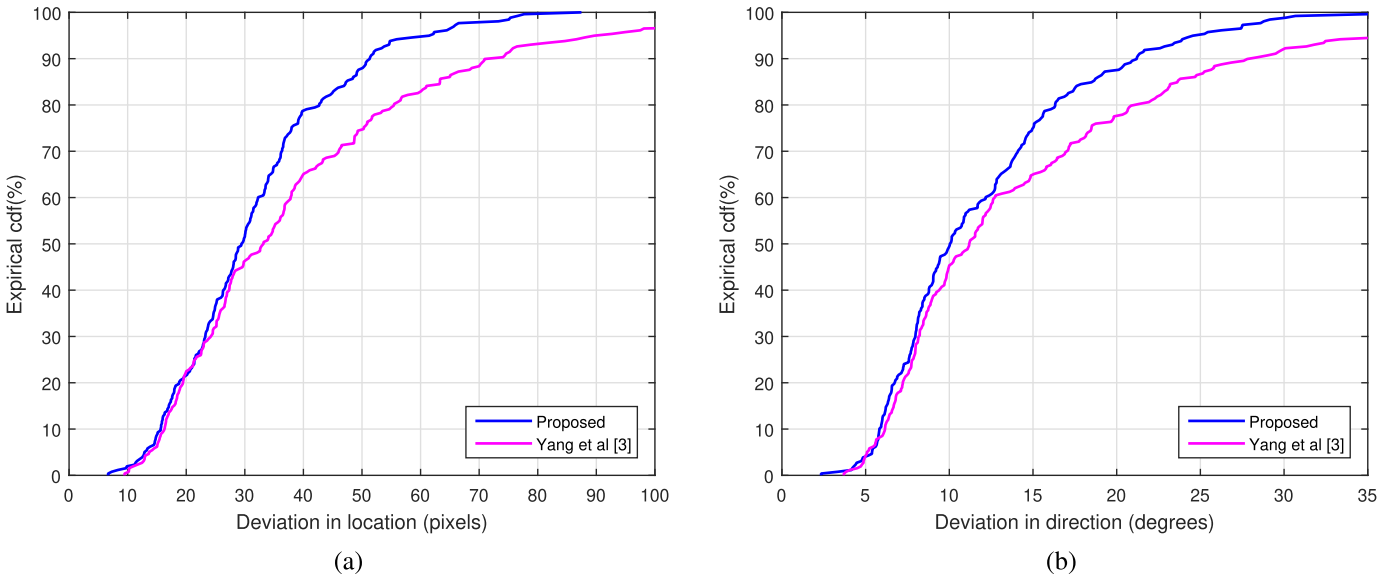

Fig. 9. The empirical cumulative distribution curves of mean location (a) and direction (b) deviation between mated minutiae pairs of all 258 pairs of fingerprints in NIST SD27 after registration by two pose estimation algorithms.

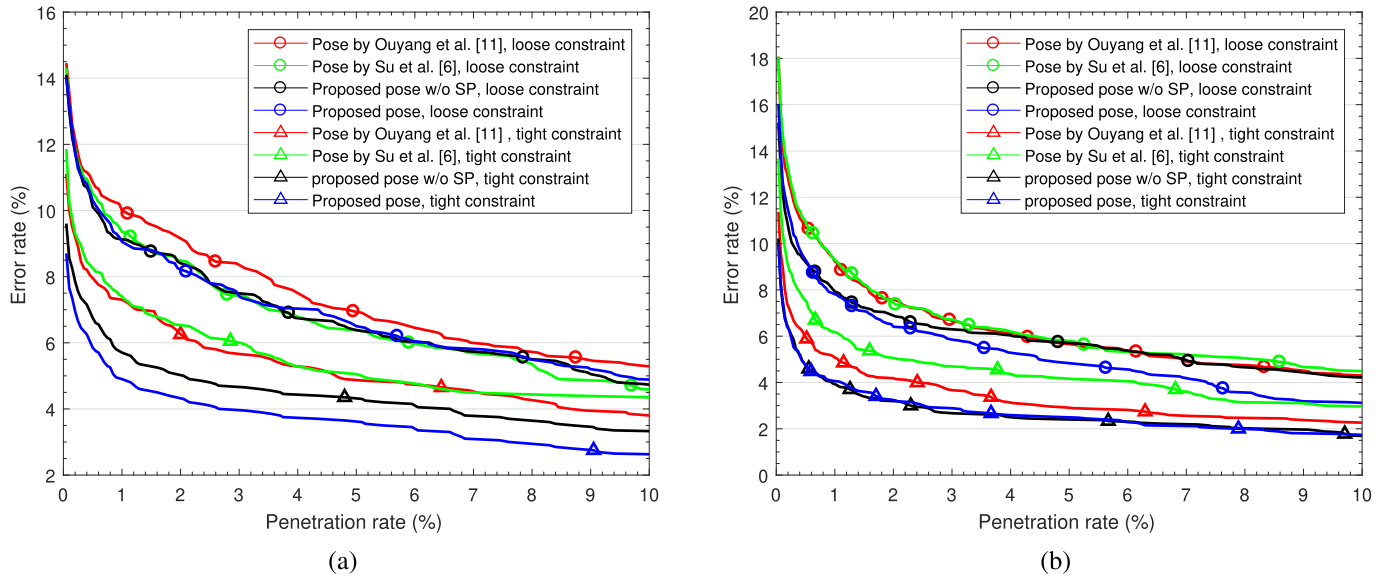

Fig. 10. The indexing results with different pose constraints by different pose estimation methods on (a) 2000 pairs of fingerprints in NIST SD4; (b) the last 2700 pairs of fingerprints in NIST SD14.

TABLE III
THE AVERAGE TIME CONSUMPTION (MS) OF INDEXING ONE QUERY FINGERPRINT IN NIST SD14

Method	Loose constraints	Tight constraints
No pose	42.34	14.40
Su et al. [6]	24.43	13.79
Ouyang et al. [11]	24.93	15.17
Proposed	24.50	15.23

the latent fingerprints. We compare our method with the method of Yang *et al.* [3]. The mated minutiae are manually marked.

The experimental results on NIST SD27 are shown in Fig. 9. As observed, our proposed method outperforms Yang *et al.*'s method with less deviation in location. About 88% fingerprints have a mean deviation in location less than 50 pixels (about the width of 5 ridges) with the pose estimated by the proposed method, while only 75% fingerprints achieve the same level with the pose estimated by Yang *et al.*'s method. Meanwhile, our method also outperforms a little better than Yang *et al.*'s method in direction.

C. Evaluation by Fingerprint Indexing

As mentioned above, the final goal of pose estimation is to better the performance of fingerprint indexing, thus we further carry out the indexing experiment to evaluate our proposed method.

The indexing algorithm in [6] is used in the experiments. Although Cao and Jain have proposed a better indexing

TABLE IV

THE SINGULAR POINT DETECTION RESULTS ON (A) NIST SD27L AND (B) NIST SD27F

Method	Precision	Recall	F1-score
VeriFinger	0.82	0.60	0.69
Proposed	0.88	0.69	0.77

(a)

Method	Precision	Recall	F1-score
VeriFinger	0.954	0.931	0.942
Proposed	0.964	0.929	0.946

(b)

algorithm in [5], the training data they used to train the network is not public and it is not simple to add restrictions of poses, thus we adopt the former indexing algorithm.

In the indexing algorithm, MCC descriptor is used for judging how similar two minutiae from two fingerprints are and calculate a matching score. In the off-line period, firstly, the pose of each gallery fingerprint is estimated and then the fingerprint is registered to the standard coordinate system. Secondly, the MCC descriptors of all minutiae are extracted and locality sensitive hashing is used to generate index table. In the on-line period, analogously, the pose of the query fingerprint is first estimated and then the query fingerprint is registered to the same coordinate system. After that, the MCC descriptors are extracted and transformed into index terms. Different from the original indexing algorithm, we add extra pose restrictions that only minutiae which are close enough are considered, namely, those have deviation less than location threshold e_l and direction threshold e_θ are available and vote for the final indexing score.

In this experiment, we set $e_l = 200, e_\theta = 40$ as loose constraints, $e_l = 60, e_\theta = 10$ as tight constraints.

Fig. 10 shows the indexing results of indexing with loose and tight pose constraints by 3 pose estimation methods on NIST SD4 (a) and NIST SD14(b). Though the proposed method is basically equal to Su's method with loose constraint on NIST SD4, with the constraint becomes tight, the performance of proposed method is obliviously superior to the other two methods, where the error rate is only about 5% at 1% penetration rate. On NIST SD14, the superiority become more oblivious. The error rate is less than 4% at 1% penetration rate while the other two methods are about 5.5% and 6.5%, respectively. It is worth noting that the performance with loose constraint by our proposed method overcomes the performance with tight constraint by Su *et al.*'s method, achieving a comparative level to the performance with tight constraint by Ouyang *et al.*'s method. This indicates that the more precise the pose is estimated, the less constraint is required in indexing, because more impostors are rejected by pose constraints. the precise pose estimation also helps boosting the indexing algorithm for the same reason. Table III shows the time consumption in the indexing experiments with different pose constraints.

Fig. 11 shows the indexing results on FVC2004 DB1A. Here the third impression of each fingerprint is used as database while the other impressions are used for query, thus we can get a total of 700 pairs of fingerprints. Considering the area of plain fingerprints are similar to latent fingerprints, we also compared the pose estimation results by network trained with

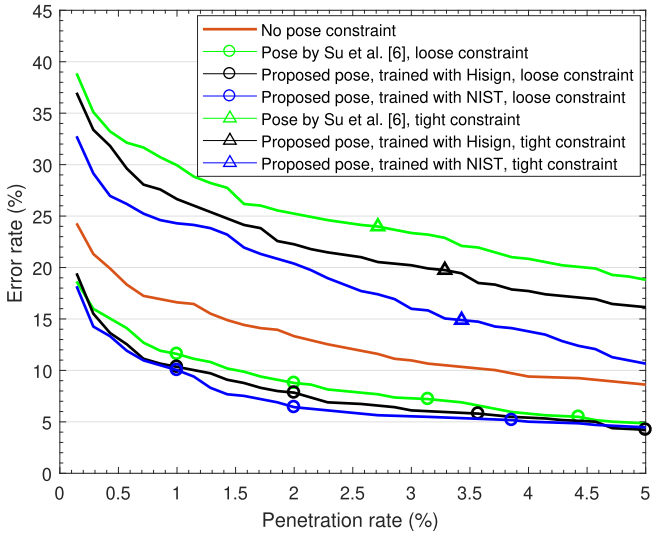

Fig. 11. The indexing results with different pose constraints by different pose estimation methods on 700 pairs of plain fingerprints in FVC2004 dB1A.

Hisign Latent Database. Different from the indexing results shown on rolled fingerprints in NIST SD4 and NIST SD14, the indexing error increases when the constraint become tight. The reason is that the pose is not as precise as that of rolled fingerprints, tight constraints drops the indexing results on the contrary. Anyhow, the loose constraints still show great improvement and our method outperforms Su *et al.*'s method under the same condition.

D. Evaluation on Singular Points Detection

We show some examples of the singular points detected by our proposed method in Fig. 6 and Fig. 7. To make the evaluation more convincing, we further carry out the following experiment in a traditional way for point detection evaluation.

NIST SD27 (both file and latent fingerprints) is used for evaluating our singular points detection to prove that the proposed network can deal with arbitrary fingerprints. We manually labeled the ground truths and compare our method with the commercial software VeriFinger SDK 11.2. If the distance between detected singular point and the ground truth is smaller than a threshold t, the detected SP is considered to be right. Here t is set to 30 pixels. The precision, recall and F1-score are listed in Table IV. The results show the proposed method outperforms VeriFinger on latent fingerprints and achieve a competitive level on rolled fingerprints.

E. Relationship Between Pose Estimation and Singular Point Detection

To further analyze the relationship between singular points and pose of fingerprint, we remove the SP heads and re-train our network to solve the pose estimation task individually. By comparing the poses estimated by networks with/without SP heads, we can tell whether the singular points can help pose estimation or not.

Fig. 12 shows some typical examples that by bringing in SPs, the pose becomes more precise. It fits our conjecture. In fact the indexing results in Fig. 10 also come to the same

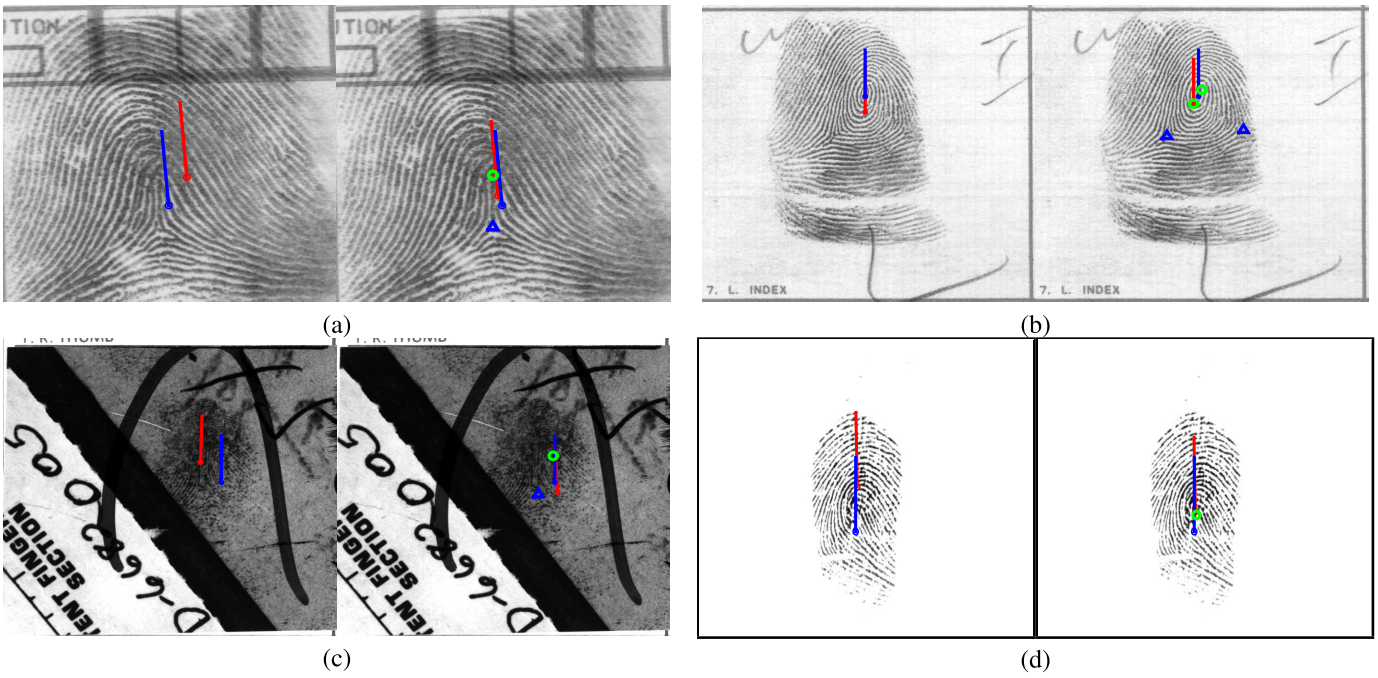

Fig. 12. Typical examples showing how SPs help pose estimation. The four fingerprints are from different databases: (a) NIST SD4; (b) NIST SD14; (c) NIST SD27; (d) FVC2004 DB1A. Among each pair, the left one shows the pose estimated without singular points, with some deviations to the ground truth. Such deviations can be removed when bringing in singular points (the right one).

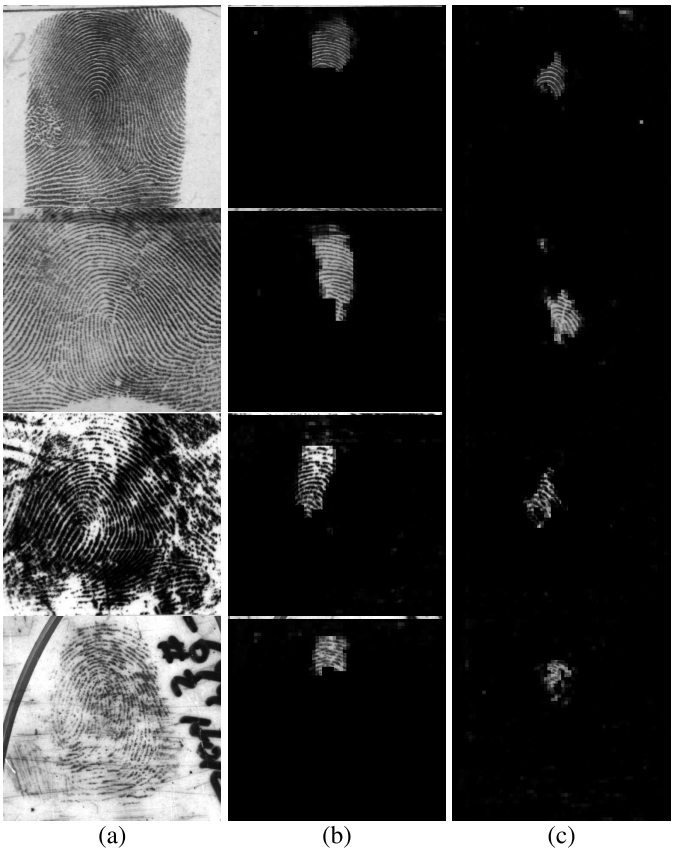

Fig. 13. The response of some channels of feature map output by the body part of the proposed network. (a) Input image; (b) 56th channel; (c) 65th channel. The top two rows are from NIST SD4 and the bottom two rows are from NIST SD27. These channels show correlation corresponding to certain orientation patterns.

conclusion. Though the improvement on NIST SD14 is quite tiny, there is a clear gap on the indexing results on NIST SD4.

F. Interpretability Analysis

In this section, we try to explain the rationality of the proposed network. Here we analyze the output of the body part, which are 90 channels of feature maps. We note that even though there is no supervision on the feature maps, some channels show high correlation with certain orientation patterns, that meets our expectation. Fig. 13 shows some typical examples. Here the feature map is resized to 8 times and multiplied to the input image as a mask. As observed, the 56th channel has a strong response at the upper region of fingerprints, where the orientation fields flow horizontally. While the 65th channel has a strong response at the region near the upper core, where the orientation fields flow like an arch. Both rolled and latent fingerprints share the same characteristic.

VI. Conclusion

In this article, we aim to solve the problem of pose estimation and singular points detection for arbitrary fingerprints by deep learning at the same time. By analyzing the conventional methods, such as Hough-voting method for pose estimation and complex filtering method for SP detection, we succeed in transforming the methods into a deep network and further proposed a novel deep learning based method for pose estimation and SP detection. Numerous experiments show great precision and robustness of the proposed method. To the best of our knowledge, the proposed method is the first pose estimation algorithm to deal with arbitrary fingerprints well.

Future work will include: (1) uniting more features like orientation fields, frequency fields or minutiae into the proposed network; (2) optimizing the architecture, including replacing the fully-connection layers to reduce the memory consumption and avoid missing information incurred by cropping the

original fingerprint image, and replacing the feature extraction body with more powerful backbones; and (3) exploring new loss function for a better representation of the relation between fingerprint's pose and singular points.

REFERENCES

[1] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, 2nd ed. London, U.K.: Springer, 2009.

[2] A. K. Jain, Y. Chen, and M. Demirkus, "Pores and ridges: High-resolution fingerprint matching using level 3 features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 15–27, Jan. 2007.

[3] X. Yang, J. Feng, and J. Zhou, "Localized dictionaries based orientation field estimation for latent fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 955–969, May 2014.

[4] R. Cappelli, "Fast and accurate fingerprint indexing based on ridge orientation and frequency," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1511–1521, Dec. 2011.

[5] K. Cao and A. K. Jain, "Fingerprint indexing and matching: An integrated approach," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 437–445.

[6] Y. Su, J. Feng, and J. Zhou, "Fingerprint indexing with pose constraint," *Pattern Recognit.*, vol. 54, pp. 1–13, Jun. 2016.

[7] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.

[8] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[9] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[10] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 149–158, Feb. 2004.

[11] J. Ouyang, J. Feng, J. Lu, Z. Guo, and J. Zhou, "Fingerprint pose estimation based on faster R-CNN," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 268–276.

[12] R. Cappelli and D. Maltoni, "On the spatial distribution of fingerprint singularities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 448–742, Oct. 2009.

[13] A. M. Bazen and S. H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 905–919, Jul. 2002.

[14] K. Nilsson and J. Bigun, "Localization of corresponding points in fingerprints by complex filtering," *Pattern Recognit. Lett.*, vol. 24, no. 13, pp. 2135–2144, Sep. 2003.

[15] M. Liu, X. Jiang, and A. C. Kot, "Fingerprint reference-point detection," *EURASIP J. Adv. Signal Process.*, vol. 2005, no. 4, pp. 498–509, Dec. 2005.

[16] V. Areekul, K. Suppasriwasuseth, and S. Jirachawang, "The new focal point localization algorithm for fingerprint registration," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4, 2006, pp. 497–500.

[17] S. Yoon, K. Cao, E. Liu, and A. K. Jain, "LFIQ: Latent fingerprint image quality," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–8.

[18] C. Deerada, K. Phromsuthirak, A. Rungchokanun, and V. Areekul, "Progressive focusing algorithm for reliable pose estimation of latent fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1232–1247, 2020.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[20] P. Schuch, J. M. May, and C. Busch, "Unsupervised learning of fingerprint rotations," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2018, pp. 1–6.

[21] Q. Yin, J. Feng, J. Lu, and J. Zhou, "Orientation field estimation for latent fingerprints by exhaustive search of large database," in *Proc. IEEE 9th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Oct. 2018, pp. 1–9.

[22] R. P. Krish, J. Bigun, D. Ramos, J. Ortega-Garcia, and J. Fierrez, "Pre-registration of latent fingerprints based on orientation field," *IET Biometrics*, vol. 4, no. 2, pp. 42–52, Jun. 2015.

[23] R. P. Krish, J. Fierrez, D. Ramos, F. Alonso-Fernandez, and J. Bigun, "Improving automated latent fingerprint identification using extended minutia types," *Inf. Fusion*, vol. 50, pp. 9–19, Oct. 2019.

[24] A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302–314, Apr. 1997.

[25] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognit.*, vol. 17, no. 3, pp. 295–303, Jan. 1984.

[26] J. Zhou, F. Chen, and J. Gu, "A novel algorithm for detecting singular points from fingerprint images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1239–1250, Jul. 2009.

[27] K. Tiwari and P. Gupta, "Meandering energy potential to locate singular point of fingerprint," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–6.

[28] A. I. Khan, "LAFIN: A convolutional neural network-based technique for singular point extraction and classification of latent fingerprints," in *Proc. Int. Conf. Soft-Comput. Netw. Secur. (ICSNS)*, Feb. 2018, pp. 1–7.

[29] L. Fan, S. Wang, H. Wang, and T. Guo, "Singular points detection based on zero-pole model in fingerprint images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 929–940, Jun. 2008.

[30] Y. Wang, J. Hu, and D. Phillips, "A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 573–585, Apr. 2007.

[31] K. Cao and A. K. Jain, "Latent orientation field estimation via convolutional neural network," in *Proc. Int. Conf. Biometrics (ICB)*, May 2015, pp. 349–356.

[32] J. Feng, J. Zhou, and A. K. Jain, "Orientation field estimation for latent fingerprint enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 925–940, Aug. 2013.

[33] P. Schuch, S. D. Schulz, and C. Busch, "ConvNet regression for fingerprint orientations," *Scand. Conf. Image Anal.*, 2017, pp. 325–336.

[34] P. Schuch, S.-D. Schulz, and C. Busch, "Deep expectation for estimation of fingerprint orientation fields," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 185–190.

[35] Y. Tang, F. Gao, J. Feng, and Y. Liu, "FingerNet: An unified deep network for fingerprint minutiae extraction," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 108–116.

[36] D.-L. Nguyen, K. Cao, and A. K. Jain, "Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge," in *Proc. Int. Conf. Biometrics (ICB)*, Feb. 2018, pp. 9–16.

[37] M. Kücken and A. C. Newell, "Fingerprint formation," *J. Theor. Biol.*, vol. 235, no. 1, pp. 71–83, Jul. 2005.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, pp. 448–456, 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[40] 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. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[41] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 2048–2057.

[42] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[43] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–15.

[44] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

Qihao Yin received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree under the supervision of Prof. Jie Zhou with the Department of Automation. His research interests include fingerprint recognition, computer vision, and pattern recognition.

Jianjiang Feng (Member, IEEE) received the B.Eng. and Ph.D. degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. From 2008 to 2009, he was a Post-Doctoral Researcher with the PRIP Laboratory, Michigan State University. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing. His research interests include fingerprint recognition and computer vision. He is an Associate Editor of the *Image and Vision Computing*.

Jiwen Lu (Senior Member, IEEE) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision, pattern recognition, and machine learning. He has authored/coauthored over 200 scientific articles in these areas, where 60 of them are IEEE TRANSACTIONS articles, including 11-TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI) and 40 of them are CVPR/ICCV/ECCV/NIPS articles. He serves an Associate Editor for several international journals including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE, and *Pattern Recognition*. He is a member of the Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, and a member of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society. He also served as Workshop Chair/Special Session Chair/Area Chair for more than 20 international conferences, such as ICIP, ICPR, ICME, ACCV and WACV. He was a recipient of the National 1000 Young Talents Program of China in 2015 and the National Science Fund of China for Excellent Young Scholars in 2018.

Jie Zhou (Senior Member, IEEE) received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1995. From 1995 to 1997, he served as a Post-Doctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a Full Professor with the Department of Automation, Tsinghua University. In recent years, he has authored over 100 articles in peer-reviewed journals and conferences. Among them, over 40 articles have been published in top journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and CVPR. His current research interests include computer vision, pattern recognition, and image processing. He received the National Outstanding Youth Foundation of China Award. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Robotics and Automation*, and two other journals.