Fingerprint Pose Estimation Based on Faster R-CNN

Jiahong Ouyang[1], Jianjiang Feng[1], Jiwen Lu[1], Zhenhua Guo[2], Jie Zhou[1]
[1]Department of Automation, Tsinghua University, Beijing, China
[2]Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
ouyangjiahong13@gmail.com, {jfeng, lujiwen, jzhou}@tsinghua.edu.cn, zhenhua.guo@sz.tsinghua.edu.cn

Abstract

Fingerprint pose estimation is one of the bottlenecks of indexing in large scale database. The existing methods of pose estimation are based on manually appointed features (e.g. special points, ridges, orientation filed). In this paper, we propose a method based on deep learning to achieve accurate pose estimation. Faster R-CNN is adopted to detect the center point and rough direction, followed by intra-class and inter-class combination to calculate the precise direction. Extensive experiments on NIST-14 show that (1) the predicted poses are close to manual annotations even when the fingerprints are incomplete or noisy, (2) the estimated poses for matching fingerprint pairs are very consistent and (3) by registering fingerprints using the estimated pose, the accuracy of a state-of-the-art fingerprint indexing system is further improved.

1. Introduction

Fingerprint is one of the most significant biometric traits. With its immutability and uniqueness, fingerprints are usually used for identification. Nowadays, the applications of fingerprint are prevalent, including access control involving a small number of users as well as searching large database for crime investigation.

For these applications, the efficiency and accuracy of fingerprint matching and indexing are crucial, in which fingerprint pose estimation takes a non-trivial role. In indexing, the search in space can be significantly reduced, if the fingerprints are in their standard pose. Meanwhile, the pose can be used as the restriction in matching, thus may increase the matching accuracy.

Traditional methods on fingerprint pose estimation can be divided into two main categories. One is based on the special points in fingerprints [10, 11, 8, 14, 1], but their accuracy largely depends on the quality of fingerprints. The other is based on machine learning methods. However, in both [15] and [16], they use features designated by human, which cannot make full use of features in images.

Recent years, deep learning has gradually brought its benefits to many area in computer vision, from image classification [7] to object detection [5, 4, 13, 12, 6]. The state-of-the-art object detection algorithms can achieve high accuracy and run in real-time. However, deep learning methods have not yet been introduced into the problem of fingerprint pose estimation.

In this paper, we propose a fingerprint pose estimation method based on object detection network— Faster R-CNN [13], followed by intra-class and inter-class combination.

Our method has two contributions below:

(1) To the best of our knowledge, we are the first to introduce deep learning to estimate fingerprint pose. The features extracted by the convolutional neural network (CNN) are based on the global structure of the fingerprint rather than the manually appointed features (e.g. singularity points, ridges, orientation field). Extensive experiments show that the proposed method is robust even when fingerprints are incomplete or noisy.

(2) By registering fingerprints using estimated poses, the accuracy of a state-of-the-art fingerprint indexing system [15] is significantly boosted.

The rest of this paper is organized as follow: in section 2, published algorithms in fingerprint pose estimation and object detection are reviewed. Our proposed method is illustrated in section 3. Experiments that evaluate our method are presented in section 4. Finally, we summarize this paper and draw the conclusion in section 5.

2. Related work

2.1. Fingerprint pose estimation

The pose of a fingerprint is given by (*x, y*) which show the position of the center point of the fingerprint and θ that illustrates the angle between the orientation of the fingerprint with the vertical direction. With the estimated pose, two fingerprints can be transferred into the standard coordinate system. The precise pose allows applying strict constraints on position and orientation in minutiae matching, which contributes to the accuracy and efficiency of fingerprint indexing. Though, there is no universal definition on fingerprint pose in the fingerprint recognition literature, as long as a pose estimation algorithm can provide consistent estimations for different images of same finger, it is already very useful for fingerprint indexing.

Existing fingerprint pose estimation algorithms can be classified into three main categories [15]: special points based, classifier based, and Hough transform based.

Special points based. Singularities were often used as reference points for fingerprint registration [10]. Complex filters were designed to detect the symmetry points in orientation field of multiple resolution scales [11]. Other types of key points were also proposed for registration. Liu *et al*. [8] presented the maximum curvature point calculated by orientation field as the reference. Focal points [14, 1] were obtained by iteratively calculating the average crossing points of normal lines to ridges patches. The precise detection of these special points depends on the quality of fingerprints and is sensitive to noises.

Classifier based. The key idea of this method is searching for a bounding box of an upright fingerprint. Su *et al*. [15] designed a set of classifiers for each direction and extracted features by the histogram of ridge orientations in each sliding window to determine the center. This method considers the global patterns of fingerprint, thus is more robust to local noise than methods based on special points.

Hough transform based. Yang *et al*. [16] proposed a fingerprint registration algorithm by Hough transform. In off-line period, the spatial distribution of orientation field patches is learned. In on-line period, for an upright fingerprint, patches from the initial orientation field vote for the location of center point. The direction of fingerprint is estimated by voting for images rotated in all possible angles. Though this method is acceptable for orientation filed estimation, the predicted poses are not consistent between different images from the same finger, which is not suitable for indexing.

2.2. Object detection

Fingerprint pose estimation can be viewed as an object detection problem. Since the distinctive success of CNN in image classification on ILSVRC 2012 [7], whether the breakthrough in convolutional network can be transferred to the domain of object detection became a hot topic. Region-based CNN (R-CNN) [5] is firstly proposed in 2014, followed by its improved version Fast R-CNN [4] and SPP-net [6], as well as the advanced algorithm Faster R-CNN [13]. A series of improvements largely increase the accuracy and the speed of object detection. Despite methods rely on region proposals, YOLO [8] breaks the input image into blocks, each of which is responsible for a fixed number of predictions, achieving the fastest speed as the image go through the convolutional network only once. SSD [9] handles a batch of small objects better by detecting in multi-scale feature map, while maintains the speed in high level. Among these approaches, Faster R-CNN stands in the front.

Evolution of Faster R-CNN. Girshick [5] first introduced the two step method—R-CNN in object detection. First, selective search is used to get around 2000 region proposals. Then, fed the proposals into a convolutional network (e.g. AlexNet [7]) and extracted the features from the last fully-connected layer to a set of SVM classifiers to get the label for each proposal. A distinct defect is that each region proposal has to go through the network separately, which leads to high computing redundancy. In Fast R-CNN [4], the first stage is maintained, while the proposals and the input image are fed into the network at the same time, thanks to the design of the new ROI pooling layer. Meanwhile, the loss function of the network includes both the classifier of the object type and the regressor of the bounding box so that the algorithm is changed into single stage. Much higher accuracy and prominent acceleration are achieved. However, Fast R-CNN still need the pre-computed region proposals. Faster R-CNN [13] proposes a Region Proposal Network (RPN) as the substitution of previous selective search, sharing the convolutional layers with Fast R-CNN network. In RPN, a small network is used to slide over the convolutional feature map, then map to a lower-dimensional feature, which is fed into box-classification layer and box-regression layer. At each sliding location, a set of anchors are responsible for variable scales and ratios of objects. Three types of training algorithms are proposed to ensure the sharing of convolutional layers between RPN and Fast R-CNN. Thanks to the RPN, high-quality region proposals are provided with dominant acceleration.

3. Proposed Method

Fingerprint pose estimation includes finding out the center point and direction. Predicting the center is equivalent to searching for the bounding box of the fingerprint, which can be solved by object detection methods. The center of the predicted bounding box is regarded as the center point of the fingerprint. The direction of the fingerprint can be seen as the category of the object, which can be achieved by the network simultaneously. At this time, the angle is just the rough value by the classifier. The angle types and the possibilities of the respective types are combined and interpolation is used to get the precise direction of the fingerprint. The flowchart of the proposed method is illustrated in Figure 1.

3.1. Definition of fingerprint pose

The direction of the fingerprint is defined as the direction that is perpendicular to the finger joint, while for the definition of center points, it varies among different fingerprint types and is based on the singularity points—core point and

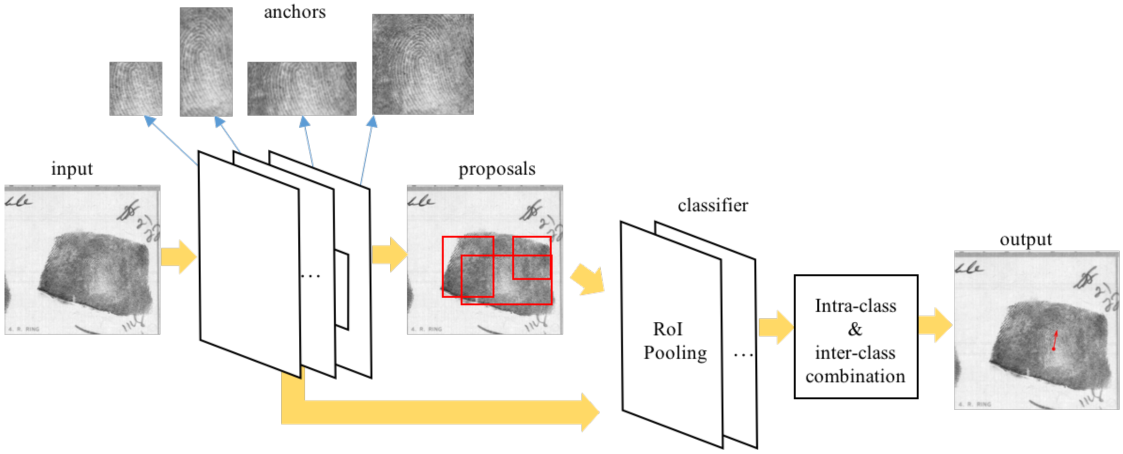


Figure 1. Flowchart of the proposed method.

delta point as the reference [15]. For arch fingerprint, the midpoint of the maximum curvature point and the northernmost straight ridge perpendicular to the finger direction is the center point. For those that have at least one core point and delta point, the upper reference point is the core point that is near to the fingertip and the lower reference point is the delta point that is far away from the upper reference point. The center is defined as the midpoint of the upper reference with the lower references projection on the line that parallels to fingerprint direction and crosses the upper reference. Examples are shown in Figure 2.

3.2. Data preparation

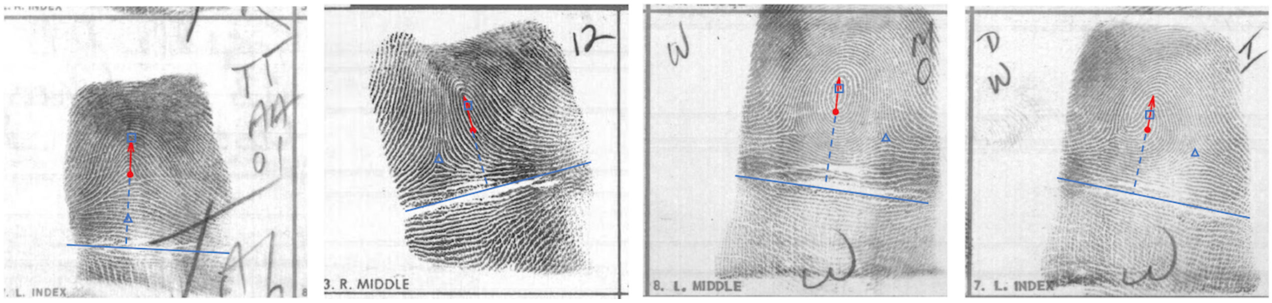
In this paper, we train the network on rolled fingerprint dataset—National Institute of Standard and Technology Special Database 14 (NIST-14). In this dataset, the image size is fixed at 800 * 768 and the resolution is 500 ppi. The raw data are the first 1500 pairs of fingerprints with manually labeled annotations of fingerprints pose. Based on that, we rotate each image every 15 degrees to generate 24 types of direction, maintaining images at its original size. Combining the direction of raw fingerprint as well as the angle of rotation, we calculate the ultimate angle of rotated fingerprint and set it to the nearest direction category. The central area of 550 * 550 around the center point is regarded as the perceptive region for a fingerprint. By doing these transformations, we generate 72,000 images of fingerprints in 24 different direction categories. 50,000 images are used as training set, while 20,000 images for validation and 2,000 images are remained to compare the result with ground truth.

3.3. Pose estimation method

We extended Faster R-CNN to the problem of fingerprint pose estimation. We fine-tune the VGG16 ImageNet pretrained model on our rolled fingerprint dataset by four-step alternating training. Faster R-CNN is used to detect the central region of fingerprint and its direction type. In general, Faster R-CNN will generate hundreds of prediction boxes, among them most direction types are approximate to the real direction. The result of Faster R-CNN is shown in Figure 3(a). Then the intra-class and inter-class combination are needed.

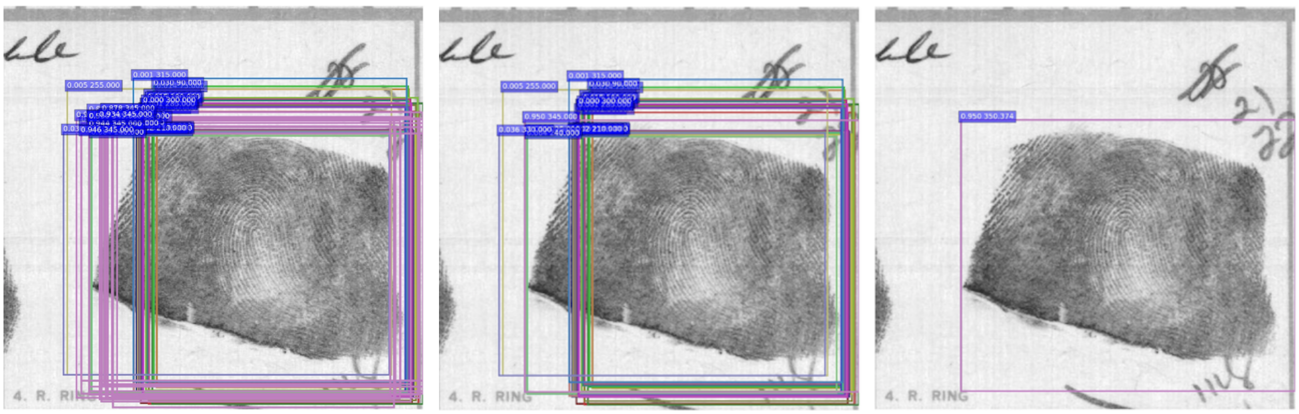
Confidence restriction and non-maximum suppression (NMS) are regarded as intra-class combination to merge the predicted bounding boxes of 24 categories respectively. Strict confidence threshold is used to narrow down the possible prediction boxes. We maintain two kinds of prediction as valid: (i) the bounding box of maximum confidence or (ii) the bounding box with the confidence level over the confidence threshold. In most situations, the second condition is sufficient, but we still adopt the first condition for the reason that it may have no result in some cases if the fingerprint is in bad condition like noisy and incomplete. After this step, direction types near to the real direction may contain several boxes and other types will leave only a single box. Then, NMS with a tight overlap threshold is used to ensure a single bounding box for each type of direction. We consider the center of each bounding box as the possible fingerprints center point and the type of the bounding box as its rough direction. Figure 3(b) illustrates the result after intra-class combination.

Inter-class combination is followed. We assume the final result as the bounding box with highest possibility among these 24 categories, then consider the second highest. If its possibility is higher than a threshold and the angle difference between those two is less than a threshold, the ultimate center point and angle are the interpolation of these two center points and angles weighted by their possibility. Otherwise, the ultimate result is the highest one. The inter-



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

Figure 2. Definition of pose for four types of fingerprint



(a) (b) (c)

Figure 3. Result of each step. Different colors represent diferent direction types. Texts on each prediction box show its possibility and type. (a) The result of Faster R-CNN network. (b) The result after intra-class combination. (c) The ultimate result after inter-class combination. 350.374 is the angle between the direction of fingerprint and the Y axis.

class combination algorithm is shown in Algorithm 1.

Algorithm 1 Inter-class Combination

Input: predicted bounding boxes of 24 direction types
for each fingerprint in the image **do**
  $box_1 \leftarrow box\ with\ highest\ confidence$
  $box_2 \leftarrow box\ with\ second\ highest\ confidence$
  $pose \leftarrow box_1$
  if $box_2.confidence < thresh_{conf}$ **then**
    $diff \leftarrow abs(box_1.angle - box_2.angle)$
    if $diff < thresh_{diff}$ **then**
      $pose \leftarrow interpolation$
    end if
  end if
end for
Output: pose

The final estimated pose is shown in Figure 3(c). Unlike traditional methods, this step ensures the consistency of the predicted direction for different images of the same fingerprint. The estimated results in [15, 16] have large quantization errors because of large angle step between each classifier or each rotated image.

3.4. Implementation details

We adopt open source code of Faster R-CNN python version created by Girshick [13]. Training and testing are implemented on NVIDIA GTX 1080 Ti.

The training of Faster R-CNN consists of RPN training and Fast R-CNN training. For RPN training, the VGG16 ImageNet pre-trained model is used to initialize RPN. Then RPN is trained end-to-end by back-propagation and stochastic gradient descent (SGD). 256 anchors from an image per batch are randomly selected as a mini-batch for computing loss function. Positive anchor is defined as anchor with the Intersection-of-Union (IoU) higher than 0.7, while negative anchors are those IoU smaller than 0.3. Those neither positive nor negative have no contribution to loss function. In a mini-batch, the ratio between positive and negative is set to 1. After generating region proposals, NMS is used to merge the redundant proposals into 2k per image. Proposals with an overlap over 0.7 are considered as repetitive proposals for the same object.

For Fast R-CNN training, same pre-trained model is used to initialize. A mini-batch consists of 256 random proposals from 2 images, 128 proposals for each. Different definitions of positive and negative sample are made. Proposals with an overlap with the ground truth over 0.5 are considered as foreground. To get hard negative, those have an overlap between 0.1 and 0.5 are defined as background. The fraction of mini-batch that is labeled foreground is 0.25. After training RPN and Fast R-CNN for the first time, the model of detector network is used to initialize second RPN training process. This time, the shared convolutional layers are fixed and only layers that are specific to RPN are trained. Then for the last step, keep the convolutional layers and train the unique layers of Fast R-CNN.

In intra-class combination, for each class, the bounding box of maximum confidence or the bounding box with the confidence level over 0.8 are regarded as valid detection results. Two valid bounding boxes with an overlap less than 0.3 are seen as prediction for different fingerprints. In general, after intra-class combination, an image with one fingerprint will remain 24 predictions.

For inter-class combination, as its shown in section 3.3, bounding box with confidence less than 0.2 is considered as invalid, leaving only several bounding box for each fingerprint. Then, if the difference between the angle of prediction with highest confidence and the one with second highest is less than 30 degrees, interpolation is needed. Like intra-class combination, bounding boxes with overlap higher than 0.3 are seen as the same fingerprint.

4. Experiments

As our model is trained on rolled fingerprint dataset, we test our result of pose estimation on the same kind of dataset. Three different approaches validate the efficiency and the accuracy of our method. In section 4.1, we introduce the dataset that we use in experiments. In section 4.2, we evaluate the pose estimation algorithm by comparing with the manually marked poses. Sine the main purpose of pose estimation is to align different images of the same finger, we use a more direct measure. In section 4.3, we use the mean difference between matching minutiae pairs to evaluate the estimation accuracy and stability. Moreover, as the indexing is the ultimate goal for pose estimation, the indexing results are used to evaluate the proposed method in section 4.4.

4.1. Dataset

We conducted experiments on rolled fingerprint dataset — NIST-14. This dataset provides 27,000 pairs of fingerprints with pattern types following the natrual distribution. A pair of fingerprints represents two different impressions of the same fingerprint. 2,000 test images from the 72,000 images obtained by rotating the first 1,500 pairs of fingerprints are used for comparing the estimated pose with the ground truth. These fingerprints have angles in (-180, 180]. The last 2,700 pairs in NIST-14 with the correspondence of minutiae pairs are used for experiments in section 4.3 and section 4.4.

4.2. Evaluation by ground truth

The results comparing the predicted pose with the ground truth is shown in Figure 6. As we can see, 80% fingerprints have the deviation less than 30 pixels in location and 3 degrees in direction. 95% fingerprints have the deviation less than 53 pixels in location and 5 degrees in direction. Considering the image size of 800×768 as well as the angle range of 360 degrees, the estimation is quite accurate. Besides, the state-of-the-art algorithm [15] only considers a narrow direction range, thus can not work in our test dataset. Another advance method [16] has to rotate the input images to search through the direction space and feed each of them into the estimation system, leading to the inefficiency.

4.3. Evaluation by matching minutiae pairs

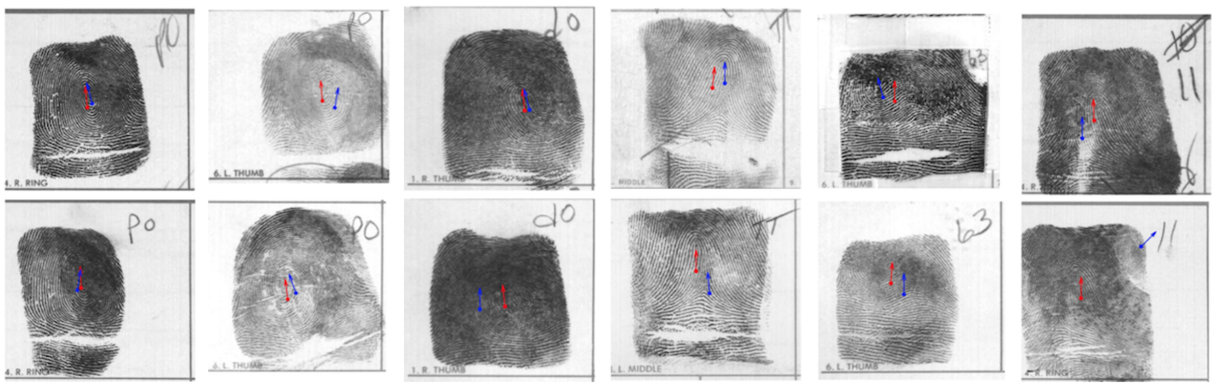
For a pair of fingerprints from the same finger, their matching minutiae should be very close after registration to the same coordinate system. Based on that, the mean deviation of minutiae pairs can be the indicator of pose accuracy and stability [15].

This method has 3 steps. First, the minutiae matching method in [2] is used to detect matching minutiae pairs between the fingerprint pairs. Second, the pose is estimated using the pose estimation algorithm and the minutiae are transformed into the standard coordinate, which is defined as below. The center point of the fingerprint is the origin. Y-axis is parallel to the direction of the estimated direction, while X-axis is decided by right-hand rule. Then, the location and direction deviations of each minutiae pair in the standard coordination are calculated. The mean deviation in location and direction shows the accuracy of the pose estimation. Figure 7 demonstrates the matching minutiae pairs and the deviation between a pair of fingerprints.

To illustrate the result of matching minutiae clearly, we only truncate the proportion from 95% $\sim$ 100%, which is shown in Figure 8. 80% fingerprints have the deviation less than 25 pixels in location and 3 degrees in direction. 95% fingerprints have the deviation less than 45 pixels and 7 degrees. Comparing with the state-of-the-art method [15], our method shows its superiority at the proportion of 99.5%, where our method is around 50 pixels and 8 degrees less than the state-of-the-art in location and direction respectively. This advantage will directly contribute to more precise and faster indexing by applying stricter threshold for matching minutiae, as we will demonstrate in details in the next section.

(a) (b) (c) (d) (e) (f)

Figure 4. Examples of pose estimation. Red dots and arrows show the result of proposed method. Blue ones illustrate the result of Su [15]. (a) and (b) shows that Su‘s method still has relatively large deviation when the fingerprint is in good condition. (c) (d) and (e) (f) illustrate the robustness of our method when the fingerprint is noisy or incomplete respectively.



(a) (b) (c) (d) (e) (f)

Figure 5. Comparison of the stability of two pose estimation methods. Each column represents a pair of impressions of the same fingerprint. Red dots and arrows show the result of proposed method. Blue ones illustrate the result of Su [15].

4.4. Evaluation by level-2 indexing

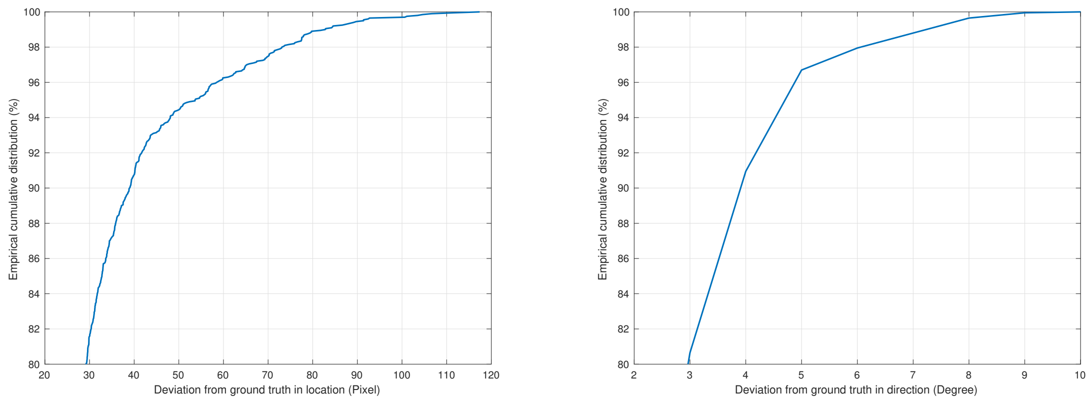
Level-2 indexing represents methods based on minutiae. In this section, we implements the indexing method proposed in [15] with various restrictions and pose estimation approaches.

In this method, MCC is used as discriptor to judge the similarities between minutiae in order to find out minutiae pairs. MCC encodes the neighborhood of minutiae by projecting its neighbors into a three dimensional space based on their relative positions and directions to the center minutiae. Locality Sensitive Hashing (LSH) is used to find the approximate nearest neighbor among MCC descriptors [3]. A set of local and global hash functions are combined to pick out several bits from the original vectors for projection methods. Inverted index table is used to store those features. Using these methods, similar vectors in the gallery are projected into the same buckets with the query vector. When searching the mated minutiae for the query one, only the matchable minutiae are considered. Matchable minutiae are defined as those having deviation less than the thresholds both in location (e_l) and orientation (e_θ) in the standard coordinate. A vote is cast for each gallery vector in this bucket by the mated minutia according to their similarities.

This algorithm can be summarized as the following steps. In off-line period, the pose of each gallery fingerprint is estimated firstly. Second, the MCC descriptors are extracted and hash functions are used to generate index terms. Finally, the features are stored in the inverted index table. In on-line period, the pose of the query fingerprint is estimated. Then, MCC features are extracted and transformed into index terms. For each minutiae in query fingerprint, the mated minutiae in each of the gallery fingerprint are found and votes are cast. At last, the score is accumulated for each gallery fingerprint.

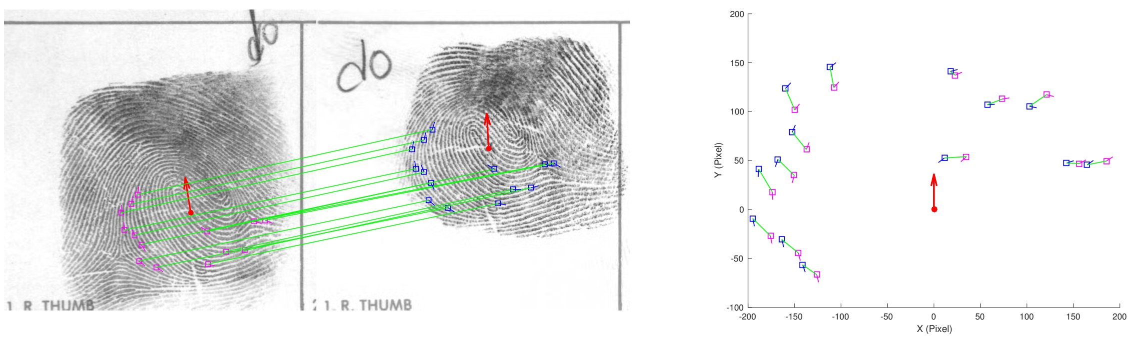
In this experiments, we set $e_l = 200, e_\theta = 40$ as loose constraints, $e_l = 120, e_\theta = 20$ as medium constraints, $e_l = 60, e_\theta = 10$ as tight constraints and $e_l = 20, e_\theta = 5$ as strict constraints. We use the fingerprints start with ‘F’ in last 2700 pairs of fingerprints in NIST-14 as gallary, while those begin with ‘S’ are for query.

Figure 9 (a) shows the results of indexing by three pose estimation methods with loose and tight constraints. The proposed method is much better than method in [16]. When the constraints are loose, the difference between our method and approach in [15] is not obvious, but when it comes to tight constraints, the superiority of precise pose is nontrival. It is worth mentioning that the method by Yang *et al.* [16] achieves better performance in loose restrictions than the tight one. The reason is that it rejects some mated minutiae as the estimated poses are not stable and consistent. Figure 9 (b) illustrates the results of the proposed method



(a) (b)

Figure 6. Empirical cumulative distribution functions of location (a) and direction (b) deviations on 2000 rotated fingerprints in test dataset on NIST-14.



(a) (b)

Figure 7. (a) The matching of minutiae pairs between two impressions of the same fingerprint. (b) The location and direction deviation of minutiae pairs in the standard coordinate.

Time (ms)	Loose	Medium	Tight
No pose	42.34	27.53	14.40
Su *et al*. [15]	24.43	25.12	13.79
Yang *et al*. [16]	39.50	24.84	16.75
Our method	24.93	23.01	15.17

Table 1. Average time consumption of indexing one image from last 2700 pairs of fingerprints in NIST-14.

with loose, medium, tight and strict constraints. We can see that, within certain range, tighter restrictions can decrease error rate and the accuracy of the estimated pose decides this range.

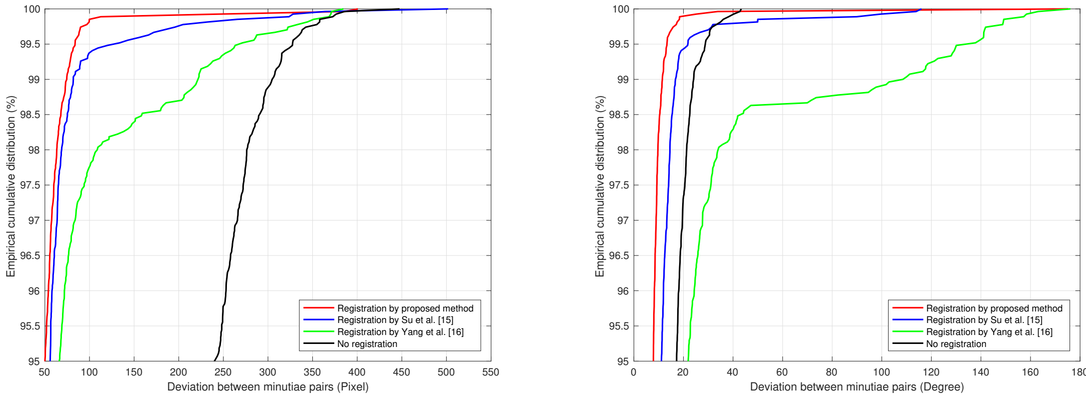
As shown in Table 1, stricter restrictions can largely reduce the average time consumption of indexing the fingerprint from the gallery. But an over-tight constraint may also slow down the speed, because some mated minutiae are rejected so that it has to search through all minutiae.

Method	Speed(/frame)	Configuration
Su *et al*. [15]	758 ms	CPU 2.50 GHz
Yang *et al*. [16]	7,300 ms	CPU 3.30 GHz
Our method	101 ms	NVIDIA GTX 1080 Ti

Table 2. Computational effciency of each pose estimation method.

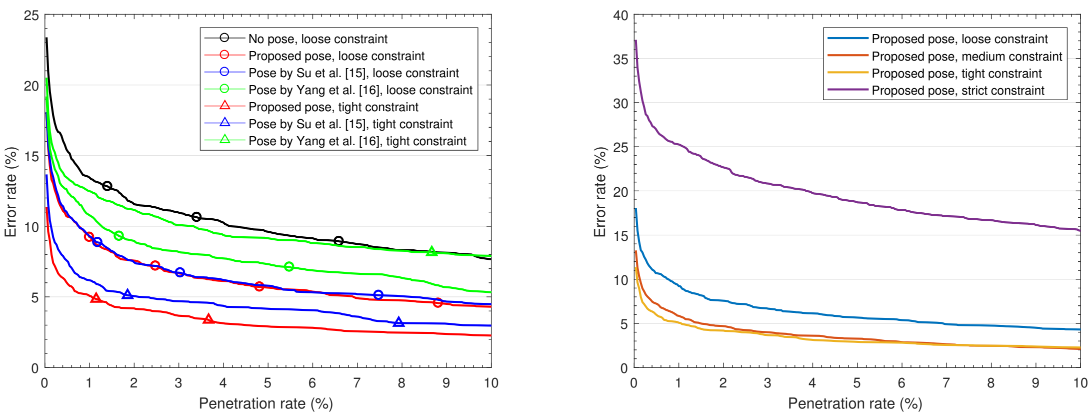
4.5. Computational efficiency

Table 2 compares the speed and computer configuration between different pose estimation methods. Both methods by Su *et al*. and Yang *et al*. work on CPU, while ours works on GPU, thus are not directly comparable. However, it should be noted that for those two methods, the image has to either be fed into a set of classifiers or be rotated by a series of angles to estimate the direction of fingerprint. Therefore, the range of fingerprint direction affects its speed, while ours is an end-to-end system that has no such restrictions.



(a) (b)

Figure 8. Empirical cumulative distribution functions of mean location (a) and direction (b) deviations between minutiae pairs before and after registration. Three approaches (classifier based by Su *et al*. [15], Hough transform based by Yang *et al*. [16] and the proposed method) are evaluated on the last 2700 pairs of fingerprints in NIST-14.



(a) (b)

Figure 9. Indexing performance on last 2700 pairs of fingerprints on NIST-14. (a) Three pose estimation approaches (classifier based by Su *et al*. [15], Hough transform based by Yang *et al*. [16] and the proposed method) are evaluated. (b) The proposed method with three levels of constraints.

5. Conclusion

In this paper, we firstly bring the deep learning into fingerprint pose estimation. Faster R-CNN is used to achieve both the center point and the direction of the fingerprint at the same time. Intra-class and inter-class combination after detection ensure a single accurate pose for each fingerprint in the image. The estimated poses are quite precise in rolled fingerprint dataset and the algorithm is fast in testing. The results comparing with the ground truth shows its accuracy. The evaluation experiment involving matching minutiae illustrates the stability among different impressions of the same finger. The indexing fingerprint experiment based on level-2 features proves the superiority of our pose estimation method.

However, the proposed method can still be improved in these aspects:

(1) extending the proposed method to plain fingerprints.

(2) using images that contain only a part of the fingerprint and strong noise to fine-tune the current model in order to handle latent fingerprints.

(3) applying other advanced deep neural network architectures as well as other the state-of-art detection algorithms [8][9] for fingerprint pose estimation.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grants 61622207, 61373074, 61527808, and 61572271.

References

[1] V. Areekul, K. Suppasriwasuseth, and S. Jirachawang. The new focal point localization algorithm for fingerprint registration. *IEEE Conference on Industrial Electronics Applications*, 4:497–500, 2006.

[2] 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*, 32(12):2128–2142, 2010.

[3] R. Cappelli, M. Ferrara, and D. Maltoni. Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1051–1057, 2011.

[4] R. Girshick. Fast R-CNN. *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions Pattern Analysis Machine Intelligence*, 37(9):1904–1916, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[8] M. Liu, X. Jiang, and A. C. Kot. Fingerprint reference-point detection. *EURASIP Journal on Advances in Signal Processing*, 2005(4):498–509, 2005.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. SSD: Single shot multibox detector. *European Conference on Computer Vision*, pages 21–37, 2016.

[10] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. Handbook of fingerprint recognition, 2nd. Springer. 2009.

[11] K. Nilsson and J. Bigun. Localization of corresponding points in fingerprints by complex filtering. *Pattern Recognition Letters*, 24(13):2135–2144, 2003.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions Pattern Analysis Machine Intelligence*, pages 1–14, 2015.

[14] K. Rerkrai and V. Areekul. A new reference point for fingerprint recognition. *International Conference on Image Processing*, pages 499–502, 2000.

[15] Y. Su, J. Feng, and J. Zhou. Fingerprint indexing with pose constraint. *Pattern Recognition*, 54:1–13, 2016.

[16] X. Yang, J. Feng, and J. Zhou. Localized dictionaries based orientation field estimation for latent fingerprints. *IEEE Transactions Pattern Analysis Machine Intelligence*, 36(5):955–969, 2014.