An Automatic Shoeprint Retrieval Method Using Neural Codes for Commercial Shoeprint Scanners

Junjian Cui(✉), Xiaorui Zhao, and Daixi Li

Dalian Everspry Sci. & Tech. Co. Ltd., No. 31 Xixian Street,
High-Tech Industrial Zone, Dalian, China
{cuijunjian,zhaoxiaorui,lidaixi}@everspry.com
http://www.footprintmatcher.com

Abstract. In this paper, an automatic shoeprint retrieval method used in forensic science is proposed. The proposed method extracts shoeprint features using recently reported descriptor called neural code. The first step of feature extraction is rotation compensation. Then, shoeprint image is divided into top region and bottom region, and two neural codes for both regions are obtained. Afterwards, a matching score between test image and reference image is calculated. The matching score is a weighted sum of cosine similarities of both regions' neural codes. Experimental results show that our method outperforms other methods on a large-scale database captured by commercial shoeprint scanners. By using PCA, the performance can be improved while the feature dimension is reduced dramatically. To our knowledge, this is the first study using the database collected by commercial shoeprint scanners, and our method obtained a cumulative match score of 88.7% at top 10.

Keywords: Shoeprint retrieval · Convolutional neural network
Neural code

1 Introduction

Shoe impressions play very important roles in forensic science since the frequency of shoe impression occurrence at the place of crime is much higher than other evidences (e.g. DNA, fingerprints, and hair) [7]. According to [11], 35% of crime scenes have useful shoeprints for further investigation, and shoe prints can be used in linking cases, knowing the brand and model of the shoes. Thus, an automated shoeprint retrieval or recognition algorithm is necessary for fast and accurate forensic investigations. Shoeprint image retrieval composed of two categories: real crime scene shoeprint retrieval and suspect's shoeprint retrieval, and this paper is focused on the latter category.

Nowadays, several commercial shoeprint scanners have been developed [1,2]. These scanners can scan suspect's shoeprint and acquire a highly detailed shoeprint image. The detailed image may then be uploaded into a database for

J. Yang et al. (Eds.): CCCV 2017, Part II, CCIS 772, pp. 158–169, 2017.
https://doi.org/10.1007/978-981-10-7302-1_14

comparison against recovered crime scene shoeprint impressions. For an application of commercial shoeprint scanners in suspect's shoeprint retrieval, please refer to [4].

Recently, several automatic shoeprint retrieval or recognition methods have been reported. Bouridane et al. [8] extracted shoeprint features using fractal transformation and compared features using Mean Square Noise Error (MSNE) method. On a database of 145 images, the recognition accuracy was 88%. A method using Fourier-Mellin transform and power spectral density (PSD) was proposed by Chazal et al. [10]. On a database of 476 images, the method achieved cumulative match scores (CMSs) of 65% and 87% at top 1 and top 5, respectively. Pavlou and Allinson [14] employed Maximally Stable Extremal Region (MSER) to detect keypoints and used SIFT descriptor to describe the keypoints. On a database containing 374 images, the method achieved 92% CMS at top 8. Zhang and Allinson [17] first calculated edge orientation histogram; and then, they used 1D DFT to obtain a rotation invariant feature. On a database of 512 images with 10% Gaussian noise, they achieved a CMS of 97.7% at top 20; however, their algorithm was not robust against rotation and noise. Patil and Kulkarni [13] utilized Radon transform to compensate image rotation and applied Gabor filter to extract multi-resolution features. The method reached a CMS of 91% at top 1 when evaluating on a database containing 200 images. Wang et al. [16] first divided shoeprint image into top region and bottom region and extracted features using Wavelet-Fourier transform. A confidence value of each region was computed, and final matching score was computed based on the confidence value and feature similarities. The method was evaluated on a very large database containing 210,000 real crime scene shoeprint images and achieved a CMS of 90.87% at top 2.

Most of methods mentioned above are suspect's shoeprint retrieval methods except for [16]. They performed pretty well only on their own databases; however, none of them used commercial shoeprint scanners to construct the database. Consequently, the applications of these methods are limited in their own databases.

Differing from these methods, this paper focuses its attention on suspect's shoeprint images collected by commercial shoeprint scanner called EverOS$^{\mathrm{TM}}$ [1]. Both test images and reference images in the database are binary images which are captured by EverOS$^{\mathrm{TM}}$. According to [16], most well-known local features (e.g. SIFT, SURF, and MSER) do not perform well on binary images. Therefore, in this paper, we use a deep Convolutional Neural Network (CNN) to extract high-level feature which is called neural code. Experimental results show that the deep CNN-based neural codes outperform other shoeprint features (e.g. [13,16,17]).

The rest of this paper is organized as follows: Sect. 2 briefly reviews CNN and neural code; Sect. 3 explains the proposed method in detail; some experimental results are given in Sect. 4, and Sect. 5 concludes this paper.

2 Related Works

Deep Convolutional Neural Networks (CNNs) have demonstrated the state-of-the-art performance in computer vision, speech, and other areas. Especially in ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), CNNs (e.g. vggNet [15], and ResNet [12]) have exhibited high classification accuracies, and ResNet [12] has outperformed human-level performance.

Our work was inspired by Babenko et al. [6]. In [6], they argued that outputs of the upper layers in a CNN could be served as good features (neural codes) for image retrieval, although the CNN was originally trained to classify images. They extracted neural codes from the last max-pooling layer (L5) and two fully connected layers (L6, and L7). Experimental results showed that the neural code extracted from L5 performed much better than the neural codes extracted from L6 and L7. They also showed that using PCA dimension reduction, the dimension of the neural code (L5) could be reduced from 9216 to 128, with no loss of retrieval accuracy.

In this paper, we employ vgg-verydeep-16 [15] to extract neural code. Vgg-verydeep-16 model is pre-trained using ImageNet and MatConvNet [3], which can be downloaded from MatConvNet. The model comprises five convolutional layers and two fully connected layers followed by a soft-max layer. Each convolutional layer is followed by a max-pooling layer. As the dimensions of neural codes extracted from other layers are too high ($\geq$100,000), in this paper, we only consider five neural codes extracted from the last max-pooling layer and two fully connected layers (each fully connected layer has a pre-activation layer and a ReLu layer).

3 The Proposed Method

The pipeline of the proposed method is illustrated in Fig. 1. The proposed method is divided into two phases: offline database feature extraction phase, and online shoeprint image retrieval phase. In Fig. 1, the dashed box indicates offline database feature extraction phase, and the other parts indicate online image retrieval phase.

In offline database feature extraction phase, we first estimate main axis of each shoeprint image in the database, and each image is rotated so that the main axis is in the vertical position. Afterwards, each image is divided into two semantic regions (top region and bottom region); then, neural codes (features) for both regions are extracted. Extracted neural codes are used in the next phase: online shoeprint retrieval.

In online shoeprint retrieval phase, neural codes for test image are extracted as described above; then, top region similarity and bottom region similarity between test image and an image in the database are calculated respectively. The matching score between two images is a weighted sum of top region similarity and bottom region similarity. After calculating all matching scores between test image and all images in the database, matching scores are sorted so that images most similar to the test image are presented at the top of ranked list. Finally, this ranked list is returned as the retrieval result.

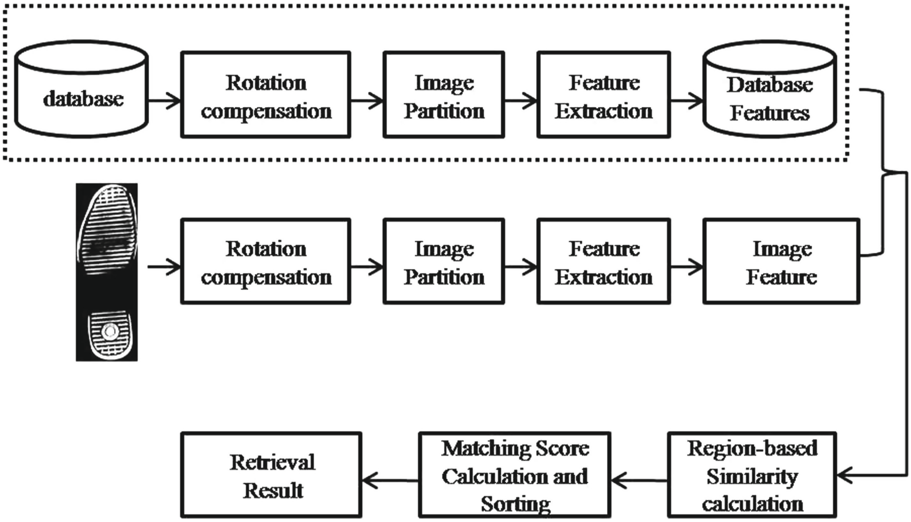

Fig. 1. Pipeline of the proposed method

3.1 Rotation Compensation

Figure 2 illustrates some shoeprint examples provided by EverOS$^{\mathrm{TM}}$. EverOS$^{\mathrm{TM}}$ provides binary images which are pre-processed interactively. The pre-processing includes three steps: rotation compensation, noise removal, and binary image generation. Among them, rotation compensation has much more effect on the retrieval result than the other two steps. Below, we will explain how to mitigate the effect caused by rotation.

In order to compensate rotation, main axis estimation is necessary. We adopt a computationally inexpensive and effective method called shape orientation algorithm [5] to find the main axis. The shape orientation algorithm is composed of three steps:

Step 1: Generate an edge map of a shoeprint image using Canny edge detector [9].
Step 2: Obtain the mass center (x_c, y_c) of the edge map.
Step 3: Find a straight line which passes through the mass center and summation of the distances from points on the edge map to this line is minimum.

We refer to the straight line described in Step 3 as main axis and will describe how to find it. The equation of a straight line passes mass center (x_c, y_c) with angle θ is given as

$$ax + by = c, \tag{1}$$

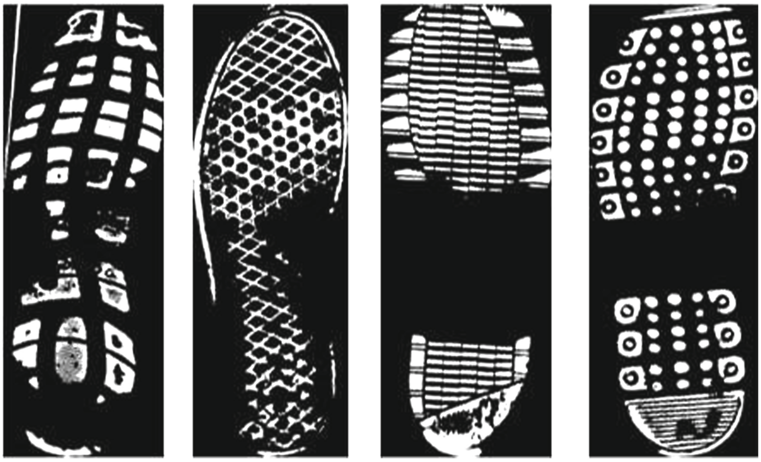

Fig. 2. Shoeprint examples provided by EverOS$^{\mathrm{TM}}$

where

$$\begin{cases} a = \tan\theta \\ b = -1 \\ c = y_c - x_c \times \tan\theta \end{cases} \quad \text{if } \theta \neq 90^\circ, \tag{2}$$

and

$$\begin{cases} a = 1 \\ b = 0 \\ c = -x_c \end{cases} \quad \text{if } \theta = 90^\circ. \tag{3}$$

The distance (d_i) from a point (x_i, y_i) on the edge map to the straight line is given as

$$d_i = \frac{|ax_i + by_i + c|}{\sqrt{a^2 + b^2}}. \tag{4}$$

Thus, the summation of the distances (D) from points on the edge map to the straight line is expressed as

$$D = \sum_i d_i. \tag{5}$$

For each angle θ_j in the range of $0 \leq \theta_j \leq \pi$ with a suitable step size (in 180 steps and 1 degree step size), we can obtain corresponding summation of distances D_j for each angle. Among these D_js, we can find the minimum, and the corresponding angle is the angle of the main axis.

After finding the main axis, we can compensate rotation by rotating the shoeprint image so that the main axis is in the vertical position.

3.2 Image Partition

A shoeprint can be divided into several parts: toe part, sole part, arch part, heel part, and back of heel part. When comparing two shoeprints, these parts can be ranked based on their importance, and the rank is as follows: sole (rank 1), heel, toe, back of heel, and arch (rank 5) [16]. We adopt the method proposed in [16] to divide a shoeprint image into two semantic regions: top region and bottom region. The top region contains toe, sole, and a part of arch, and the bottom region contains heel, back of heel, and the rest part of arch. The ratio between top region and bottom region is set to 6:4. Details are illustrated in Fig. 3.

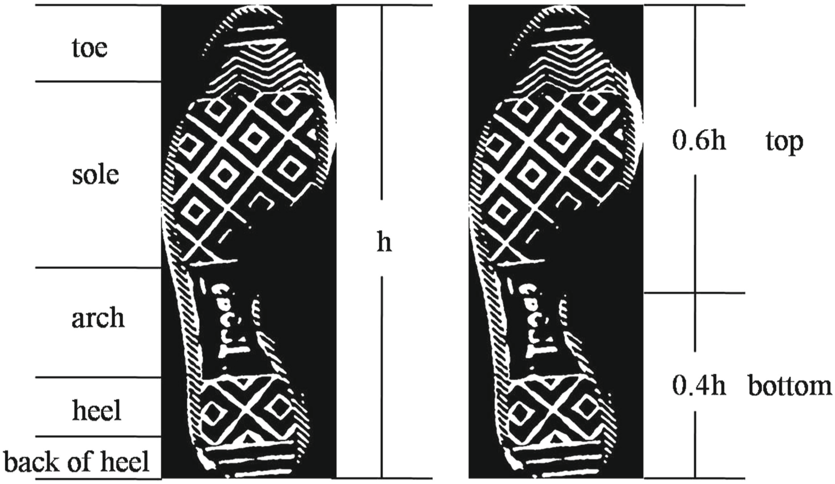

Fig. 3. Shoeprint image partition

3.3 Feature Extraction

The CNN model used in this paper is vgg-verydeep-16 net. The model has five convolutional layers (layer 1–layer 5), and each convolutional layer is followed by a max-pooling layer. At the top of the model, there are two fully connected layers (layer 6, layer 7) and a soft-max layer. Each fully connected layer includes a pre-activation layer (layer 6p for layer 6, and layer 7p for layer 7) and a ReLu transform layer (layer 6r for layer 6, and layer 7r for layer 7). The output of a certain layer can be flattened into a vector, and we call this vector a neural code. Since the neural codes obtained from first four max-pooling layers are of too high dimensions (e.g. a neural code extracted from the forth max-pooling layer is $14 \times 14 \times 512 = 100352$ dimension); therefore, we only consider five neural codes: a neural code extracted from the last max-pooling layer (layer 5m), and four neural codes extracted from fully connected layers (layer 6p, layer 6r, layer 7p, and layer 7r). These five neural codes are denoted by NC5m, NC6p, NC6r, NC7p, and NC7r. The dimension of NC5m is $7 \times 7 \times 512 = 25088$, and the dimensions of the rest neural codes are 4096.

The CNN model is applicable to 224×224 images; thus, top region and bottom region of a shoeprint image are resized to 224×224. Then, for each shoeprint image, we obtain two neural codes for top region and bottom region respectively. We consider these two neural codes as shoeprint image features.

Since the dimension of NC5m is much higher than the other neural codes used in this paper, we use PCA to reduce NC5m's dimension. In Sect. 4, we will show PCA dimension reduction can improve retrieval performance compared to original NC5m.

3.4 Region-Based Similarity Calculation

After feature extraction, feature similarity calculation is needed. While comparing two shoeprints, two similarities (top region similarity and bottom region similarity) are calculated.

In this paper, we use cosine similarity to compare two neural codes. For two neural codes n_1 and n_2, cosine similarity (sim) between n_1 and n_2 is defined as

$$sim = \frac{n_1 \bullet n_2}{\|n_1\|_2 \|n_2\|_2}. \tag{6}$$

3.5 Matching Score Calculation and Sorting

After calculating top region similarity and bottom region similarity, a matching score between two shoeprint images can be calculated. The matching score is a weighted sum of two similarities. Since the ratio between top region and bottom region was set to 6:4, the matching score is calculated using

$$score = 0.6 \times sim_t + 0.4 \times sim_b, \tag{7}$$

where, $score$, sim_t and sim_b are matching score, top region similarity and bottom region similarity, respectively.

Real shoeprints have left prints, right prints, and upside down prints. While comparing two shoe prints, they are compared four times. For a test image, its original version, mirror version, up-down flipped version, and up-down flipped + mirror version are compared to an image in the database and four matching scores are calculated. Among the four matching scores, we pick the maximum as the final matching score between the test image and the image in the database. Figure 4 demonstrates the diagram of final matching score calculation.

Once all matching scores between the test image and all images in the database are calculated, these matching scores are sorted so that images most similar to the test image are presented at the top of the ranked list. Finally, the ranked list is returned as the final retrieval result.

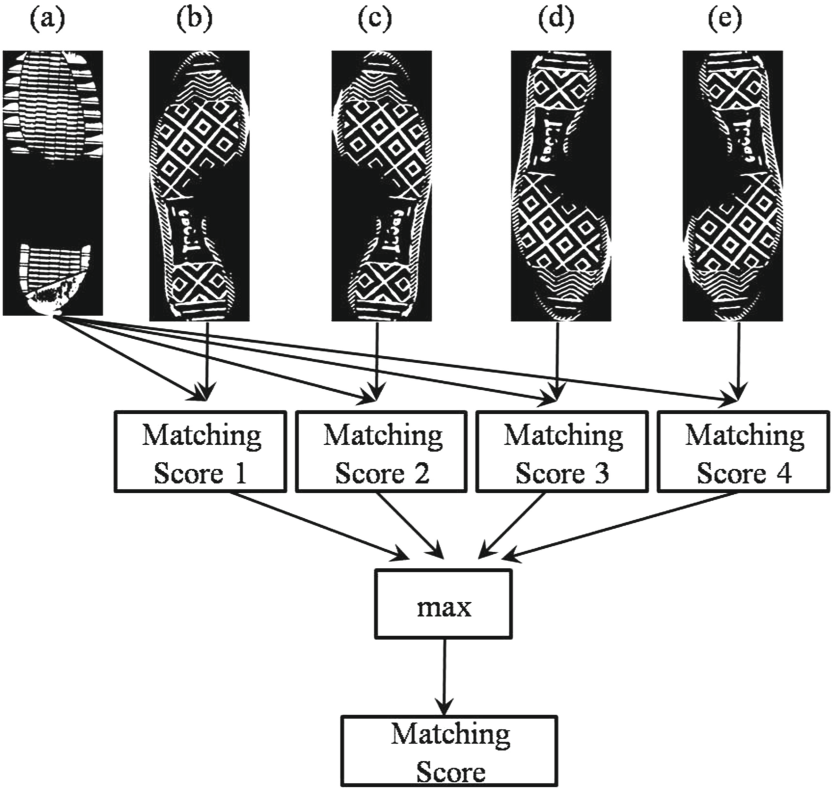

Fig. 4. Diagram of final matching score calculation. (a) An image in the database, (b) test image (original version), (c) test image (mirror version), (d) test image (up-down flipped version), (e) test image (up-down flipped + mirror version)

4 Experimental Results

4.1 Dataset and Evaluation Metrics

In order to evaluate the proposed method, we conducted several experiments. The dataset used in the experiments is composed of the test set and the database. The test set includes 1,000 images and the database includes 37,886 images belonging to 37,886 classes. For each image in the test set, there exists only one matching image in the database. Performance of the shoeprint image retrieval accuracy was evaluated in terms of cumulative match score (CMS).

4.2 Performance of Different Neural Codes

We first evaluated the performance of five neural codes described in Sect. 3. CMSs for top 1, top 5, and top 10 are shown in Table 1.

Table 1. CMS (%) of different neural codes

Neural code	Dimension	Top 1	Top 5	Top 10
NC5m	25088	64.4	83.5	87.9
NC6p	4096	59.0	77.6	81.7
NC6r	4096	59.0	77.4	81.4
NC7p	4096	57.3	76.9	81.3
NC7r	4096	56.7	76.1	80.2

As can be seen from Table 1, the neural code extracted from layer 5m performs much better than the neural codes extracted from fully connected layers. This result is consistent with [6], in which they showed that neural codes extracted from convolutional layers are more discriminative.

4.3 Performance of Different Dimensions of NC5m

In previous subsection, we have shown that neural code extracted from convolutional layer performed much better than neural codes extracted from fully connected layers. From Table 1, we can see that dimension of NC5m is much higher than the other neural codes. Therefore, we use PCA to reduce the dimension of NC5m, and CMSs for top 1, top 5, and top 10 are shown in Table 2.

Table 2. CMS (%) of different dimensions of NC5m

Dimension	top 1	top 5	top 10
Original	64.4	83.5	87.9
95%	64.9	84.2	88.7
4096	64.8	84.2	88.7
2048	64.5	83.9	88.4
1024	64.0	83.0	87.7
512	62.5	82.5	86.8
256	60.6	80.6	85.1
128	57.5	77.8	82.0
64	53.1	73.7	78.5
32	44.4	66.4	72.4
16	33.2	51.5	57.8

In Table 2, "95%" in the third line means that we use 95% of the total principle component covariance. As can be seen from Table 2, with PCA dimension reduction, performance has been improved (e.g. 95%, 4096, and 2048) compared

Table 3. CMS (%) of different methods

Method	top 1	top 5	top 10
[17]	12.8	20.9	24.5
[13]	23.5	32.2	36.6
[16]	54.8	69.8	74.5
NC5m	64.4	83.5	87.9
NC5m + PCA (95%)	64.9	84.2	88.7

to original NC5m. We also can see that 256 dimensional neural code still outperforms neural codes extracted from fully connected layers; however the performance degrades considerably when dimension is reduced to 32.

4.4 Comparison with Other Methods

We also compared our method with three state-of-the-art methods [13,16,17], and the results are illustrated in Fig. 5 and Table 3. From Fig. 5 and Table 3, we can see that the proposed methods (NC5m and NC5m + PCA) reached much higher CMSs than other methods. Among [13,16,17], only [16] has been

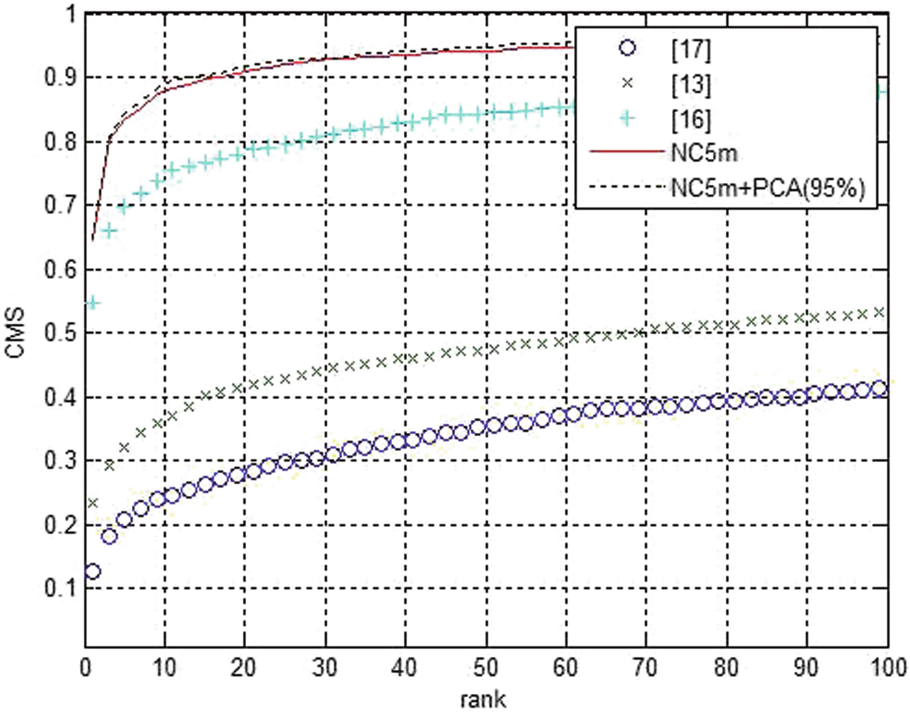

Fig. 5. CMS (%) with respect to rank for different methods

evaluated on a large-scale dataset, and none of them has been evaluated on images captured by commercial shoeprint scanners. Therefore, this experiment also showed that the proposed method is applicable to the databases collected by commercial shoeprint scanners.

5 Conclusion

In this paper, we have proposed a suspect's shoeprint retrieval method using neural codes. The proposed method used neural codes to extract shoeprint features. By using rotation compensation, shoeprint image partition, region-based similarity calculation, and weighted sum of similarities, the proposed method is simple but performs well on shoeprint images captured by commercial shoeprint scanners. As far as we know, this is the first study using the shoeprint database collected commercial scanners. Performance of the proposed method has been evaluated on a database containing 37,886 shoeprint images, and the proposed method performed much better than other state-of-the-art methods. We also showed that by using PCA dimension reduction method, performance can be improved while using a short neural code. Since we used pre-trained CNN model to extract neural codes, training a CNN model using shoeprint images and expanding its application to real crime scene shoeprint images will be the future scopes of this work.

References

1. Dalian Everspry Sci. & Tech. Co. Ltd.: http://www.footprintmatcher.com
2. Hangzhou Chancel Electronic Technology Co. Ltd.: http://www.hzchancel.cn
3. Matconvnet: http://www.vlfeat.org/matconvnet/pretrained
4. Treadfinder Homepage: https://www.treadfinder.uk
5. Abdel-Kader, R.F., Ramadan, R.M., Zaki, F.W., El-Sayed, E.: Rotation-invariant pattern recognition approach using extracted descriptive symmetrical patterns. Int. J. Adv. Comput. Sci. Appl. **3**(5), 151–158 (2012)
6. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Proceedings of ECCV, pp. 584–599 (2014)
7. Bodziak, W.J.: Footwear Impression Evidence: Detection. Recovery and Examination. CRC Press, Boca Raton (2000)
8. Bouridane, A., Alexander, A., Nibouche, M., Crookes, D.: Application of fractals to the detection and classification of shoeprints. In: Proceedings of IEEE International Conference on Image Processing, pp. 474–477 (2000)
9. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
10. Chazal, P.D., Flynn, J., Reilly, R.: Automated processing of shoeprint images based on the Fourier transform for use in forensic science. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 341–350 (2005)
11. Girod, A.: Computer classification of the shoeprint of burglar soles. Forensic Sci. Int. **82**, 59–65 (1996)

12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
13. Patil, P., Kulkarni, J.: Rotation and intensity invariant shoeprint matching using gabor transform with application to forensic science. Pattern Recogn. **42**(7), 1308–1317 (2009)
14. Pavlou, M., Allinson, N.: Automated encoding of footwear patterns for fast indexing. Image Vis. Comput. **27**, 402–409 (2009)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of International Conference on Learning Representations (2015)
16. Wang, X., Sun, H., Yu, Q., Zhang, C.: Automatic shoeprint retrieval algorithm for real crime scenes. In: Proceedings of ACCV, pp. 399–413 (2014)
17. Zhang, L., Allinson, N.: Automatic shoeprint retrieval system for use in forensic investigations. In: UK Workshop on Computational Intelligence (2005)