IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 23, 2019, accepted February 18, 2019, date of publication February 25, 2019, date of current version April 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901335

An Iterative Deep Neural Network for Hand-Vein Verification

HUAFENG QIN[1,2], MOUNIM A. EL YACOUBI[3], JIHAI LIN[1,2], AND BO LIU[1,2]

[1]Chongqing Engineering Laboratory of Detection Control and Integrated System, Chongqing 400067, China
[2]School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China
[3]Department of Electronics and Physics, Telecom-SudParis, 91011 Évry, France

Corresponding author: Huafeng Qin (qinhuafengfeng@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402063, in part by the Natural Science Foundation Project of Chongqing under Grant cstc2017jcyjAX0002, Grant cstc2018jcyjAX0095, Grant cstc2017zdcy-zdyfX0067, and Grant cstc2013kjrc-qnrc40013, and in part by the Institut Mines-Telecom.

ABSTRACT Hand-vein biometrics as a high-security pattern has received more and more attention. One of the open issues in hand-vein verification is the lack of robustness against image quality degradation, which may comprise the verification accuracy. To achieve robust verification, vein feature extraction approaches, especially vein texture segmentation, have been extensively investigated. In recent years, deep neural networks have achieved promising results in medical image segmentation and have been brought into vein verification, but current solutions suffer from two challenges for vein segmentation: 1) lacking the labeling data, which is expensive to obtain and 2) the incorrect label data obtained by manual labeling scheme or automatic labeling scheme may strongly influence parameters when the network is trained, which may degrade the verification performance. This paper proposes an iterative deep belief network (DBN) to extract vein features based on the initial label data, which are automatically generated using a very limited *a priori* knowledge and iteratively corrected by our DBN. First, a known handcrafted vein image segmentation technique is employed to automatically label vein pixel and background pixel. A training dataset is constructed based on the patches centered on the labeled pixels. Second, a DBN is trained on the resulting database to predict the probability of each pixel to belong to be a vein pixel given a patch centered on it. The vein patterns are segmented using a probability threshold of 0.5. The resulting vein features are employed to reconstruct the training dataset, based on which the network is retrained. During the iterative procedure, the incorrect labels of training data are statistically corrected, which enables DBN to effectively learn what a finger-vein pattern is by learning the difference between vein patterns and background ones. The experimental results on two public hand-vein databases show a significant improvement in terms of hand-vein verification accuracy.

INDEX TERMS Hand biometrics, palm-vein verification, deep learning, iterative deep neural network, representation learning.

I. INTRODUCTION

With the tremendous ubiquity of Internet and increasing security awareness, traditional authentication, such as passwords, personal identification numbers, smart cards, is hard to meet the requirements of convenience, reliability, and security in practical applications. For example, passwords are easy to forgot, and smart cards are easily lost, copied and forged. Under such circumstances, automatic personal verification using physiological and/or behavioral characteristics of humans, has received increasing attention. Currently, various biometrics characteristics have been investigated and applied in practical life. Broadly, they can be categorized in two categories: (1) extrinsic biometric features, i.e. face [1], fingerprint [2], iris [3], palmprint [4], hand shape [5], and handwriting [6], and (2) intrinsic biometric features, i.e. finger-vein [7], hand-vein [8] and palm-vein [9]. The extrinsic biometrics characteristics such as face, fingerprint and iris have been successfully employed by biometric verification systems for immigration clearance, financial payments, access control systems, and consumer electronics, but they are also prone to spoof attack, which degrades the

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Munir.

security level of the verification system. For example, the face biometric trait is easy to be captured and the fake photographs, recorded videos, and three-dimensional artificial face models can be employed to attack the facial recognition systems [10]. Similarly, iris recognition systems and fingerprint recognition systems are also vulnerable to be fooled by their fake version [11], [12]. Therefore, the usage of these extrinsic biometric traits results in the problems of privacy and security.

Compared to extrinsic biometric features, using the vein pattern for verification has the following advantages [13], [14]. (1) liveness verification. The vein pattern is only collected from a lively body and further achieves effective verification. (2) high security and high privacy. The vein patterns are concealed in our body; they are much harder to forge and also difficult to acquire without the consent of an individual. Therefore, vein verification provides higher security and privacy for the user.

A. MOTIVATION AND RELATED WORK

Vein patterns are the network structure of blood vessels and are concealed beneath the human skin. Some medical research works [14]–[16] have shown that the texture of blood vessels has high uniqueness for each individual [14], even for identical twins [14], [15]. Therefore, the automatic personal identification/verification using vein biometrics have been widely investigated in the past years [17]–[35]. Generally, vein patterns are difficult to observe in visible lighting. As different skin layers e.g. outermost epidermis, dermis, and subcutaneous layers have different absorption rates of infrared light, the vein patterns can be acquired by infrared illumination, which effectively avoids possible external damage, spoof attacks, and impersonation. To acquire vein images with high contrast, the infrared light with above 850 nm wavelength is usually employed in current works [7]–[9], [18]. Normally, blood vessels extend in our body, showing a clear network and good connectivity. However, during the vein capturing procedure, various factors such as environmental illumination [20]–[22], ambient temperature [7], [22], [23], light scattering in imaging finger tissues [24], [25] affect the imaging quality of vein patterns. Therefore, the acquired vein images include not only vein patterns but also noise and irregular shadowing, so it is very difficult to completely segment the vein features. Usually, some vein patterns are missed and false vein patterns are generated in some regions from the vein image, which degrades the distinctiveness, resulting in low verification accuracy. Currently, a number of methods have been proposed to segment the vein network from vein image for verification. For example, various handcrafted descriptors are designed based on the prior knowledge to extract the vein geometry structure. In these works, some researchers observe that the cross-profile of finger-vein patterns show a valley shape and propose to detect the valley for finger-vein pattern segmentation. The representative approaches include the line tracking methods [26]–[29], [36] and curvature-based measures [9], [22], [23], [30], [34], [37], [38]. Other ones assume that a vein pattern is supposed to be a line-like texture in a predefined neighborhood region. Many works based on Gabor filters [7], [24], [31], [33], matched filters [39], wide line detector [21] and neural networks [32] are proposed to extract the line-like texture for finger-vein verification. Among the approaches described above, the handcrafted methods depend on the assumption distributions such as valleys and line segments, so they suffer from following problems [17]. (1) These assumptions are not always effective to extract the vein patterns. (2) it is impossible to describe attributes of all distributions created by the pixels. (3) It is difficult to develop a mathematical model to effectively model the distributions such as valleys or line segments. As an effective solution, the deep learning based approaches, not requiring any assumption, have been directly used to learn robust features from raw pixel images and successfully applied for various computer vision tasks [40]–[44]. Also, some researchers brought it into medical image segmentation [45]–[49] such as retina image segmentation, brain segmentation, and neuronal membranes segmentation. As Deep Neural Networks such as Convolutional Neural Network (CNN) [45], Deep Belief Network (DBN) [48], and Auto-Encoder (AE) [49], have the powerful capacity of feature representation, the promising results are achieved if the learning objective (ground truth) is provided for them (DBN is an unsupervised model but it can initialize an MLP which is supervised). Therefore, they have shown super performance compared to hand-crafted approaches after training or fin-tuning them using ground truth. Different from the medical image segmentation tasks, vein feature extraction for verification faces a challenge that there is no ground-truth for existing works. To overcome this problem, in work [17], a scheme is proposed to combine seven vein segmentation baselines to produce ''ground-truth'' (vein and background labels) for finger-vein verification. Combining as many existing baselines as possible may generate more accurate labels. However, there still exist many incorrect labels in the ''ground truth'' even if all baselines are employed for labeling or the data is manually labeled, so current performance may be limited for vein verification. In addition, there may be not enough baselines for generation of accurate labels in other application regions. Therefore, how to learn a good representation of the vein patterns, from weakly accurate ''ground truth'' is still an issue for vein segmentation, which has motivated us to develop a more robust learning approach for real-world vein verification.

B. OUR WORK

To overcome this problem, in this paper, a deep learning based model is trained by unsupervised learning and by reinforcement learning from self-correcting for vein segmentation. Our approach can predict the probability of a pixel to belong to a vein pattern using very limited knowledge. The main paper contributions are summarized as follows: (1) This paper

proposes a deep learning based approach which is iteratively trained to extract hand-vein features and achieve promising performance. First, different from existing approaches based on manual labeling schemes, we employ an existing hand-crafted image segmentation technique to extract the vein network from an image and the resulting binary image is used to automatically label each pixel. Secondly, a training set is constructed based on patches centered on labeled pixels and then input into DBN for training. In the testing phase, the patch of each pixel is taken as input for DBN to predict the probability of the pixel to belong to a vein pattern. The vein pattern is segmented using a probability threshold of 0.5. Thirdly, each pixel is reassigned a label by the resulting segmented vein image. Similarly, a new training dataset is built to re-train the DBN model. Repeatedly, the DBN is iteratively trained to correct the labels and the resulting labeling training data enable it to learn robust features for vein segmentation. Therefore, the proposed approach can achieve high accuracy even if there exists few incorrect labels for the training patches. (2) We carry out rigorous experiments to investigate the capacity of self-correcting in our work. In the experiments, we employ several baseline approaches to generate the initial training patches, which may include few incorrect labels. Then, the performance of DBN after iteratively correcting is reported. The experimental results show that the proposed approach statistically correct the incorrect labels and significantly improve hand-vein verification accuracy. To the best of our knowledge, there has not been any work to study self-correcting capacity of DBN. (3) This paper present a systematic and comparative analysis of the proposed approach in both contact and contactless hand-vein imaging environment and evaluate the performance of our method over several baselines. Experimental results show that the proposed model is able to extract the vein patterns from raw images in a robust way and achieve better performance than existing approaches.

II. PALM-VEIN ROI EXTRACTION

Contactless hand-vein identification has numerous advantages, such as more hygienic, non-contact acquisition, and enhanced palm layout freedom, so it is easy to be accepted and used by users. However, the acquired hand-vein images in contactless imaging not only present scale, translational and rotational variations, but also contain a large background without discriminative patterns. Generally, matching these images with such variations and background contributes a lot of errors and increases time cost. To achieve accurate hand-vein verification, it is necessary to extract the region of interest (ROI) and aliment it with respect to scale, translation and rotation. Fig.1, Fig.2 and Fig.3 show the preprocessing procedure.

A. HAND REGION SEGMENTATION

As shown in Fig.1(a), the acquired finger-vein image contains not only the hand but also large background. For most hand images, as there exists large contrast between the hand region and the background region, a global threshold for each image is computed by OTSU [50] to extract the hand region. However, in some regions of few acquired hand images, the background and foreground (hand) have similar contrast and it is difficult to separate the hand from the background. Therefore, an edge operator i.e. Sobel edge detector [51] is employed to separate palm and background. First, we extract the edges from the acquired image (Fig.1(b)). Second, the acquired image is subject to binarization using a threshold obtained from OTSU method (shown in Fig.1(b)), and the edge map is subtracted from the binarized image (Fig.1(c)). Third, the binary mask image (Fig.1(e)) only including hand region is obtained by filling holes and filtering the isolated blobs (if any) in the resulting images. Finally, we process the binary mask image (Fig.1(e)) by a median filter to obtain the smoothness boundary (Fig.1(f)) for segmentation of ROI.

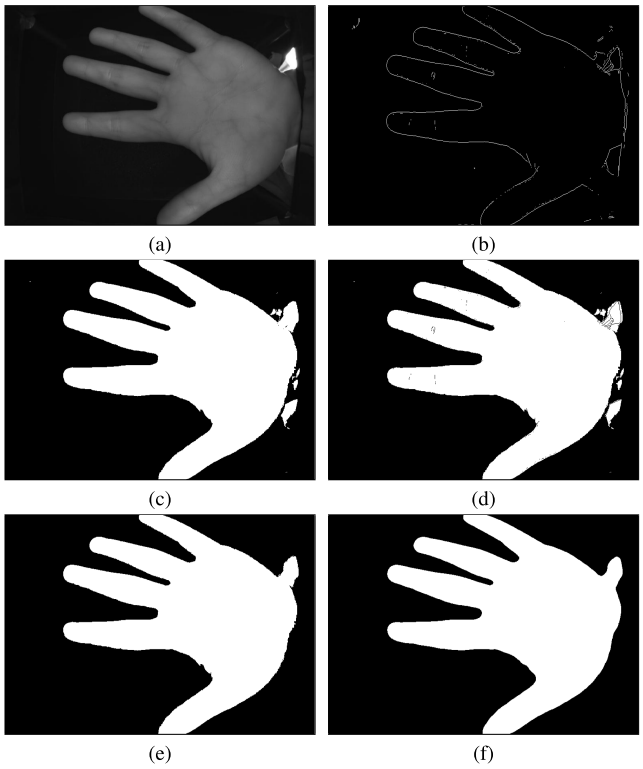

FIGURE 1. Segmenting results for hand region. (a) Original hand image; (b) The edge image extracted from (a). (c) Binary image segmented from (a). (d) Edge image subtracted from (c). (e) Binary image obtained from (d) by filtering isolated blobs and filling hole. (f) Hand region obtained from (e) by median filtering.

B. KEY POINT EXTRACTION

To extract the palm ROI, existing approaches [9], [34] proposed to detect the key points in the hand image. In these works, the pixels connecting two adjacent finger regions are defined as key points for ROI extraction (show in Fig.2(f)). In other words, the key-points are the tangent points of the boundary of two adjacent finger regions. So, five fingers result in four key points. To detect them, the approaches [9], [34] transform the hand contour into

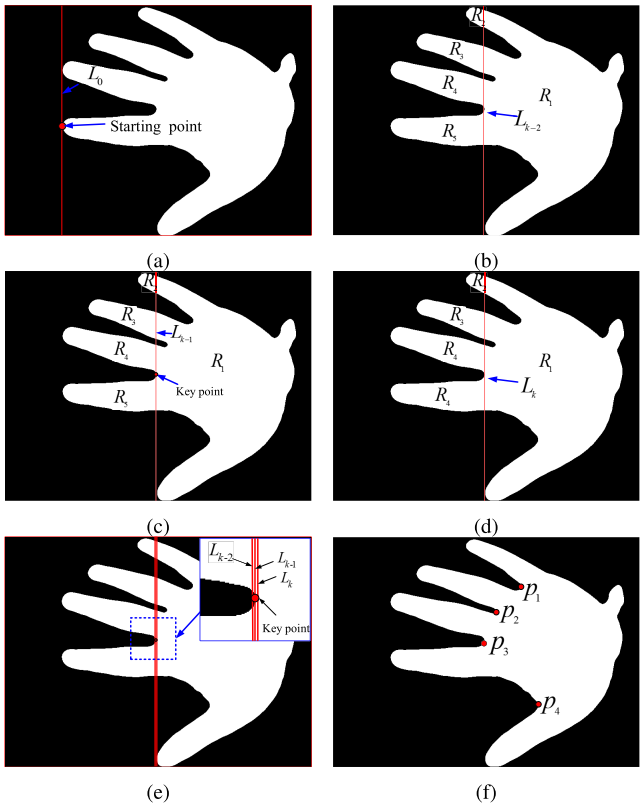

FIGURE 2. Key point detection results. (a) Original hand image (The corruption will start from first non-zeros pixel). (b) The image corrupted by L_{k-2}. (c) The image corrupted by L_{k-1}. (d) The image corrupted by L_k. (e) Spatial relations of L_{k-2}, L_{k-1} and L_k. (f) Detected four key points in hand image. Note that (d) denotes current corrupted image, and (b) and (c) are its last two corruption images. In (b) and (c), there are five connected regions (R_1, R_2, R_3, R_4 and R_5) while there are four connected regions (R_1, R_2, R_3 and R_4) in (d). In (e), the rectangle region in upper right corner is the enlargement of blue dashed box.

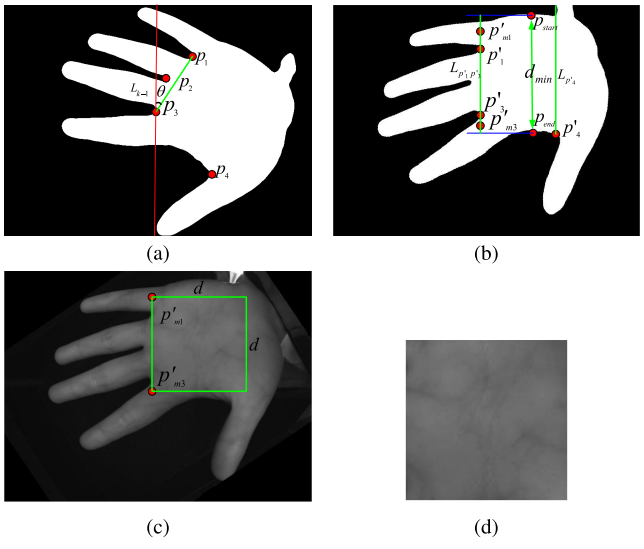

FIGURE 3. Normalized results. (a) Computing orientation based on two key points. (b) Orientation normalization of binary hand image. (c) ROI extraction. (d) Scale normalization of ROI.

a curve in a X-Y axis plane by computing the distances between all points on the hand contour and a predefined pixel. There, we proposed to directly detect the four key points in the two-dimensional hand image. First, a vertical line is employed to corrupt the mask image. To reduce detection time, the corruption operator starts from first non-zeros pixels on the left (Fig.2(a)) and ends after detecting all key points i.e. 4 points. For each step, the mask image is corrupted into several regions (R_1, R_2, R_3, R_4 and R_5 in Fig.2(b)-Fig.2(c)). To facilitate description, the current corrupted image is denoted as f_k (Fig.2(d)) and its last two corrupted images are presented as f_{k-2} (Fig.2(b)) and f_{k-1} (Fig.2(c)). Similarly, the corrupted lines in f_k, f_{k-1} and f_{k-2} are denoted as L_k, L_{k-1} and L_{k-2} (the red lines in Fig.2(b)-Fig.2(d)), which result in a number of isolated regions (e.g. five regions in Fig.2(b) and Fig.2(c), and four regions in Fig.2(d)). Comparing f_k with f_{k-1}, if some regions in f_{k-1} are merged together in f_k, there exist one or more key points in L_{k-1}; Otherwise, continue to corrupt remaining region. For example, the R_4 and R_5 in Fig.2(c) are merged into one region R_4 in Fig.2(d), so there is a key point in corrupted line L_{k-1}. To extract it, the corrupted positions L_k, L_{k-1} and L_{k-2} are shown in same image to facilitate comparison (as shown in Fig.2(e)). From Fig.2(e), we observe that the key point in L_{k-1} (tangent points) and its left adjacent point in L_{k-2} have different pixel values (the values are 1 for key point and 0 for its left adjacent point, respectively). So it is detected by computing difference of pixel values in L_{k-1} and L_{k-2}. Repeate the same operator until extracting all key points. The approach is detailed in algorithm 1. Fig.2(f) shows the detected key points (tangent points).

C. ORIENTATION NORMALIZATION AND ROI EXTRACTION

After obtaining the key points p_1, p_2, p_3, and p_4 (Fig.2(f)), similar to approaches [9], [34], we employ two points p_1 and p_3 for orientation normalization. As shown in Fig.3(a), a line is determined based on two key points and the angle θ between it and the vertical line is computed by

$$\theta = \arg\tan(x_{p1} - x_{p3})/(y_{p1} - y_{p3}) \tag{1}$$

where (x_{p1}, y_{p1}) and (x_{p3}, y_{p3}) are the coordinates of points p_1 and p_3. The orientation variation is normalized by rotating the hand image in an angle θ, as shown in Fig.3(b).

Generally, the scale variations in the contactless image may be quite large. To normalize all images into same scale, it is better to adaptively select the location and size of the ROI according to reference points in the palm instead of employing any empirical parameters. To include rich discriminative information, we compute the minimal width of the palm regions. First, a straight line $L_{p'_1p'_3}$ is determined based on p'_1 and p'_3 (as shown in Fig.3(b)). In addition, a vertical straight line $L_{p'_4}$ is determined by point p'_4. Then, we compute the shortest distance d_{min} between upper boundary and lower boundary of palm region from $L_{p'_1p'_3}$ to $L_{p'_4}$ and obtain two points p_{start} and p_{end}, as shown in Fig.3(b). Let $(x_{p'_1}, y_{p'_1})$ and $(x_{p'_3}, y_{p'_3})$ be the coordinates of points p'_1 and p'_3, and $(x_{p_{start}}, y_{p_{start}})$ and $(x_{p_{end}}, y_{p_{end}})$ be the coordinates of points p_{start} and p_{end}. The coordinate of point p'_{m1} is computed by

Algorithm 1 Key point detection for palm image

Input: The preprocessing palm-vein image $f(x, y)$ (Fig.1(f)); The number of key point K.

Output: Palm-vein key points matrix $M(x, y)$;

Step 1: Corrupting input palm-vein image $f(x, y)$ using a vertical line from current pixel. The current pixel is initialed by first non-zeros pixel on the left (as shown Fig.2(a)).

Step 2: Labeling each connected region in the resulting corrupting image (Figs.2(b)-(d)) by morphological processing, and comparing current corrupted image (Fig.2(d)) with last corrupting image (Fig.2(c)). If some regions in Fig.2(c)) are merged in current corrupted image (Fig.2(d))), go to step 3. Otherwise, go to step 1.

Step 3: Detecting the key points.

Step 3.1: Recording the corrupted image and pixel values in corruption lines. To facilitate description, we denote the current corruption line as L_k (Fig.2(d)) and its last two corruption lines as L_{k-1} and L_{k-2} (as shown in Fig.2(b) and Fig.2(c)). Similarly, the current corrupted image and its last two corrupted image are presented by f_k, f_{k-1} and f_{k-2} (Fig.2(b), Fig.2(c) and Fig.2(d)). The pixel values in L_k, L_{k-1} and L_{k-2} are stored in vectors V_k, V_{k-1} and V_{k-2}.

Step 3.2: Computing the difference between vectors V_{k-1} and V_{k-2} to detect key point. The vector V_{k-2} is subtracted from vectors V_{k-1} and the resulting vector V_d is subject to binarization to obtain a binary vector V_b. The no-zeros pixel points in V_b is labeled as connected components by morphological processing.

Step 3.3: Assume there are N connected components V_b^n $(n-1, 2 \ldots N)$ in V_b. We select all pixel values in connected component V_b^n and add them into the corresponding corruption position in f_{k-1} (red line L_{k-1} in Fig.2(c)).

Step 3.4: We compare the amount of connected components in f_{k-1} before and after adding connected components V_b^n. If the amount of connected components decreases, there exists key point in V_b^n, and go to step 3.5; Otherwise, go to step 3.3.

Step 3.5: Selecting a point from V_b^n as key point. The pixel points in V_b^n are sorted in ascending order along their Y-axis (vertical direction) to obtain vector $V_b'^n$. Then the middle point in $V_b'^n$ is selected as key point and its corresponding coordinate in two dimensional image is recorded in a matrix $M(x, y)$.

Step 3.6: Repeated step 3.3 to step 3.5 until the key points in N connected components are stored in matrix $M(x, y)$, and then we go to step 4.

Step 4: Computing the number of key points in matrix $M(x, y)$. If the number of key point is equal to K, stop iteration and output matrix $M(x, y)$; Otherwise, go to Step 5.

Step 5: The corruption line continues to move from the current position (Fig.2(d)) to remaining region and then go to step 2.

Return $M(x, y)$;

$(x_{p_1'}, (y_{p_1'} + y_{p_{start}})/2)$. Similarly, the coordinate of point p'_{m3} is $(x_{p_3'}, (y_{p_3'} + y_{p_{end}})/2)$. The distance d between p'_{m1} and p'_{m3}, as shown in Fig. 3(b) is calculated as follows

$$d = \sqrt{(x_{p_1'} - x_{p_3'})^2 + ((y_{p_1'} + y_{p_{start}})/2 - (y_{p_3'} + y_{p_{end}})/2)^2} \tag{2}$$

A ROI with size of $d \times d$ is segmented from the hand image (Fig.3(c)). To facilitate verification, all images are resized to 100×100 for scale normalization, as shown in Fig.3(d).

III. ITERATIVE DEEP NEURAL NETWORK FOR HAND-VEIN PATTERN EXTRACTION

Recently, deep neural networks such as Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN) have shown a strong ability to learn effective feature representations from input data and have been successfully applied for computation vision tasks [40]–[42], [48]. However, current deep learning based solutions mainly depend on domain knowledge such as manually labeling. We will propose a new scheme to train them for hand-vein verification with very limited domain knowledge. In this section we take DBN as an example to introduce our approach because it involves less hyper-parameters than CNN. Firstly, we show the theoretical background of DBN and a practical architecture of our deep neural network model. Then, an iterative deep learning model is proposed to extract the hand-vein features for verification.

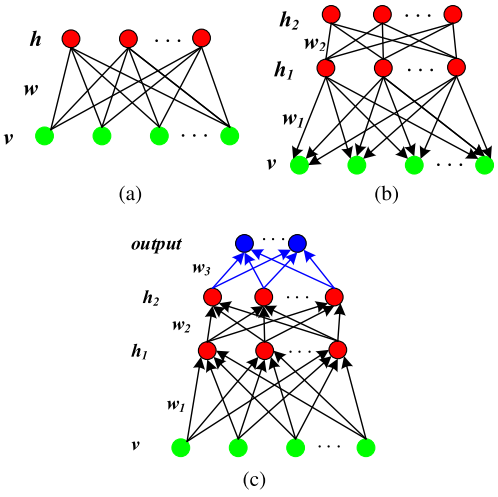

FIGURE 4. The graphical model representations for (a) an RBM, (b) a two hidden layer DBN and (c) a neural network.

A. DBN ARCHITECTURE

A Deep Belief Network (DBN) [52] is a probabilistic generative model with deep architecture. A DBN is constructed by stacking a predefined number of restricted Boltzmann machines(RBMs), as shown in Fig.4. An effective greedy layer-wise algorithm is used to pre-train each layer

of networks. In a binary RBM, an energy is defined as follows

$$E(v, h; \theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{i,j} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} b_j h_j \quad (3)$$

where model parameters $\theta = \{W, b, a\}$ and $v_i, h_j \in \{0, 1\}$. W is the weight with $V \times H$ dimensions, where w_{ij} is the symmetric interaction term between visible unit v_i and hidden unit h_j. b_i and a_j are the bias term, I and J are the numbers of visible and hidden units. The network assigns a probability to every possible visible-hidden vector pair via the energy function.

$$p(v, h; \theta) = \frac{\exp(-E(v, h; \theta))}{Z} \quad (4)$$

The normlization term Z is computed by

$$Z = \sum_{v,h} \exp(-E(v, h)) \quad (5)$$

The conditional probabilities can be obtained as

$$P(h_j = 1|v; \theta) = \sigma(\sum_{i=1}^{I} w_{i,j} v_i + a_j) \quad (6)$$

$$P(v_i = 1|v; \theta) = \sigma(\sum_{j=1}^{J} w_{i,j} h_j + b_i) \quad (7)$$

where $\sigma(x) = 1/(1 + exp(x))$

After computing the gradient of the log likelihood function $logp(v; \theta)$, we update the parameters of the RBM by the following rule.

$$\Delta W_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \quad (8)$$

here ϵ is the learning rate,$\langle v_i h_j \rangle_{data}$ is the expectation observed in the training set and $\langle v_i h_j \rangle_{model}$ is the expectation of the distribution defined by the model. It is exponentially expensive to compute the exact $\langle v_i h_j \rangle_{model}$ so the Contrastive Divergence (CD) technique is employed to approximate the gradient by replacing $\langle v_i h_j \rangle_{model}$ with $\langle v_i h_j \rangle_{recon}$, which is easier and faster to compute [53].

We stack RBMs to construct DBN (as shown in Fig.4(b)) and train it using a greedy layer-wise pre-training method [54]. Then the bottom up weights of the resulting DBN can be used to initialize the weights of a multi-layer feed-forward neural network with a Softmax output layer (as shown in Fig.4(c)), which can then be discriminatively fine-tuned by back-propagating error derivatives.

B. ITERATIVE DBN FOR FINGER-VEIN EXTRACTION

In many applications [40]–[49], the labels for classification are labeled manually and a robust feature representation is extracted by training the DBN. So, once the networks are successfully trained, the performance mainly depends on the prior knowledge associated with the manual labels. However, the labels are not always provided or are expensive to obtain. Besides, the manual or automatic labeling schemes are prone to errors, so the performance of DBN is still limited even when it is trained successfully. To overcome this problem, in this section, the DBN are iteratively trained to model the distribution of vein pixels with very limited prior knowledge. In other words, the performance of the proposed model weakly depends on the prior knowledge i.e. labels. First, we label each pixel of a training image as either vein or background using one baseline verification system. For each pixel, the window centered on it is input to DBN for training. The missing pixels are synthesized by mirroring if a window extends outside of the image boundary. The output of the last DBN layer is interpreted as the probability of the patch center pixel to belong to a vein pattern. Applying DBN to all the pixels in this way, the vein features are extracted based on a probability threshold of 0.5. The resulting binarized images are subsequently used to re-label pixels (label correction) and a new training set is constructed to retrain DBN for segmentation. This training process is repeated until the DBN achieves satisfying verification accuracy on a validation dataset (detailed in Section IV-E). During the iterative training procedure, the incorrect pixel labels are statistically corrected which enables DBN to achieve better performance. The proposed DBN model is detailed in Algorithm 2.

1) LABELING FINGER-VEIN PATTERN

In this section, we assign a label for each pixel using one baseline. For an image f (Fig.5(a)), we segment it into vein features and background by four baseline algorithms i.e. Repeated line tracking [19], Gabor filter [4], Maximum principle curvature [16], and Hessian phase [14]. Fig.5 shows an example of the extracted patterns by the various approaches. For a pixel (x, y) in each binary image, its value can be treated as its label. As shown in Fig.5(b)-Fig.(d), the value of vein pixel (white region) is 1 and the value of background pixel (black region) is 0.

We label pixels (0 and 1 denote background and vein pixels respectively) in a grayscale finger-vein image (Fig.5(a)) by a binary image (as shown in Fig.5(b), Fig.5(c), Fig.5(d), or Fig.5(e)). Based on resulting labels, an image is split into different patches with size of $N \times N$ which are input into DBN to learn to extract vein features.

2) TRAINING AND TEST OF DBN

For DBN training, the input data consists of patches with size $N \times N$. The dimension of output is decided according to the number of classes to predict. After forward propagation through the network layers, the output from each layer of DBN is the representation of an input image. After training, given a patch either from a training or a test image, the DBN computes the probability of its center to belong to a vein feature, and labels it according to the winning class (i.e. a probability threshold of 0.5). In our experiments, we build a database consisting of 100000 patches for each class to train the DBN. First, all pixels are labeled based on vein network (binary image) from any one baseline. 100000 vein patches and 100000 non-vein patches are randomly selected and a

Algorithm 2 Iterative DBN for hand-vein feature extraction.

Input: The original hand-vein image $f(x, y)$, training dataset Ω, and validation dataset C;

Output: Hand-vein segmentation image $F(x, y)$;

Step 1:Segmenting the hand-vein feature based on one baseline.

Step 2: Assigning a label for each pixel based on the resulting binary vein image (0 and 1 denote background and vein pixels respectively), and selecting some patches centered on the vein pixels as positive samples and patches centered on the background pixels as negative samples to form the training set $\mathbb{A}$.

Step 3: Training DBN by stochastic gradient descent.

Step 4: Input images from C into DBN to obtain probability maps, which are binarlized to extract vein structure using a threshold of 0.5.

Step 5: Matching the resulting binary images to compute the recognition accuracy on validation dataset. If the highest verification accuracy on a validation dataset is achieved, stop iteration and go to Step 7; Otherwise, go to Step 6.

Step 6: Input an image from Ω into DBN to obtain a probability map $P'(x, y)$, based on which we re-label each pixel and construct training set. For example, we select some patches with high probability belonged to vein as positive samples ($P'(x, y)$>0.6) and patches with high probability belonged to background as negative samples ($P'(x, y)$<0.4) to construct the training set $\mathbb{A}'$. Then, go to Step 3.

Step 7: Input image $f(x, y)$ into DBN to obtain enhanced image and the resulting image is subject to binarilization to generate the vein feature image $F(x, y)$.

Return $F(x, y)$;

patch based training set is constructed for DBN training. For an input patch, the DBN outputs the probability of its center to belong to a vein pattern. When computing the probability of all pixels in an image, the DBN outputs a probability map. Secondly, we select 100000 patches with higher probability (i.e. larger than 0.6) as belonging to vein patterns and 100000 patches with lower probability (i.e. less than 0.4) as belonging to vein patterns and build a new training set to retrain our DBN. Subsequently, a probability map obtained from DBN is employed to reconstruct the training set. The training is done in a recursive way. The iterative process will be stopped by verification accuracy on validation dataset (as detailed in Section IV-E). Finally, the vein network is extracted by binarizing the output of last DBN layer using a threshold of 0.5. The detailed training process is depicted in Algorithm 2.

C. MATCHING

We compute the matching score between an enrollment image and an unlabeled testing image for verification. Let us assume

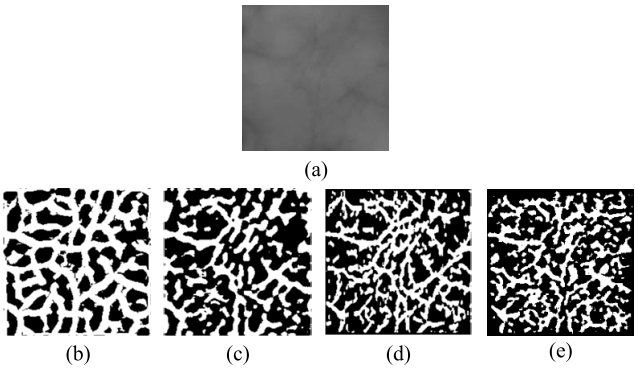

FIGURE 5. Segmented results. (a) Original hand-vein image, and (b) vein features extracted from (a) using Gabor filter, (c) vein features extracted from (a) using Hessian phase, (d) vein features extracted from (a) using Maximum principle curvature, and (e) vein features extracted from (a) using Repeated line tracking.

E and F are enrollment and test binarized feature images with size of $x \times y$, respectively. The width and height of E are extended to $2w + x$ and $2h + y$, and then its expanded image $\bar{E}$ is obtained and expressed as:

$$\bar{E}(i,j) = \begin{cases} E(i-w, j-h) & \text{if } 1+w \le i \le x+w, \\ & 1+h \le j \le y+h \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

The matching score between E and F is obtained by

$$d(E,F) = \min_{0 \le m \le 2w, 0 \le n \le 2h} \frac{\sum_{i=1}^{x}\sum_{j=1}^{y} hamdis(\bar{E}(i+m, j+n), F(i,j))}{\sum_{i=1}^{x}\sum_{j=1}^{y} \ominus(\bar{E}(i+m, j+n), -1)} \tag{10}$$

where *hamdist* denotes the hamming distance between two templates, i.e., summation of the number of pixels with different values and

$$\ominus(U,V) = \begin{cases} 1 & if\ U \ne V \\ 0 & otherwise. \end{cases} \tag{11}$$

The parameters w and h which control the distance of translation in horizontal and vertical directions are experimentally set to 30 and 30, respectively.

However, the matching score computed from Eq.(10) is not robust to local variations such as local rotation and translation, which usually exist in contactless captured hand-vein images. To overcome this problem, local matching scores are generated to achieve hand-vein verification. For template E and testing image F, we divide each of them into various local patches. The obtained kth patches from E and F are presented as E_k and F_k, $k = 1, 2, \ldots, K$. Then, we employ Eq.(10) to match the corresponding partitions and generate local matching scores. Finally, the local matching scores are combined to obtain the global matching score for verification by following Eq.(12).

$$d'(E,F) = \frac{1}{K}\sum_{k=1}^{K} d(E_k, F_k) \tag{12}$$

IV. EXPERIMENTS AND RESULTS

To evaluate the performance of the proposed approach for vein verification, we carry out experiments on two public hand-vein databases which are collected using contactless and contact device at different times. A classical approach (i.e. Repeated line tracking [26]) and a state of the art (i.e. Gabor filters [7]) methods have shown promising results on finger-vein databases. Also, the extraction of hand-vein features using Hessian phase [9] and Maximum principle curvature [34] achieved high performance for hand-vein verification. To test our approach, we employ four baselines to label pixels and build four training sets, each of which is employed as initial training set to iteratively train our DBN for palm-vein verification. To simplify the description, we respectively denote them as ''The propose approach+ Repeated line tracking'', ''The proposed approach+ Maximum principle curvature'', ''The proposed approach + Gabor filters'' and ''The proposed approach+ Hessian phase''. Then, the corresponding performance is shown in the following experiments. Also, we compare all four baselines mentioned above to get more insights into the problem of hand-vein verification.

A. DATABASE

1) DATABASE A

The CASIA Multi-Spectral Palmprint Image Database [55] includes 7,200 palm images captured from 100 subjects using a contactless imaging device. The images were captured in two separate sessions with a minimum interval of one month. In each session, the left hand and middle hand of each subject provided 3 image samples respectively. Each sample contains six palm images which are captured at the same time where the six spectra are 460nm, 630nm, 700nm, 850nm, 940nm and white light respectively. Therefore, there are 18 (3 samples $\times$ 6 spectra) images for each subject in one session. As the focus of our work is on palm-vein verification, and the palm-vein images are largely observed in NIR, only the images that were acquired under 850nm wavelength illumination are employed to evaluate the performance of our approach. The palm-vein dataset consists of 1200 palm-vein images (100 subjects $\times$ 2 hands $\times$ 3 samples $\times$ 2 sessions) with size of 768×576. As the acquired image using a contactless imaging device includes more variations such as translation, rotation, scale and uneven illumination, the acquired finger vein images are firstly subjected to preprocessing steps before feature extraction. In our experiments, the region of interest (ROI) image is extracted, and then translation, orientation and scale alignment are carried out using our preprocessing method described in Section II. Fig.9(a) and Fig.9(b) show the original palm-vein image and normalized palm-vein image.

2) DATABASE B

The PolyU multispectral palmprint Database [4] was collected from 250 volunteers under four illuminations e.g. blue, green, red and near-infrared (NIR), in two sessions, separated by about 9 days. Each volunteer provided 24 images from left hand and right hand (12 for each hand) in each session under each illumination. 6000 NIR images (12 NIR images $\times$ 2 hands $\times$ 250 subjects) are employed to test our approach as we aim to perform vein verification. Original images have the spatial resolution of 288×352. In their work, as the captured images include background, our ROI extraction algorithm is applied to the Blue band to find the ROI coordinate system.

B. EXPERIMENTS SETTING

All the experiments are implemented in Matlab and conducted on a high performance computer with 8 Core E3-1270v3 3.5 GHz processor, 16GB of RAM, and a NVIDIA Quadro GTX1070 graphics cards. The public database is divided into three sub-datasets for training, validation and testing, respectively, to test our approach. For database A, there are only 100 subjects associated with 200 hands to provide 1200 images in two sessions, as described in IV-A. The different hands are treated as different classes, based on which we split the dataset into three subsets: training dataset with 300 (50 right-hands $\times$ 6 images) images, validation dataset with 300 (50 right-hands $\times$ 6 images) images and test dataset with 600 (100 left-hands $\times$ 6 images) images. To simplify the description, the three datasets are denoted as dataset A1, A2 and A3, respectively. For database B, there are 500 palms associated with 6000 images. Similarly, there are 2400 (200 hands $\times$ 12 images) images in training set B1, 600 (50 hands $\times$ 12 images) images in testing set B3, and 3000 (250 hands $\times$ 12 images) images in testing set B3. To extract vein patterns, the DBN is trained as follows: Firstly, the vein and background image pixels from the training set are labeled using the scheme described in subsection III-B-1). Then, to train the DBN, we select patches centered on vein pixels as positive samples and patches centered on background pixels as negative samples. In the experiments, for each image in database A and database B, about 334 and 43 vein pixels are select as positive examples, and we use the same amount of pixels randomly sampled (without repetitions) among all background pixels, which results in 200,000 training examples in total (100,000 positive examples and 100,000 negative examples). The validation sets A2 and B2 are used to select parameters, and the datasets A3 and B3 are employed for testing.

C. PARAMETER SELECTION FOR DBN

In this section, we modify the parameters of a basic DBN network to investigate the best trade-off between accuracy and speed. In our experiment, the basic network consists of two hidden layers, one input layer, and one output layer (as shown in Fig.4(c)). The number of units in the two hidden layers are 100 and 100 respectively. The number of neurons in the input layer is decided by the dimension of the input vector, and the dimension of output is decided upon according to the number of classes to be predicted. The size of training patch is initialized to 15×15 (resulting in a 225-dimensional

input vector) and vein segmentation is a binary classification problem (vein and background), so the number of units in the input layer and output layer are 225 and 2, respectively. To simplify description, we denote the basic DBN as 225-100-100-2. In addition, to determine the architecture of DBN, the iteration step is fixed to 1 and the classic repeated line tracking approach [19]) is employed to label pixels for parameter selection of DBN.

1) NUMBER OF LAYERS

In this section, to determine the layers of DBN, we shrink and extend the basic network by reducing and adding some hidden layers. For example, two networks (225-200-200-200-2,225-200-2) are created by adding and reducing a hidden layer to the basic network (225-200-200-2), respectively. The two new DBNs and the basic one are compared by the following experiments. The three DBNs above (225-200-2, 225-200-200-2, 225-200-200-200-2) are trained to extract the vein patterns of images in dataset A2. The genuine matching scores are computed by matching the images from same palm while the impostor matching scores are computed by matching images from different palms. Therefore, there are 150 (50 × 3) genuine scores and 3675 (50 × 49 × 3/2) impostor scores. The False Rejection Rate (FRR) is computed by genuine scores and the False Acceptance Rate is computed by impostor scores. The Equal Error Rate (EER) is the error rate when FAR is equal to FRR. Table 1 shows the EER and the average feature extraction processing time from one hand-vein image for different DBNs. From Table 1, we observe that, compared to the two-hidden layers network, the three-hidden layers networks achieve the same accuracy on the validation set, but the time cost increases while the two-hidden layers network achieves similar performance with less time.

TABLE 1. Results of using different layer numbers.

Methods	225-200-200 -200-2	225-200 -200-2	225-200 -2
EER(%)	1.41	1.41	1.78
Time(sec)	1.21	1.05	0.79

2) NUMBER OF KERNELS

In general, increasing neurons in hidden layers results in better verification accuracy, but it will lead to large time cost accordingly. We change the neuron number in the basic network and create a larger network 225-400-400-2, and two smaller networks 225-100-100-2, 225-50-50-2. Then, the ERRs and computation time of the three networks on dataset A2 are reported in this section. Also, the experimental results of the basic network are illustrated in Table 2 to facilitate comparison. From the experimental results in Table 2, we see that the networks 225-200-200-2 and 225-100-100-2 achieve the same EER. However, the smaller network 225-100-100-2 requires less the computation time.

TABLE 2. Results of using different neuron numbers.

Methods	225-400 -400-2	225-200 -200-2	225-100 -100-2	225-50 -50-2
EER(%)	1.41	1.41	1.41	1.85
Time(sec)	1.40	1.05	0.85	0.81

TABLE 3. Results of using different patches.

Patch Size	19×19	15×15	11×11	7×7
EER(%)	2.00	1.41	1.33	1.85

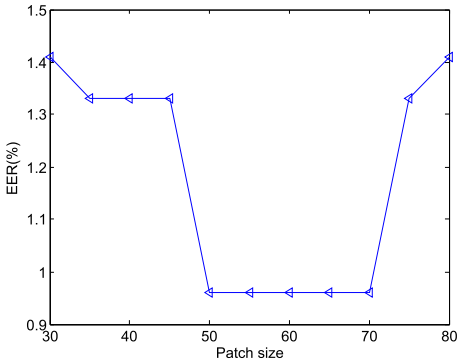

FIGURE 6. Relationship between patch size and EER.

3) PATCH SIZE

The selection of the patch size for DBN is critical for achieving high performance. If the selected size is too small, the patch may represent noise but dominate vein feature. In other words, the extracted image includes more noise which produces mismatch errors. On the contrary, too large patches include more global information than needed as vein patterns that are far away from the center pixel may actually confuse DBN training. In this section, an appropriate patch size for DBN is determined experimentally and Table 3 depicts the relationship between the patch size and the EER. It can be observed that a smaller error is achieved when patch size is equal to 11.

D. PATCH SIZE FOR MATCHING

To overcome local variations such as local translation and local rotation, the enrollment template and testing image are divided into non-overlapping regions for matching. The selection of appropriate patch size is important to improvement of verification performance. If the patch size is too small, it will not include enough discriminative vein patterns to achieve high verification accuracy. In other words, patches with a too small size result in higher genuine matches, but they also lead to higher impostor matches which degrades the performance. By contrast, a large patch may include enough discriminative information, but it emphasizes on the global features and is not robust to local variations such as local translation and rotation which results in high verification error. Fig.6 illustrates the relationships between partition size

and EER on the CASIA database. It can be observed from Fig.6 that a smaller equal error rate is achieved when patch size ranges from 50 to 70. As the larger patch leads to large matching time, we fixed patch size to 50 × 50 for matching.

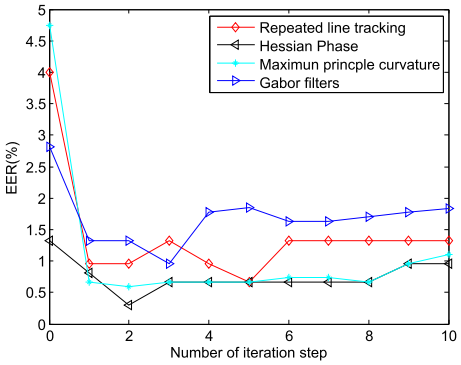

FIGURE 7. Relationship between iteration step and EER.

E. ITERATION NUMBER

In our approach, the DBN takes a patch labeled by one baseline as input and provides its probability to belong to a vein pattern. Then, this output of DBN is employed to generate a more accurate label, based on which a training dataset is reconstructed by the scheme described in Section IV-B. If the patches from the reconstructed training set are taken as input of DBN, more robust vein patterns can be extracted, which enables the segmentation to be done in a recursive way. The number of iteration steps is important to achieve low verification error. If the number of the iteration steps is small, some incorrect labels may not be statistically corrected by DBN. By contrast, if the number of iteration steps is large, the DBN is likely to be over-fitting. In this section, we optimized the number of the iteration steps by evaluating the performance on a validation dataset. In our experiment, the number of iterative steps is optimized by computing the EER on validation set A2. First, four baselines i.e. Hessian phase, Repeated line tracking, Maximum principle curvature, and Gabor filters are respectively employed to label each pixel, based on which the training set is obtained (as shown in Section III-B-2)). Second, the DBN is iteratively trained and the binary vein network is extracted at different iterative steps. Third, we select some images captured in first session for training and remaining ones for testing. 300 (50 × 6) genuine matching score and 3675 (50 × 49 × 3/2) impostor scores are generated by matching images from same and different hands, respectively. ERR is computed by genuine and impost scores. Fig.7 illustrates the EER at different numbers of iterative steps using different baselines for verification. From the experimental results in Fig.8, it can be seen that the EERs for Hessian phase, Repeated line tracking, Maximum principle curvature, and Gabor filters are 1.33, 4.0, 4.37, and 2.81, respectively, which are significantly reduced by the proposed approach. The lower EERs for five baselines are achieved when the numbers of iteration steps are 2, 2, 5, and 3, respectively. So, the iteration steps of 2, 2, 3, and 5 are respectively determined for Hessian phase, Repeated line tracking, Maximum principle curvature, and Gabor filters. From the experimental results in Fig.7, we observe that the verification accuracy is significantly improved using our model at one iteration. The reason may be explained by the following fact. There may be many incorrect labels in the segmented images from the four hand-crafted baselines (i.e. the four baselines achieve high EERs 1.33, 4.0, 4.37, and 2.81) because they segment images base on image processing techniques, only considering its pixels and their correlations. On the contrary, the DBN relies on rich statistics on nonlinear pixels correlations because it is trained using a large training set, which enables it to learn a hierarchical feature representation. However, more iteration steps can not result in more improvement of EERs because the incorrect labels are effectively corrected by DBN after each iteration.

F. VISUAL ASSESSMENT

In this section, we visually analyze and assess the performance of various approaches, so that more insights into the proposed approach are obtained. Fig.8 illustrates the hand-vein extraction results using the proposed approach on CASIA database. Fig.8(a) shows an original image from CASIA database. The sub figures Fig.8(b) shows the vein features segmented by four baselines, e.g. Gabor filters, Maximun Curvature points, Hessian Phase, and Repeated line tracking. Fig.8(c), Fig.8(d), Fig.8(e) and Fig.8(f) illustrates vein feature images obtained at different iteration steps i.e. 1,2,3,4.

From the obtained results in Fig.8, we can see that, there is a large difference between the obtained vein features by five baselines (as shown Fig.8(b)). For example, the Gabor filters (segmentation results at first row in Fig.8(b)) and Hessian Phase (segmentation results at third row in Fig.8(b)) extract smoother vein features while there are more noises and corrupted vein features in the images obtained by Maximun Curvature points (segmentation results at second row in Fig.8(b)) and Repeated line tracking (segmentation results at last row in Fig.8(b)). The larger difference is explained by the following fact. (1) the five handcrafted methods are designed based on different assumptions. For example, in [7], a vein pattern is defined as a line-like texture in a predefined neighborhood region and Gabor filters are employed to extract vein features. However, Hessian Phase, Maximun Curvature points, and Repeated line tracking are proposed to extract vein feature by observation that the cross-sectional profile of a vein pattern shows a valley shape. (2) the mathematical models for vein feature extraction are different.

Compared to the experimental results from four baselines, the proposed method effectively suppresses the noise and extracts smoother and continuous vein features from raw finger-vein images. Surprisingly, with the increasing of iteration steps, vein patterns extracted by the proposed approach show similar noise distribution, similar smoothness and similar continuous features even if its initial labels are from four different baselines. This may be explained by the fact that

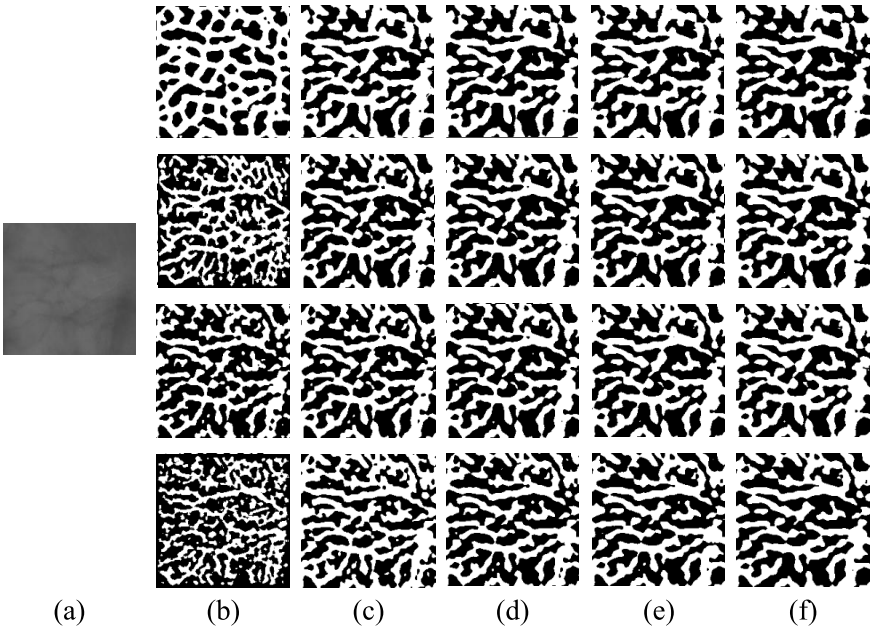

FIGURE 8. Extracted results of a finger image; (a) original image;(b) vein features extracted by four baselines (From top to bottom, the vein features are obtained by Gabor filters, Maximun Curvature points, Hessian Phase, and Repeated line tracking.); (c) vein features extracted by the proposed approach (the number of iteration steps is 1);(d) vein features extracted by the proposed approach (the number of iteration steps is 2);(e) vein features extracted by the proposed approach (the number of iteration steps is 3); and (f) vein features extracted by the proposed approach (the number of iteration steps is 4).

the DBN learns statistical distributions of vein patterns based on a number of patches which are not easy to be learned by human.

To investigate how to correct the labels of pixels by the proposed approach, we show the corrected pixels (the colored region in Fig.9) at each iteration step. In Fig.9, the color regions in each column are differences between vein feature images in two adjacent columns from Fig.8. For example, the color regions in Fig.9(a) denote the different pixel points between vein feature images from four baselines (Fig.8(b)) and ones at one iteration step (Fig.8(c)). From Fig.8 and Fig.9, it can be observed that the difference (colored region) is reduced with the increase of the iteration step. The large color regions in Fig.9(a) imply that more vein pixels are corrected at the first iteration. In other words, there are more incorrect labeling pixels in Fig.8(a). From Figures.8 and 9, we also see that the amount of different pixels (as shown in colored region Fig.9(b)-(d)) will decrease after several iterations. For example, it is difficult to observe the colored region when the number of iteration step is 4. These experimental results in this section are consistent with the trends in Section IV-E.

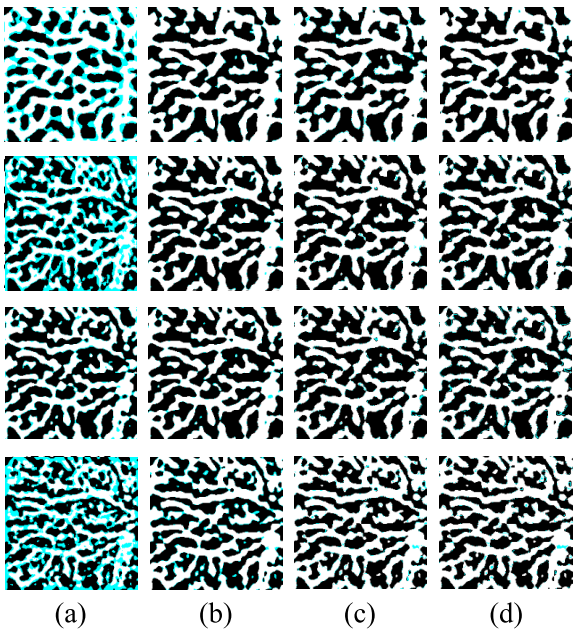

FIGURE 9. Corrected pixels (marked in color); (a) difference (colored region) between image from Fig.8(b) and image from Fig.8(c); (b) difference (colored region) between image from Fig.8(c) and image from Fig.8(d); (c) difference (color region) between image from Fig.8(d) and image from Fig.8(e); and (d) difference (color region) between image from Fig.8(e) and image from Fig.8(f).

G. VERIFICATION RESULTS BASED ON IMAGE DATASET FROM TWO SESSIONS

In this section, we carry out rigorous experiments to evaluate the performance of the proposed approach based on CASIA database and PolyU database collected from both sessions. For testing subset A3, 100 hands provided 600 images at two sessions, 300 images for one session. In dataset B3, there are 3000 images from 250 hands for verification. We select hand-vein images collected in the first session as training set while the remaining images acquired in the second session

are employed as testing data to assess the verification performance. Matching images from the same finger produces genuine scores while the impostor scores are created by matching images from different hands. This results in 300 (100×3) genuine scores and 1, 500 (250×6) genuine scores for CASIA database and PolyU database, respectively. The computation for impostor matching score is time consuming because there are 178, 200 ($6 \times 6 \times 100 \times 99$ /2) matching groups and 4, 482, 000 ($12\times 12 \times 250 \times 249$/2) matching groups for A3 and B3. To reduce computation time, similar to [34], all hands are randomly split into 10 groups and then the impostor matching scores are computed for each group. For example, we divide the 100 hands from A3 into 10 groups and each group includes 60 (6×10) images from 10 hands. For each group, matching the i-th sample at different sessions from different hands ($i = 1, 2, 3, 4, 5, 6$) produces impostor matching scores, which results in 270 ($10 \times 9 \times 3$) impostor matching scores. Hence, there are totally 2, 700 (270×10 groups) matching scores for 10 groups on A3. Similarly, there are 36, 000 ($25 \times 24 \times 6 \times 10$ groups) impostor matching scores for B3. The experimental results from various approaches are listed in Table 4 and the corresponding receiver operating characteristics (ROC) curves (the Genuine acceptance rate (GAR = 1-FAR) against the FAR) are illustrated in Fig.10.

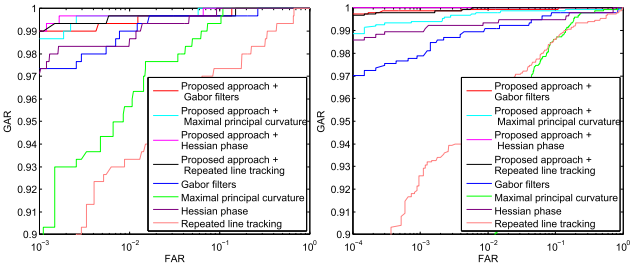

FIGURE 10. Receiver operating characteristics from (a) Database A and (b) Database B.

It can be observed from the presented results that our approaches achieve more than 98% GAR at FAR = 0.1% on database A and more than 99% GAR at FAR = 0.1% on database B, which are significantly higher than accuracies achieved by corresponding baseline. The experimental results in Table 4 shows similar trend that the proposed approach achieves lower EER than corresponding baseline when the vein features from baseline are employed to label pixels for the training of our model. In our experiments, compared to the Hessian Phase and Repeated line tracking, the Gabor filters and Maximun Curvature points show lower EER, but the proposed approaches trained by labeling data from four baselines show similar verification accuracy and achieve the state of the art level on both datasets. The lowest EERs 0.33% for database A and 0.015% for database B are achieved by the proposed approach + Hessian phase which is trained by labeling data obtained from Hessian phase.

TABLE 4. EER of various approaches on image data from two different sessions.

Methods	Database A	Database B
Repeated line tracking [26]	4.00	2.96
The proposed approach +Repeated line tracking	0.67	0.16
Maximum principle curvature [34]	2.33	3.87
The proposed approach + Maximum principle curvature	**0.33**	0.33
Gabor filters [7]	1.00	0.93
The proposed approach +Gabor filters	0.67	0.13
Hassian phase [9]	1.33	0.78
The proposed approach + Hassian phase	**0.33**	**0.015**

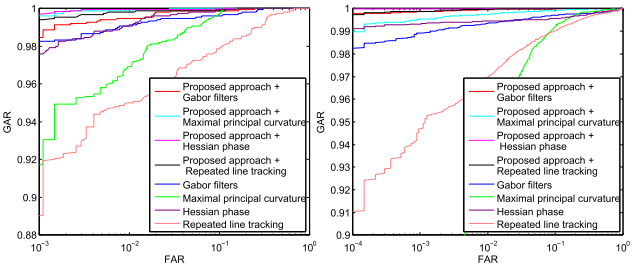

FIGURE 11. Receiver operating characteristics from (a) Datbase A and (b) Database B.

H. VERIFICATION RESULTS BASED ON IMAGE DATASET FROM TWO MIXED SESSIONS

In this section, the performance of mixed scores from same session and two sessions is reported. For A3, 600 images are collected from 100 hands at two sessions and each hand provided 3 images at one session. In dataset B3, there are 3000 images captured from 250 hands at two sessions and 6 images are acquired from each hand at one session. We match images from same hand to generate 1, 500 ($C_2^6 \times 100$) genuine matching scores for A3 and 16, 500 ($C_2^{12} \times 250$) genuine scores for B3. We adopt a similar experimental protocol to that described in the previous section to generate impostor scores, which results in 2, 700 (10 hands $\times$ 9 hands $\times$ 3 images $\times$ 10 groups) impostor matching scores for A3 and 36, 000 (25 hands $\times$ 24 hands $\times$ 6 images $\times$ 10 groups) impostor matching scores for B3, respectively. The experimental results for various approaches are summarized in Table 5. The receiver operating characteristics (ROC) curves for the corresponding performances are illustrated in Fig.11.

The experimental results summarized in Fig.11 and Table 5 are quite consistent with the trends from the experiments in the previous section. The proposed approach significantly outperforms the existing baselines in terms of EER improvement and achieve the best performance among all the approaches considered in this work, i.e. 0.20% for database A and 0.007% for database B. Another observation that can be noticed from the results in Fig.11 is that the proposed approach (almost) consistently achieves higher GAR, especially in the lower FAR region, as compared to four baselines.

TABLE 5. EER of various approaches on image data from two mixed sessions.

Methods	Database A	Database B
Repeated line tracking [26]	3.37	2.13
The proposed approach +Repeated line tracking	0.33	0.13
Maximum principle curvature [34]	1.87	2.96
The proposed approach + Maximum principle curvature	0.19	0.31
Gabor filters [7]	0.93	0.93
The proposed approach +Gabor filters	0.63	0.13
Hassian phase [9]	0.93	0.24
The proposed approach + Hassian phase	**0.20**	**0.007**

V. DISCUSSION

The experimental results depicted in Table 4 and Table 5 (and Fig.10 and Fig.11) show that using the vein feature images obtained from different baselines as labels to train our DBN achieves higher performance than the baselines mentioned above. Surprisingly, the proposed approach still achieves state-of-the-art EER level even when the baselines for labeling have higher verification error. For example, as illustrated in Table 4, the two baselines Repeated line tracking achieves 4.00% EER on database A and Maximum principle curvature achieves 3.87% EER on database B. However, taking labeling data from the two baselines as input of our iterative DBN, the verification errors are reduced to 0.67% and 0.33% (in Table 4).

Such a good performance may be explained as follows: 1) the handcrafted segmentation-based approaches do not infer any knowledge from the different images because they segment each image independently from the others. On the contrary, we employ handcrafted segmentation-based approaches to generate huge patch sets from different images for DBN training, which enables our approach to harness a rich prior knowledge so that it is capable of distinguishing vein patterns from background. 2) the handcrafted segmentation-based methods extract explicitly some image processing-based features (low level features) that might discard relevant information for vein pattern classification. By contrast, DBN can automatically learns high level features that are directly related to vein patterns. 3) the label of each pixel is corrected by iteratively training our DBN (as shown Fig.8 and Fig.9) and the corrected labels enable our DBN to learn more robust features for vein pattern representation. Therefore, the proposed approach is capable of predicting the probability of a pixel to belong to a vein pattern even if some baselines provide incorrect labels for some training patches.

From the experimental results (Table 4, Table 5, Figure 10, and Figure 11), we see that all approaches achieve improvement in terms of verification accuracy on image datasets acquired in two mixed sessions. Such a good performance can be attributed to the fact that the genuine matching from images captured at the same session are considered in the second verification experiments (Section IV-H). The images collected at the same session have smaller within-class variations because the capturing environment and user behavior are almost the same during finger-vein image acquisition within a short duration, which results in better verification performance. On the contrary, in the first verification experiments (Section IV-G), the genuine matching is computed between the images collected at two separate sessions. However, there are large within-class variations in such images, matching from which will ultimately compromise the performance of the authentication system.

TABLE 6. Summary of EERs derived from recently published palm-vein recognition papers on CASIA Database and PolyU Database.

Reference	Year	Methodology	Accuracy (%)	
			CASIA Database	PolyU Database
[4]	2010	Gabor filters for feature extraction and score level fusion strategy for matching	Null [1]	0.012
[56]	2011	Contour Code representation	0.758	0.1030
[9]	2011	Feature extraction and matching using NMRT and Hessian phase	(NMRT) 0.51 (Hessian phase) 1.44	0.004 0.43
[57]	2013	Combining curvelet transform and Gabor filter for feature extraction,and score level fusion for identification	Null [1]	0.102
[58]	2014	Gaussian-Radon transform for the principal direction extraction and matching feature bins for identification	0.67	0.14
[34]	2014	Imporved LBP method on mutual foreground	0.267	Null [1]
[59]	2017	Quality-specific discriminative LBP for feature extraction and improved Chi-square for verification	Null [1]	0.079
[60]	2017	DBN model fine-tuned by "coarse-to-fine" scheme for feature extraction and LDM for identification	Null [1]	0.058
	2018	The proposed approach	(Two separate sessions) **0.33** (Two mixed sessions) **0.20**	**0.015** **0.007**

[1] "Null" means that the corresponding parameter is not given in the reference

As a whole, the proposed approach achieves state-of-the-art recognition results on database B w.r.t on database A. Certainly, the verification accuracies may be further enhanced by incorporating other deep learning models into our approach. In the experiments, we have also embedded CNN into our approach by replacing DBN and evaluated this approach on both databases. As a result, the CNN based approach achieves similar performance compared to our approach, but it requires more computation time to segment an image. Like other works [9], [34], [60], [59], we summarize the representative palm-vein approaches in the existing literature in Table 6 to estimate the performance of our approach. The experimental protocol employed by the works in Table 4 are different and can be broadly categorized in two categories: (1) in the first category, the hand-vein images acquired during the first imaging session are selected as training data while the remaining images acquired during the second session are employed as testing data to assess the verification performance [4], [9], [58]. The matching is performed between images collected at first session and at second session. (2) In the second protocol, each image from two mixed sessions is selected for training and the remaining images are employed for testing [34], [59], [56]. The genuine matching is produced using all images from same hand regardless of the variation between the two sessions. In our experiments, we employ both protocols to estimate the performance of our approach and the best verification performance is listed in Table 6 to facilitate comparison. The proposed approach achieves 0.33% on CASIA database and 0.015% on PloyU database using fist protocol, and 0.20% on CASIA database and 0.007% on PloyU using second protocol

(as listed in Table 6), which implies that the proposed method has a powerful capacity to achieve state-of-the-art EER level compared with others.

VI. CONCLUSION

This paper proposed an iterative deep learning approach to predict probability of pixels to belong to veins or to background by learning a deep feature representation. First, the pixel in an image is labeled as vein and background by a baseline method. The patches centered in each labeled pixel are employed to construct the training dataset. Then, a DBN is proposed to extract the vein feature for vein segmentation. The DBN is iteratively trained to correct the incorrect labels, so it avoids the tedious and prone-to-error automatic labeling and achieves good performance. Experimental results on two public databases show that the proposed approach significantly improves the verification error rate and achieves a state of the art level performance.

REFERENCES

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 586–591.
[2] 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.
[3] J. Daugman, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 21–30, Jan. 2004.
[4] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An online system of multispectral palmprint verification," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 480–490, Feb. 2010.
[5] A. Kumar and D. Zhang, "Personal recognition using hand shape and texture," *IEEE Trans. Image Process.*, vol. 15, no. 8, p. 2454–2461, Aug. 2006.
[6] A. El-Yacoubi, R. Sabourin, C. Y. Suen, and M. Gilloux, "An hmm-based approach for off-line unconstrained handwritten word modeling and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 752–760, Aug. 1999.
[7] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.
[8] A. Kumar and K. V. Prathyusha, "Personal authentication using hand vein triangulation and knuckle shape," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 2127–2136, Sep. 2009.
[9] Y. Zhou and A. Kumar, "Human identification using palm-vein images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1259–1274, Dec. 2011.
[10] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 762–777, Apr. 2015.
[11] V. Ruiz-Albacete, P. Tome-Gonzalez, F. Alonso-Fernandez, J. Galbally, J. Fierrez, and J. Ortega-Garcia, "Direct attacks using fake images in iris verification," in *Proc. Eur. Workshop Biometrics Identity Manage. (BioID)*. Roskilde, Denmark: Springer, May 2008, pp. 181–190.
[12] D. Menotti *et al.*, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 864–879, Apr. 2015.
[13] J.-D. Wu and S.-H. Ye, "Driver identification using finger-vein patterns with radon transform and neural network," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5793–5799, 2009.
[14] T. Tanaka and N. Kubo, "Biometric authentication by hand vein patterns," in *Proc. Sice. Conf.*, vol. 1, Aug. 2004, pp. 249–253.
[15] M. A. Shahin, A. M. Badawi, and M. Kamel, "Biometric authentication using fast correlation of near infrared hand vein patterns," *Int. J. Biomed. Sci.*, vol. 2, no. 3, pp. 141–148, 2007.
[16] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.
[17] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.
[18] Y. Lu, S. J. Xie, S. Yoon, J. Yang, and D. S. Park, "Robust finger vein roi localization based on flexible segmentation," *Sensors*, vol. 13, no. 11, pp. 14339–14366, 2013.
[19] M. S. M. Asaari, S. A. Suandi, and B. A. Rosdi, "Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3367–3382, 2014.
[20] J. Hashimoto, "Finger vein authentication technology and its future," in *Proc. Symp. VLSI Circuits, Dig. Tech. Papers*, Jun. 2006, pp. 5–8.
[21] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1269–1272.
[22] W. Song, T. Kim, H. C. Kim, J. H. Choi, H.-J. Kong, and S.-R. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, 2011.
[23] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 8, pp. 1185–1194, 2007.
[24] J. Yang and Y. Shi, "Towards finger-vein image restoration and enhancement for finger-vein recognition," *Inf. Sci.*, vol. 268, pp. 33–52, Jun. 2014.
[25] E. C. Lee and K. R. Park, "Image restoration of skin scattering and optical blurring for finger vein recognition," *Opt. Lasers Eng.*, vol. 49, no. 7, pp. 816–828, 2011.
[26] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, no. 4, pp. 194–203, 2004.
[27] T. Liu, J. B. Xie, W. Yan, P. Q. Li, and H. Z. Lu, "An algorithm for finger-vein segmentation based on modified repeated line tracking," *Imag. Sci. J.*, vol. 61, no. 6, pp. 491–502, 2013.
[28] P. Gupta and P. Gupta, "An accurate finger vein based verification system," *Digit. Signal Process.*, vol. 38, pp. 43–52, Mar. 2015.
[29] H. Qin, L. Qin, and C. Yu, "Region growth-based feature extraction method for finger-vein recognition," *Opt. Eng.*, vol. 50, no. 5, 2011, Art. no. 057208.
[30] H. Qin, L. Qin, L. Xue, X. He, C. Yu, and X. Liang, "Finger-vein verification based on multi-features fusion," *Sensors*, vol. 13, no. 11, pp. 15048–15067, 2013.
[31] W. Yang, X. Huang, F. Zhou, and Q. Liao, "Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion," *Inf. sci.*, vol. 268, pp. 20–32, Jun. 2014.
[32] Z. Zhang, S. Ma, and X. Han, "Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4, Aug. 2006. pp. 145–148.
[33] C.-B. Yu, H.-F. Qin, Y.-Z. Cui, and X.-Q. Hu, "Finger-vein image recognition combining modified hausdorff distance with minutiae feature matching," *Interdiscipl. Sci., Comput. Life Sci.*, vol. 1, no. 4, pp. 280–289, 2009.
[34] W. Kang and Q. Wu, "Contactless palm vein recognition using a mutual foreground-based local binary pattern," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 11, pp. 1974–1985, Nov. 2014.
[35] X. Xi, L. Yang, and Y. Yin, "Learning discriminative binary codes for finger vein recognition," *Pattern Recognit.*, vol. 66, pp. 26–33, Jun. 2017.
[36] J. Yang, Y. Shi, and G. Jia, "Finger-vein image matching based on adaptive curve transformation," *Pattern Recognit.*, vol. 66, pp. 34–43, Jun. 2017.
[37] H. Qin, X. He, X. Yao, and H. Li, "Finger-vein verification based on the curvature in radon space," *Expert Syst. Appl.*, vol. 82, pp. 151–161, Oct. 2017.
[38] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis,"*IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp.1892–1905, Aug. 2017.
[39] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," *IEEE Trans. Med. Imag.*, vol. 8, no. 3, pp. 263–269, Sep. 1989.
[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2, 2012, pp. 1097–1105.
[41] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.

[42] D. C. Ciresan, U. Meier, L. M. Gambardella, and S. Juergen, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.
[43] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.
[44] K. He, X. Zhang, S. Ren, and J. Sun. (Dec. 10, 2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385
[45] P. Liskowski and K. Krawiec, "Segmenting retinal blood vessels with deep neural networks," *IEEE Trans. Med. Imag.*, vol. 35, no. 11, pp. 2369–2380, Nov. 2016.
[46] D. Ciresan, A. Giusti, L. M. Gambardella, and S. Juergen, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 2843–2851.
[47] W. Zhang *et al.*, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," in *Proc. IEEE Int. Symp. Biomed. Imag.*, vol. 108, Apr. 2015, pp. 1342–1345.
[48] H. Jang, S. M. Plis, V. D. Calhoun, and J. H. Lee, "Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: Evaluation using sensorimotor tasks," *Neuroimage*, vol. 145, pp. 314–328, Jan. 2017.
[49] Y. Guo, Y. Gao, and D. Shen, "Deformable MR prostate segmentation via deep feature learning and sparse patch matching," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1077–1089, Apr. 2016.
[50] N. Ohtsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[51] I. Sobel, "An isotropic 3×3 image gradient operator," in *Machine Vision for Three-demensional Scences*. ScienceDirect, 1990.
[52] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[53] G. E. Hinton, *Training Products of Experts by Minimizing Contrastive Divergence*. Cambridge, MA, USA: MIT Press, 2002.
[54] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[55] *CASIA MS Palmprint V1 Database*. Accessed: Apr. 2013. [Online]. Available: http://www. cbsr.ia.ac.cn/MSPalmprint
[56] Z. Khan, A. Mian, and Y. Hu, "Contour code: Robust and efficient multispectral palmprint encoding for human recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1935–1942.
[57] J. Sun and W. Abdulla, "Palm vein recognition by combining curvelet transform and Gabor filter," in *Biometric Recognition*. 2013, pp. 314–321.
[58] Y. Zhou, Y. Liu, Q. Feng, F. Yang, J. Huang, and Y. Nie, "Palm-vein classification based on principal orientation features," *PLoS ONE*, vol. 9, no. 11, 2014, Art. no. e112429.
[59] J. Wang and G. Wang, "Quality-specific hand vein recognition system," *IEEE Trans. Inf. Forensics Secur.*, to be published.
[60] J. Wang, G. Wang, and M. Zhou, "Bimodal vein data mining via cross-selected-domain knowledge transfer," *IEEE Trans. Inf. Forensics Secur.*, vol. PP, no. 99, p. 1, 2017.

Authors' photographs and biographies not available at the time of publication.