Pattern Recognition Letters 77 (2016) 58–65

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Deep belief network based statistical feature learning for fingerprint liveness detection☆

Soowoong Kim[a], Bogun Park[b], Bong Seop Song[b], Seungjoon Yang[a,*]

[a] School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 689-798, Republic of Korea
[b] Suprema Inc., 16F Parkview Office Tower of Jeongja-dong, Bundang-gu Seongnam, Gyeonggi 463-863, Republic of Korea

ARTICLE INFO

Article history:
Received 14 July 2015
Available online 8 April 2016

MSC:
41A05
41A10
65D05
65D17

Keywords:
Fingerprint liveness detection
Fingerprint anti-spoofing
Deep belief network
Deep learning
Statistical feature extraction
Livdet2013

ABSTRACT

Fingerprint recognition systems are vulnerable to impersonation by fake or spoof fingerprints. Fingerprint liveness detection is a step to ensure whether a scanned fingerprint is live or fake prior to a recognition step. This paper presents a fingerprint liveness detection method based on a deep belief network (DBN). A DBN with multiple layers of restricted Boltzmann machine is used to learn features from a set of live and fake fingerprints and also to detect the liveness. The proposed method is a systematic application of a deep learning technique, and does not require specific domain expertise regarding fake fingerprints or recognition systems. The proposed method provides accurate detection of the liveness with various sensor datasets collected for the international fingerprint liveness detection competition.

1. Introduction

Fingerprint recognition is one of the biometric authentication methods that has been widely used in forensic, civilian, and commercial applications [17]. Fingerprint recognition systems usually consist of relatively inexpensive components, and yet can provide highly accurate authentication [4]. However, fingerprint recognition systems are vulnerable to impersonation by fake, or spoof, fingerprints. Fake fingerprints can be made of materials such as silicon or play-doh under cooperative or non-cooperative scenarios using various methods [18]. Countermeasures to prevent false authentication by detecting the liveness of fingerprints are being developed [8]. Methods to detect the liveness of fingerprints can be divided into hardware-based and software-based methods.

Hardware-based systems utilize auxiliary hardware components such as oxygen saturation sensors [24], multispectral imaging systems [22], or optical coherence tomography scanning systems [27] to detect features that separate live fingers from fake fingers. The addition of hardware components usually increases costs of otherwise inexpensive recognition systems significantly. Hardware-based systems that employ additional modality are found to be venerable to false authentication scenarios in which a fake fingerprint is used for the fingerprint sensor and a live finger is used for the other modality [1].

Software-based methods utilize dynamic or static features that are available using existing hardware. Dynamic features such as changes of skin elasticity [13] or perspiration pattern [23] during the fingerprint scanning process can be observed to detect the liveness of a given fingerprint. The use of dynamic features has a disadvantage that detection time is increased to extract features that change over time. Static features that separate live fingerprints from fake fingerprints can be observed from a single impression of a fingerprint to detect the liveness. For example, active sweat pores can be detected by a high pass filtering to detect live fingerprints [19]. One can utilizes the fact that materials used to fabricate artificial fingerprints have different textural details and hence different high frequency spectrum components [21]. Features that do not change during the fabrication of fake fingerprints can also be used for liveness detection [7,9]. The performance of static feature based liveness detection methods depend heavily on the choice of features selected in a hand-designed manner according to domain knowledge and experience.

Recently, it has become more common to attempt to learn features through supervised or unsupervised learning [2]. Features

☆ This paper has been recommended for acceptance by Ajay Kumar.

* Corresponding author. Tel.: +82 52 217 2110; fax: +82 52 217 2299.
E-mail address: syang@unist.ac.kr (S. Yang).

http://dx.doi.org/10.1016/j.patrec.2016.03.015
0167-8655/

automatically derived via machine learning have advantages of not requiring specific domain expertise and potentially yielding a much larger set of features for a classifier. There have been researches on learning features for fingerprint liveness detection. In [6], the independent component analysis (ICA) is used to automatically learn a set of filters that extract textural features to be used in classification by a support vector machine (SVM). The filters obtained via ICA are learned using a set of natural images not by a set of fingerprint images. Examples of learned filters are mostly edge detectors at various angles and phases, and are not specifically trained for fingerprint applications. In [5], a convolutional network (CN) is used to extract features, whose dimension is reduced by the principal component analysis (PCA), to be used by a SVM for fingerprint liveness detection. The weights of the CNN are set to random values without any learning process. Even though the CN can work with random weights, the CN is not specifically trained for fingerprint applications. The learned principal components are eigen images that are suitable for applications that require energy compaction.

This paper presents a fingerprint liveness detection method based on a deep belief network (DBN). DBN is a deep learning architecture that has been used in various applications such as character recognition [26], speech recognition [20], facial expression recognition [16], music information retrieval [10], and so forth. DBN has been also used in fingerprint applications for enhancement of scanned fingerprint images [25]. The generative properties of DBN allow better understanding of the performance, and provide a simpler solution for the classification tasks [12]. We use a DBN to learn features from a set of live and fake fingerprints and also to determine the liveness of fingerprints. The restricted Boltzmann machine (RBM) [12] is used at each layer of the DBN. The RBMs are trained sequentially from the first layer to the second to last layer by unsupervised learning with layer-wise greedy algorithm. Then the DBN, including the last layer, is trained by supervised learning with back propagation with a labeled set of live and fake fingerprints. The DBN output is the posterior probability that a patch taken from a given fingerprint belongs to a live or fake fingerprint. A hypothesis test is set up with outputs of the DBN with multiple patches taken from a given fingerprint as inputs to determine the liveness.

The proposed method is a systematical application of deep learning to the problem of fingerprint liveness detection. Liveness detection as well as learning of features appropriate for the liveness detection is performed within an established framework of deep learning without relying on specific domain expertise. In particular, the proposed method is trained for multiple datasets without any information about the sensors, images, or pre-processing steps. Different features are learned for each dataset to be used in the liveness detection with a particular fingerprint recognition system.

The proposed fingerprint liveness detection method is evaluated with datasets collected using four different optical sensors for the international fingerprint liveness detection competition (LivDet2013) [8]. The liveness detection results show that the proposed method can provide high performance in terms of the false rejection rate (FRR) and the false acceptance rate (FAR). With the DBN topology being simple, the computational complexity of the detection method is relatively low. The proposed method provides efficient and effective software based fingerprint liveness detection.

This paper is organized as follow. Section 2.1 explains a preprocessing step to prepare patches of inputs from a scanned fingerprint image. Section 2.2 presents the topology and modeling of the proposed DBN. Section 2.3 provides a training method including pre-training with unsupervised learning and fine-tuning with supervised learning. Section 2.4 provides a hypothesis testing with outputs of the DBN with multiple inputs from a given fingerprint. Experimental results and discussions are given in Section 3. Section 4 concludes the paper.

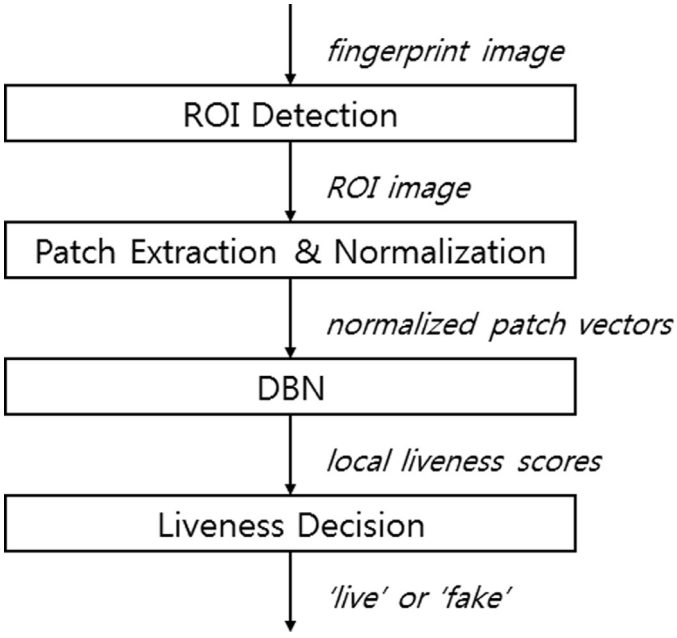

Fig. 1. Flowchart of the proposed fingerprint liveness detection process.

2. Proposed fingerprint liveness detection algorithm

The proposed fingerprint liveness detection system consists of the following steps. From a query fingerprint image, region of interest (ROI) is detected. Small size patches from the ROI image are extracted, normalized, and reshaped as vectors. The outputs of the DBN with the patches as inputs are collected and fed into a liveness detection routine. A flowchart of the proposed fingerprint liveness detection system is shown in Fig. 1. The following sections describe detail operations.

2.1. ROI detection and input patch Ppeparation

A query image composed of a fingerprint on scanner platen background is captured by a fingerprint sensor. The fingerprint image is not always at the center of a captured image. The ROI that contains only the fingerprint image is extracted from a captured image. Salient points of the fingerprint image are detected using the two dimensional Harris corner detector. The average location of the detected salient points is calculated. Region inside a rectangle of a fixed size centered at the average location is used as an ROI. Fig. 2 shows an example of the detected ROI. The blue points indicate the locations of salient points detected by the Harris corner detector. The red point is the average location of the salient points. The image inside the red rectangle is the ROI of the captured image.

Patches of the size $H_p \times W_p$ are extracted from the image inside the ROI. Patches are allowed to overlap the other patches by t pixels. Pixels in each patch are reshaped as a vector to form a vector $\mathbf{p}$ of the size $M \times 1$, where $M = H_p \times W_p$. The vectors are normalized to have zero mean and unit variance.

2.2. DBN

A DBN is used for the fingerprint liveness detection. Fig. 3 shows a schematic of the DBN for the proposed liveness detection. There are K layers of RBMs superimposed such that the visible unit

Fig. 2. An example of ROI placement on a fingerprint image. The blue markers are the salient points detected by the 2D Harris corner detector. The red marker is the average location of the salient points. And the red rectangle is the ROI of size 160 × 160. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

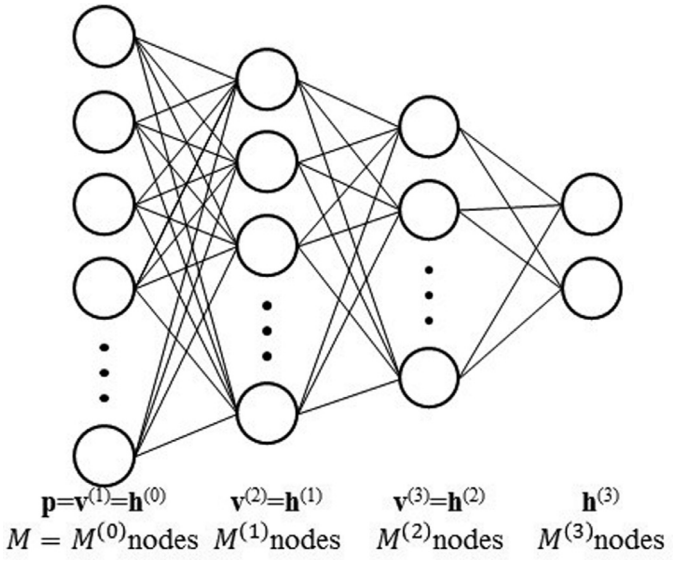

Fig. 3. Topology of the proposed DBN.

at the kth layer is the hidden unit of the $(k-1)$th layer. The first layer has M nodes for the visible unit. Each RBM has $M^{(k)}$ nodes for the hidden layer, except for the Kth layer which has two output nodes.

Given a visible unit $\mathbf{v}^{(k)}$, a hidden unit $\mathbf{h}^{(k)}$, and the model parameter $\theta^{(k)}$ of the kth layer, the joint Boltzmann distribution of the RBM is defined by

$$P(\mathbf{v}^{(k)}, \mathbf{h}^{(k)}; \theta^{(k)}) = \frac{1}{z}\exp(-E(\mathbf{v}^{(k)}, \mathbf{h}^{(k)}; \theta^{(k)})), \tag{1}$$

where z is a normalization constant and the parameters of the kth layer are

$$\theta^{(k)} = (\mathbf{W}^{(k)}, \mathbf{b}^{(k)}, \mathbf{c}^{(k)}). \tag{2}$$

The matrix $\mathbf{W}^{(k)}$ describes the relations between the ith visible node and the jth hidden node, and $\mathbf{b}^{(k)}$ and $\mathbf{c}^{(k)}$ are the bias for visible and hidden units, respectively, of the kth layer. The energy function $E(\mathbf{v}, \mathbf{h}; \theta)$ is given by

$$E(\mathbf{v}^{(k)}, \mathbf{h}^{(k)}; \theta^{(k)}) = -\sum_{i=1}^{M^{(k-1)}}\sum_{j=1}^{M^{(k)}} W_{ij}^{(k)} v_i^{(k)} h_j^{(k)} - \sum_{i=1}^{M^{(k-1)}} b_i^{(k)} v_i^{(k)} - \sum_{j=1}^{M^{(k)}} c_j^{(k)} h_j^{(k)}, \tag{3}$$

where $W_{ij}^{(k)}$ is the (i, j)th element of $\mathbf{W}^{(k)}$, and $b_i^{(k)}$ and $c_j^{(k)}$ are ith and jth elements of $\mathbf{b}^{(k)}$ and $\mathbf{c}^{(k)}$.

The first layer of the DBN is a RBM with the vector $\mathbf{p}$ obtained from the patch images as an input. It is assumed that each element of input vector follows the Gaussian distribution with zero mean and unit variance. Since the visible unit receives the input vector $\mathbf{p}$, the visible unit follows the Gaussian distribution. The hidden unit follows the Bernoulli distribution. The first layer is the Gaussian-Bernoulli (GB) RBM [29]. The conditional probabilities are given by

$$Q(h_j^{(1)} = 1|\mathbf{v}^{(1)}; \theta^{(1)}) = \phi\left(\sum_{i=1}^{M} W_{ij}^{(1)} v_i^{(1)} + c_j^{(1)}\right)$$
$$P(v_i^{(1)}|\mathbf{h}^{(1)}; \theta^{(1)}) = \mathcal{N}\left(v_i^{(1)}; \sum_{j=1}^{M^{(1)}} W_{ij}^{(1)} h_j^{(1)} + b_i^{(1)}, 1\right), \tag{4}$$

where the activation function $\phi(\cdot)$ is the sigmoid function $\phi(x) = \frac{1}{1+e^{-x}}$ and $\mathcal{N}$ is the Gaussian probability density function with mean $\sum_{j=1}^{M^{(1)}} W_{ij}^{(1)} h_j^{(1)} + b_i^{(1)}$ and variance one. With the input vector obtained from the patch images, we have $\mathbf{v}^{(1)} = \mathbf{p}$.

All the layers except for the first and the last layers in the multi-layered DBN share the same input and output characteristics. The visible and hidden units follow the Bernoulli distributions. Each layer is the Bernoulli-Bernoulli (BB) RBM [11]. The conditional probabilities are given by

$$Q(h_j^{(k)} = 1|\mathbf{v}^{(k)}; \theta^{(k)}) = \phi\left(\sum_{i=1}^{M^{(k-1)}} W_{ij}^{(k)} v_i^{(k)} + c_j^{(k)}\right)$$
$$P(v_i^{(k)} = 1|\mathbf{h}^{(k)}; \theta^{(k)}) = \phi\left(\sum_{j=1}^{M^{(k)}} W_{ij}^{(k)} h_j^{(k)} + b_i^{(k)}\right). \tag{5}$$

with the superimposed RBMs, we have $\mathbf{v}^{(k)} = \mathbf{h}^{(k-1)}$.

2.3. Training of DBN

The DBN is trained in two steps. The first step is to train the individual RBMs using the layer-wise greedy training algorithm [3]. The second step is to fine-tune the DBN with labeled inputs.

2.3.1. Pre-training

The RBM's are trained sequentially from the first layer to the $(K-1)$th layer. With a training set of unlabeled vectors, parameters are updated using the contrastive divergence [11]. For the RBM in the first layer, the data samples for the visible unit are taken from the input vector $\mathbf{p}$. The data samples for the hidden unit is obtained with the conditional probability $Q(\mathbf{h}_{\text{data}}^{(1)}|\mathbf{v}_{\text{data}}^{(1)})$. The samples drawn from the model for the visible and hidden units are obtained through a small number of iterations of (4). The gradient of parameters of the hidden layer are found by

$$\begin{aligned}\nabla W_{ij}^{(k)} &= E[v_i^{(k)} h_j^{(k)}]_{\text{data}} - E[v_i^{(k)} h_j^{(k)}]_{\text{model}}\\ \nabla b_i^{(k)} &= E[v_i^{(k)}]_{\text{data}} - E[v_i^{(k)}]_{\text{model}}\\ \nabla c_j^{(k)} &= E[h_j^{(k)}]_{\text{data}} - E[h_j^{(k)}]_{\text{model}},\end{aligned} \tag{6}$$

where the expectations $E[\cdot]_{\text{data}}$ and $E[\cdot]_{\text{model}}$ are calculated with the data samples and samples obtained from the model, respectively. Then the parameters are updated by

$$\begin{aligned} W_{ij}^{(k)} &\leftarrow W_{ij}^{(k)} + \epsilon \nabla W_{ij}^{(k)} \\ b_i^{(k)} &\leftarrow b_i^{(k)} + \epsilon \nabla b_i^{(k)} \\ c_j^{(k)} &\leftarrow c_j^{(k)} + \epsilon \nabla c_j^{(k)}, \end{aligned} \tag{7}$$

where ϵ is the learning rate.

The kth RBM, for $k = 1$ to $K-1$, is trained with the output of hidden unit of the $(k-1)$th layer as the input. The data samples for the visible unit are taken from the hidden unit $\mathbf{h}^{(k)}$. The data samples for the hidden unit is obtained with the conditional probability $Q(\mathbf{h}_{\text{data}}^{(k)} | \mathbf{v}_{\text{data}}^{(k)})$. The samples drawn from the model for the visible and hidden units are obtained through a small number of iterations of Eq. (5). The parameters are updated with Eqs. (6) and (7).

2.3.2. Fine-tuning

Once all the layers from the first to the $(K-1)$th are trained with the layer-wise greedy training algorithm, the DBN is fine-tuned with a training set of labeled inputs. The input vector $\mathbf{p}$ is labeled by the label vector $\mathbf{y}$ such that $\mathbf{y} = [1, 0]^{\mathrm{T}}$ when $\mathbf{p}$ is form the live fingerprint class, and $[0, 1]^{\mathrm{T}}$ when $\mathbf{p}$ is from the fake fingerprint class.

The last layer of the DBN consists of two hidden nodes, which are the two output node of the DBN. The first and second output nodes represent the posterior probabilities that a given input vector $\mathbf{p}$ is from the live and fake fingerprint classes, respectively. The conditional probability is given by

$$Q(h_j^{(K)} = 1 | \mathbf{v}^{(K)}; \theta^{(K)}) = \psi \left(\sum_{i=1}^{M^{(K-1)}} W_{ij}^{(K)} v_i^{(K)} + c_j^{(K)} \right), \tag{8}$$

where the activation function $\psi(\cdot)$ is the softmax function $\psi(\mathbf{x})_j = e^{3x_j} / (\sum_{i=1}^{2} e^{3x_i})$. The visible unit of the Eq. (8) is the hidden layer of the $(K-1)$th layer, or $\mathbf{v}^{(K)} = \mathbf{h}^{(K-1)}$.

For a given input vector $\mathbf{p}$ from a labeled training set, the input to the visible unit of the kth layer is found sequentially by

$$h_j^{(k)} = \phi \left(\sum_{i=1}^{M^{(k-1)}} W_{ij}^{(k)} h_i^{(k-1)} + c_j^{(k)} \right) \tag{9}$$

for $k = 1$ to $(K-1)$ with $\mathbf{v}^{(1)} = \mathbf{h}^{(0)} = \mathbf{p}$. The output of the hidden layer of the Kth layer is found by

$$h_j^{(K)} = \psi \left(\sum_{i=1}^{M^{(K-1)}} W_{ij}^{(K)} h_i^{(K-1)} + c_j^{(K)} \right). \tag{10}$$

The error vector is computed with the label vector $\mathbf{y}$ of the input vector $\mathbf{p}$ by

$$\delta_j^{(K)} = h_j^{(K)} - y_j. \tag{11}$$

The error vector at kth layer is calculated sequentially by

$$\delta_i^{(k)} = \left(\sum_{j=1}^{M^{(k+1)}} W_{ij}^{(k+1)} \delta_j^{(k+1)} \right) d_i^{(k)} \tag{12}$$

for $k = (K-1)$ to 1, where

$$d_i^{(k)} = h_i^{(k)} (1 - h_i^{(k)}). \tag{13}$$

The parameters of the gradients of parameters are found sequentially by

$$\nabla W_{ij}^{(k)} = h_i^{(k-1)} \delta_j^{(k)} \tag{14}$$

Table 1
Datasets of LivDet2013 database used in the experiments.

Sensor		Biometrika	Italdata	Crossmatch	Swipe
DPI		569	500	500	96
Image size		315 × 372	640 × 480	800 × 750	208 × 1500
Training	Live	1000	1000	1250	1250
Samples	Fake	1000	1000	1000	1000
Test	Live	1000	1000	1250	1250
Samples	Fake	1000	1000	1000	1000

$$\nabla c_j^{(k)} = \delta_j^{(k)} \tag{15}$$

for $k = 1$ to $(K-1)$, and the parameters are updated by

$$W_{ij}^{(k)} \leftarrow W_{ij}^{(k)} + \epsilon \nabla W_{ij}^{(k)} \tag{16}$$

$$c_j^{(k)} \leftarrow c_j^{(k)} + \epsilon \nabla c_j^{(k)}. \tag{17}$$

2.4. Liveness decision

The DBN provides the posterior probability that a given input $\mathbf{p}$ belongs to the live and fake fingerprint classes. Given a query fingerprint image, many input vectors can be extracted. The liveness of a query image is determined from the extracted input vectors. Let $\mathbf{p}_l$ for $l = 1, 2, \ldots, L$ be the L extracted input vectors. The output of the first output node of the Kth layer of the DBN with $\mathbf{p}_l$ as an input is denoted by x_l. The value of x_l is close to one when the input vector $\mathbf{p}_l$ is from a live fingerprint, and close to zero when from a fake fingerprint. A hypothesis test on whether x_l's are from a live or fake fingerprint is performed by

$$\begin{aligned} &\sum_{l=1}^{L} x_l \geq \tau \quad : \text{live fingerprint} \\ &\sum_{l=1}^{L} x_l < \tau \quad : \text{fake fingerprint}, \end{aligned} \tag{18}$$

where τ is a threshold.

3. Experiments and results

The parameters used for the experiments are as follow. From a given image, the ROI of 160 × 160 pixels is detected. The patch images of 16 × 16 pixels are extracted, from which input vectors of size 256 × 1 are prepared. The elements of the input vectors are normalized to zero mean and unit variance. The DBN has three layers, or $K = 3$. The first layer receives the 256 × 1 input vectors, hence the number of visible nodes of the first layer M is 256. The number of units in each layers, $M^{(k)}$, are 128, 64, and 2 for $k = 1, 2, 3$, respectively. The number of inputs used for the hypothesis testing, L, is 100.

The DBN is trained and evaluated using a database collected for LivDet2013 [8]. The LivDet2013 database consists of four fingerprint image datasets collected with different optical sensors: Biometrika, Italdata, Crossmatch, and Swipe. The detail descriptions of datasets are given in Table 1. In the training phase, the dropout rate of 0.5 is used to reduce problems associated with overfitting [28]. The learning rate ϵ is set to 0.001. Two DBN's are prepared for each dataset. One trained using only the provided LivDet2013 dataset, and the other trained using artificially augmented LivDet2013 dataset that is 10 times larger than the provided dataset. The data augmentation is done using the same method given in [5].

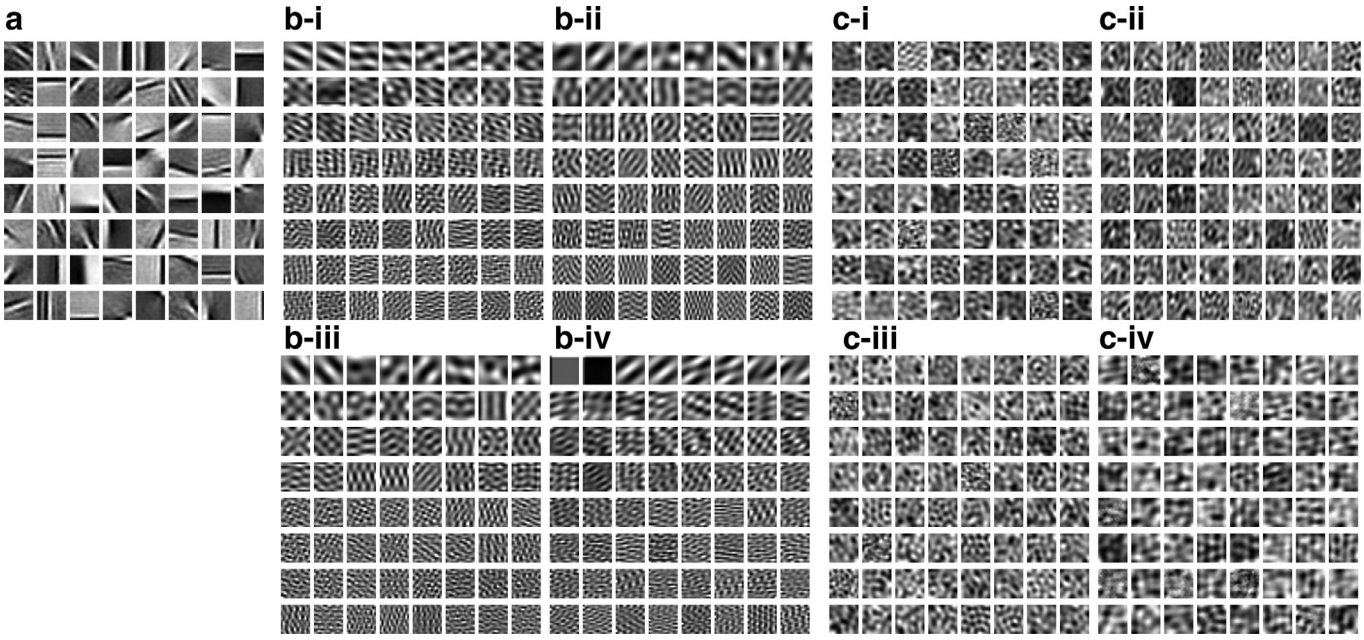

Fig. 4. Features learned by various methods, (a) ICA with natural images, (b) CN with random filters followed by PCA, and (c) the second layer of the proposed DBN, for (i) Biometrika, (ii) Italdata, (iii) Crossmatch, and (iv) Swipe datasets.

Fig. 4 shows examples of features learned by various methods. Fig. 4 (a) shows features learned by the ICA with natural images, following the method in [6]. 24 images from Kodak Photo CD are used. The features are mostly edge detectors with various angles and phases. Fig. 4 (b) shows features learned by the CN with random filters followed by the PCA, following the method in [5] using (i) Biometrika, (ii) Italdata, (iii) Crossmatch, and (iv) Swipe datasets. The features are principal components of fingerprints, which can pack energy into a small set of components. The first 64 components of the ICA and PCA results are shown. Fig. 4 (c) shows the features learned by the proposed DBN using (i) Biometrika , (ii) Italdata, (iii) Crossmatch, and (iv) Swipe datasets. The second layer features are shown using the method described in [15]. It can be seen that for the proposed method the features are learned specifically for the images in the dataset for the liveness detection. In contrast, the learned features obtained by random filtering and PCA are very similar for four different datasets.

Fig. 5 shows the ROC curves for Biometrika, Italdata, Crossmatch, and Swipe datasets. The false rejection rate (FRR) and the false acceptance rate (FAR) are shown. The FRR is the rate that a live fingerprint is detected as a fake fingerprint, and the FAR is the rate a fake fingerprint is detected as a live fingerprint. The threshold value, τ, that provides the equal error rate (EER) is selected for the detector.

The performance of the liveness detection is evaluated using the average classification error (ACE). The ACE is the the mean of the rate of misclassified live fingerprints and the rate of misclassified fake fingerprints. The detection accuracy can be obtained by 1-ACE. Table 2 shows the detection accuracy measured with Biometrika, Italdata, Crossmatch, and Swipe datasets. The detection accuracy of the four liveness detection methods, Dermalog and UniNap1, which are available on [8], the pore analysis based approach in [14], and the random filter convolution network (CN) based approach in [5], are shown for comparison. The detection accuracy of the random filter CN based approach and the proposed DBN are shown for two cases: when trained with only the provided dataset and when trained with data augmentation.

Fake fingerprints in Biometrika and Italdata datasets are collected under a non-cooperative scenario using latent fingerprints,

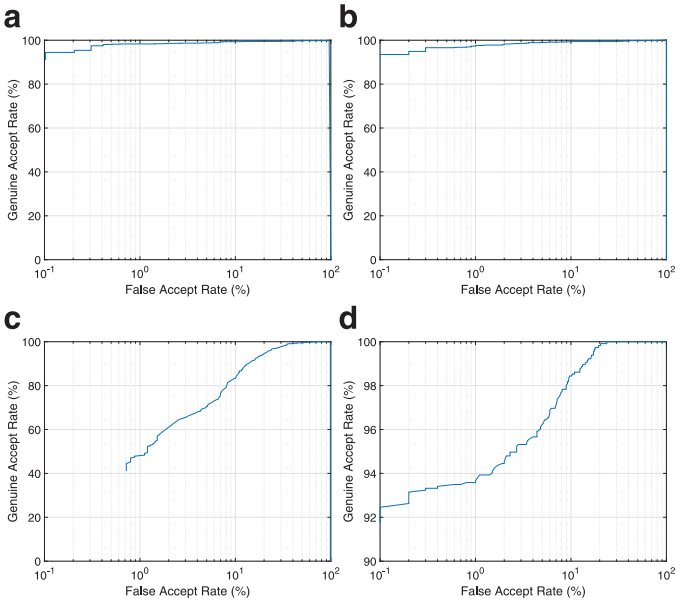

Fig. 5. ROC curves of the proposed fingerprint liveness detection method, (a) Biometrika $EER = 1.16$ at $\tau = 76.7$, (b) Italdata Biometrika $EER = 1.85$ at $\tau = 68.2$, (c) Crossmatch $EER = 10.32$ at $\tau = 13.9$, and (d) Swipe datasets $EER = 3.43$ at $\tau = 60.1$.

whereas those in Crossmatch and Swipe datasets are collected under a cooperative scenario [8]. Generally, a fake fingerprint created under a non-cooperative scenario is less similar to the original fingerprint than the one created under a cooperative scenario. Fingerprints of Swipe dataset are acquired while users are swiping their fingers across the sensor surface. Fingerprints are deformed by uneven swiping speed. Moreover, the resolution of the sensor in terms of dots per inch (dpi) is significantly lower than the other sensors. Sweat pore are seldom seen in fingerprints of Swipe dataset. Examples of live and fake fingerprints are shown in Fig. 6. It can be seen that the quality of fake fingerprints of Biometrika and Italdata datasets is significantly poorer than those of

Table 2
Detection accuracy for LivDet2013 datasets. Dermalog and UniNap1 results are from [8]. Result for a pore analysis based approach is from [14]. Result for random filter convolutional network (CN) is from [5].

	Dermalog	UniNap1	Pore Analysis	CN-PCA-SVM w/o AUG	CN-PCA-SVM with AUG	DBN w/o AUG	DBN with AUG
Biometrika	98.30	95.30	97.80	95.45	99.20	98.85	98.85
Italdata	99.20	96.50	99.00	52.35	97.55	98.15	99.40
Crossmatch	44.53	68.80	65.10	94.80	96.71	89.68	93.05
Swipe	96.47	85.93	–	94.03	92.33	96.50	96.56
Average	84.63	86.58	87.30	84.16	96.45	95.80	97.10

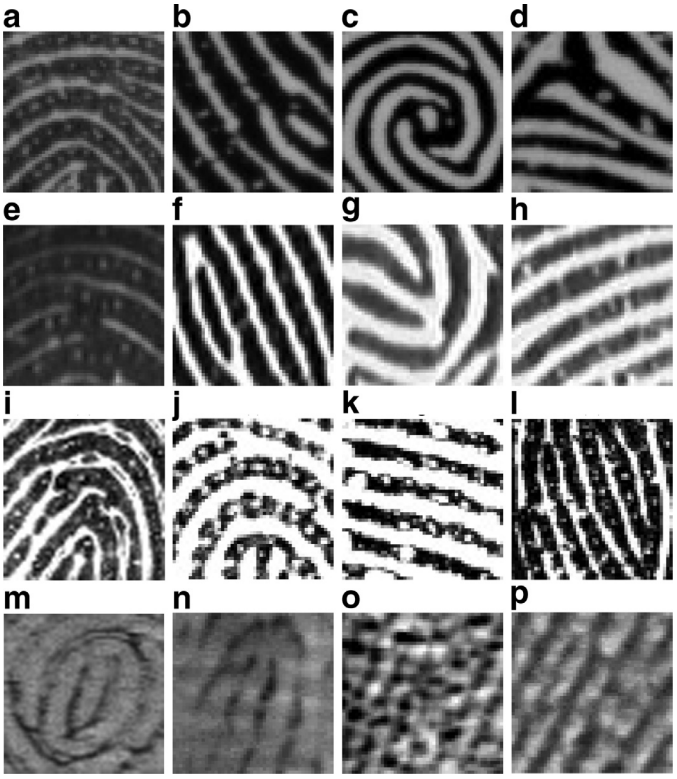

Fig. 6. Example fingerprint images (50 × 50 cropped) in the LivDet2013 dataset [8]. (a) live, (b) latex, (c) ecoflex, and (d) gelatin fingerprint images from Biometrika dataset, (e) live, (f) latex, (g) modasil, and (h) wood glue fingerprint images from Italdata dataset, (i) live, (j) latex, (k) body double, and (l) playdoh fingerprint images from Crossmatch dataset, and (m) live, (n) latex, (o) body double, and (p) wood glue fingerprint images from Swipe dataset.

Crossmatch dataset. The fake fingerprints of the Crossmatch dataset closely resemble the live ones. In particular, the characteristics of sweat pores, such as pore spacing, pore region gray level variance, and pore region maximum gray level difference, are retained in the fake fingerprints. It can be also seen that there are no sweat pores in either live or fake fingerprints of the Swipe dataset.

Detail descriptions of Dermalog and UniNap1 algorithms are not available. Dermalog, UniNap1, and the pore analysis based algorithms perform better for Biometrika and Italdata datasets and poorer for Crossmatch dataset. It is relatively easier to differentiate fake fingerprints from live ones for Biometrika and Italdata datasets than for Crossmatch dataset. Dermalog algorithm performs well and UniNap1 algorithm performs poorly for Swipe data. The results of the pore analysis based algorithm for the Swipe dataset is not available.

The random filter CN based algorithm in [5] consists of CN with randomly generated filters, PCA, and SVM. Parameters that defines the topology of the CN for each dataset, such as the number of layers, the size of filters in each layers, the number of principal components, are determined through experiments. The selected parameters are different for each dataset. The number of layers varies from two to five, the number of filters in each layer varies from 64, 128, up to, 2048. The size of the filters varies as 5×5, 7×7, 9×9, 14×5, and 21×5. Even though the filter coefficients are randomly selected, the CN's are prepare to match the characteristics of the datasets through the careful selections of the topology. However, the relationship between the CN topology and dataset is not available.

The performance of the CN based algorithms are poor when trained with only the provided datasets without data augmentation. However, with data augmentation, the algorithms are properly trained and provide better performance. It is noticeable that the CN based algorithm trained with data augmentation performs well with the Crossmatch dataset. However, the topology of the CN is very large, with five convolutional layers and 64 filters at the first layer and 1024 filters at the last layer. In CN based methods, each filter output is an image. The memory requirement to store intermediate filter outputs grows rapidly with the number of layers and the number of filters. The result with the Swipe dataset is relatively poor. And the result with data augmentation is lower than that without data augmentation. The CN topology prepared through experiment may not be selected to properly match the characteristics of the Swipe dataset.

The proposed method provides accurate detection for all four datasets. The average detection accuracy is better than Dermalog, UniNap1, and the pore analysis based algorithm in both cases when trained with and without data augmentation. The average detection accuracy of the proposed method is better than the CN based method for each case when trained with and without data augmentation. The detection methods can access only the images in the dataset without any information about the sensors, images, or pre-processing steps. The proposed method can learn and utilizes features specific for the datasets without any information or domain expertise for accurate liveness detection. The ability to learn features from each dataset is an advantage of applying deep learning to the problem of liveness detection. The proposed method is able to learn useful features from the Crossmatch dataset, for which the characteristics of pores in live and fake fingerprints are very similar, and also from the Swipe dataset, for which pores are seldom seen in either live or fake fingerprints.

The accuracy of the proposed method is slightly lower with Crossmatch dataset than with the other three datasets. The fingerprint images in Crossmatch dataset are obtained after excessive contrast enhancement [8]. Fig. 7 shows the histograms of pixel values for all the fingerprint images in the datasets. The images inside the ROIs are used. It can be seen that fingerprint images in Biometrika, Italdata, and Swipe datasets have grayscale values spread out between zero and one, whereas images in the Crossmatch dataset have grayscale values mostly at the extreme ends of zero and one. This observation is most likely due to excessive contrast enhancement algorithm in the pre-processing step. We collected 2000 live fingerprint images from 50 people, four images each of all ten fingers, and 2000 fake fingerprint images from 200 spoof fingers, ten images per each spoof finger. The images were divided into two equal datasets, training and testing. The spoof

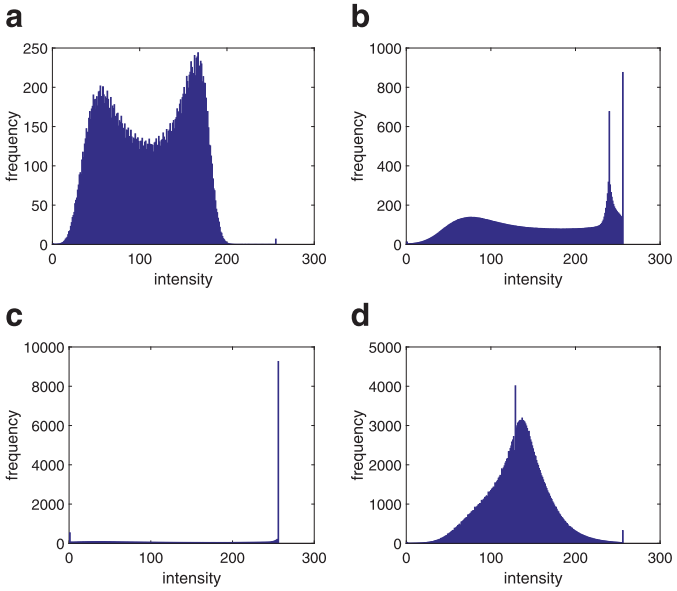

Fig. 7. The average histograms of fingerprint images inside ROI region for the LivDet2013 (a) Biometrika, (b) Italdata, (c) Crossmatch, and (d) Swipe datasets.

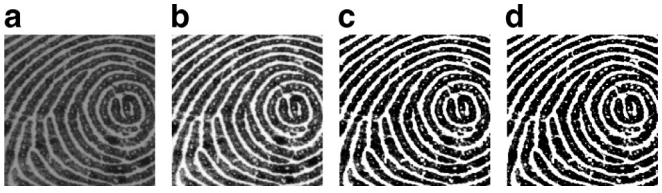

Fig. 8. Example images pre-processed by adaptive contrast enhancement at different parameters, (a) $\beta = 0.4$, (b) $\beta = 0.5$, (c) $\beta = 0.6$, and (d) $\beta = 0.7$.

material were Playdoh, gelatin, and silicone. All of live and fake fingerprint images were collected on Suprema RS-D scanner without any pre-processing. A simple contrast enhancement routine is applied by

$$I(i,j)' = \bar{I}(i,j) + \beta\left(\frac{I(i,j) - \bar{I}(i,j)}{\sigma(i,j)}\right) \tag{19}$$

where $I(i,j)'$ and $I(i,j)$ are the (i,j)th pixel values of contrast enhanced and scanned images, respectively, and $\bar{I}(i,j)$ and $\sigma(i,j)$ are mean and variance inside a small window centered at (i,j). The parameter β controls the amount of contrast enhancement. Examples of contrast enhanced images and the histograms of fingerprint images of collected fingerprints are shown on Fig. 8. It can be seen that with stronger contrast enhancement, images and their histogram becomes more and more like the ones of the Crossmatch dataset. The detection accuracy at EER with the scanned fingerprint images are 97.10, 95.65, 94.15, and 91.80 for $\beta = 0.4, 0.5, 0.6$, and 0.7. Excessive contrast enhancement in the pre-processing step is a possible explanation for relatively lower accuracy for Crossmatch dataset. It is reasonable for a liveness detection routine to be placed before the pre-processing and to be able to access images without contrast enhancement for improved liveness detection accuracy.

Liveness detection methods based hand-designed with domain knowledge usually require sophisticated feature extractions steps. The major arithmetic operations in the proposed liveness detection are matrix vector multiplications and calculations of activation functions. The numbers of nodes for the proposed network topology are 256, 128, 64, and 2. The sizes of the weight matrices are 256×128, 128×64, and 64×2 for $\mathbf{W}^1$, $\mathbf{W}^2$, and $\mathbf{W}^3$, respectively. L patches are used to determine the liveness of a fingerprint. Hence, it requires L matrix vector multiplications with the 256×128, 128×64, and 64×2 size matrices, and L times 256, 128,64, and 2 nonlinear activation function calculations are required for the liveness detection. Once the DBN is trained, the detection can be performed with straightforward operations of multiplications, additions, and evaluations of activation functions. The average detection time of the proposed method for LivDet2013 test samples is 76ms on a workstation with 3.40 GHz 8-core CPUs and 32 GB memory. The training times of the proposed DBNs without and with artificially augmented training sets are one and seven days, respectively, on the same workstation.

4. Conclusion

This paper presents a fingerprint liveness detection method developed through a systematic application of a deep learning technique. A DBN with multiple layers of RBM is trained with a set of live and fake fingerprints. The DBN is fine-tuned such that it provides the conditional probability that a patch from a given fingerprint image belongs to a live or fake fingerprint. A hypothesis test is set up to determine the liveness of a given fingerprint based on the outputs of the DBN with multiple patches from the fingerprint. The performance evaluated with datasets in the LivDet database shows that the proposed method accurately determines the liveness of fingerprints for all the database without requiring any information on the recognition system. With the topology of the DBN being simple, the proposed method can provide efficient and effective fingerprint liveness detection prior to the recognition to prevent spoofing by fake fingerprints.

Acknowledgment

This work is supported by Suprema Inc.

References

[1] A. Al-Ajlan, Survey on fingerprint liveness detection, in: Proceedings of the IEEE International Workshop on Biometrics and Forensics (IWBF), 2013, pp. 1–5.
[2] Y. Bengio, Foundations and Trends:Learning Deep Architectures for AI, Machine Learning 2 (1) (2009) 1–127.
[3] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2007) 153.
[4] K. Delac, M. Grgic, A survey of biometric recognition methods, in: Proceedings of the 46th International Symposium Electronics in Marine (ELMAR), IEEE, 2004, pp. 184–193.
[5] R. Frassetto Nogueira, R. de Alencar Lotufo, R. Campos Machado, Evaluating software-based fingerprint liveness detection using convolutional networks and local binary patterns, in: Proceedings of the IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS), IEEE, 2014, pp. 22–29.
[6] L. Ghiani, A. Hadid, G.L. Marcialis, F. Roli, Fingerprint liveness detection using binarized statistical image features, in: Proceedings of the IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), IEEE, 2013, pp. 1–6.
[7] L. Ghiani, G.L. Marcialis, F. Roli, Fingerprint liveness detection by local phase quantization, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR), IEEE, 2012, pp. 537–540.
[8] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G.L. Marcialis, F. Roli, S. Schuckcrs, Livdet 2013 fingerprint liveness detection competition 2013, in: Proceedings of the International Conference on Biometrics (ICB), IEEE, 2013, pp. 1–6.
[9] C. Gottschlich, E. Marasco, A.Y. Yang, B. Cukic, Fingerprint liveness detection based on histograms of invariant gradients, in: Proceedings of the IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2014, pp. 1–7.
[10] P. Hamel, D. Eck, Learning features from music audio with deep belief networks, in: Proceedings of the 11th International Society for Music Information Retrieval Conf.,, Utrecht, The Netherlands, 2010, pp. 339–344.
[11] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.
[12] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[13] J. Jia, L. Cai, K. Zhang, D. Chen, A new approach to fake finger detection based on skin elasticity analysis, in: Advances in Biometrics, Springer, 2007, pp. 309–318.

[14] P. Johnson, S. Schuckers, Fingerprint pore characteristics for liveness detection, in: Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, 2014, pp. 1–8.
[15] H. Lee, C. Ekanadham, A.Y. Ng, Sparse deep belief net model for visual area v2, in: Advances in Neural Information Processing Systems, 2008, pp. 873–880.
[16] P. Liu, S. Han, Z. Meng, Y. Tong, Facial expression recognition via a boosted deep belief network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1805–1812.
[17] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer Science & Business Media, 2009.
[18] E. Marasco, A. Ross, A survey on antispoofing schemes for fingerprint recognition systems, ACM Comput. Surv. (CSUR) 47 (2) (2014) 28.
[19] S.A. Memon, Novel Active Sweat Pores Based Liveness Detection Techniques for Fingerprint Biometrics, Brunel University School of Engineering and Design, 2012 Ph.D. thesis.
[20] A.-r. Mohamed, D. Yu, L. Deng, Investigation of full-sequence training of deep belief networks for speech recognition., in: INTERSPEECH, 2010, pp. 2846–2849.
[21] S.B. Nikam, S. Agarwal, Local binary pattern and wavelet-based spoof fingerprint detection, Int. J. Biometrics 1 (2) (2008) 141–159.
[22] K.A. Nixon, R.K. Rowe, Multispectral fingerprint imaging for spoof detection, in: Defense and Security, International Society for Optics and Photonics, 2005, pp. 214–225.
[23] S.T. Parthasaradhi, R. Derakhshani, L. Hornak, S.A. Schuckers, Time-series detection of perspiration as a liveness test in fingerprint devices, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 35 (3) (2005) 335–343.
[24] P.V. Reddy, A. Kumar, S. Rahman, T.S. Mundra, A new antispoofing approach for biometric devices, IEEE Trans. Biomed. Circuits Syst. 2 (4) (2008) 328–337.
[25] M. Sahasrabudhe, A.M. Namboodiri, Fingerprint enhancement using unsupervised hierarchical feature learning, in: Proceedings of the Indian Conference. on Computer Vision Graphics and Image Processing, ACM, 2014, p. 2.
[26] M.M.R. Sazal, S.K. Biswas, M.F. Amin, K. Murase, Bangla handwritten character recognition using deep belief network, in: Proceedings of the International Conference on Electrical Information and Communication Technology (EICT), IEEE, 2014, pp. 1–5.
[27] C. Sousedik, C. Busch, Quality of fingerprint scans captured using optical coherence tomography, in: Proceedings of the IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2014, pp. 1–8.
[28] N. Srivastava, Improving Neural Networks with Dropout, University of Toronto, 2013 Ph.D. thesis.
[29] D. Yu, M.L. Seltzer, Improved bottleneck features using pretrained deep neural networks, in: INTERSPEECH, vol.237, 2011, p. 240.