Journal of Visual Communication and Image Representation 54 (2018) 182–192

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

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

Face spoofing detection with local binary pattern network[☆]

Lei Li[a,*], Xiaoyi Feng[a], Zhaoqiang Xia[a], Xiaoyue Jiang[a], Abdenour Hadid[a,b]

[a] *School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi, China*
[b] *Center for Machine Vision and Signal Analysis (CMVS), University of Oulu, Oulu, Finland*

ARTICLE INFO

Keywords:
Face spoofing detection
Deep learning
Local binary pattern

ABSTRACT

Nowadays, face biometric based access control systems are becoming ubiquitous in our daily life while they are still vulnerable to spoofing attacks. So developing robust and reliable methods to prevent such frauds is unavoidable. As deep learning techniques have achieved satisfactory performances in computer vision, they have also been applied to face spoofing detection. However, the numerous parameters in these deep learning based detection methods cannot be updated to optimum due to limited data. Local Binary Pattern (LBP), effective features for face recognition, have been employed in face spoofing detection and obtained promising results. Considering the similarities between LBP extraction and convolutional neural network (CNN) that the former can be accomplished by using fixed convolutional filters, we propose a novel end-to-end learnable LBP network for face spoofing detection. Our network can significantly reduce the number of network parameters by combing learnable convolutional layers with fixed-parameter LBP layers that are comprised of sparse binary filters and derivable simulated gate functions. Compared with existing deep leaning based detection methods, the parameters in our fully connected layers are up to $64x$ savings. Conducting extensive experiments on two standard spoofing databases, i.e., Relay-Attack and CASIA-FA, our proposed LBP network substantially outperforms the state-of-the-art methods.

1. Introduction

Face recognition system has become prevalent in a broad range of applications [1,2]. However, face spoofing attacks are always considered as serious threats to these systems [3,4]. Specially, improvements on information acquisition make spoofing algorithms much easier to be designed, leading to the situation even more severe. In this sense, it is inevitable to develop effective spoofing detection (also called face anti-spoofing) methods and integrate them with biometric systems to prevent such frauds.

A face spoofing attack occurs when a person tries to masquerade as someone else by falsifying face and thereby attempting to gain illegitimate access and advantages. Based on different fake faces, four types of attacks can be considered: (i) printed face photos, (ii) displayed face images, (iii) replayed videos and (iv) 3D masks. In printed face photo attacks, the attacker prints the face photo on paper and puts it in front of the camera. In both displayed image and replayed video attacks, a digital screen is used to show the face images or replayed videos. For 3D mask scenario, the attacker uses a 3D mask of authorized person to enter the system. Among above mentioned face spoofing attack scenarios, the printed face photo attacks, displayed face image attacks, and 3D mask attacks cannot exhibit facial aliveness signals, such as eye blinking, lip movements and facial expression changes. Therefore, detecting replayed video attacks and distinguishing them from real faces are more challenging. However, it is also possible to include some aliveness information even in the printed face photos and 3D masks. For instance, the attacker may cut the eyes area in the mask in order to imitate the eye blinking pattern of the authorized subject. Fig. 1 shows an example for different face spoofing attacks.

Due to the printing defects and noises in cameras' systems, fake faces may have some disadvantages such as lower image quality and lack of high frequency information, which can be presented by texture property [2,7]. Therefore, facial texture analysis has attracted the attention of research community to tackle the problem of face spoofing attacks [8–14]. Especially with the success of deep learning in computer vision and multimedia analysis tasks [15–19], deep texture analysis based detection algorithms have also been applied in the face anti-spoofing problems [20,21]. However, the fake face data is limited, which makes it difficult to train a deep network and severely constrains the detection performances. Compared with deep learning, the handcrafted Local Binary Pattern (LBP) [22] features and its variations have stronger characterization ability in texture analysis [23–26] and have

[☆] This paper has been recommended for acceptance by Jiwen Lu.
[*] Corresponding author.
E-mail address: lilei_npu@mail.nwpu.edu.cn (L. Li).

https://doi.org/10.1016/j.jvcir.2018.05.009
Received 14 December 2017; Received in revised form 6 May 2018; Accepted 11 May 2018
Available online 16 May 2018
1047-3203/ © 2018 Elsevier Inc. All rights reserved.

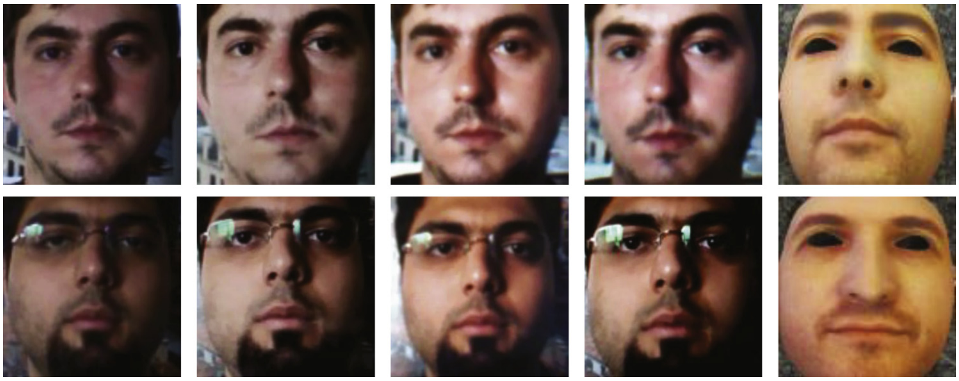


Fig. 1. Some samples of real and fake faces. From the left to the right: real faces, printed face photos, digital face images, recorded face videos and 3D masks. The first column to the fourth column are from the Replay-Attack database [5] and the last column is from the 3DMAD database [6].

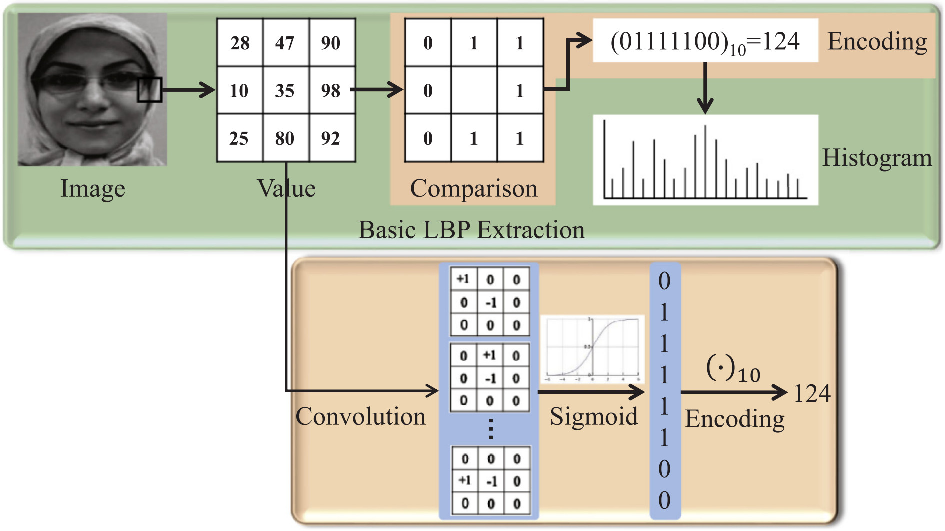


Fig. 2. The similarities between convolutional operation and LBP extraction process.

been applied in face spoofing detection [5,13,14]. In fact, the essence of LBP extraction is to convolve the image with a set of fixed convolutional filters. As illustrated in Fig. 2, it shows how to use the convolutional filters to realize the process of LBP extraction. Considering the similarities between convolutional network and LBP extraction, we propose a new network structure for face spoofing detection.

Usually, training a deep network with fully learnable parameters and limited data is computationally expensive and prone to over-fitting. So, some variants of convolutional neural networks, such as using binary and fixed weights, have been proposed to tackle this problem [27,28]. However, the parameters in fully connected layers are ignored. More importantly, there are more parameters in fully connected layers compared to convolutional layers. For instance, in VGG-face model [29], the parameter magnitude of fully connected layer is up to tens of millions (4096 × 4096 = 16777,216), whereas there are only 1728 parameters in the first convolutional layer. Apart from adopting binary and fixed weights, Hubara et al. [30] introduced some binary activate functions in their network. Based on binary weights and binary activate functions, their network can drastically reduce the memory consumption. However, unlike them, the purpose of our network is to reduce the number of training parameters.

The architecture of our proposed network is illustrated in Fig. 3. In our network, we save the parameters in convolutional layers and fully connected layers by introducing completely LBP layers. More specifically, in LBP layers, we use fixed sparse binary filters to save the parameters in convolutional layers and utilize a novel statistical histogram function to save the parameters in fully connected layers. We train and test our proposed method on two public available databases: Replay-Attack and CASIA-FA [7]. The experimental results demonstrate the effectiveness of our proposed method in face spoofing detection outperforming the state-of-the-art approaches.

The remainder of the paper is organized as follows: Section 2 reviews the existing state-of-the-art methods of face spoofing detection. Then our proposed learnable LBP network is introduced in Section 3. Section 4 provides the details of experimental setup. After that, we show and analyze the detection results in Section 5. Finally, in Section 6, we conclude this paper and discuss the directions for future research.

2. Related work

Since the early 2000s, many face spoofing detection methods have been proposed [5–7,13,31–37]. Based on different clues, we categorize these methods into four categories: (i) texture analysis [5,13,31], (ii) motion analysis [32,33,38], (iii) image quality analysis [7,34,35], and (iv) hardware based methods [6,36,37].

2.1. *Texture analysis based methods*

Face spoofing detection methods based on texture analysis usually analyze the printing failures, blurring and other effects caused by limitations of the printers and display devices. A major trend of these methods is the analysis of micro-texture pattern descriptor. Maatta

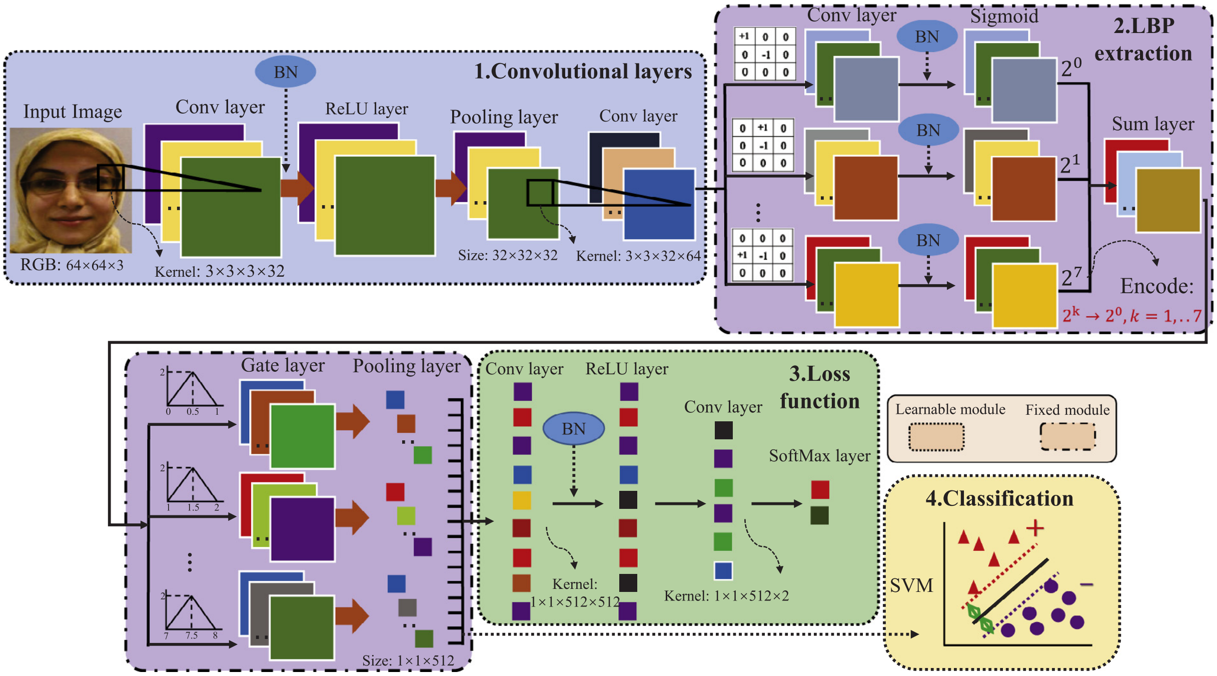


Fig. 3. The architecture of our proposed face anti-spoofing network. It consists of four modules, which showed with different colors (blue, purple, green and yellow). *Conv* is the abbreviation of convolutional.

et al. [31] extracted multi-scale LBP features to describe the differences between the real and fake faces. Chingovska et al. [5] utilized different kinds of LBP features and classifiers to detect fake faces. In [5], experiments showed that the Support Vector Machine (SVM) [39] is the best classifier for spoofing detection. Moreover, Akshay et al. extracted Haralick features [40] from video frames and obtained some good detection results. To capture the color differences in luminance and chrominance, Boulkenafet et al. [13] proposed a method by analyzing color-textures. They extracted LBP features from different color spaces, such as RGB, HSV and YCbCr, and concatenated them into a single feature vector. In this way, the fake faces can be distinguished from the real ones. In another work [14], Boulkenafet et al. extracted the multi-scale textural features by constructing Gaussian pyramids [41] and used them to detect fake faces. With the recent advances of deep learning in computer vision [15,16], some face anti-spoofing works have also begun to focus on analyzing deep textures. Yang et al. [21], proposed an end-to-end CNN model for spoofing detection. They fed their proposed model with face images of different scales and found that the better detection accuracy can be achieved when considering some regions of background rather than pure face regions. Instead of using fully connected layers, Li et al. [42] extracted the hand-crafted features from convolutional feature maps and showed some promising performances. In another work, Xu et al. [43] proposed a long short memory network (LSTM) [44] to detect face spoofing, which can obtain temporal texture variations from video sequences.

2.2. Motion analysis based methods

Apart from texture variations, motion is also a vitally important characteristic, especially for printed photo and displayed image attacks. Anjos et al. [38] analyzed the motion correlation coefficient between face region and background. They hypothesized that the fake faces and background have lower correlation coefficient than the real faces. Pan et al. [32] used an undirected conditional random field framework to detect blinking. This is because the involuntary eyes blinking typically occurs in the interval of 2 to 4 s [45]. Moreover, Pereira et al. [9] and Phan et al. [46] extracted the features of LBP-TOP [47] and LDP-TOP [48] to describe the facial motion variations, respectively. Santosh et al. [49] used the dynamic mode decomposition (DMD) to capture the dynamics of movements. On the other side, planar object movements can also be analyzed for face anti-spoofing task. Therefore, Bao et al. [50] addressed the problem of detecting planar photo and displayed attacks. There is another fact that the 3D shapes of faces have different motion deformation patterns compared with planar objects. So Tan et al. [33] used Difference-of-Gaussians (DoG) to extract the differences in motion information between the real and fake faces. Although motion analysis based methods are effective to static image attacks and 3D mask attacks, these methods can still be easily deceived by replayed video attacks. Therefore, it is necessary to request the subject to perform specific movements [51,52].

2.3. Image quality analysis based methods

Since reproduced face images and videos usually have lower quality in comparison with real ones, some methods [7,34,35,53] took advantage of high frequency components of the data to recognize fake faces. Specially, Wen et al. [2] extracted specular reflection, blurriness, chromatic moment and color diversity caused by the medium of liquid crystal display (LCD) screen to describe the light reflection differences between the real and -fake faces. The low quality of image acquisition sensors is another important factor for image quality. So, Feng et al. [54] fused several different advanced image-quality features with dense optical flow features to capture the fake faces with low quality. Moreover, Pinto et al. [53] analyzed the Fourier spectrum [55] of the noise signature to obtain the features that can distinguish the real faces from printed fake faces. Such image quality analysis based approaches are expected to work well for low-resolution printed photo attacks and when using crude face masks, but are likely to fail for high quality spoof artifacts.

2.4. Hardware based methods

In addition to the analysis of face images and videos, advanced hardwares, e.g., depth, multi-spectral and light-field cameras, have been utilized for face anti-spoofing task. For instance, Erdogmus et al. [6] proposed to detect face spoofing by analyzing depth information and achieved promising results. This is because there is no any depth information in spoofing attacks using 2D mediums. Instead of extracting the light reflection differences by intrinsic image decomposition, Pavlidis et al. [36,37] achieved it by computing the upper-band of near-infrared (NIR) spectrum. More recently, light-field cameras allow exploiting disparity and depth information from a single capture. Therefore, Kim et al. [56–58] introduced these kinds of cameras into face anti-spoofing. Even though these hardware-based methods can get good performances, some of them might present operation restrictions in certain conditions. For instance, the sunlight can cause severe perturbations for NIR and depth sensors; wearable 3D masks are obviously challenging for those methods relying on depth data.

3. Proposed method

3.1. Overall architecture

In order to tackle the problem of insufficient training data and excessive network parameters, we propose a novel end-to-end learnable LBP network for face spoofing detection. By integrating fixed sparse binary filters and derivable statistical histogram functions, our proposed network has three distinctive advantages: (i) drastically reducing the network parameters in convolutional and fully connected layers; (ii) effectively training the parameters directly with limited data; (iii) completely realizing the basic LBP extraction process. As shown in Fig. 3, our network comprises of four modules: convolutional layers, LBP layers, loss function and classification layers. In the network, convolutional layers and LBP layers are used to extract the simulated LBP features, loss function module is exploited to train the network parameters and classification layers are utilized to predict whether the input face is a spoofing attack.

Towards original images, the convolutional feature maps are obtained by forward-propagating the raw pixels through the first module. More specifically, the first module contains two convolutional layers, one pooling layer and one restricted linear unit (ReLU) layer. Additionally, to eliminate the impact of filter parameters initialization, a batch normalization (BN) layer [59] is adopted before the ReLU layer as

$$\widehat{X} = \gamma\cdot\frac{X-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta \tag{1}$$

where X is the input of BN layer. μ and σ are the mean and variance of X, respectively. γ and β are scale and shift factors, and ϵ is a constant added to the variance for numerical stability.

In LBP layers, the simulated LBP features are extracted to describe the texture information of convolutional feature maps obtained by the first module. Compared with existing deep learning based spoofing detection methods [20,21], the fixed sparse binary filters and the novel statistical histogram function are used to connect the convolutional feature maps instead of simply combining convolutional and ReLU layers. By introducing LBP layers, the parameters in convolutional layers of this module and the parameters in fully connected layers of loss function module can be drastically saved. To make the back propagation (BP) algorithm [60] work well, in the next section, we will explain how to extract LBP features based on convolutional network in details.

For face spoofing detection, its essence is to classify the real face and fake face.

Algorithm 1 The flow of classification module.

```
Input:
  The trained LBP network, denoted as $M_{LBP}$.
  The training face videos and labels, denoted as $V_{train}$ and $L_{train}$,
 respectively.
  The testing face videos, denoted as $V_{test}$.
Do:
  (1) Train a SVM classifier
    (a) Divide each training video into some continuous frames,
   denoted as $f_i=$
    $[f_{i_1},f_{i_2},...,f_{i_{i'}},...,f_{i_{n'}}]$, where $f_{i_{i'}}$ is $i'_{th}$ frame of $i_{th}$ video in $V_{train}$;
    (b) Feed $f_{i_i}$ into $M_{LBP}$ and extract the LBP features
 $l_i = [l_{i1},l_{i2},...,l_{i_{i'}},...,$
    $l_{i_{n'}}]$, where $l_{i_{i'}}$ is the LBP features of $f_{i_{i'}}$;
    (c) Average $l_i$ and get a feature vector $\overline{l_i}$, where $\overline{l_i} = \frac{1}{n'}\sum_{i'}^{n'} l_{i_{i'}}$;
    (d) Use the averaged feature vectors $[\overline{l_1},\overline{l_2},...,\overline{l_i},...\overline{l_v}]$ and $L_{train}$ to
 train a
    SVM classifier, denoted as $M_{SVM}$.
  (2) Test the trained SVM classifier
    (a) As training videos, get the averaged LBP features of the $j_{th}$
 test video,
     denoted as $\overline{l_j}$, where $\overline{l_j} = \frac{1}{n'}\sum_{j'}^{n'} l_{j_{j'}}$;
  (b) Feed $\overline{l_j}$ into $M_{SVM}$ and get the classification results.
```

Therefore, after LBP features extraction, the most commonly used *SoftMax* loss function is employed to measure the classification error [29,61]. In training stage, the *Softmax* loss function can maximize the probability of the right class and fine tune the network parameters based on the algorithm of BP, as shown in Section 6,

$$\mathscr{F}(Y) = \sum_{i=1}^{n} \{log(e^{y_{i1}} + e^{y_{i2}} + \cdots + e^{y_{iv}}) + y_{ir}\} \tag{2}$$

where i is the index of training samples, and n is the number of training samples. $Y = [Y_1,Y_2,...,Y_i,...,Y_n]$ is the label set, $Y_i = [y_{i1},y_{i2},...,y_{ir},...,y_{ik}]$ is the predict vector of the i_{th} training sample, y_{ir} is the predict value of the i_{th} class, and v is the number of classes.

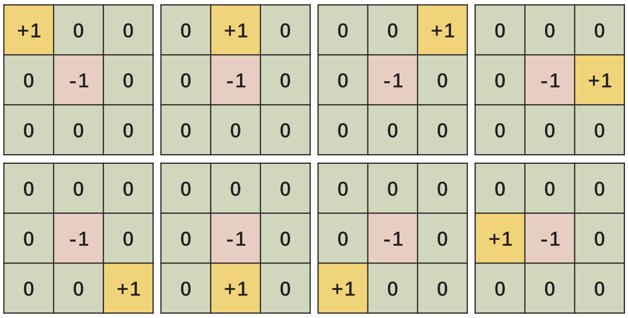
In testing stage, the LBP features are extracted from a video sequence and averaged into a feature vector to combine the facial movement information. Then a SVM classifier is used to classify the averaged features. The main flow of classification module is described in Algorithm 1.

3.2. LBP Layers

The process of basic LBP extraction can be divided into three steps: (i) comparing the pixel value with its neighborhoods and binarizing the results, (ii) encoding the binarization results, (iii) obtaining the statistical histogram of the encoded results. Based on this, we realize them in our network.

Our proposed approach and the one proposed in [27] share some similarities but also major differences. A detailed comparison is as follows:

(1) Similarities: Both networks have realized the first and the second steps of LBP extraction and use the same function (i.e. sigmoid) to binarize the comparison results.
(2) Differences: (a) In the process of encoding, our network sets all coding weights to 2^0 instead of using different encoding weights (2^7,2^6,2^5,2^4,2^3,2^2,2^1,2^0), which can prevent unbalanced network and make the extracted features have rotational invariance. (b) Instead of cascaded convolutional layers following encoding operation, we



+1	0	0	0	+1	0	0	0	+1	0	0	0
0	-1	0	0	-1	0	0	-1	0	0	-1	+1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	-1	0	0	-1	0	0	-1	0	+1	-1	0
0	0	+1	0	+1	0	+1	0	0	0	0	0

Fig. 4. The fixed sparse binary convolutional filters for comparison.

directly extract the statistical histogram features [62,63] to describe the encoded results. Compared with cascaded convolutional layers, the statistical histogram can effectively save the parameters in fully connected layers of loss function module.

3.2.1. Comparison and binarization

In our network, we compare the value of the center pixel with its D neighborhoods. The comparison can be realized by D different convolutional filters. In these filters, the parameters are fixed and initialized with sparse binary values. In our LBP network, we set $D = 8$, as illustrated in Fig. 4. After comparison, each convolutional feature map generated from the first module can get 8 comparison results.

For binarizing the comparison results, a straightforward way is using a step function to activate the comparison results. The step function is defined as $b(x) = \{0,x \leq 0;1,\mathit{otherwise}\}$, where x is the pixel of the comparison results. However, in training stage, this will causes the gradients vanish. Therefore, the *sigmoid* function is selected $s(x) = \frac{1}{1+e^x}$. Unfortunately, the range of activation region of *sigmoid* function is from -5 to $+5$. To limit the comparison results in an appropriate range, a BN layer is introduced before the Sigmoid layer, as shown in Fig. 3.

3.2.2. Encoding

In basic LBP, the encoding operation is illustrated in Eq. (3).

$$s(\hat{x}) = \sum_{c=1}^{8} 2^{c-1}\cdot x_c, \quad s.\ t. \quad c = 1,2,...,8 \tag{3}$$

where c is the index of the neighborhood of center pixel $\hat{x}$,2^{c-1} is the encoding weight, and x_c is the value of the pixel. However, it is unsuitable in our proposed network. More specifically, when the back propagation algorithm is used to compute the parameters' gradients, encoding operation will lead to the result: The input gradients in Sigmoid layer of LBP extraction module have different artificial weights, as illustrated in Eq. (4).

$$\delta(x_c) = \frac{\partial g(x_c)}{\partial x_c}\cdot\delta^{up}(x_c) = 2^{c-1}\cdot\delta^{up}(x_c) \tag{4}$$

where $\delta^{up}(x_c)$ the gradients of Sum layer. After that, the gradients are transferred into the second convolutional layer and combined together based on the fixed sparse binary filters. Although we can combine them together, due to different magnitudes of encoding weights, the feature map with the weight 2^7 would obtain $2^7/2^8$ of the back-propagated gradients while the feature map with the weight 2^0 only can obtain $2^0/2^8$ of the back-propagated gradients. Therefore, we set all encoding weights to 2^0 to avoid the gradients vanishing in the feature map with small encoding weights, as shown in Fig. 3.

3.2.3. Statistical Histogram

After the encoding operation, we should calculate the statistical distribution of each encoded map, denoted by M_c. In basic LBP, a statistical function can achieve this; however, it cannot work in our proposed network. This is because there is no partial derivative of the statistical function. Therefore, we have to find a new way to get the statistical histogram.

The main process of a statistical function includes two steps: (i) finding the pixels that belong to a specific value range, (ii) counting the number of these pixels and computing their proportion. Inspired by this, we introduce a Gate layer and a Pooling layer to get statistical histogram. More specifically, each encoded feature map will be filtered by a series of derivable gate functions. Then the filtered feature maps are pooled by averaging operation and connected them into a statistical histogram. Since the encoded results M_c are in a range of 0–8, we divide the range into several K intervals. Here, we empirical set $K = 8$. This means that 8 gate functions are used to filter M_c, as illustrated in Eq. (5). Then each encoded map of M_c will get 8 different filtered feature maps, where M_G is noted as one feature map of them.

$$f(x_M) = \begin{cases} \varnothing(x_{M_c}) & \mathit{if} \quad 0 \leqslant x_{M_c} < 1 \\ \varnothing(x_{M_c}-1) & \mathit{if} \quad 1 \leqslant x_{M_c} < 2 \\ \cdots & \cdots \\ \varnothing(x_{M_c}-7) & \mathit{if} \quad 7 \leqslant x_{M_c} \leqslant 8 \end{cases} \tag{5}$$

where $\varnothing(\cdot)$ is the activate function, x_{M_c} is the pixel of M_c, and $\varnothing(\cdot)$ is illustrated in Eq. (6).

$$\varnothing(t) = \begin{cases} 4t & \mathit{if} \quad 0 \leqslant t < 0.5 \\ -4(t-1) & \mathit{if} \quad 0.5 < t \leqslant 1 \\ 0 & \mathit{otherwise} \end{cases} \tag{6}$$

After the process of gate functions, the statistical function is used to obtain the proportion of the specific value range in M_G, as illustrated in Eq. (7).

$$P(M_G) = \frac{\sum_{i=1}^{32}\sum_{j=1}^{32} x_{ij}}{32^2} \tag{7}$$

where x_{ij} is the pixel of M_G, the size of M_G is 32 × 32. In Fig. 5, it shows an example of computing the proportion of the value range from 2 to 3 in M_G. In our network, we use an average pooling layer with the kernel size 32 × 32 to realize the statistical function.

3.3. Implementation details

The parameters of our proposed LBP network are configured in Table 1. For the first convolutional layer, a convolutional filter with the

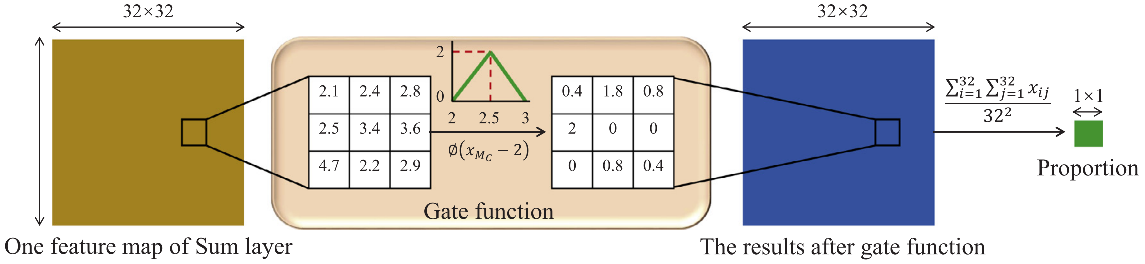

32×32
32×32
2
0
2.1 2.4 2.8
2.5 3.4 3.6
4.7 2.2 2.9
2 2.5 3
$\varnothing(x_{M_C} - 2)$
0.4 1.8 0.8
2 0 0
0 0.8 0.4
Gate function
$\frac{\sum_{i=1}^{32}\sum_{j=1}^{32} x_{ij}}{32^2}$
1×1
Proportion
One feature map of Sum layer
The results after gate function

Fig. 5. The process of computing the proportion of the value range from 2 to 3.

Table 1
The configuration parameters in our proposed network. *mPool* is the operation of max pooling, *aPool* is the operation of average pooling and FC is the fully connected layer.

layer	1	2	3	4	5	6	7	8
module	1	1	1	1	1	2	2	2
type	Conv	BN	ReLU	mPool	Conv	Conv	BN	Sigmoid
support	3	1	1	2	3	3	1	1
filt dim	3	–	–	–	3	–	–	–
num filts	32	–	–	–	64	–	–	–
stride	1	1	1	2	1	1	1	1
pad	1	0	0	0	1	1	0	1
layer	9	10	11	12	13	14	15	16
module	2	2	2	3	3	3	3	3
type	Sum	Gate	aPool	FC	BN	ReLU	FC	SoftMax
support	1	2	32	1	1	1	1	1
filt dim	–	–	–	1	–	–	1	–
num filts	–	–	–	512	–	–	512	–
stride	1	2	0	1	1	1	1	1
pad	0	0	0	0	0	0	0	0

size 3 × 3 is used and the total number of filters is 32. In the second convolutional layer, total 64 convolutional filters with the size 3 × 3 are used to extract more detailed information from the output of the first pooling layer. In the third convolutional layer, we fix the parameters of convolutional filters with the size 3 × 3 to compare the center pixel with its neighborhood. Except the 11_{th} layer, all pooling layers adopt the downsampling factor 2 × 2 and the dimensionality of two FC layers is 512.

For the filters in the first convolutional layer, the second convolutional layer and FC layers, we initialize them based on [64], as illustrated in Eq. (8).

$$W = \frac{rand(n_C)}{\sqrt{2/n_C}} \tag{8}$$

where $rand(\cdot)$ samples from a zero mean, unit standard derivation gaussian, and n_C is the channel number of its inputs. This ensures that all neurons in the network initially have approximately the same output distribution and empirically improves the rate of convergence.

Before performing the convolution and downsampling operations, the raw face images would be aligned and cropped to the size 64 × 64 × 3 for extracting the features. In training stage, the stochastic gradient descent (SGD) algorithm [65] is used to learn the network parameters. The momentum is set to 0.9 and weight decay 0.0005. The learning rate is set to 10^{-3} in the beginning and will be multiplied with damping factor 0.5 when all mini-batches are traversed and re-allocated randomly.

4. Experimental setup

4.1. Experiment data

We validate our proposed method on two public available face anti-spoofing databases: Replay-Attack [5] and CASIA Face Anti-spoofing [7]. Table 2 gives a comparison about these two databases, and a description of each database is given below.

4.1.1. Replay-attack

The IDIAP Replay-Attack database [1] [5] consists of 1300 video clips of real and attack attempts to 50 clients. These clients are divided into 3 subject-disjoint subsets for training, development and testing (15, 15 and 20, respectively). The real face videos are recorded under two different lighting conditions: *controlled* and *adverse*. Three types of attacks are created: printed photos, displayed images and replayed videos. In displayed image and replayed video attacks, high quality images and videos of the real client are replayed on iPhone 3GS and iPad display devices. For printed photo attacks, high quality images were printed on A4 papers and presented in front of the camera. Fig. 6 shows some examples of real and fake faces.

Table 2
A summary of two public-domain databases.

Database	Lighting scenarious	Subjects	Attack type	Subject gender	Subject age
Replay-Attack	2	50	Printed Photos Displayed images Replayed videos	Male 86% Female 14%	20 to 40 years
CASIA-FA	1	50	Warped photos Cut photos Replayed videos	Male 86% Female 14%	20 to 35 years

4.1.2. CASIA-FA

The CASIA Face Anti-Spoofing Database (CASIA-FA) [2] [7] consists of 600 video recordings of real and attack attempts to 50 clients, which are divided into two subject-disjoint subsets for training and testing (20 and 30, respectively). Two types of fake faces are created: printed photo and replayed video attacks. For the printed photo attacks, they include two different kinds: warped photos and cut photos. The real and the attack attempts were recorded using three camera devices with: low, normal and high resolution. Fig. 7 shows some examples of real and fake faces.

4.2. Evaluation protocol

For the performance evaluation, we follow the overall protocol associated with the two databases. For each database, we use the training set to learn our LBP network and the testing set to evaluate the performance. On CASIA-FA, the results are evaluated in term of Equal Error Rate (EER). Replay-Attack also provides a development set to tune the model parameters. Thus, the results are reported in term of EER on the development set and Half Total Error Rate (HTER) on the test set, illustrated in Eq. (9).

$$HTER = \frac{FRR(\varkappa,\mathscr{D}) + FAR(\varkappa,\mathscr{D})}{2} \tag{9}$$

where $\mathscr{D}$ denotes the used database and the value of $\varkappa$ is estimated on EER. $FRR(\varkappa,\mathscr{D})$ is the false rejection rate for the real faces and $FAR(\varkappa,\mathscr{D})$ is the false acceptance rate for the fake faces.

5. Results and discussion

5.1. Tested on replay-attack and CASIA-FA databases

To evaluate the effectiveness of our proposed LBP network, we perform experiments on Replay-Attack and CASIA-FA databases. More specifically, we test on three kinds of attacks: printed photos, displayed images and replayed videos. For CASIA-FA, there are two kinds of print fake faces: warped photos and cut photos. Caused by the impact of eye blinking, the experiments of CASIA-FA are conducted on warped photo attacks and cut photo attacks, respectively.

Table 3 shows the detection results of different attacks. For Replay-

[1] https://www.idiap.ch/dataset/replayattack/download-proc.

[2] http://www.cbsr.ia.ac.cn/english/FaceAntiSpoofDatabases.asp.

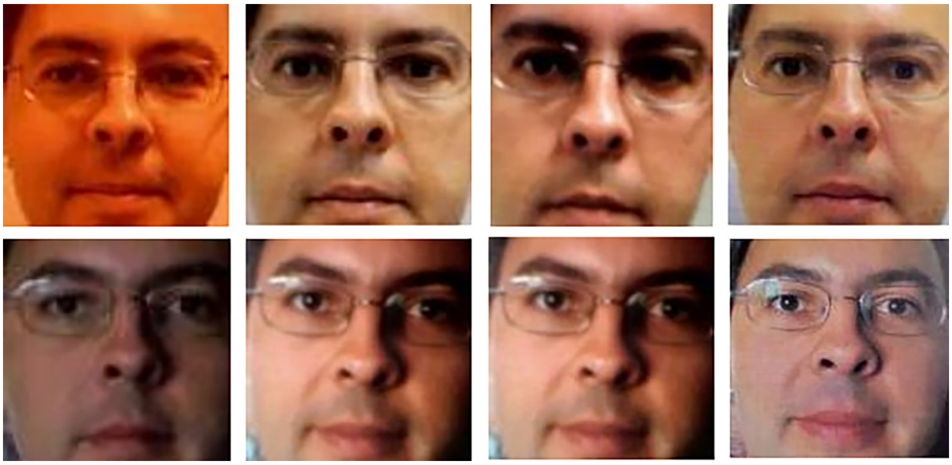


Fig. 6. Samples from Replay-Attack database. The first row presents images taken from the controlled scenario, while the second row corresponds to the images from the adverse scenario. From the left to the right: real faces and the corresponding displayed image, replayed video and printed photo attacks.

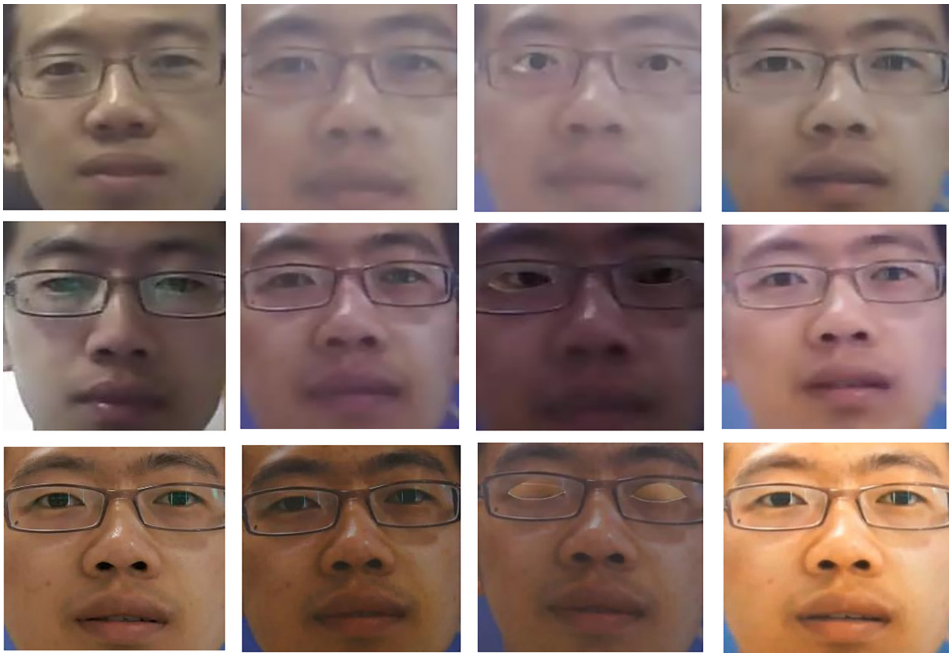


Fig. 7. Samples from CASIA-FA database. From top to bottom: low, normal and high quality images. From the left to the right: real faces and the corresponding warped photo, cut photo and replayed video attacks.

Table 3
The results of our LBP network tested on Replay-Attack and CASIA-FA databases.

Spoofing Attacks	Replay-Attack		CASIA-FA	
	EER(%)	HTER(%)	EER(%)	
Printed photos	2.8	1.9	Wraped photos Cut photos	17.4 11.1
Displayed images	7.8	9.6	-	
Replayed videos	5.1	3.8	14.3	
All attacks	0.6	1.3	2.5	

Attack database, the best results are obtained on printed photo attacks with EER = 2.8% and HTER = 1.9%, which is twice lower than replay video attacks. When tested on CASIA-FA, the EER of warped photo and replayed video attacks are up to 17.4% and 14.3%, respectively. Comparing the warped photo attacks with cut photo attacks, the former contains eye blinking and it should be more difficult to be detected. However, it is surprisingly that the EER of cut photo attacks is better than the warped photo attacks. After careful observation, we find that there some significant differences between the cut eye and the real eye, as illustrated in Fig. 8.

To evaluate the performance more comprehensively, Fig. 9 and Fig. 10 provide the ROC curves of Relay-Attack and CASIA-FA databases, respectively. From these curves, we can find our proposed method can get better detection results on replayed video attacks than printed photo and displayed image attacks.

At end, we train and test our LBP network on all attacks, which means that the training and testing data include print photo, displayed image and replayed video attacks. The detection results are illustrated in Table 3. From the detection results, it can be clearly see that increasing the number and diversity of the training data can significantly improve the performance in comparison to using only single kind of face spoofing. For instance, the EER and HTER of Replay-Attack decreased from 7.8% to 0.6% and decreased from 9.6% to 1.3%, respectively. For CASIA-FA, the EER decreased more than 6 times (from

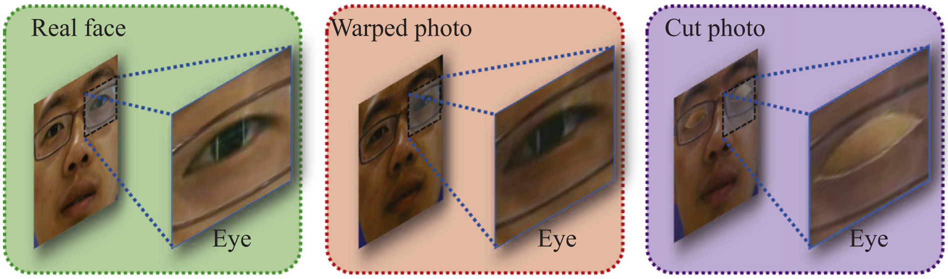


Fig. 8. The visible difference of the eye region in CASIA-FA database. Compared with warped photos, the eye region of cut photos is different with the real eye, which means more easily to be detected.

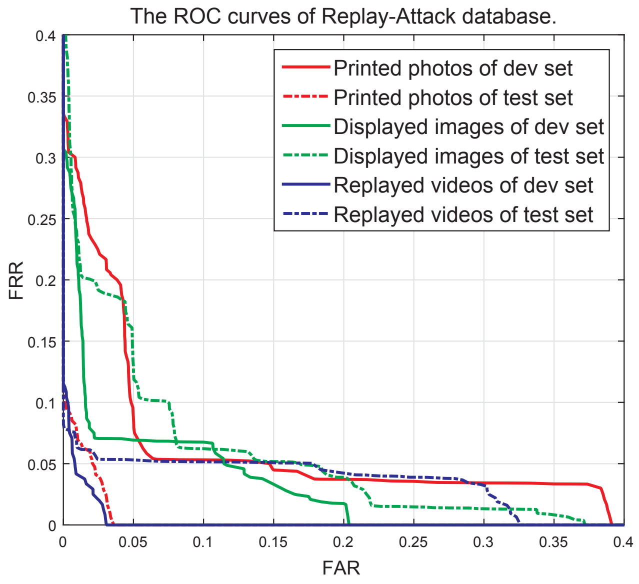


Fig. 9. Trained on different attacks of Replay-Attack database.

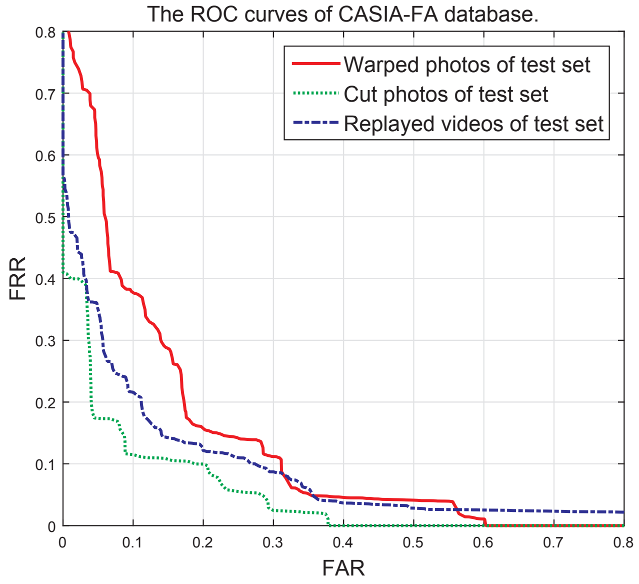


Fig. 10. Trained on different attacks of CASIA-FA database.

17.4% to 2.5%). Fig. 11 shows the ROC curves of Replay-Attack and CASIA-FA. From these curves, it can be observed that the fake faces of Replay-Attack are much easier to be detected than CASIA-FA.

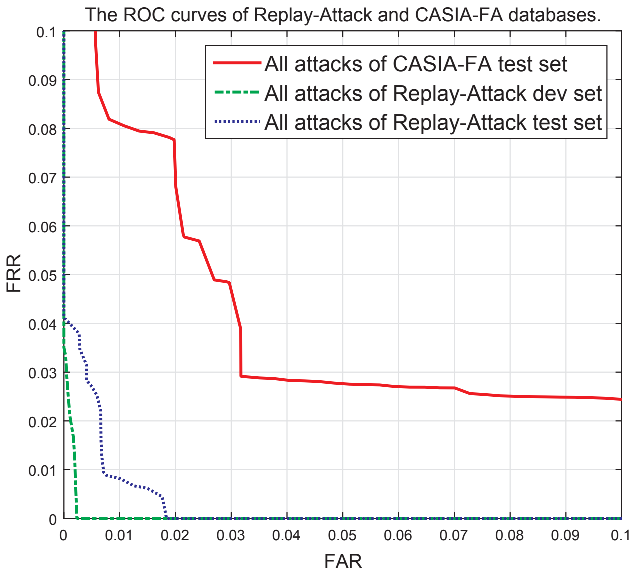


Fig. 11. Trained on all attacks of Replay-Attack and CASIA-FA, respectively.

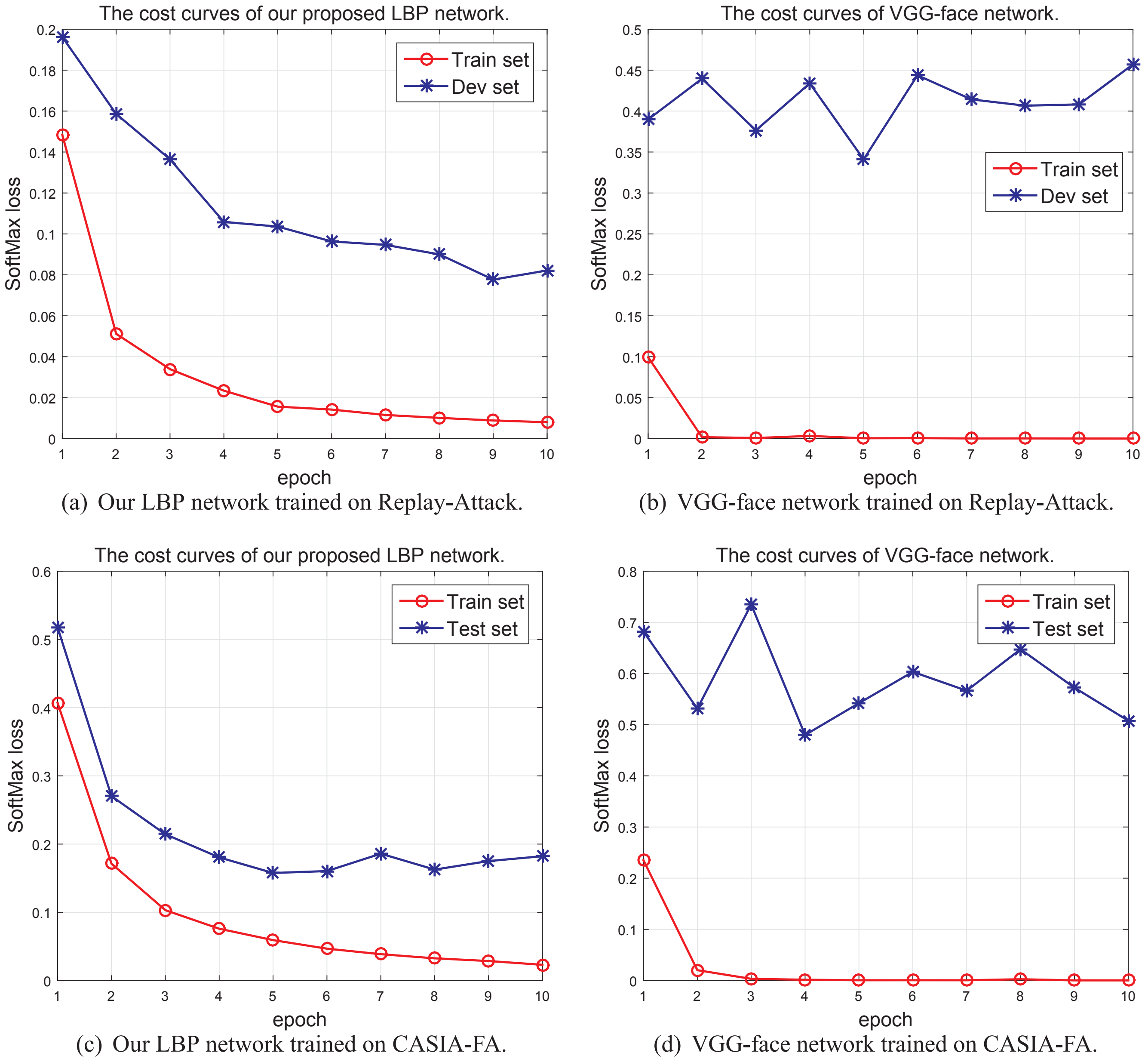
5.2. Time and memory consumptions

In order to compare the performance of our network in time and memory consumptions, we select VGG-face network [29] as the baseline. Because VGG-face is used for face recognition, we have to fine tune it for face spoofing detection. To be fair, in both VGG-face network and our proposed LBP network, we use the same training data, training parameters, and loss function. We perform the comparison experiments on a HP workstation with the following configuration: Windows 10 Enterprise Edition operating system, Matlab 2015a, 64G memory, two Intel E-52620 v3 CPUs and one NVIDIA GeForce GTX 1080 Ti GPU device.

The time and memory comparison results are illustrated in Table 4. As can be seen from the table, our proposed network only needs 122 MB memory during training stage, which is up to 25x savings compared with VGG-face network. Moreover, our network is faster than the VGG-face network in their training iterations. Based on the size of the input

Table 4
Comparison the mean time and memory cost between our network and VGG-face network.

20ptNetwork	Training Stage		Testing Stage	
	Time (s/epoch)	Memory (MB/image)	Time (s/image)	Memory (MB/image)
VGG-face	1155.04	3051	0.021	1123
Our Proposed Network	838.76	122	0.03	87



(a) Our LBP network trained on Replay-Attack.

(b) VGG-face network trained on Replay-Attack.

(c) Our LBP network trained on CASIA-FA.

(d) VGG-face network trained on CASIA-FA.

Fig. 12. The cost curves of Replay-Attack and CASIA-FA databases.

Table 5
The detection results of our network and VGG-face network.

Network	Replay-Attack		CASIA-FA
	EER(%)	HTER(%)	EER(%)
VGG-face	8.4	4.3	5.2
Our Proposed network	0.6	1.3	2.5

image (64 × 64 vs. 224 × 224) and the depth of the network (16 vs. 37), our proposed network actually should be much faster than VGG-face network. This is because our network takes a lot of time to extract the simulated statistical histograms. This also can explain why our network runs slowly compared against VGG-face network in testing stage. For the loss curves shown in Fig. 12, our network takes longer time to achieve convergence. In spite of this, our network has not been over fitted compared with VGG-face network, which can be revealed from the blue curves of Fig. 12 and the detection results of Table 5.

5.3. Comparison against state-of-the-art methods

Table 6 provides a comparison against the state-of-the-art methods. It can be seen that our proposed detection method outperforms them on both Replay-Attack and CASIA-FA databases. For Replay-Attack database, the best EER of these methods is 0.7%, which is very close to our 0.6%. However, our HTER has decreased their corresponding HTER from 3.1% to 1.3%. For CASIA-FA database, our proposed method yields in the best EER, which further illustrates the effectiveness of our method. Compared with deep learning methods [21,42,43], our proposed method has greatly improved the performances (Replay-Attack: EER decreased from 6.1% to 0.6%, Replay-Attack: HTER decreased from 4.3% to 1.3%, and CASIA-FA: EER decreased from 7.3% to 2.5%).

6. Conclusion

We proposed to approach the problem of face spoofing detection by a novel LBP network. The proposed network combines the hand-crafted features with deep learning and can reduce the network parameters by

Table 6
Comparison between the performance of our proposed method and the state-of-the-art methods.

Methods	Replay-Attack		CASIA-FA
	EER(%)	HTER(%)	EER(%)
Motion + LBP [12]	4.5	5.1	-
Motion [11]	11.6	11.7	26.6
LBP [5]	13.9	13.8	18.2
LBP-TOP [9]	7.9	7.6	10.0
Spectral Cubes [53]	-	2.8	14.0
IQA [35]	-	-	13.3
Haralick Features [66]	-	-	6.7
LDP-TOP [46]	2.5	1.7	8.9
DMD [49]‡	5.3	3.7	21.7
Color LBP [13]‡	0.9	4.9	7.1
Scale LBP [14]	0.7	3.1	4.2
Deep CNN [21]	6.1	2.1	7.3
Partial CNN [42]	2.9	4.3	4.5
LSTM-CNN [43]	-	-	5.2
Our Proposed Method	**0.6**	**1.3**	**2.5**

‡ was reported in [14].

obtaining the statistical histograms. Extensive experiments on Replay-Attack and CASIA-FA databases showed interesting results. More importantly, unlike most of the state-of-the-art methods, our proposed method can achieve stable performances on the both databases. However, our proposed network still has some limitations. For instance, our method cannot achieve good detection results for some specific attack. Therefore, as future work, we will tackle how to improve the detection performances on a specific kind of attack and build a larger database with more intrusions. Moreover, we will explore the impact of the size of fixed convolution kernels and extract the simulated LBP features from different convolutional layers for face anti-spoofing. In addition, we will verify the effectiveness of our network structure in other computer vision tasks, such as face recognition and target recognition.

Conflict of interest

There is no conflict of interest.

Acknowledgment

This work is partly supported by the National Aerospace Science and Technology Foundation and the National Nature Science Foundation of China (No. 61702419).

References

[1] Z. Akhtar, G. Fumera, G.L. Marcialis, F. Roli, Evaluation of serial and parallel multibiometric systems under spoofing attacks, in: IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems, 2012, pp. 283–288. doi: 10.1109/BTAS.2012.6374590.

[2] D. Wen, H. Han, A.K. Jain, Face spoof detection with image distortion analysis, IEEE Trans. Inform. Forensics Security 10 (4) (2015) 746–761, http://dx.doi.org/10.1109/TIFS.2015.2400395.

[3] Y. Li, K. Xu, Q. Yan, Y. Li, R.H. Deng, Understanding osn-based facial disclosure against face authentication systems, in: ACM Symposium on Information, Computer and Communications Security, 2014, 413–424. doi: 10.1145/2590296.2590315.

[4] L. Omar, I. Ivrissimtzis, Evaluating the resilience of face recognition systems against malicious attacks, in: Seventh UK British Machine Vision Workshop, 2015, pp. 5.1–5.9. doi: 10.5244/C.29.BMVW.5.

[5] I. Chingovska, A. Anjos, S. Marcel, On the effectiveness of local binary patterns in face anti-spoofing, in: Biometrics Special Interest Group, 2012, pp. 1–7.

[6] N. Erdogmus, S. Marcel, Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect, in: IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems, 2013, pp. 1–6. doi: 10.1109/BTAS.2013.6712688.

[7] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, S.Z. Li, A face antispoofing database with diverse attacks, in: International Conference on Biometrics, 2012, pp. 26–31. doi: 10.1109/ICB.2012.6199754.

[8] N. Kose, J.L. Dugelay, Classification of captured and recaptured images to detect photograph spoofing, in: International Conference on Informatics, Electronics Vision, 2012, pp. 1027–1032. doi: 10.1109/ICIEV.2012.6317336.

[9] T.D.F. Pereira, A. Anjos, J.M.D. Martino, S. Marcel, Lbp-top based countermeasure against face spoofing attacks, in: Asian Conference on Computer Vision Workshops, 2012, pp. 121–132. doi: 10.1007/978-3-642-37410-4_11.

[10] W. Kim, S. Suh, J.J. Han, Face liveness detection from a single image via diffusion speed model, IEEE Trans. Image Processing 24 (8) (2015) 2456–2465, http://dx.doi.org/10.1109/TIP.2015.2422574.

[11] T.D.F. Pereira, A. Anjos, J.M.D. Martino, S. Marcel, Can face anti-spoofing countermeasures work in a real world scenario?, in: International Conference on Biometrics, 2013, pp. 1–8. doi: 10.1109/ICB.2013.6612981.

[12] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, Complementary countermeasures for detecting scenic face spoofing attacks, in: International Conference on Biometrics, 2013, pp. 1–7. doi: 10.1109/ICB.2013.6612968.

[13] Z. Boulkenafet, J. Komulainen, A. Hadid, Face anti-spoofing based on color texture analysis, in: IEEE International Conference on Image Processing, 2015, pp. 2636–2640. doi: 10.1109/ICIP.2015.7351280.

[14] Z. Boulkenafet, J. Komulainen, X. Feng, A. Hadid, Scale space texture analysis for face anti-spoofing, in: International Conference on Biometrics, 2016, pp. 1–6. doi: 10.1109/ICB.2016.7550078.

[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computing Research Repository abs/1409.1556. arXiv:1409.1556.

[17] J. Lu, V.E. Liong, J. Zhou, Deep hashing for scalable image search, IEEE Trans. Image Processing 26 (5) (2017) 2352–2367, http://dx.doi.org/10.1109/TIP.2017.2678163.

[18] Z. Xia, X. Peng, X. Feng, A. Hadid, Deep convolutional hashing using pairwise multi-label supervision for large-scale visual search, Signal Processing Image Commun. 59 (2017) 109–116, http://dx.doi.org/10.1016/j.image.2017.06.008.

[19] Z. Xia, X. Peng, X. Feng, A. Hadid, Scarce face recognition via two-layer collaborative representation, IET Biometrics 7 (1) (2018) 56–62, http://dx.doi.org/10.1049/iet-bmt.2017.0193.

[20] K. Patel, H. Han, A.K. Jain, Cross-database face antispoofing with robust feature representation, in: Chinese Conference on Biometric Recognition, 2016, pp. 611–619. doi: 10.1007/978-3-319-46654-5_67.

[21] J. Yang, Z. Lei, S.Z. Li, Learn convolutional neural network for face anti-spoofing, Computing Research Repository abs/1408.5601 (2014) 373–384.

[22] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987, http://dx.doi.org/10.1109/TPAMI.2002.1017623.

[23] G. Zhang, X. Huang, S.Z. Li, Y. Wang, X. Wu, Boosting local binary pattern (lbp)-based face recognition, in: Advances in Biometric Person Authentication, 2005, pp. 179–186. doi:10.1007/978-3-540-30548-421.

[24] M.A. Rahim, M.S. Azam, N. Hossain, M.R. Islam, Face recognition using local binary patterns (lbp), Global J. Comput. Sci. Technol. 13 (4) (2013) 1–8.

[25] J. Lu, V.E. Liong, J. Zhou, Learning compact binary face descriptor for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015) 2041–2056, http://dx.doi.org/10.1109/TPAMI.2015.2408359.

[26] J. Lu, V.E. Liong, J. Zhou, Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2018) 1–14, http://dx.doi.org/10.1109/TPAMI.2017.2737538.

[27] F. Juefei-Xu, V.N. Boddeti, M. Savvides, Local binary convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4284–4293. doi:10.1109/CVPR.2017.456.

[28] M. Courbariaux, Y. Bengio, J.-P. David, Binaryconnect: Training deep neural networks with binary weights during propagations, Adv. Neural Inf. Processing Syst. 28 (2015) 3123–3131.

[29] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: British Machine Vision Conference, 2015, pp. 41.1–41.12.

[30] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks, Adv. Neural Inf. Processing Syst. (2016) 4107–4115.

[31] J. Määttä, A. Hadid, M. Pietikäinen, Face spoofing detection from single images using micro-texture analysis, in: International Joint Conference on Biometrics (IJCB), 2011, pp. 1–7. doi: 10.1109/IJCB.2011.6117510.

[32] G. Pan, L. Sun, Z. Wu, S. Lao, Eyeblink-based anti-spoofing in face recognition from a generic webcamera, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[33] Y. Li, X. Tan, An anti-photo spoof method in face recognition based on the analysis of fourier spectra with sparse logistic regression, in: Chinese Conference on Pattern Recognition, 2009, pp. 1–5. doi: 10.1109/CCPR.2009.5344092.

[34] X. Tan, Y. Li, J. Liu, L. Jiang, Face liveness detection from a single image with sparse low rank bilinear discriminative model, in: European Conference on Computer Vision, 2010, pp. 504–517. doi: 10.1007/978-3-642-15567-3_37.

[35] H. Li, S. Wang, A.C. Kot, Face spoofing detection with image quality regression, in: International Conference on Image Processing Theory Tools and Applications, 2016, pp. 1–6. doi: 10.1109/IPTA.2016.7821027.

[36] I. Pavlidis, P. Symosek, The imaging issue in an automatic face/disguise detection system, in: IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications, 2000, pp. 15–24. doi: 10.1109/CVBVS.2000.855246.

[37] Z. Zhang, D. Yi, Z. Lei, S.Z. Li, Face liveness detection by learning multispectral reflectance distributions, in: IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, 2011, pp. 436–441. doi: 10.1109/FG.2011.5771438.

[38] A. Anjos, S. Marcel, Counter-measures to photo attacks in face recognition: a public database and a baseline, in: International Joint Conference on Biometrics, 2011, pp. 1–7. doi: 10.1109/IJCB.2011.6117503.

[39] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, http://dx.doi.org/10.1007/BF00994018.

[40] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. smc-3 (6) (1973) 610–621, http://dx.doi.org/10.1109/TSMC.1973.4309314.

[41] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden, Pyramid methods in image processing, Rca Eng. 29 (6) (1984) 33–41.

[42] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, A. Hadid, An original face anti-spoofing approach using partial convolutional neural network, in: International Conference on Image Processing Theory Tools and Applications, 2016, pp. 1–6. doi: 10.1109/IPTA.2016.7821013.

[43] Z. Xu, S. Li, W. Deng, Learning temporal features using lstm-cnn architecture for face anti-spoofing, in: Asian Conference on Pattern Recognition, 2015, pp. 141–145. doi: 10.1109/ACPR.2015.7486482.

[44] F.A. Gers, N.N. Schraudolph, J. Schmidhuber, Learning precise timing with lstm recurrent networks, J. Mach. Learn. Res. 3 (2002) 115–143.

[45] C.N. Karson, Spontaneous eye-blink rates and dopaminergic systems, Brain 106 (3) (1983) 643–653, http://dx.doi.org/10.1093/brain/106.3.643.

[46] Q.T. Phan, D.T. Dang-Nguyen, G. Boato, F.G.B.D. Natale, Face spoofing detection using ldp-top, in: IEEE International Conference on Image Processing, 2016, pp. 404–408. doi: 10.1109/ICIP.2016.7532388.

[47] G. Zhao, M. Pietikainen, Dynamic texture recognition using local binary patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928, http://dx.doi.org/10.1109/TPAMI.2007.1110.

[48] B. Zhang, Y. Gao, S. Zhao, J. Liu, Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor, IEEE Trans. Image Processing 19 (2) (2010) 533–544, http://dx.doi.org/10.1109/TIP.2009.2035882.

[49] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, A.T.S. Ho, Detection of face spoofing using visual dynamics, IEEE Trans. Inf. Forensics Security 10 (4) (2015) 762–777, http://dx.doi.org/10.1109/TIFS.2015.2406533.

[50] W. Bao, H. Li, N. Li, W. Jiang, A liveness detection method for face recognition based on optical flow field, in: International Conference on Image Analysis and Signal Processing, 2009, pp. 233–236. doi: 10.1109/IASP.2009.5054589.

[51] D.F. Smith, A. Wiliem, B.C. Lovell, Face recognition on consumer devices: Reflections on replay attacks, IEEE Trans. Inf. Forensics Security 10 (4) (2015) 736–745, http://dx.doi.org/10.1109/TIFS.2015.2398819.

[52] G. Pan, L. Sun, Z. Wu, Y. Wang, Monocular camera-based face liveness detection by combining eyeblink and scene context, Telecommun. Syst. 47 (3–4) (2011) 215–225, http://dx.doi.org/10.1007/s11235-010-9313-3.

[53] A. Pinto, H. Pedrini, W.R. Schwartz, A. Rocha, Face spoofing detection through visual codebooks of spectral temporal cubes, IEEE Trans. Image Processing 24 (12) (2015) 4726–4740, http://dx.doi.org/10.1109/TIP.2015.2466088.

[54] L. Feng, L.M. Po, Y. Li, X. Xu, F. Yuan, C.H. Cheung, K.W. Cheung, Integration of image quality and motion cues for face anti-spoofing, J. Visual Commun. Image Representation 38 (2016) 451–460, http://dx.doi.org/10.1016/j.jvcir.2016.03.019.

[55] S.W. Smith, The scientist and engineer's guide to digital signal processing, California Technical Publishing, 1997.

[56] S. Kim, Y. Ban, S. Lee, Face liveness detection using a light field camera, Sensors 14 (12) (2014) 22471–22499, http://dx.doi.org/10.3390/s141222471.

[57] Z. Ji, H. Zhu, Q. Wang, Lfhog: A discriminative descriptor for live face detection from light field image, in: IEEE International Conference on Image Processing, 2016, pp. 1474–1478. doi:10.1109/ICIP.2016.7532603.

[58] F.P.A. Sepas-Moghaddam, P. Correia, Light field local binary patterns description for face recognition, in: IEEE International Conference on Image Processing, 2017, pp. 3815–3819. doi:10.1109/ICIP.2017.8296996.

[59] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[60] Y. Chauvin, D.E. Rumelhart, Backpropagation: Theory, Architectures, and Applications, Psychology Press, New York, 1995.

[61] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: ACM International Conference on Multimedia, 2015, pp. 689–692. doi:10.1145/2733373.2807412.

[62] Y. Duan, J. Lu, J. Feng, J. Zhou, Context-aware local binary feature learning for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2017) 1139–1153, http://dx.doi.org/10.1109/TPAMI.2017.2710183.

[63] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition, IEEE Trans. Image Processing 16 (1) (2007) 57–68, http://dx.doi.org/10.1109/TIP.2006.884956.

[64] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026–1034. doi:10.1109/ICCV.2015.123.

[65] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Physica-Verlag HD, 2010, pp. 177–186. doi: 10.1007/978-3-7908-2604-316.

[66] A. Agarwal, R. Singh, M. Vatsa, Face anti-spoofing using haralick features, in: IEEE International Conference on Biometrics Theory, Applications and Systems, 2016, pp. 1–6. doi:10.1109/BTAS.2016.7791171.