This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895212, IEEE Transactions on Information Forensics and Security

Replayed Video Attack Detection Based on Motion Blur Analysis

Lei Li, Zhaoqiang Xia*, *Member*, *IEEE*, Abdenour Hadid, *Senior Member*, *IEEE*, Xiaoyue Jiang, Haixi Zhang, and Xiaoyi Feng*

***Abstract*—Face presentation attacks are main threats to face recognition system, and many presentation attack detection (PAD) methods have been proposed in recent few years. Although these methods have achieved significant performance in some specific intrusion modes, difficulties still exist in addressing replayed video attacks. Thats because replayed fake faces contain a variety of aliveness signals such as eye blinking and facial expression changes. Replayed video attacks occurred when attackers try to invade biometric systems by presenting face videos in front of cameras, and these videos are often launched by a liquid-crystal display (LCD) screen. Due to the smearing effects and movements of LCD, videos captured from real and replayed fake faces present different motion blurs, which mainly reflected in blur intensity variation and blur width. Based on these descriptions, a motion blur analysis based method is proposed to deal with replayed video attack problem. We first present a 1D convolutional neural network (CNN) for motion blur intensity variation description in time domain, which consists of a serial of 1D convolutional and pooling filters. Then, a local similar pattern (LSP) feature is introduced to extract blur width. Finally, features extracted from 1D CNN and LSP are fused to detect replayed video attacks. Extensive experiments on two standard face PAD databases, i.e., Relay-Attack and OULU-NPU, indicate that our proposed method based on motion blur analysis significantly outperforms the state-of-the-art methods and show excellent generalization capability.**

***Index Terms*—Replayed Video Attack, Motion Blur Analysis, 1D CNN, Local Similar Pattern**

I. Introduction

WITH the maturity of face recognition technology, it has been widely applied to many biometric systems [1], [2]. Although these biometric systems have achieved high accuracy on recognizing customer faces, face PAD is still an extensive problem [3], [4]. What is even worse, with the popularity of Internet communication and social media, criminals can easily access to people's biological information, such as face pictures and videos, and it is not difficult for them to use the information to invade personal biometric system. Therefore, considering this urgent security situation, an effective and reliable face PAD method must be developed for identifying such threats.

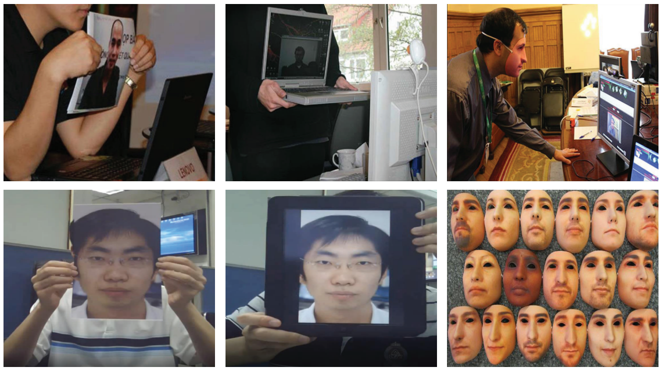


Fig. 1. Samples of face presentation attacks. From left to right: printed face photos, displayed face images or replayed videos, and 3D masks.

Based on different artefact models, four types of face presentation 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 face photos on paper and puts them in front of the camera. In both displayed image and replayed video attacks, a digital screen is used to show face images or videos. For 3D mask scenario, the attacker uses a 3D mask of authorized person to fool the system. Compared with replayed video attacks, the printed photo attacks, displayed image attacks and 3D mask attacks cannot exhibit facial aliveness signals (e.g. eye blinking, pulse and facial expression changes). Hence, detecting replayed video attacks and distinguishing them from real faces are more challenging on common cameras. For instance, Li *et al.* [5] proposed a 3D mask PAD method by computing the pulse from face videos. Even though such method can effectively detect 3D mask attacks, it will become ineffective when replayed video attacks occur. Fig. 1 shows an example of different face presentation attacks.

In the last decade, many face PAD approaches have been proposed [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. However, most of them did not specifically analyze the attributes of replayed video attacks but classified all different attacks into a same category. Especially with the popularization of high-definition screens, the overall analysis mode cannot effectively detect replayed video attacks. Therefore, Keyurkumar *et al.* [18] detected the replayed video attacks by analyzing moiré pattern, which is an optical phenomenon and appears when the screen is close to the camera. Although this method can achieve good detection performance, the generation of moiré pattern needs some objective conditions,

L. Li, Z. Xia, X. Jiang, H. Zhang, and X. Feng are with Northwestern Polytechnical University, Xi'an 710129, China. e-mail: (lilei_npu@mail.nwpu.edu.cn; zxia@nwpu.edu.cn; xjiang@nwpu.edu.cn; dennisbang@live.cn; fengxiao@nwpu.edu.cn).

A. Hadid is with Northwestern Polytechnical University, Xi'an 710129, China, and also with the University of Oulu, FI-90014, Finland. e-mail: (hadid@ee.oulu.fi).

* Corresponding authors.

Manuscript received May 25, 2018.

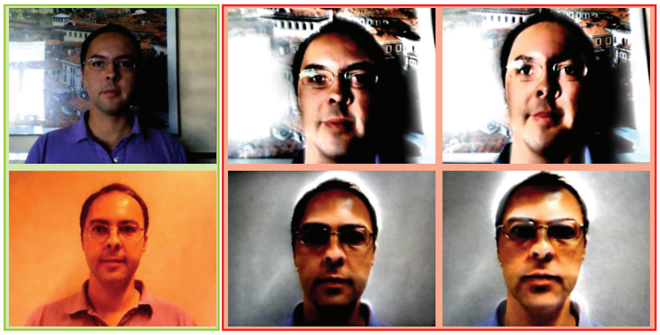


Fig. 2. The motion blur after video magnification. From left column to right column: real face videos, replayed face videos showed by iPAD and replayed face videos showed by iPhone 3GS. Compared with real faces, the motion blurs of fake faces are more obvious, especially in the edge regions.

for instance, the camera cannot be too far away from the screen. Galbally *et al.* [19] extracted different kinds of image quality assessment features to describe the quality of real and replayed fake faces. Even though this method can work well in the case of coarse videos (e.g. low-resolution screen), it will gradually become invalid as the screen resolution increases. In another work, Bharadwaj *et al.* [20] computed Histogram of Oriented Optical Flow (HOOF) [21] features for describing motion differences between the real and replayed faces. Although some satisfactory results were obtained, the generalization ability of their method is poor. Recently, Aziz *et al.* [22] built a deep learning network and extracted features from replayed video frames. In spite of the promising detection results, like [19], it can also be spoofed by a high-resolution screen invasion.

For replayed video attacks, attackers usually use an LCD screen to play face videos [9], [16]. When a motion occurs, the camera's Charge-Coupled Device (CCD) of biometric system can record the trace of the motion. In this process, caused by the smearing effects and movements of LCD, the captured real face video and replayed fake face video have different motion blurs, which are mainly reflected in blur intensity variation and blur width. Fig. 2 shows an example of different motion blurs. From the figure, we can clearly see that the motion blurs produced by replay video attacks are different from that of real faces.

Based on the differences of motion blurs between the real faces and fake faces, we propose a new method for replayed video attack detection. First, we preprocess the captured videos based on motion magnification algorithm [23]. Then, 1D CNN feature for motion blur intensity variation and LSP feature for motion blur width are extracted. Finally, feature fusion mechanisms and Support Vector Machine (SVM) [24] classifier are used to identify replayed video attacks. We train and test our proposed method on two public available databases: Replay-Attack [6] and OULU-NPU [25]. The experimental results demonstrate the effectiveness and excellent generalization capabilities of the proposed method in replayed video attack detection compared to the state-of-the-art approaches.

Among the significant contributions of this present paper, we can cite:

1) While most previous works on face PAD are based on analyzing only the visible cues (i.e. texture) of the face images, we propose a novel and appealing approach using motion blur analysis and demonstrate that the blur intensity variation and blur width can be very useful in discriminating replayed fake faces from genuine ones.
2) We exploit the intensity distribution histogram to describe the brightness of face image and design a novel 1D CNN for extracting intensity variation information from time domain. Compared with other existing deep learning networks, our 1D CNN consists a series of 1D convolutional filters and 1D pooling kernels, which can extract variation features from intensity distribution histograms.
3) We utilize LSP feature to capture the width of motion blur. Unlike traditional texture features that compare the differences between image pixels, the LSP feature encodes the pixels that have same brightness values and counts the distribution of encoded results.
4) Two different fusion mechanisms (i.e. early fusion and late fusion) are explored for 1D CNN and LSP features, and the fused features significantly improve detection performance compared to individual feature.
5) Extensive experimental analysis is conducted on the two latest and challenging face PAD databases using their pre-defined publicly well-defined experimental evaluation protocols ensuring the reproducibility of the results and a fair comparison with the state-of-the-art methods. Furthermore, in our cross-database evaluation, the proposed method shows promising generalization capabilities.

The remainder of the paper is organized as follows: Section II reviews the existing state-of-the-art methods of face presentation attack detection. Section III introduces the preliminaries needed in our method. After that, our motion analysis based detection method is described in Section IV. Section V provides the details of experimental setup and Section VI discusses the obtained results. Finally, in Section VII, we conclude the paper and discuss some directions for future research.

II. Related Work

Since the early 2000s, many methods have been proposed for addressing the problem of face PAD [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Based on different clues, we categorize these methods into four categories: (i) texture analysis [6], [8], [9], (ii) motion analysis [10], [11], [26], (iii) image quality analysis [12], [13], [14], [15], [16], and (iv) hardware based methods [7], [15], [16], [27], [28], [29].

1) Texture analysis based methods: Due to the limitations of the printers and display devices, face PAD methods based on texture analysis mainly analyze the printing failures, blurring and other effects on fake faces. In these methods, the analysis of micro-texture pattern descriptor is a mainstream. For instance, Maatta *et al.* [8] computed the texture differences between the real and fake faces in a multi-scale local binary pattern (LBP) feature space. Chingovska *et al.* [6] utilized different variations of LBP features and classifiers to detect face presentation attacks. Moreover, Akshay *et al.* [30] extracted

Haralick features [31], a kind of textural features for image classification, from video frames to detect the presented fake faces. To capture the color differences in chrominance and luminance caused by printing or displaying failures, Boulkenafet *et al.* [9], [32] proposed a method by extracting LBP features from different color spaces and analyzing those color-textures in an SVM classifier. In another work [33], Boulkenafet *et al.* extracted multi-scale textural features from a Gaussian pyramids and handled the key problem of the variation in the input image quality and resolution in face PAD. With the recent successes of deep learning in computer vision [34], [35], some face PAD works have also begun to introduce deep texture analysis into face PAD. Yang *et al.* [36] proposed an end-to-end CNN model for face PAD and fed the model with different scale face images for considering background information. In [37], [38], the hand-crafted features were extracted from convolutional feature maps to distinguish the real and fake faces rather than invoking fully-connected layers. Instead of using hand-crafted features, Li *et al.* [39] proposed a novel learnable LBP network for face spoofing detection, which can significantly reduce the network parameters. Moreover, in order to capture the temporal texture variations from a video sequence, Xu *et al.* [40] proposed a long short memory network (LSTM) and Li *et al.* [41] designed a 3D CNN to detect face presentation attack respectively. These texture analysis based methods have good detection performances when the artificial traces are obvious, such as rough face texture. However, with the development of high-definition screens (especially retina screens), their detection performance for replayed video attacks and displayed face image attacks tend to decrease drastically.

2) Motion analysis based methods: Apart from texture variations, motion is also a vitally important clue for face PAD, especially for the printed photo and displayed image attacks. Due to the involuntary eye blinking typically occurs in the interval of two to four seconds, Pan *et al.* [10] tackled face PAD task by detecting eyelid motion and realized it in an undirected conditional random field framework. In another work, Anjos *et al.* [26] computed the motion correlation coefficient between face region and background and classified the face image whose motion correlation coefficient is less than the threshold into a presentation attack. Moreover, Pereira *et al.* [42] and Phan *et al.* [43] extracted the features of LBP-TOP [44] and LDP-TOP [45] to describe the texture motion in face region and utilized a classifier to detect presentation attack. Santosh *et al.* [46] 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 PAD. Therefore, Bao *et al.* [47] addressed the problem of face PAD by analyzing the light differences in optical flow fields generated by movements of two-dimensional presentation attacks and three-dimensional real faces. Tan *et al.* [11] utilized Difference-of-Gaussian (DoG) filters to extract the differences in motion deformation patterns caused by different object dimensions. Since there are no aliveness signals in 3D mask attacks, Li *et al.* [5] proposed a 3D mask PAD method by detecting the pulse from face videos. 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 [48], [49].

3) Image quality analysis based methods: Since reproduced face images and videos usually have lower quality in comparison with the original ones, some methods took advantage of high frequency components of the data to recognize fake faces. For instance, in [12] and [13], the high frequency features were extracted by the DOG filters and analyzed for face PAD. Instead of directly using high frequency information, Li *et al.* [14] mapped this information into a more discriminative feature space to classify the real and presented fake faces. In another work, Wen *et al.* [2] connected four kinds of features to describe the specular reflection, blurriness, chromatic moment and color diversity caused by LCD screen. The low quality of image or video display devices is another important factor for face quality. So, Feng *et al.* [50] detected the fake faces by invoking both advanced image-quality feature and dense optical flow feature. Moreover, Pinto *et al.* [51] analyzed the Fourier spectrum [52] 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 or when using crude face masks, but are likely to fail for high quality displayed images or replayed videos.

4) Hardware based methods: Apart from the analysis of face images and videos, various advanced hardwares, e.g., depth, multi-spectral and light-field cameras, have also been utilized for face PAD task. For instance, Erdogmus *et al.* [7] used a Kinect camera to obtain the depth information and effectively detected face presentation attacks by analyzing the differences between the real and fake faces in the depth information. The reason is that there is no any depth information in the presentation attacks using 2D mediums. In order to capture the differences in light reflection, Pavlidis *et al.* [15] achieved it by computing the upper-band of near-infrared (NIR) spectrum and Zhang *et al.* [16] invoked two photodiodes to receive the reflectance light instead of using intrinsic image decomposition algorithms. More recently, light-field cameras allow exploiting disparity and depth information from a single capture. Therefore, Kim *et al.* [27], [28], [29] introduced these kinds of cameras into face PAD. Even though these hardware-based methods can achieve good performances for replayed video attacks, 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.

III. Background and Motivation

Before describing our method, we analyze the differences of motion blur between the real face and replayed fake face in a quantitative view. More specifically, we first describe the smearing of LCD, which induces the differences in blur-intensity variation. Then, the principle of motion blur generation is introduced and the reasons for blur-width differences are explained.

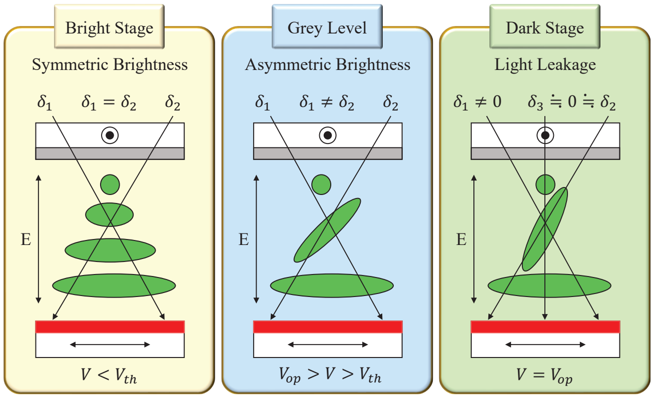

Bright Stage
Symmetric Brightness
δ_1 $\delta_1 = \delta_2$ δ_2
E
$V < V_{th}$
Grey Level
Asymmetric Brightness
δ_1 $\delta_1 \neq \delta_2$ δ_2
E
$V_{op} > V > V_{th}$
Dark Stage
Light Leakage
$\delta_1 \neq 0$ $\delta_3 \doteqdot 0 \doteqdot \delta_2$
E
$V = V_{op}$

Fig. 3. Schematic drawing of the liquid crystal director configuration with increasing voltage [54]. The amount of transmitted light is controlled by different liquid crystal states. From left to right, the figure shows the relationship between the three different brightness of the liquid crystal (i.e. bright, grey and dark) and the TN cell voltage.

A. Screen Smearing

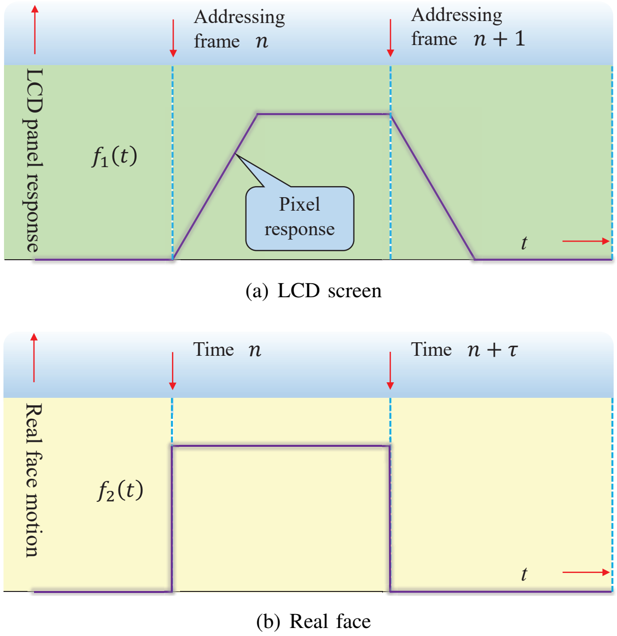
A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. In LCD, the twisted nematic (TN) mode is the basic existing mode of liquid crystals. The liquid crystal does not emit lights directly. Instead, it uses a backlight or reflector to produce images in color or monochrome [53]. In this process, the states of liquid crystals are changed from top to bottom substrate or from bottom to top substrate under crossed polarizers and vertical field. Fig. 3 shows an example of liquid-crystal state change as the increasing of voltage in a TN cell. When the used voltage (V) below the threshold (V_{th}) and the phase retardation between the left (δ_1) and right (δ_2) is almost same, the change brightness rate of the liquid crystal is about same along azimuthal direction, resulting in symmetry in brightness. However, when $V > V_{th}$ and $\delta_1 \neq \delta_2$, the liquid crystal tilts up in one direction that different along azimuthal direction. In this case, the bright state of liquid crystal is changed into the grey level. On the other hand, the liquid crystal changes into dark state when a driving voltage (V_{op}) for dark state is applied.

However, in the process of state change, the LCD screen needs a response time to change the state of liquid crystal as shown in Fig. 4(a) [55]. Compared to LCD, the real facial movement is rapid and without delaying, shown in Fig. 4(b). Caused by the response time, the current-time video frame often has similar intensity values with the previous one, leading to the phenomenon of smearing.

B. Motion Blur

Caused by physical and technical limitations, the videos captured by digital cameras are not perfect and have various types of degradations. The motion blur is one of the frequent degradations, which can be illustrated by Eq. 1 [56]

$$z(x,y) = \int I(x-s, y-t)h(x-s, y-t; s,t)dsdt \quad (1)$$



(a) LCD screen

(b) Real face

Fig. 4. Different temporal responses of LCD panel [55] and real face motion. When the brightness of the pixel changes, LCD requires a response time to complete the state change of the liquid crystal, which is different from the real face.

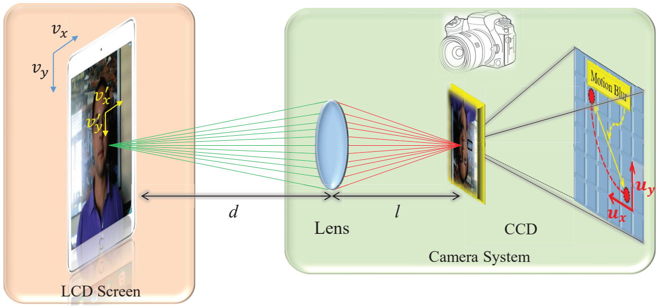


Fig. 5. The motion blur generated by a camera. When the camera captures a moving target, motion blur is generated on the CCD sensor and affected by the speed of the target motion.

where I is the original image. For a biometric system, I is the replayed fake face or real face. h is *point-spread* function, and z is the blurred image. More specifically, the intensity of blurred image is controlled by original fake face video or real face and the degree of blur depends on the *point-spread* function. In addition, both of them have a positive correlation with the exposure time.

1) Intensity Difference: Since LCD requires a response time to complete the state change of the liquid crystal, the intensity variations of replayed face videos are different from the real ones. As shown in Fig. 4, the change in the brightness of a replayed fake face is smoother than the real one. Consequently, the intensity variations of replayed fake face and real face are different. More specifically, in the exposure time, the CCD sensor of camera in the biometric system accumulates the intensity of $f_1(t)$ and $f_2(t)$, respectively. As a result, the motion blur generated by replayed fake face and real face have different intensity variations.

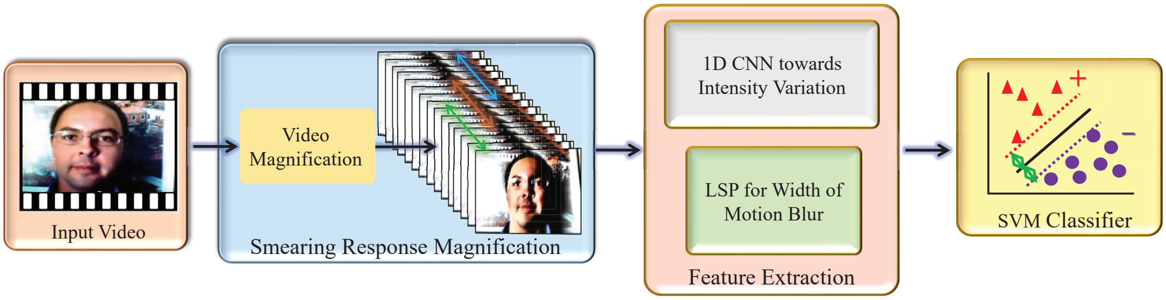

Input Video
Video Magnification
Smearing Response Magnification
1D CNN towards Intensity Variation
LSP for Width of Motion Blur
Feature Extraction
+
−
SVM Classifier

Fig. 6. Flow chart of our proposed motion blur analysis based replayed video attacks detection method. First, the input face video is magnified for magnifying motion blur in intensity and width. Then, the feature extraction module extracts two kinds of features from the magnified video. Finally, an SVM classifier is invoked to classify whether the extracted features are extracted from a valid access or a face presentation attack.

2) Blur Difference: As aforementioned, attackers often use LCD panels (e.g. mobile phones and tablets) to replay face videos. Assuming the camera is stationary, there are two types of motions during fake face attacks: the motion of the screen (denoted as v) and the face motion in the replayed video (denoted as v'). However, for the real face, it includes only the second kind of motion. When the camera captures the motion in replayed video or real face, the motion blur will be generated based on the *point-spread* function h and original image I. Meanwhile, the *point-spread* function depends on the speed of motion. Fig. 5 shows an example of motion blur generated from a replayed video attack. During the camera's exposure, the CCD continuously collects outside light information, which will record the trajectory of movement. The width of the trajectory is illustrated in Eq. 2

$$w = (u_x + u_y) \times \tau \tag{2}$$

where τ is the expose time of camera. u_x and u_y are the motion speed of a point in horizontal and vertical directions, respectively.

For the motion speed of u_x and u_y, they depend on the object's speed and the distance between the object and camera [57]. Taking a replayed video attack as an example, shown in Fig. 5, u_x and u_y can be calculated as Eq. 3

$$\begin{aligned} u_x &= -\frac{l}{d} \times (v_x + v_x^{'}) \\ u_y &= -\frac{l}{d} \times (v_y + v_y^{'}) \end{aligned} \tag{3}$$

where d and l are the distances between the lens and object and the distance between the lens and CCD sensor, respectively. v is the motion speed of LCD medium and $v^{'}$ is the motion speed of replayed face. The subscripts x and y denote the horizontal and vertical directions, respectively. Therefore, the width of motion blur caused by replayed video attacks can be calculated as Eq. 4

$$w_{fake} = ||(-\frac{l}{d} \times ((v_x + v_x^{'}) + (v_y + v_y^{'}))) \times \tau||_1 \tag{4}$$

where $||\cdot||_1$ is 1-norm operation. However, the real face has only facial motion compared to replayed video attacks. Its blur width is calculated as follows

$$w_{real} = ||(-\frac{l}{d} \times (v_x^{'} + v_y^{'})) \times \tau||_1 \tag{5}$$

Comparing Eq. 4 with Eq. 5, we can clearly conclude that the widths of motion blurs caused by real-face and replayed-video attacks are different. Fig. 2 shows some examples of motion blurs. In the figure, the motion blur widths of replayed faces are different from the real ones.

IV. Proposed Method

In this part, we present the pipeline of our proposed detection method for replayed video attacks. The proposed method consists of three modules, i.e., video magnification, feature extraction and SVM classification. The overall process pipeline is shown in Fig. 6. First, a captured face video is processed by the video magnification algorithm, which can magnify motion blur in intensity and width. After that, the blur-intensity variation features and blur-width features are extracted from a 1D CNN network and LSP mode respectively. Finally, the feature fusion mechanisms and SVM classifier are used to distinguish whether the processed video is a valid access or a face presentation attack.

A. Video Magnification

Before feature extraction, we first perform motion amplification on the captured videos. In our proposed method, we use Eulerian video magnification (EVM) algorithm [58] to preprocess the captured face videos. This procedure is able to amplify temporal changes of brightness and small motions [23]. Given a video signal V, the frame changes can be expressed as $V(x,y,t) = f(x+\delta_x(t), y+\delta_y(t))$, where $V(x,y,0) = f(x,y)$ and $\delta_x(t)$, $\delta_y(t)$ are variation amplitudes in x and y directions respectively. The goal of video magnification is to synthesize the signal

$$\hat{V}(x,y,t) = f(x + (1+\alpha)\delta_x(t), y + (1+\alpha)\delta_y(t)) \tag{6}$$

where α is the amplification factor. Based on the first-order Taylor-series expansion towards x and y, $V(x,y,t)$ can be expanded as

$$V(x,y,t) = f(x,y) + \delta_x(t)\frac{\partial f}{\partial x} + \delta_y(t)\frac{\partial f}{\partial y} \tag{7}$$

To get the partial derivatives of f about x and y, a broadband temporal bandpass filter $B(x,y,t)$ is applied on $V(x,y,t)$ at every position (x,y). The filter is expressed as

$$B(x,y,t) = \delta_x(t)\frac{\partial f}{\partial x} + \delta_y(t)\frac{\partial f}{\partial y} \tag{8}$$

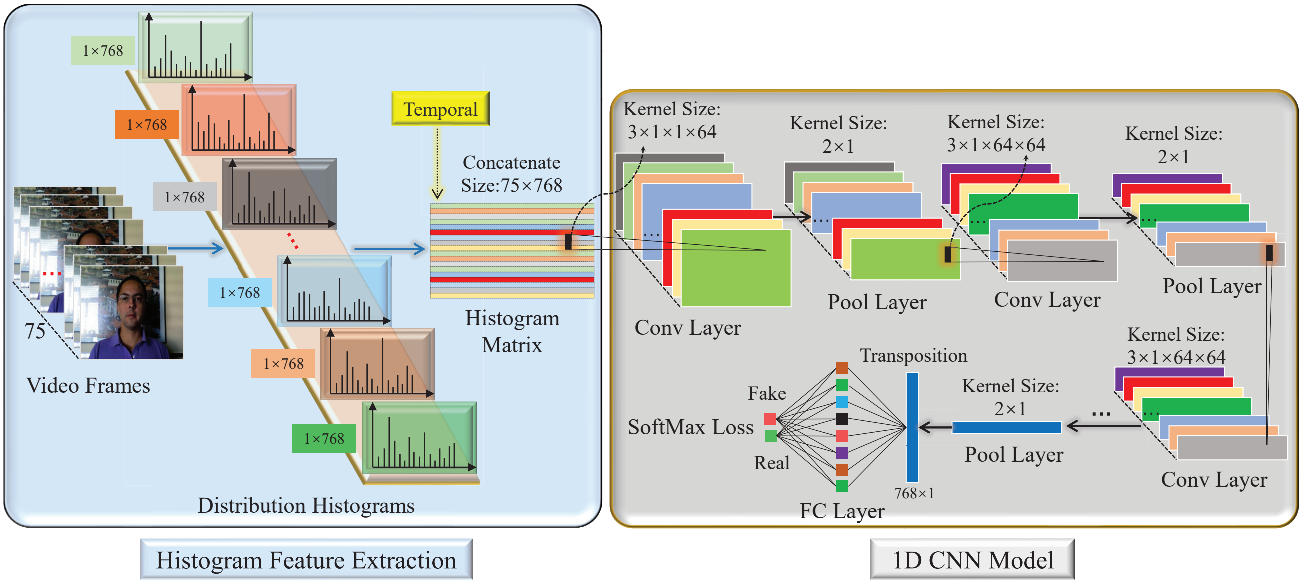


Fig. 7. The architecture of CNN for intensity variation extraction. It includes two modules: histogram feature extraction and 1D CNN model. The former is used to extract the intensity histograms of video sequence, and the latter is used to extract brightness variation feature. Conv layer is the abbreviation of convolutional layer and FC layer is the abbreviation of fully connected layer.

Then Eq. 7 can be represented as

$$\hat{V}(x,y,t)=f(x,y)+\alpha B(x,y,t) \tag{9}$$

Combining Eqs. 7, 8 and 9, the magnified video frame will be obtained based on 10.

$$\hat{V}(x,y,t)=f(x,y)+(1+\alpha)[\delta_x(t)\frac{\partial f}{\partial x}+\delta_y(t)\frac{\partial f}{\partial y}] \tag{10}$$

B. 1D CNN for Intensity Variation

Towards original magnified videos, we extract intensity histograms to describe the brightness information. More specifically, given a video frame $f(x,y)$, the distribution histograms are calculated from R, G and B color channels denoted as h_R, h_G and h_B, respectively. After that, h_R, h_G and h_B are concatenated into one histogram h_f with the dimension 1×768.

In order to extract temporal information from brightness changes in intensity histograms, we propose a novel convolutional neural network. The main architecture of our proposed network is shown in Fig. 7. It includes two modules. First, we intercept m successive video frames, extract their histograms and concatenate the histograms into a histogram matrix, which can be written as Eq. 11. Then, the histogram matrix with size $m\times768$ will be fed into our proposed network. Following the face PAD work [32] that capturing texture variations by computing the feature descriptions within a time window of three seconds, in this paper, we set $m=75$.

$$H=\begin{bmatrix} h_{f_1} \\ h_{f_2} \\ ... \\ h_{f_m} \end{bmatrix} \tag{11}$$

In our network, we use 1D operation to extract intensity variation information in H. Compared with other deep networks [36], [59], our solution differs mainly in three aspects: (i) 1D convolutional filters are adopted to extract the temporal information of neighboring histograms; (ii) 1D pooling layers are introduced and can effectively reduce the temporal dimension after convolutional layers; (iii) The receptive field size of the output features is 75 corresponding to the number of video frames and the output features do not fuse the information between different brightness values.

The parameters of our proposed network are summarized in Table I. For all convolutional layers except the 16_{th} layer, the size of convolutional filters is set to 3×1. This means the convolutional filter can extract the brightness variation in three adjacent histograms and save network parameters as [60]. After each convolutional layer, we use Rectified Linear Units (ReLU) to activate the convolutional outputs [61]. Then the pooling layers with size 2×1 are utilized to gradually downsample the results from ReLU layer. In the last pooling layesr, the dimension of outputs is 1×768. Before feeding the outputs of last pooling layer into a fully connected (FC) layer, we introduce a transposition layer to reshape the size of outputs into 768×1.

For face PAD, its essence is to classify whether the input is a real face or a presented fake face. Therefore, after FC layer, the most commonly used $SoftMax$ loss function is employed to measure the classification error [62], [63]. 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 back propagation (BP) [64], as shown in Eq. 12

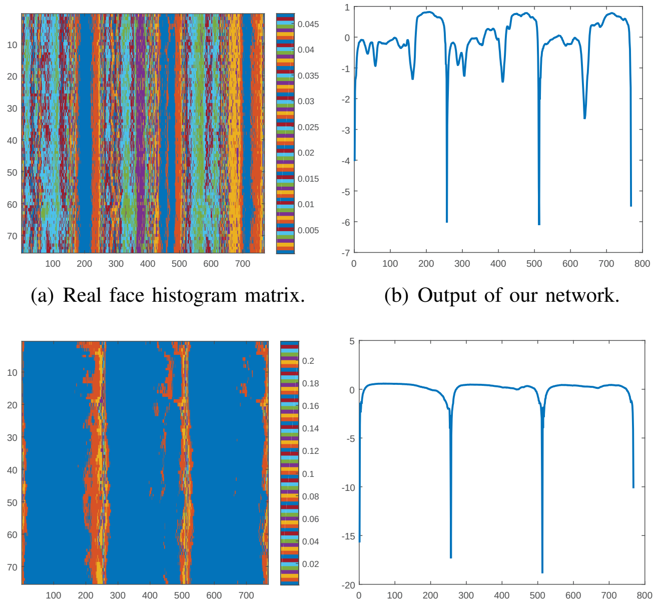
$$\mathcal{F}(Y)=\sum_{i=1}^{n}\{log(e^{y_{i1}}+e^{y_{i2}}+...+e^{y_{iv}})+y_{ir}\} \tag{12}$$

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}

TABLE I
THE CONFIGURATION PARAMETERS IN OUR PROPOSED 1D CNN MODEL. *aPool* IS THE OPERATION OF AVERAGE POOLING AND *Trans* IS THE OPERATION OF MATRIX TRANSPOSE.

layer type	1 Conv	2 ReLU	3 aPool	4 Conv	5 ReLU	6 aPool	7 Conv	8 ReLU	9 aPool	10 Conv	11 ReLU
filt size	[3, 1]	–	–	[3, 1]	–	–	[3, 1]	–	–	[3, 1]	–
filt dim	1	–	–	64	–	–	64	–	–	64	–
num filts	64	–	–	64	–	–	64	–	–	64	–
stride	1	1	[2, 1]	1	1	[2, 1]	1	1	[2, 1]	1	1
pad	[1, 0]	0	0	[1, 0]	0	0	[1, 0]	0	0	[1, 0]	0

layer type	12 aPool	13 Conv	14 ReLU	15 aPool	16 Conv	17 ReLU	18 Conv	19 Trans	20 FC	21 SoftMax	22 –
filt size	–	[3, 1]	–	–	[2, 1]	–	[1, 1]	–	[1, 1]	–	–
filt dim	–	64	–	–	64	–	64	–	64	–	–
num filts	–	64	–	–	64	–	64	–	2	–	–
stride	[2, 1]	1	1	[2, 1]	1	1	1	1	1	1	–
pad	0	[1, 0]	0	0	0	0	0	0	0	0	–



(a) Real face histogram matrix.

(b) Output of our network.

(c) Fake face histogram matrix.

(d) Output of our network.

Fig. 8. The input and output of our proposed network. The first row is the result of a real face video and the second row is the result of a fake face video. To make the histogram matrix better visualized, we use lines colormap in (a) and (c). (b) and (d) plot the envelope of the output of our 1D CNN network.

training sample, y_{ir} is the predict value of the i_{th} class, and v is the number of classes. In testing stage, we extract the outputs of the last pooling layer as the finally features of H denoted as C_f.

Fig. 8 shows the intensity histogram matrices and the learned brightness variation features of real and replayed fake face videos. From the distribution changes in Figs. 8(a) and 8(c), we can find that the variation in the brightness of real face video is more obvious than the replayed one. This also verifies that our intensity variation hypothesis holds. Moreover, the envelope lines of real and replayed fake faces also have differences in amplitudes, which are shown in Figs. 8(b) and 8(d).

C. LSP for Motion Blur Width

Based on the differences of motion blur between the real face and replayed fake face, a rotation invariant texture descriptor is used to describe them. Motion blur often occurs at the edge of a moving object and usually has same or closeness brightness values. Local binary pattern (LBP) [65] is the most commonly used feature in texture analysis. The main idea of LBP is to determine whether the value of the image pixel is smaller than its neighboring pixels and encode the comparison results. Inspired by this, we utilize the feature called local similar pattern (LSP) [66] to describe the width of motion blur. Instead of comparing the value of the image pixel as in LBP, we compare whether it is equal to the neighboring pixels. The LSP pattern of a pixel extracted from the video frame $f(x, y)$ at band c can be represented as follows:

$$LSP^{(c)}_{P,R}(x, y) = \sum_{p=0}^{P-1} \delta(r^{(i)}_p - r^{(i)}_c) \times 2^p \tag{13}$$

where $\delta(x) = 1$ if $x = 0$, otherwise 0. r_c and $r_p(p = 0, ..., P-1)$ denote the intensity values of central pixel (x, y) and its P neighborhood pixels located at the circle of radius $R(R > 0)$. Fig.9 shows an example of LSP feature extraction with $P = 8$.

After LSP encoding, we compute the statistical histograms from the encoded maps. At end, the LSP feature of $f(x, y)$ can be defined by

$$L_f = [L^{(1)}_f ... L^{(M)}_f] \tag{14}$$

where M is the channel number.

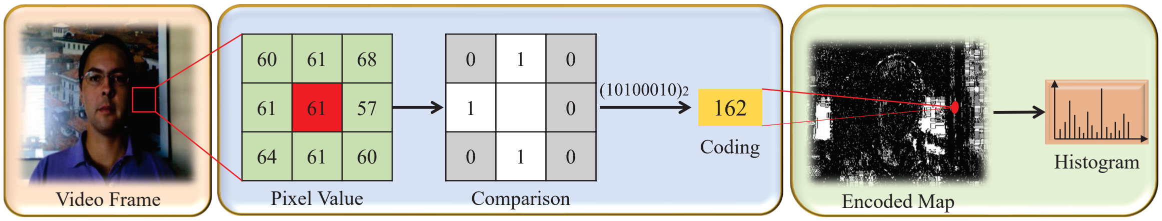
In Fig. 10, we visualize some encoded LSP maps of real and replayed face images. From these maps, we can clearly find that regions with the same brightness have the same encoding value. Therefore, our proposed LSP feature can be used to describe the width of motion blur.

D. Implementation Details

For the filters in convolutional layers of 1D CNN network, we initialize them based on [67] as illustrated in Eq. 15.

$$W = \frac{rand(n_C)}{\sqrt{2/n_C}} \tag{15}$$

where $rand(\cdot)$ samples from a zero mean, unit standard derivation gaussian function, and n_C is the channel number of



60	61	68
61	61	57
64	61	60

0	1	0
1		0
0	1	0

Fig. 9. Flow chart of LSP feature extraction. The LSP compares whether the center pixel is equal to the neighborhood pixels, rather than comparing whether it is smaller than the neighborhood pixels.

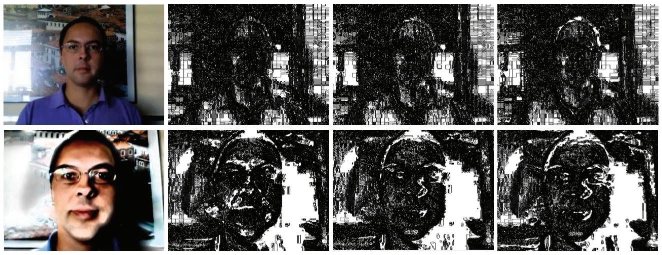


Fig. 10. Exemplar Visualization of encoded maps of LSP. The first row shows the results of a magnified real face video while the second row corresponds to the magnified replayed face video. From left to right: magnified video frame, R color channel maps, G color channel maps and B color channel maps.

Fig. 11. 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 left column to right column: real faces and the corresponding printed photo, displayed image and replayed video attacks.

the inputs in convolutional layer. This ensures that all neurons in the network initially have the approximately same output distribution and empirically improves the rate of convergence. In training stage, the stochastic gradient descent (SGD) algorithm [68] 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} and all mini-batches are traversed and reallocated randomly.

Before feature extraction, we magnify the face videos in different frequencies with an interval of $20Hz$ and ranging from $0Hz$ to $240Hz$. For motion blur width descriptor, we extract LSP feature from RGB color space. After 1D CNN and LSP features extraction, we fuse C_f and L_f and use an SVM classifier to predict whether the input face is a real client. In our paper, we realize the 1D CNN and SVM based on the toolboxes of MatConvNet with the version 1.0-beta20 [1] and liblinear with the version 1.96 [2] [69], respectively. The code for this work can be downloaded at https://github.com/lileiNPU/MotionBlurAnalysis.

V. Experimental Setup

A. Experimental Data

We validate our proposed method on two publicly available face PAD databases: Replay-Attack [6] and OULU-NPU [25]. Table II summarizes these two databases, and detailed descriptions are given below.

1) Replay-Attack: The IDIAP Replay-Attack database [3] [6] consists of 1300 video clips of real and attack and 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 real clients 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. 11 shows some examples of real and fake faces.

2) OULU-NPU: The OULU-NPU Database [4] [25] consists of 4950 real access and attack videos and attempts 55 clients. Similar to Replay-Attack database, all clients are divided into 3 subject-disjoint subsets for training, development and testing (20, 15 and 20, respectively). These videos were recorded using the front cameras of six mobile devices in three sessions with different illumination conditions and background scenes. Two types of fake faces are created: printed photo and replayed video attacks. The attacks were created using two printers and two display devices. For the replayed video attacks, the original face videos were recorded by 6 different cell phones. Fig. 12 shows some examples of real and fake faces.

B. Evaluation Protocol

For performance evaluation, the results are reported in term of recently standardized ISO/IEC 30107-3 metrics [70]: Attack Presentation Classification Error Rate (APCER) and Bona Fide Presentation Classification Error Rate (BPCER). In principle, these two metrics correspond to the False Acceptance Rate (FAR) and False Rejection Rate (FRR) commonly used in

[1] http://www.vlfeat.org/matconvnet/

[2] https://www.csie.ntu.edu.tw/∼cjlin/liblinear/

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

[4] https://sites.google.com/site/oulunpudatabase/welcome

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895212, IEEE Transactions on Information Forensics and Security

TABLE II
A SUMMARY OF TWO PUBLIC-DOMAIN DATABASES.

Database	Released year	Lighting scenarios	Subjects	Attack type	Subject gender	Subject age
Replay-Attack	2012	2	50	Printed photos Displayed images Replayed videos	Male 86% Female 14%	20 to 40 years
OULU-NPU	2017	3	55	Printed photos Replayed videos	Male 69% Female 31%	20 to 60 years

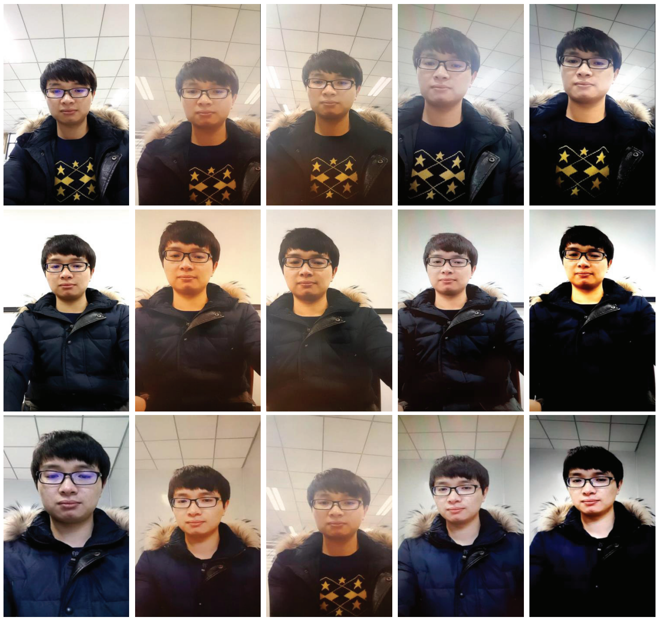


Fig. 12. Samples from OULU-NPU database. The first row presents images taken from the first scenario, the second row corresponds to the images from the second scenario, and the last row corresponds to the images from the third scenario. From left column to right column: real faces and the corresponding printed photo, printed photo, replayed video and replayed video attacks.

the PAD related literature. However, different with the FAR and FRR, the attacker's potential (such as expertise, resources and motivation) in the worst case scenario are taken into considered by APCER and BPCER. It is noted that the APCER and BPCER depend on the decision threshold. Therefore, the development set is used to fine tune the system parameters and estimate the threshold value. To evaluate the overall system performance in a single value, the BPCER20, which calculates the BPCER when APCER is 5%, is employed in the following experiments.

C. Network Selection

In the training stage of 1D CNN, we empirically iterate 100 epochs on all training data. Since Replay-Attack and OULU-NPU also provide a development set, we select the network with the lowest cost of the development set in the iteration of 100 epochs as the model in testing stage, instead of selecting the network corresponding to the lowest cost of the training set. This means that the number of iterations to get the optimal network model at different frequencies may be different.

VI. Experimental Results and Discussion

Due to the factors such as the liquid-crystal refresh frequency, the motion of the LCD screen and the external environment (e.g. light), it is difficult to determine the motion frequency at a fixed point. Therefore, in this section, we present and discuss the detection results of different motion blur features that obtained in the magnified frequencies from $0Hz$ to $240Hz$. More specifically, we begin our experiments by analyzing the performances of our proposed 1D CNN feature and designed LSP feature. Then, we discuss how to fuse the 1D CNN and LSP features. Finally, the performance of our method is compared against the state-of-the-art algorithms and the generalization capabilities of the proposed approach are evaluated by conducting cross-database experiments.

A. Effectiveness of 1D CNN Feature

In this part, we present the performances of our 1D CNN feature and compare them with original unlearned intensity histogram feature on different magnified frequencies. For the unlearned intensity histogram, we first extract the intensity histograms from successive 75 frames of the video sequence and then average them to get the final feature vector. The results are shown in Table III. It can be clearly seen that our proposed 1D CNN can effectively extract intensity variation information from original intensity histogram matrix. More specifically, in the magnified frequency from $180Hz$ to $200Hz$, the APCER, BPCER and BPCER20 of Replayed-Attack database are 0.3%, 0.3% and 0.0%, respectively. For OULU-NPU database, the APCER is 18.7%, BPCER is 14.2%, and BPCER20 is 43.9%. Although these indicators of OULU-NPU are not as good as that obtained in Replay-Attack, the BPCER is still twice as low as the original unlearned histograms. Moreover, we can also find that the performance in the magnified frequency from $180Hz$ to $200Hz$ is significantly better than other frequencies. We conjecture that the reason may lie in that the network models selected for evaluation are originated from different training epochs. In the table, we also compare the detection results of the original videos with the results of the videos that been processed by EVM. By comparing the detection results, we found that the EVM-processed face video can improve the detection performance, while the detection performance is degraded in some frequency ranges.

B. Effectiveness of LSP Feature

Table IV provides the detection performance of our designed LSP feature. From these results, we can find the best APCER, BPCER and BPCER20 of Replay-Attack database are obtained

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895212, IEEE Transactions on Information Forensics and Security

TABLE III
COMPARE INTENSITY VARIATION FEATURES OF OUR PROPOSED 1D CNN WITH ORIGINAL UNLEARNED INTENSITY HISTOGRAMS.

Frequency(Hz)	Method	Replay-Attack			OULU-NPU		
		APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	Unlearned Hist	5.1	4.3	4.4	31.8	28.3	77.5
	1D CNN	1.6	2.1	0.1	21.5	26.4	51.0
20-40	Unlearned Hist	3.4	5.8	4.9	28.3	30.2	69.6
	1D CNN	2.3	2.4	0.8	19.0	22.6	45.0
40-60	Unlearned Hist	4.2	6.7	6.4	26.5	35.3	68.6
	1D CNN	3.0	4.6	4.3	21.3	29.8	55.6
60-80	Unlearned Hist	4.0	7.1	6.8	25.7	39.2	69.3
	1D CNN	4.9	4.7	4.7	25.2	31.8	62.4
80-100	Unlearned Hist	4.0	7.2	6.7	26.2	40.8	69.8
	1D CNN	3.2	5.3	5.1	23.0	34.5	59.9
100-120	Unlearned Hist	4.8	11.0	10.9	26.3	40.9	71.0
	1D CNN	4.4	6.5	5.8	24.8	29.9	56.9
120-140	Unlearned Hist	4.6	11.5	11.2	26.4	40.1	69.4
	1D CNN	3.9	5.1	5.0	20.7	17.3	52.8
140-160	Unlearned Hist	4.1	11.0	10.5	27.5	40.3	69.1
	1D CNN	2.5	5.1	5.0	22.8	25.8	54.6
160-180	Unlearned Hist	3.3	10.9	10.4	27.7	39.9	69.2
	1D CNN	2.7	5.1	5.0	24.3	33.9	60.0
180-200	Unlearned Hist	3.4	11.0	10.5	26.3	40.2	68.0
	1D CNN	**0.3**	**0.3**	**0.0**	**18.7**	**14.2**	**43.9**
200-220	Unlearned Hist	2.8	11.7	10.9	26.5	40.3	67.3
	1D CNN	0.6	3.9	3.4	26.5	37.4	64.9
220-240	Unlearned Hist	2.3	10.7	9.9	26.8	39.7	67.4
	1D CNN	3.3	6.3	6.3	31.0	37.7	66.3
Original Video	Unlearned Hist	2.6	9.0	8.9	27.3	38.6	63.5
	1D CNN	1.7	3.8	3.8	24.6	33.1	58.3

TABLE IV
THE DETECTION RESULTS OF OUR DESIGNED LSP FEATURE.

Frequency(Hz)	Replay-Attack			OULU-NPU		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	2.5	5.5	2.7	24.0	30.4	63.3
20-40	3.0	3.4	2.7	14.7	13.1	33.8
40-60	4.6	1.1	1.1	9.8	7.2	16.6
60-80	4.6	0.2	0.2	9.2	7.0	15.6
80-100	2.6	**0.0**	**0.0**	6.9	4.1	6.1
100-120	**1.7**	**0.0**	**0.0**	6.5	3.7	6.3
120-140	1.9	**0.0**	**0.0**	5.6	**3.6**	3.9
140-160	2.1	**0.0**	**0.0**	6.6	4.1	4.8
160-180	1.8	**0.0**	**0.0**	4.9	4.6	4.6
180-200	2.1	**0.0**	**0.0**	3.6	4.6	**3.3**
200-220	2.0	**0.0**	**0.0**	3.9	4.5	3.5
220-240	2.4	**0.0**	**0.0**	**3.4**	5.2	3.9
Original Video	4.0	2.5	2.5	8.1	6.7	12.5

TABLE V
THE DETECTION RESULTS OF EARLY FUSION OF 1D CNN AND LSP FEATURES.

Frequency(Hz)	Replay-Attack			OULU-NPU		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	**0.0**	3.2	**0.0**	12.1	16.1	25.2
20-40	0.2	0.9	0.3	9.0	5.7	9.7
40-60	0.3	1.0	0.2	10.0	2.9	7.7
60-80	**0.0**	5.0	0.1	8.8	1.6	4.9
80-100	**0.0**	5.0	**0.0**	9.6	1.7	6.1
100-120	**0.0**	3.2	**0.0**	10.8	1.4	4.1
120-140	**0.0**	1.7	**0.0**	10.0	1.6	3.6
140-160	**0.0**	1.6	**0.0**	10.0	1.3	2.4
160-180	0.1	1.1	**0.0**	9.3	0.9	1.5
180-200	0.1	1.0	**0.0**	9.0	0.7	1.3
200-220	0.1	**0.8**	**0.0**	8.7	0.8	1.1
220-240	0.1	**0.8**	**0.0**	8.1	**0.6**	**1.0**
Original Video	1.0	1.0	**0.0**	**7.4**	2.2	3.8

TABLE VI
THE DETECTION RESULTS OF LATE FUSION OF 1D CNN AND LSP FEATURES.

Frequency(Hz)	Replay-Attack			OULU-NPU		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	0.4	0.5	0.1	12.2	13.8	33.4
20-40	0.2	0.4	0.2	11.4	10.2	24.8
40-60	0.1	0.2	**0.0**	8.0	7.4	13.8
60-80	0.2	0.1	**0.0**	9.5	6.9	14.5
80-100	0.2	0.1	**0.0**	8.1	5.4	9.1
100-120	**0.0**	**0.0**	**0.0**	7.2	5.0	7.8
120-140	0.1	**0.0**	**0.0**	6.0	5.7	7.1
140-160	**0.0**	**0.0**	**0.0**	6.0	5.3	6.1
160-180	**0.0**	**0.0**	**0.0**	5.9	5.2	6.7
180-200	**0.0**	**0.0**	**0.0**	5.8	6.1	7.1
200-220	**0.0**	**0.0**	**0.0**	**5.3**	**4.7**	**5.3**
220-240	**0.0**	**0.0**	**0.0**	5.7	4.8	5.8
Original Video	0.4	2.5	2.5	12.0	12.0	32.8

TABLE VII
COMPARISON BETWEEN OUR PROPOSED COUNTERMEASURE AND STATE-OF-THE-ART METHODS ON THE TWO BENCHMARK DATABASES USING THE BASED EVALUATION.

Methods	Replay-Attack			OULU-NPU		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
Moiré Analysis[18]†	15.6	12.9	29.9	20.0	12.3	56.8
Image Quality[19]‡	13.9	13.6	25.8	42.6	24.2	77.8
HOOF[20]‡	7.3	6.8	10.4	38.8	45.7	88.0
RGB LBP[9]‡	5.3	3.8	4.1	16.1	16.7	40.8
YCbCr LBP[9]‡	2.4	6.8	5.4	19.5	13.5	36.9
HSV LBP[9]‡	6.4	6.8	7.8	20.6	13.5	50.1
Proposed Early Fusion	0.1	0.8	**0.0**	8.1	**0.6**	**1.0**
Proposed Late Fusion	**0.0**	**0.0**	**0.0**	**5.3**	4.7	5.3

† we tested the method on original replayed video set of Replay-Attack instead of recapturing the video data as [18].
‡ we retested the method on the replayed video set of Replay-Attack and OULU-NPU databases respectively.

in the frequency range from $100Hz$ to $120Hz$ with APCER = 1.7%, BPCER = 0.0% and BPCER20 = 0.0%. For OULU-NPU database, the best detection results are obtained in the frequency range from $180Hz$ to $200Hz$ with BPCER20 = 3.3%. Similar to Replay-Attack, the best APCER and BPCER are also obtained in different magnified frequencies. Comparing Table III with Table IV, we can see that LSP feature performs better than 1D CNN feature. We can also conclude that the OULU-NPU database is more difficult than Replay-Attack database due to more different replayed videos and more complex external environment.

C. Fusion of 1D CNN and LSP Features

As can be seen in previous experiments, motion blur intensity and motion blur width seem to be effectively for detecting replayed video attacks. In order to benefit from both blur intensity variation and blur width, we explore how to fuse 1D CNN and LSP features. In this part, we fuse the 1D CNN feature extracted from $180Hz$ to $200Hz$ with the LSP feature extracted from $0Hz$ to $240Hz$ in two different mechanisms: early fusion and late fusion. For early fusion, the features are concatenated into one feature vector and fed into an SVM classifier. For late fusion, 1D CNN feature and LSP feature are first fed into two different SVM classifiers. Then, the scores from the SVMs are added together to get the final classification score. Tables V and VI show the results of early fusion and late fusion respectively. From the two tables, we can see that early fusion is more suitable to OULU-NPU database with APCER = 8.1%, BPCER = 0.6% and BPCER = 1.0% and late fusion is more suitable to Replay-Attack database with APCER = 0.0%, BPCER = 0.0% and BPCER20 = 0.0%. In addition, by comparing Tables III, IV and V, we can conclude that the 1D CNN and LSP features complement perfectly each other. Finally, even though the average of APCER and BPCER of fusion results for Replay-Attack are not better than 1D CNN system and the APCER of fusion results for OULU-NPU are not better than LSP system, the fusion is effective considering both databases together.

D. Comparison with the State of the Art

In this part, we implement the Image Quality [19], HOOF [20] and Moiré Analysis [18] based methods and compare our APCER, BPCER and BPCER20 with them. The comparison results are shown in Table VII. It can be seen that our proposed motion blur analysis based method outperforms the-state-of-the-art algorithms on Replay-Attack database. Especially for the evaluation indicators, the APCER, BPCER and BPCER20 of our method are 0.0%. On OULU-NPU database, our proposed method also significantly surpasses these baselines. However, some samples of OULU-NPU database are still incorrectly classified compared to Replay-Attack database. Figure 13 visualizes some video sequences in the frequency range from $220Hz$ to $240Hz$ that are classified incorrectly. From these failed cases we can see that the background

TABLE VIII
THE EARLY FUSION PERFORMANCE OF CROSS-DATABASE EVALUATION IN TERMS OF APCER(%), BPCER(%) AND BPCER20(%) ON REPLAY-ATTACK AND OULU-NPU.

Frequency(Hz)	Train on: Replay-Attack Evaluate on: OULU-NPU								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	12.9	53.4	73.3	17.5	56.4	78.0	21.6	49.9	84.4
20-40	23.1	35.9	71.6	27.3	33.0	68.0	30.4	32.3	77.6
40-60	27.5	32.7	71.4	30.9	29.0	68.5	33.7	27.3	74.8
60-80	18.6	45.0	76.6	20.7	47.1	73.5	24.3	41.1	76.2
80-100	20.4	40.4	73.2	22.0	41.7	74.2	26.7	37.7	72.7
100-120	24.7	35.3	71.7	25.3	35.3	73.3	30.0	32.2	69.7
120-140	27.7	32.9	71.0	28.0	31.3	73.3	33.1	28.6	69.2
140-160	28.1	31.4	70.3	28.1	30.8	74.3	34.2	27.3	70.3
160-180	28.9	30.5	69.9	29.2	29.7	75.5	35.4	26.9	69.9
180-200	29.1	30.2	70.0	29.5	29.7	76.4	35.9	26.7	69.7
200-220	30.1	29.4	70.0	30.3	29.4	76.5	37.0	26.0	69.4
220-240	29.6	29.6	69.8	30.4	30.0	77.2	36.6	26.6	68.1
Original Video	27.5	59.0	90.1	26.0	62.8	90.0	35.7	52.2	90.7

Frequency(Hz)	Train on: OULU-NPU Evaluate on: Replay-Attack								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	5.5	33.3	33.5	6.0	38.9	40.0	6.8	51.0	54.4
20-40	8.0	13.0	16.5	4.7	13.6	13.4	6.8	13.7	15.6
40-60	23.0	8.5	25.5	19.1	4.3	25.7	26.8	7.0	20.0
60-80	30.0	10.6	27.7	21.9	4.6	24.4	35.3	6.6	21.3
80-100	38.1	10.3	40.6	38.6	2.9	38.8	47.7	6.8	36.7
100-120	42.3	13.6	38.2	43.7	2.2	38.1	53.8	6.2	39.6
120-140	47.2	12.7	62.0	49.6	1.4	58.4	57.2	6.4	55.6
140-160	51.0	11.4	73.7	53.1	4.5	76.7	60.8	6.3	70.3
160-180	53.8	10.2	71.9	56.2	5.1	79.3	62.8	6.8	71.0
180-200	54.3	9.2	79.3	56.4	5.2	80.0	63.4	6.7	76.5
200-220	56.0	7.3	80.0	58.8	4.8	80.0	65.4	6.4	77.9
220-240	56.5	8.4	80.0	59.7	4.3	80.1	65.6	6.3	78.3
Original Video	66.2	17.1	80.0	66.4	14.8	83.3	70.3	8.2	85.0

illumination in these video frames is very strong. Therefore, we speculate that the illumination interferes with the detection accuracy of the proposed method.

E. Cross-Database Analysis

In real-world applications, face PAD techniques are operated in open environments, where the conditions and attack scenario are unknown. To gain insight into the generalization capabilities of our motion blur analysis based detection method, we conduct a cross-database evaluation. To be more specific, the countermeasure was trained and tuned on one of the databases and then tested on another database. The obtained detection performances are summarized in Table VIII and Table IX.

When the proposed late fusion countermeasure is trained on OULU-NPU and tested on Replay-Attack database, we notice that the best averaged BPCER20 is got in the frequency from $20Hz$ to $40Hz$ with BPCER20 = 43.8%. However, the best averaged BPCER20 of early fusion is close to the late fusion with BPCER20 = 43.6%. When the proposed countermeasure is trained on Replay-Attack and tested on OULU-NPU database, the best averaged BPCER20s of early fusion and late fusion are 71.2% and 65.8%, respectively. From these results, we observe that the model trained on Replay-Attack database is not able to be generalized as good as the model trained on OULU-NPU. The reason behind this is that the OULU-NPU contains more variations in the collected data (e.g., light and imaging quality) compared to the Replay-Attack database. Therefore, the model tuned for Replay-Attack has difficulties to perform well in the new acquisition conditions. In addition, we also find that the classifier achieved the best results from intra database test does not achieve the best cross-database evaluation results. For instance, the best BPCER20 of intra OULU-NPU test is 1.0% while the averaged BPCER20 of cross database test is 79.5%. Finally, compared with the-state-of-the-art methods, our proposed method is more stable as shown in Table X.

Although our proposed algorithm can detect well replayed video attacks, it is difficult to distinguish fake face intrusions with no brightness variation. This is due to the fact that our detection algorithm is based on motion blur changes in brightness. In addition to brightness variation, the movement of the screen is also an important factor in the effectiveness of our detection method. Specifically, when the screen is completely stationary relative to the camera, our proposed motion blur description feature may not be applicable. Since the proposed detection method needs to rely on the response time of the liquid crystal, a screen with better contrast range and a higher frame rate or resolution would be harder to be detected for the proposed system.

VII. CONCLUSION

In this article, we addressed the problem of replayed face PAD from the viewpoint of the motion blur analysis. We

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895212, IEEE Transactions on Information Forensics and Security

TABLE IX
THE LATE FUSION PERFORMANCE OF CROSS-DATABASE EVALUATION IN TERMS OF APCER(%), BPCER(%) AND BPCER20(%) ON REPLAY-ATTACK AND OULU-NPU.

Frequency(Hz)	Train on: Replay-Attack Evaluate on: OULU-NPU								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	36.7	21.2	60.8	35.7	14.1	68.9	47.5	17.5	67.8
20-40	38.7	19.9	66.1	38.5	19.4	73.2	46.8	16.2	71.2
40-60	44.5	19.1	66.3	47.8	18.5	72.5	55.2	15.4	71.2
60-80	49.6	17.5	67.5	53.2	15.8	71.6	58.0	13.2	66.0
80-100	52.5	14.6	65.0	55.8	10.8	68.4	60.2	9.9	64.9
100-120	57.8	12.6	65.1	58.1	8.4	68.6	63.1	7.4	65.1
120-140	59.1	12.1	67.1	60.1	8.3	70.6	64.5	7.0	67.2
140-160	59.3	12.2	67.5	60.3	8.6	71.0	65.0	6.7	67.8
160-180	59.3	12.7	76.8	59.2	7.9	71.5	62.4	6.4	73.1
180-200	58.8	13.8	79.3	58.4	9.7	73.7	61.7	6.7	72.5
200-220	57.1	14.1	77.9	57.5	10.6	72.8	60.3	7.2	71.7
220-240	57.4	15.1	78.0	57.3	11.7	74.2	59.8	7.5	73.1
Original Video	51.6	39.1	92.4	56.9	26.9	89.7	60.7	21.4	92.6

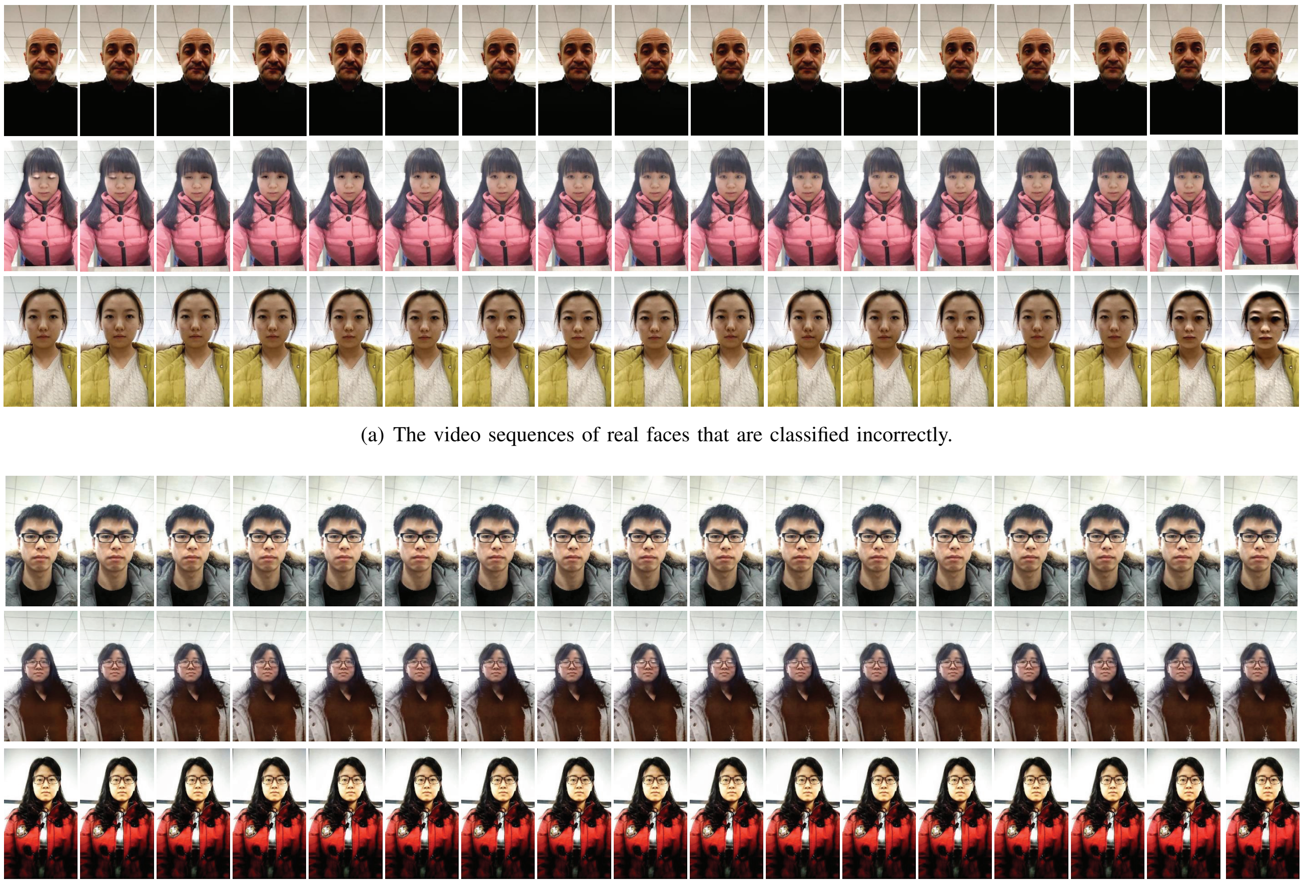
Frequency(Hz)	Train on: OULU-NPU Evaluate on: Replay-Attack								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
0-20	7.1	30.8	36.5	5.4	32.4	32.7	4.4	46.0	44.6
20-40	9.0	5.6	9.4	7.8	10.7	17.1	8.8	12.9	24.6
40-60	16.9	10.2	24.7	13.5	20.3	37.4	19.7	19.3	36.8
60-80	34.4	14.7	37.7	30.2	22.6	48.1	43.1	21.3	49.1
80-100	50.7	10.9	55.3	55.1	19.4	61.5	60.0	14.8	70.9
100-120	60.0	11.4	61.1	64.6	14.0	67.4	68.5	13.2	81.3
120-140	66.3	16.9	90.1	69.8	19.9	87.2	72.6	17.4	96.8
140-160	74.6	6.3	85.2	77.6	3.1	80.1	80.1	9.5	98.8
160-180	78.3	13.4	100	84.6	20.0	99.2	83.9	13.4	100
180-200	81.8	15.9	100	87.0	20.8	100	88.0	16.9	100
200-220	82.8	10.3	99.7	85.4	14.5	99.9	86.9	12.7	100
220-240	84.1	8.8	98.1	87.5	12.8	100	87.7	10.2	100
Original Video	51.5	61.6	93.3	55.5	60.1	91.3	51.6	58.7	93.6

TABLE X
THE PERFORMANCE OF CROSS-DATABASE EXPERIMENT ON THE REPLAY-ATTACK AND OULU-NPU DATABASES COMPARED WITH BASELINE METHOD.

Methods	Train on: Replay-Attack Evaluate on: OULU-NPU								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
Moiré Analysis[18]	38.4	53.7	93.9	39.0	49.0	95.7	44.2	43.3	93.6
Image Quality[19]	41.5	42.3	86.6	32.2	48.5	89.0	41.0	43.9	90.3
HOOF[20]	28.8	64.2	93.6	25.7	63.2	92.9	33.6	68.4	95.9
RGB LBP[9]	47.2	27.9	90.3	36.5	35.2	91.4	39.7	32.5	92.0
YCbCr LBP[9]	78.7	**3.4**	88.4	68.4	**5.0**	88.6	70.5	**3.6**	87.4
HSV LBP[9]	**15.9**	61.3	85.3	**17.7**	64.1	92.3	**20.2**	60.1	86.8
Proposed Early Fusion	23.1	35.9	71.6	27.3	33.0	**68.0**	30.4	32.3	77.6
Proposed Late Fusion	38.7	19.9	**66.1**	38.5	19.4	73.2	46.8	16.2	**71.2**

Methods	Train on: OULU-NPU Evaluate on: Replay-Attack								
	Train Set			Dev set			Test set		
	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)	APCER(%)	BPCER(%)	BPCER20(%)
Moiré Analysis[18]	52.4	38.6	91.2	41.1	50.9	99.2	44.1	46.4	94.7
Image Quality[19]	51.4	**2.4**	50.8	47.9	**2.7**	47.1	57.7	**1.5**	55.0
HOOF[20]	36.2	49.4	81.9	32.1	47.5	80.7	34.1	48.7	84.2
RGB LBP[9]	59.7	26.5	90.4	53.1	34.0	98.2	49.1	40.9	92.5
YCbCr LBP[9]	26.0	35.6	67.9	26.7	40.1	85.0	22.0	40.0	68.8
HSV LBP[9]	49.1	26.1	89.5	45.1	37.3	89.9	38.3	36.0	97.2
Proposed Early Fusion	**8.0**	13.0	16.5	**4.7**	13.6	**13.4**	**6.8**	13.7	**15.6**
Proposed Late Fusion	9.0	5.6	**9.4**	7.8	10.7	17.1	8.8	12.9	24.6

designed a new 1D CNN for motion blur intensity description and introduced a feature descriptor of LSP for motion blur width. Apart from that, we also investigated how to fuse the 1D CNN feature and LSP feature for describing the intrinsic disparities in the motion blur between genuine faces and replayed fake ones. Extensive experiments on the two latest and challenging face PAD databases (the Replay-Attack database and OULU-NPU database) showed excellent results. On OULU-NPU database, the proposed motion blur representation based on the 1D CNN and LSP outperformed the state



(a) The video sequences of real faces that are classified incorrectly.

(b) The video sequences of fake faces that are classified incorrectly.

Fig. 13. Examples of video sequences in OULU-NPU database that are classified incorrectly.

of the art, while very competitive results were achieved on Replay-Attack database. Furthermore, after feature fusion, our proposed method showed promising detection performance, thus suggesting that 1D CNN and LSP complement each other.

From the results on Replay-Attack and OULU-NPU databases, we find that external-environment factors (e.g. illumination) limit the effectiveness of our proposed detection method. Thus, we will study how to eliminate the influence of external environmental factors and improve the robustness of our method. In addition, the best fusion mechanism of Replay-Attack and OULU-NPU are different. Therefore, we will also try to find an optimal fusion way for 1D CNN and LSP features.

Acknowledgment

This paper is partly supported by the National Natural Science Foundation of China (No.61702419), the Natural Science Basic Research Plan of Shaanxi Province of China (No. 2018JQ6090), and the Fundamental Research Funds for the Central Universities (No.3102015BJ(II)ZS016).

References

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

[2] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, April 2015.

[3] Y. Li, K. Xu, Q. Yan, Y. Li, and R. H. Deng, "Understanding osn-based facial disclosure against face authentication systems," *ACM Symposium on Information, Computer and Communications Security*, pp. 413–424, June 2014.

[4] L. Omar and I. Ivrissimtzis, "Evaluating the resilience of face recognition systems against malicious attacks," in *Seventh UK British Machine Vision Workshop*, Sept 2015, Conference Proceedings, pp. 5.1–5.9.

[5] X. Li, J. Komulainen, G. Zhao, P. C. Yuen, and M. Pietikäinen, "Generalized face anti-spoofing by detecting pulse from face videos," in *International Conference on Pattern Recognition*, Dec 2017, Conference Proceedings, pp. 4244–4249.

[6] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Biometrics Special Interest Group*, Sept 2012, Conference Proceedings, pp. 1–7.

[7] N. Erdogmus and 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*, Sept 2013, Conference Proceedings, pp. 1–6.

[8] J. Määttä, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *International Joint Conference on Biometrics*, Oct 2011, Conference Proceedings, pp. 1–7.

[9] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *IEEE International Conference on Image Processing*, Sept 2015, Conference Proceedings, pp. 2636–2640.

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

[11] Y. Li and 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*, Nov 2009, Conference Proceedings, pp. 1–5.

[12] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *International Conference on Biometrics*, March 2012, Conference Proceedings, pp. 26–31.

[13] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision*, Sept 2010, Conference Proceedings, pp. 504–517.

[14] H. Li, S. Wang, and A. C. Kot, "Face spoofing detection with image quality regression," in *International Conference on Image Processing Theory Tools and Applications*, Dec 2016, Conference Proceedings, pp. 1–6.

[15] I. Pavlidis and 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*, June 2000, Conference Proceedings, pp. 15–24.

[16] Z. Zhang, D. Yi, Z. Lei, and S. Z. Li, "Face liveness detection by learning multispectral reflectance distributions," in *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, May 2011, Conference Proceedings, pp. 436–441.

[17] L. Li, P. L. Correia, and A. Hadid, "Face recognition under spoofing attacks: countermeasures and research directions," *IET Biometrics*, vol. 7, no. 1, pp. 3–14, 2018.

[18] K. Patel, H. Han, A. K. Jain, and G. Ott, "Live face video vs. spoof face video: Use of moire patterns to detect replay video attacks," in *International Conference on Biometrics*, May 2015, Conference Proceedings, pp. 98–105.

[19] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 710–724, Feb 2014.

[20] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Computer Vision and Pattern Recognition Workshops*, Sept 2013, Conference Proceedings, pp. 105–110.

[21] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Conference on Computer Vision and Pattern Recognition*, June 2009, Conference Proceedings, pp. 1932–1939.

[22] A. Alotaibi and A. Mahmood, "Enhancing computer vision to detect face spoofing attack utilizing a single frame from a replay video attack using deep learning," in *International Conference on Optoelectronics and Image Processing*, June 2016, Conference Proceedings, pp. 1–5.

[23] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, "Riesz pyramids for fast phase-based video magnification," in *IEEE International Conference on Computational Photography*, May 2014, Conference Proceedings, pp. 1–10.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, Feb 1995.

[25] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *IEEE International Conference on Automatic Face and Gesture Recognition*, May 2017, Conference Proceedings, pp. 612–618.

[26] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: A public database and a baseline," in *International Joint Conference on Biometrics*, Oct 2011, Conference Proceedings, pp. 1–7.

[27] S. Kim, Y. Ban, and S. Lee, "Face liveness detection using a light field camera," *Sensors*, vol. 14, no. 12, pp. 22 471–22 499, Nov 2014.

[28] Z. Ji, H. Zhu, and Q. Wang, "Lfhog: A discriminative descriptor for live face detection from light field image," in *IEEE International Conference on Image Processing*, Sept 2016, Conference Proceedings, pp. 1474–1478.

[29] F. P. A. Sepas-Moghaddam, P. Correia, "Light field local binary patterns description for face recognition," in *IEEE International Conference on Image Processing*, Sept 2017, Conference Proceedings, pp. 3815–3819.

[30] A. Agarwal, R. Singh, and M. Vatsa, "Face anti-spoofing using haralick features," in *IEEE International Conference on Biometrics Theory, Applications and Systems*, Sept 2016, Conference Proceedings, pp. 1–6.

[31] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov 1973.

[32] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, Aug 2016.

[33] Z. Boulkenafet, J. Komulainen, X. Feng, and A. Hadid, "Scale space texture analysis for face anti-spoofing," in *International Conference on Biometrics*, June 2016, Conference Proceedings, pp. 1–6.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, Dec 2012, Conference Proceedings, pp. 1097–1105.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository*, vol. abs/1409.1556, Sept 2014.

[36] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *Computing Research Repository*, vol. abs/1408.5601, pp. 373–384, Aug 2014.

[37] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *International Conference on Image Processing Theory Tools and Applications*, Dec 2016, Conference Proceedings, pp. 1–6.

[38] L. Li, X. Feng, X. Jiang, Z. Xia, and A. Hadid, "Face anti-spoofing via deep local binary patterns," in *IEEE International Conference on Image Processing*, Sept 2018, Conference Proceedings, pp. 101–105.

[39] L. Li, X. Feng, Z. Xia, X. Jiang, and A. Hadid, "Face spoofing detection with local binary pattern network," *Journal of Visual Communication and Image Representation*, vol. 54, pp. 182 – 192, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1047320318301044

[40] Z. Xu, S. Li, and W. Deng, "Learning temporal features using lstm-cnn architecture for face anti-spoofing," in *Asian Conference on Pattern Recognition*, Nov 2015, Conference Proceedings, pp. 141–145.

[41] H. Li, P. He, S. Wang, A. Rocha, X. Jiang, and A. C. Kot, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, pp. 1–14, April 2018.

[42] T. D. F. Pereira, A. Anjos, J. M. D. Martino, and S. Marcel, "Lbp-top based countermeasure against face spoofing attacks," in *Asian Conference on Computer Vision Workshops*, Nov 2012, Conference Proceedings, pp. 121–132.

[43] Q. T. Phan, D. T. Dang-Nguyen, G. Boato, and F. G. B. D. Natale, "Face spoofing detection using ldp-top," in *IEEE International Conference on Image Processing*, Sept 2016, Conference Proceedings, pp. 404–408.

[44] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 915–928, June 2007.

[45] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 533–544, Feb 2010.

[46] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki, and A. T. S. Ho, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, April 2015.

[47] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *International Conference on Image Analysis and Signal Processing*, April 2009, Conference Proceedings, pp. 233–236.

[48] G. Pan, L. Sun, Z. Wu, and Y. Wang, "Monocular camera-based face liveness detection by combining eyeblink and scene context," *Telecommunications Systems*, vol. 47, no. 3-4, pp. 215–225, Aug 2011.

[49] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 736–745, April 2015.

[50] L. Feng, L. M. Po, Y. Li, X. Xu, F. Yuan, C. H. Cheung, and K. W. Cheung, "Integration of image quality and motion cues for face anti-spoofing," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, July 2016.

[51] A. Pinto, H. Pedrini, W. R. Schwartz, and A. Rocha, "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, Dec 2015.

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

[53] "Lcd," https://www.merriam-webster.com/dictionary/LCD, accessed March 9, 2018.

[54] S. H. Lee and M. H. Lee, "Liquid crystal displays with high image quality and fast response time," *Journal of the Korean Physical Society*, vol. 39, pp. S42–S48, Dec 2001.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895212, IEEE Transactions on Information Forensics and Security

[55] A. A. S. Sluyterman, "What is needed in lcd panels to achieve crt-like motion portrayal?" *Journal of the Society for Information Display*, vol. 14, no. 8, pp. 681–686, Aug 2006.

[56] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 105–116, Feb 2008.

[57] H. Y. Lin and K. J. Li, "Motion blur removal and its application to vehicle speed detection," in *International Conference on Image Processing*, Oct 2004, Conference Proceedings, pp. 3407–3410.

[58] H. Y. Wu, E. Shih, E. Shih, J. Guttag, and W. Freeman, "Eulerian video magnification for revealing subtle changes in the world," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, July 2012.

[59] K. Patel, H. Han, and A. K. Jain, "Cross-database face antispoofing with robust feature representation," in *Chinese Conference on Biometric Recognition*, Sept 2016, Conference Proceedings, pp. 611–619.

[60] M. S. M. Sajjadi, B. Schölkopf, and ichael Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," *CoRR*, vol. abs/1612.07919, 2016. [Online]. Available: http://arxiv.org/abs/1612.07919

[61] X. Glorot, A. Bordes, Y. Bengio, X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics*, April 2011, Conference Proceedings, pp. 315–323.

[62] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, Sept 2015, Conference Proceedings, pp. 1–12.

[63] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *ACM International Conference on Multimedia*, Oct 2015, Conference Proceedings, pp. 689–692.

[64] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, architectures, and applications*. New York: Psychology Press, 1995.

[65] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, July 2002.

[66] H. R. Pourreza, M. Masoudifar, and M. ManafZade, "Lsp: Local similarity pattern, a new approach for rotation invariant noisy texture analysis," in *18th IEEE International Conference on Image Processing*, Sept 2011, Conference Proceedings, pp. 837–840.

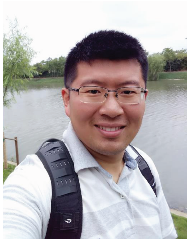
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, Conference Proceedings, pp. 1026–1034.

[68] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Physica-Verlag HD, Sept 2010, pp. 177–186.

[69] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. 9, pp. 1871–1874, Aug 2008.

[70] "Iso/iec jtc 1/sc 37 biometrics. information technology - biometric presentation attack detection - part 1: Framework," International Organization for Standardization, Tech. Rep., 2016.

Lei Li received the M.S. degree in electronics and communication engineering from Shanghai Maritime University, Shanghai, China. He is currently pursuing the Ph.D. degree with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include signal and image processing, biometrics, and spoofing detection.



Zhaoqiang Xia received the B.E. degree and Ph.D. degree from the Northwestern Polytechnical University, Xi'an, China, in 2008 and 2014, respectively. He is currently an associate professor in the School of Electronics and Information, Northwestern Polytechnical University. He has authored or co-authored more than 30 papers in journals and conferences, and has served as a reviewer for journals and conferences. His current research interests include multimedia retrieval, statistical machine learning and computer vision.

Abdenour Hadid received the D.Sc. degree in electrical and information engineering from the University of Oulu, Finland, in 2005. He is currently an Academy Research Fellow with the Center for Machine Vision and Signal Analysis, University of Oulu. He is regularly visiting the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, machine learning, and pattern recognition with a particular focus on biometrics.

Xiaoyue Jiang received the M.S. and Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2003 and 2006, respectively. Then she worked as an associated researcher in Vrijie University of Brussels (Belgiumn), the University of Birmingham (UK), and Queen's University of Belfast (UK) from 2006 to 2012. She is currently an associate professor with the School of Electronics and Information, Northwestern Polytechnical University since 2012. Her research interests include image processing, computer vision and machine learning.

Haixi Zhang received the B.E. and M.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2011 and 2013, respectively. He is currently pursuing the Ph.D degree with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include machine learning, computer vision and image processing.

Xiaoyi Feng received the M.S. degree from the Northwest University, Xi'an, China, in 1994. She received her Ph.D. degree from the Northwestern Polytechnical University, Xi'an, China, in 2001. She is currently a professor with the School of Electronics and Information, Northwestern Polytechnical University since 2008. Her current research interests include computer vision, image process, radar imagery and recognition.