Journal of
Electronic Imaging

JElectronicImaging.org

Face antispoofing based on frame difference and multilevel representation

Azeddine Benlamoudi
Kamal Eddine Aiadi
Abdelkrim Ouafi
Djamel Samai
Mourad Oussalah

SPIE.

IS&T
imaging.org

Azeddine Benlamoudi, Kamal Eddine Aiadi, Abdelkrim Ouafi, Djamel Samai, Mourad Oussalah, "Face antispoofing based on frame difference and multilevel representation," *J. Electron. Imaging* **26**(4), 043007 (2017), doi: 10.1117/1.JEI.26.4.043007.

Face antispoofing based on frame difference and multilevel representation

Azeddine Benlamoudi,[a,*] Kamal Eddine Aiadi,[a] Abdelkrim Ouafi,[b] Djamel Samai,[a] and Mourad Oussalah[c]
[a]University of Ouargla, Faculté des Nouvelles Technologies de l'information et de la communication,
Laboratoire de Génie Électrique (LAGE), Ouargla, Algeria
[b]University of Biskra, Laboratory of LESIA, Algeria
[c]University of Oulu, Center for Ubiquitous Computing, Finland

Abstract. Due to advances in technology, today's biometric systems become vulnerable to spoof attacks made by fake faces. These attacks occur when an intruder attempts to fool an established face-based recognition system by presenting a fake face (e.g., print photo or replay attacks) in front of the camera instead of the intruder's genuine face. For this purpose, face antispoofing has become a hot topic in face analysis literature, where several applications with antispoofing task have emerged recently. We propose a solution for distinguishing between real faces and fake ones. Our approach is based on extracting features from the difference between successive frames instead of individual frames. We also used a multilevel representation that divides the frame difference into multiple multiblocks. Different texture descriptors (local binary patterns, local phase quantization, and binarized statistical image features) have then been applied to each block. After the feature extraction step, a Fisher score is applied to sort the features in ascending order according to the associated weights. Finally, a support vector machine is used to differentiate between real and fake faces. We tested our approach on three publicly available databases: CASIA Face Antispoofing database, Replay-Attack database, and MSU Mobile Face Spoofing database. The proposed approach outperforms the other state-of-the-art methods in different media and quality metrics. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.4.043007]

Keywords: antispoofing; multilevel; frame difference; local binary pattern; local phase quantization; binarized statistical image features.

Paper 170022 received Jan. 11, 2017; accepted for publication Jun. 27, 2017; published online Jul. 21, 2017.

1 Introduction

The human face has a significant role in our life by conveying people's identity. It is used in biometric face recognition technology as a key to security. Face-based biometric systems have many advantages over other biometrics systems, such as fingerprint, palm-print, and iris. The most important advantage is that face images can be captured from a distance without a physical contact with the person to be identified. However, such advantages can be problematic when the intruder uses an image of the authentic user taken using a simple cell phone or from a social media, such as Facebook, which is widely published and easy to possess. This problem is referred to as attack (spoof).

The face recognition systems can be spoofed using different types of attacks: print photo, video, mask, make-up, or plastic surgery.[1,2] The most popular attacks are video and print attacks. In addition to their simple implementations, many studies showed the effectiveness of these attacks against face recognition systems.[3] Recently, many researchers are focused on detecting these types of attacks using different face antispoofing methods.

The face antispoofing methods allow the system to separate between a genuine face and fake one using a single image or a sequence of images (video). These methods analyze the appearance or the dynamic properties of the face images.[1,4–9]

Motivated by the background subtraction, we remarked in our research that unlike individual frames, the frame difference (FD) provides useful insights for distinguishing between genuine face and that captured using printed photos or mobile devices.[10] Therefore, we exploit in this work both dynamic and static information (texture and motion) to differentiate between real and fake faces. We used the temporal information that is present in video to analyze the dynamics of facial texture by applying the texture descriptors on the FD.

More specifically, the approach proceeds in the following way. First, we detected the face and the eyes to rotate and crop the region of interest (ROI) in each frame. After the normalization of the ROI, we extracted the foreground using the difference between two frames to compute the movement. Then, we divided the ROI into multilevel (ML). In each block, we extracted the features using one of these feature extractors: local phase quantization (LPQ), local binary patterns (LBP), and binarized statistical image features (BSIF). After that, we calculated the mean of the feature vectors over a time window of 150 frames; then we used the Fisher score (FS) method to rank the features. Finally, the support vector machine (SVM) was used for classification.

We conducted extensive experiments on three publicly available databases, namely, CASIA Face Antispoofing database (CASIA-FASD)[11], Replay-Attack database,[12] and MSU Mobile Face Spoofing database (MSU-MFSD),[13] which show that our approach gives good results compared with those of the state of the art.

*Address all correspondence to: Azeddine Benlamoudi, E-mail: azeddine.benlamoudi@univ-ouargla.dz

1017-9909/2017/$25.00 © 2017 SPIE and IS&T

The rest of the paper is organized as follows: Sec. 2 describes related works on face antispoofing. Section 3 shows and highlights the main contribution of the paper. Section 4 presents the proposed approach. Section 5 contains the databases that we used in our experiments, the experimental evaluation of our proposed method in the previous databases, and the comparison with the other methods. Finally, Sec. 6 concludes the paper.

2 Related Work

There are many ways to detect spoof attacks. In this paper, we focus on only two types of face antispoofing methods, which are hardware- and software-based techniques. In this section, we present all previous work in face antispoofing techniques, but we focus only on those that are thematically closer to our objectives and contributions (see Table 1).

2.1 *Hardware-Based Techniques*

The hardware-based techniques advocate incorporating extra hardware devices to differentiate between the real and the fake faces. Ng and Chia[34] used randomized temporal affective cues in the form of facial expressions to verify the liveness of users. Pavlidis and Symosek[35] showed that the band of the near-infrared (1.4 to 2.4 μm) is particularly advantageous for disguise detection purposes. Chetty and Wagner[36] combined acoustic and visual feature vectors to distinguish live

Table 1 A summary of published methods on face spoof detection.

Authors	Methods	Databases	Years
Kollreider et al.[5]	Motion	MITCMU YALE Recaptured	2007
Biggio et al.[14]	Multimodal	LivDet11 Photo Attack Personal Photo Attack Print Attack	2011
Chingovska et al.[15]	Texture	Replay-Attack CASIA-FAS NUAA photograph imposter	2012
Määttä et al.[16]	Texture	Yale Recaptured PRINT ATTACK	2012
Kose and Dugelay[17]	Texture	NUAA photograph imposter	2012
Erdogmus and Marcel[18]	3-D	Morpho 3-D Mask Attack	2013
Yang et al.[1]	Texture	NUAA photograph imposter CASIA-FAS PRINT ATTACK	2013
Komulainen et al.[19]	Motion	Replay–Attack	2013
Galbally and Marcel[20]	IQA	CASIA-FAS Replay-Attack	2014
Galbally et al.[21]	Multimodal	Iris spoof, Iris-Synthetic LivDet REPLAY-ATTACK	2014
Bharadwaj et al.[22]	Motion	PRINT ATTACK Replay-Attack CASIA-FAS	2014
de Freitas Pereira et al.[2]	Motion	Replay-Attack CASIA-FAS	2014
Menotti et al.[23]	Deep learning	Warsaw, Biosec & MobBIOfake Replay-Attack & 3-DMAD Biometrika, CrossMatch, Italdata & Swipe	2015
Garcia and de Queiroz[24]	Moiré pattern	Replay-Attack Moiré	2015
Yang et al.[25]	Person-specific	CASIA-FAS Replay-Attack	2015
Chingovska and Anjos[26]	Person-specific	Replay–Attack	2015
Wen et al.[27]	Motion	Replay-Attack CASIA-FAS MSU-MFS	2015
Pinto et al.[28]	Motion	CASIA-FAS Replay-Attack UVAD 3-DMAD	2015
Tirunagari et al.[29]	Motion	PRINT ATTACK Replay-Attack CASIA-FAS	2015
Arashloo et al.[30]	Texture	Replay-Attack CASIA-FAS NUAA photograph imposter	2015
Boulkenafet et al.[31]	Color texture	CASIA-FAS Replay–Attack	2015
Patel et al.[32]	Color texture	Replay-Attack CASIA-FAS MSU-MFS	2015
Galbally and Satta[33]	3-D	3-DFS-DB EURECOM MASK-ATTACK DB IDIAP MASK-ATTACK DB	2016

synchronous audio-video recordings from Replay-Attacks that use audio with a still photo. Erdogmus and Marcel[37] used depth information to discriminate between the real and the two-dimensional spoofing attacks. Smith et al.[38] proposed an approach for face recognition systems that can counter the attacks using the color reflected from the user face, which is displayed on the mobile devices. These reflections are used to determine whether the images were captured in real time. Wang et al.[39] proposed a face liveness detection approach to counter spoofing attacks by recovering sparse three-dimensional (3-D) facial structure. Other methods used different visual spectrum (complementary infrared, near-infrared, etc.)[40–42] to distinguish between the genuine faces and the spoof attacks.

2.2 *Software-Based Techniques*

The software-based techniques use the simple RGB images to detect the spoof attacks. These methods can be divided into static- and dynamic-based techniques. The static-based techniques are applied on a single image, while the dynamic-based techniques are applied on video sequences.

Most methods that differentiate between the real faces and the fake ones are based on texture analysis. Chingovska et al.[15] and Maatta et al.[16] used LBP as a descriptor to detect the spoof attack. Kose and Dugelay[17] used another variant of the LBP descriptor, which is LBP variance to differentiate between the real and the fake faces. Yang et al.[1] introduced a face recognition based on pooling the features extracted from the different face components using the Fisher criterion. Arashloo et al.[30] used kernel discriminant analysis fusion to combine two spatial–temporal descriptors multiscale BSIF on three orthogonal planes and multiscale LPQ on three orthogonal planes. de Freitas Pereira et al.[2] also worked with the dynamic texture based on LBP histograms on three orthogonal planes (LBP-TOP) to differentiate between real and fake people. This last method showed better performances compared with the simple LBP methods proposed in Refs. 15–17. The reason for the good results of LBP-TOP is that temporal information plays an important role in face antispoofing. Pinto et al.[28] proposed a method based on temporal and spectral information, which used the time-spectral features as low-level descriptors and used the visual codebook concept to find midlevel features descriptors. Tirunagari et al.[29] proposed an algorithm called dynamic mode decomposition (DMD) to capture the visual dynamics while LBP is used to capture the dynamic patterns. Wen et al.[27] proposed a method based on image distortion analysis (IDA). Four different features, specular reflection, blurriness, chromatic moments, and color diversity, were used to represent the face images. These features can capture the differences between the real and the fake images without capturing the detail informations related to the user-identity.

Bharadwaj et al.[22] used the Eulerian motion magnification to enhance the motion cues. It was found that extracting histogram of oriented optical flow from the enhanced video yields an enhanced result with respect to the state-of-the-art results on the Replay-Attack database. Komulainen et al.[19] also used a fusion between the motion and the texture features to enhance the classification performances. Kollreider et al.[5] proposed strategies to avert advanced spoofing attempts, such as replayed videos, by analyzing the motion of the lips only.

Patel et al.[32] studied the effect of the different channels of the RGB color spaces (R, G, B, and grayscale) and the different face regions on the performance of the LBP- and dense scale invariant feature transform-based methods. Their experiments show that extracting the texture from the red channel gives the best results. Boulkenafet et al.[31] proposed a method of face antispoofing based on color texture analysis. After representing the RGB images in two color spaces, HSV and YCbCr, they used the LBP descriptor to extract the texture features from each channel, and then they concatenated these features to differentiate between real and fake faces.

Galbally and Marcel[20] proposed an image quality assessment (IQA) using 14 quality measures to distinguish between the real and the fake faces. Galbally et al.[21] evaluated 25 different quality measures, which were also used for fingerprint and iris antispoofing. Recently, some methods, such as those in Refs. 25 and 26, used the user-specific information to enhance the performance of the texture-based face antispoofing methods. Biggio et al.[14] addressed the problem of spoof attacks on biometrics using two modals: face and fingerprint. They tested different score-fusion rules, such as sum, product, weighted sum by linear discriminant analysis, likelihood ratio (LLR), and extended LLR.

Garcia and de Queiroz[24] proposed face spoofing detection by searching for moiré patterns due to the overlap of the digital grids. Their detection is based on peak detection in the frequency domain. They used SVM with radial basis function kernel for the classification. They conducted their experiments on Replay-Attack Corpus and Moir databases.

Other techniques in face antispoofing are based on textures on 3-D modals, such as Refs. 33 and 18. In 3-D modals, the attacker uses a mask to spoof the system, so the use of wrinkles would be a great assistant to detecting the attack. In Ref. 33, they presented a study that addresses the spoofing issue by analyzing the feasibility of performing low-cost attacks with self-manufactured 3-D printed models to 2.5-D and 3-D face recognition systems. Erdogmus and Marcel[18] inspected the spoofing potential of subject-specific 3-D facial masks for different recognition systems and addressed the detection problem of this more complex attack type. Also, the authors performed experiments on two different databases.

Recently, deep learning approaches have been used in face antispoofing, especially using convolutional neural network (CNN). For instance, Menotti et al.[23] focused on two general-purpose approaches to building image-based antispoofing systems using convolutional networks. Their systems deal with several attack types in three biometric modalities, namely, iris, face, and fingerprint. The first approach consists of learning suitable convolutional network architectures for each domain, while the second approach focuses on learning the weights of the network via backpropagation.

3 Research Contributions

In our work, we propose an algorithm for face spoofing detection based on an extended FD algorithm. We also combine this extended FD algorithm with an ML representation to take into account both dynamic and static information. This combination gave us multiple parts of the foreground image (see Fig. 1).

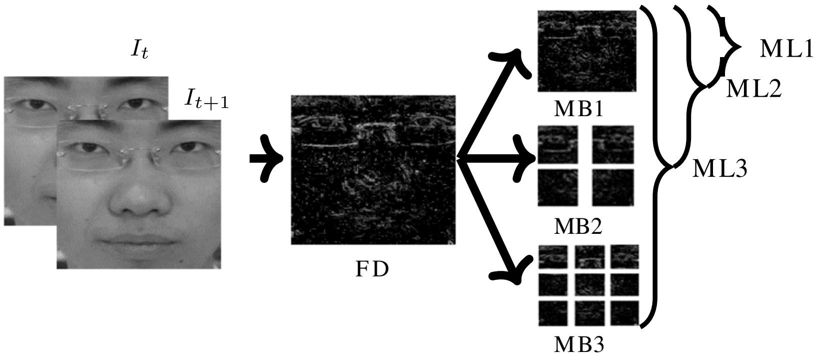

Fig. 1 Principle of FD and ML.

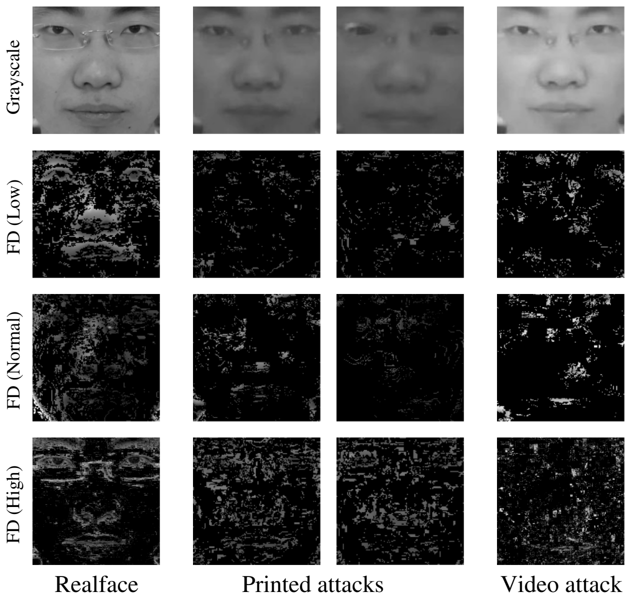

Fig. 2 Example of a genuine face and corresponding print and video attacks in grayscale and FD.

Commonly, the FD technique computes the difference between the current and the previous frame. A threshold is then used to obtain the foreground, which is a binary image. The equation of the FD is given as

$$F_t = |I_t - I_{t-1}|, \tag{1}$$

where F_t is the difference between two frames, I_t is the current frame, and I_{t-1} is the previous frame.

In our case, we used a threshold only to eliminate the unchanged pixels values between the two successive frames. If there is motion, the foreground pixels take the value of the current frame. However, if there is no motion, the foreground pixel is set to zero (see the below equation)

$$F'(i,j)_t = \begin{cases} I_t(i,j) & \text{if } F(i,j)_t > T \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where F'_t is the foreground and T is the threshold $= 0$.

Figures 2 and 3 and Tables 2 and 3 demonstrate the effectiveness of using our FD approach. In Table 2, we computed the entropy on the real and fake face of the same person. The entropy describes the quantity of information of the image, and the image entropy equation is given as

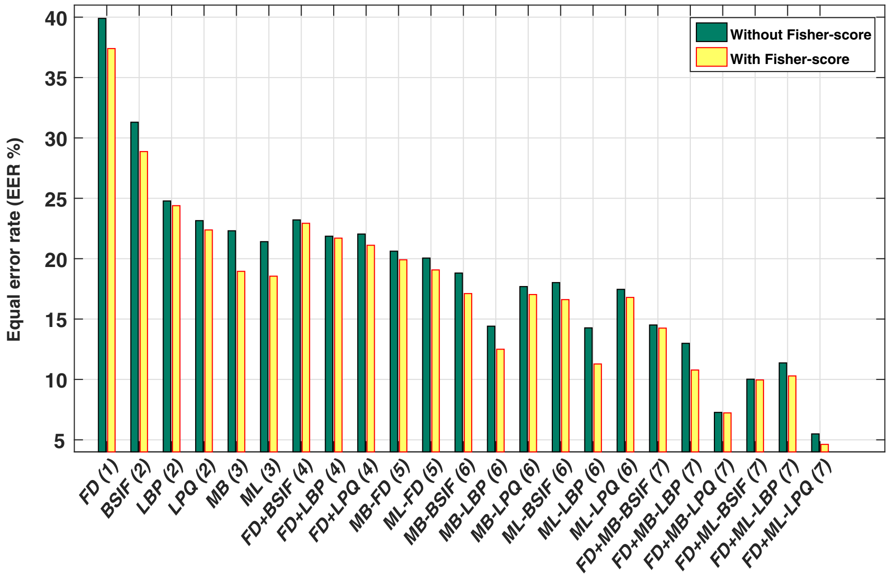

Fig. 3 Impact of FD (motion) on representation and texture across the CASIA-FASD.

Table 2 Entropy.

Qualities	Real	Printed	Attacks	Video attack
Low	4.16	1.76	1.58	1.96
Normal	4.13	2.17	1.57	1.79
High	4.92	3.87	3.35	4.68

Table 3 Results in EER (%) on CASIA for motion (FD), representation (ML and MB), and texture (LBP, LPQ, and BSIF).

Types of methods	Methods	Without Fisher	With Fisher
Motion (1)	FD	39.90	37.40
Texture (2)	BSIF	31.30	28.87
	LBP	24.78	24.39
	LPQ	23.15	22.38
Representation (3)	MB	22.31	18.95
	ML	21.41	18.55
Motion + texture (4)	FD-BSIF	23.21	22.93
	FD-LBP	21.86	21.70
	FD-LPQ	22.04	21.11
Motion + representation (5)	FD-MB	20.62	19.91
	FD-ML	20.05	19.07
Representation + texture (6)	MB-BSIF	18.81	17.11
	MB-LBP	14.41	12.50
	MB-LPQ	17.69	17.03
	ML-BSIF	18.02	16.61
	ML-LBP	14.27	11.28
	ML-LPQ	17.46	16.79
Motion + representation + texture (7)	FD-MB-BSIF	14.51	14.25
	FD-MB-LBP	12.99	10.78
	FD-MB-LPQ	07.27	07.23
	FD-ML-BSIF	10.02	09.96
	FD-ML-LBP	11.37	10.29
	FD-ML-LPQ	05.49	04.62

$$E = -\sum_{i=0}^{255} P_i . \log_2(P_i), \tag{3}$$

where E is the entropy of F' and P_i is the probability of color i. Both real and fake faces have three qualities (low, normal, and high). We took into account two types of attacks, printed and video. We observe that the entropy is greater in the case of real faces compared with fake faces in all scenarios.

Visually, we observe from Fig. 2 that when computing the foreground of the real faces, the facial features are more visible compared with the case of fake ones in all qualities and all types of attack. From these remarks, we have been motivated to use FD in face antispoofing. This FD allows us to extract motion in the foreground and illustrate the fake faces. We applied then the ML representation on the foreground of the FD that permits to obtain multiple blocks. This latter is then followed by texture descriptor. The use of FD (motion) combined with ML (representation)[43] and texture description improves the results. This was proved experimentally (see Fig. 3 and Table 3).

We observe from Table 3 and Fig. 3, when we used texture (2) descriptors (LBP, LPQ, and BSIF), we obtained an improvement in equal error rate (EER) compared with using motion (1) FD only by computing the histogram of FD directly. The results are better when we combined motion and texture (4). Another aspect is when we used ML and multiblock (MB) representations (3); this improves the results of both motion (1) and texture (2). We can remark also that combining representation with motion (5) or with texture (6) clearly improves the results. In the case of combining motion (FD) with texture and representation (7), the results are better compared with all previous methods. Finally, adding the FS to any method improves the EER. This is why we choose to use (motion) (FD) + texture (LPQ) + representation (ML) + FS in this paper as a new approach.

4 Proposed Framework

Figure 4 shows the general structure of our approach. First, we detect the face and localize the eye center coordinates to normalize the ROI. Second, we extract the motion using the FD between consecutive faces. Then, we apply ML representation to get multiple blocks to be used in features extraction. Features of all blocks are concatenated to get one feature vector. We used all the previous steps for an input video of 6 s (150 frames); then we averaged the feature vectors of all these frames. After that, we ranked the average feature vector by the FS. Finally, we used Library of SVM (Lib-SVM) as a classifier to differentiate between real and fake faces. In the following, we will discuss all these steps in details.

4.1 *Face Preprocessing*

In our approach, face preprocessing is performed in three steps: face detection, eye localization (pose correction), and face normalization. Face detection is a significant step in face antispoofing. We used the Viola and Jones algorithm[44] to detect the face region and the pictorial structure model[45] to localize the eye positions; then, the coordinates of the eyes are used to correct the face pose. In Fig. 5, we explain how to rotate and crop the face[46] using the eye coordinates where $R1$ is the coordinates of right eye, $L1$ is the coordinates of left eye, $R2$ is the coordinates of right eye

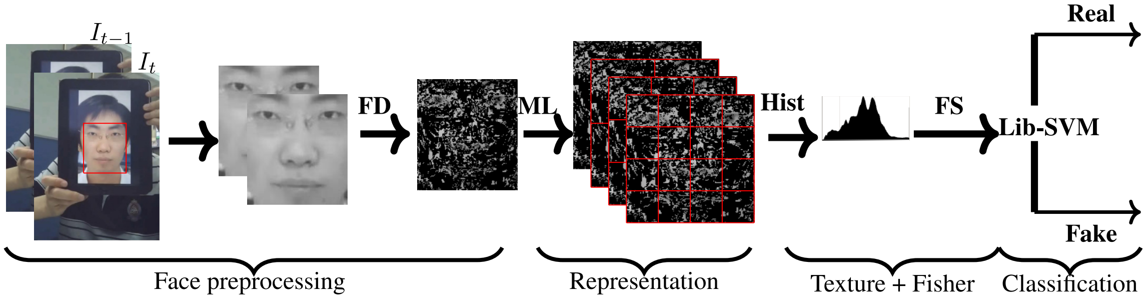

Fig. 4 Framework of our proposed approach.

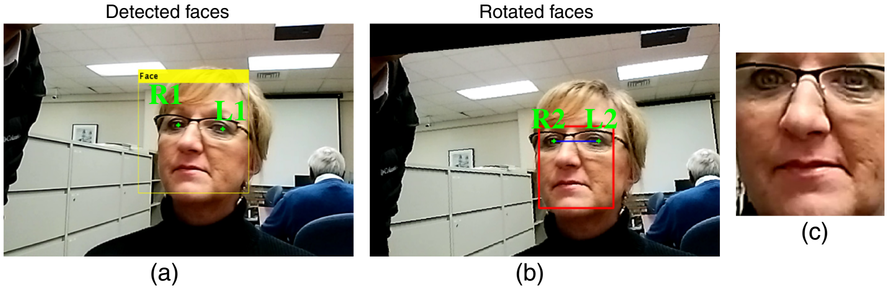

Fig. 5 Example of face alignment: (a) face and eye detection, (b) pose correction, and (c) cropped face (ROI).

after rotating the image, and $L2$ is the coordinates of left eye after rotating the image. FD (Sec. 3) is then applied on cropped faces for motion extraction. We will explain how to apply the representation (ML) on the FD in Sec. 4.2.

4.2 *Representation*

In our work, we choose to use the MLs representation that is extended from the MBs representation.

- MB is a technique that divides the face ROI into $(n \times n)$ blocks. On each block, we apply a texture descriptor to get more features of the face ROI. Figure 6(b) shows how we divide an image in MBs.
- ML representation[47] is a technique that combines the features extracted from consecutive different MBs. In other terms, we extract features of the whole image, and then we divide it into different blocks of different sizes and extract features of each block as illustrated in Fig. 6(c). The whole features are then concatenated into one vector.

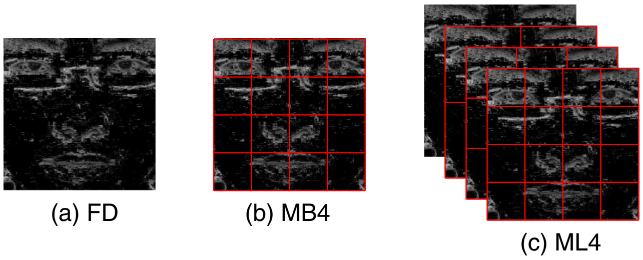

Fig. 6 Representation of MBs and MLs: (a) FD (whole image), (b) MB representation, and (c) ML representation.

4.3 *Texture Descriptors*

In our work, we used three popular texture descriptors: LBP, LPQ, and BSIF on FD-ML to extract the features to distinguish between the real and the fake faces. In this section, we will describe these techniques:

- LBP gives one feature vector for each image. During the LBP operation, every image pixel acts as a threshold to its neighborhood to obtain binary numbers. By scanning those binary numbers in a clockwise direction, we convert them to a decimal number. In the sequel, the neighborhood corresponds to the sampling point spaced on a circle that is centered at the pixel; next, the sampling points are interpolated using bilinear interpolation. The LBP is given by Eqs. (4) and (5), where the notation (P, R) denotes a neighborhood of P sampling points on a circle of radius R

$$S(X) = \begin{cases} 1 & \text{if } X \geq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c)2^p. \tag{5}$$

- LPQ was proposed by Ojansivu and Heikkilä[48] to deal with blurred images. The spatial blurring is given by convolution between two matrices, which are the image intensity and a point spread function (PSF) such as

$$g(x) = (f * h)(x), \tag{6}$$

where $g(x)$ is the blurred image, $f(x)$ is the true image, $h(x)$ is the PSF, and x is a vector of coordinates $[x, y]^T$.

In LPQ, the phase is examined in local neighborhoods N_x at each pixel position $x = [x_1, x_2]^T$ of the image $f(x)$. These local spectra are computed using a discrete short-term Fourier transform[48] defined as

$$F(u, x) = \sum_{y} f(y) w_R(y - x) e^{-2j\pi u^T y}. \tag{7}$$

- BSIF was proposed by Kannala and Rahtu.[49] The BSIF is represented by binary code string for the pixels of a given image. The code value of a pixel is considered a local descriptor of the image. Given an image patch X of size $(l \times l)$ pixels and a linear filter W_i of the same size, the filter response s_i is obtained as

$$s_i = \sum_{u,v} W_i(u, v) X(u, v) = W_i^T x, \tag{8}$$

where vector notation is introduced in the latter stage. Given n linear filters W_i, we stack them into a matrix W and compute all responses at once

$$S = W_x. \tag{9}$$

Next, given a random sample of natural image patches, we compute the filters W_i so that the elements s_i of s are as independent as possible when considered random variables (see Ref. 49).

4.4 *Fisher Score*

The FS[50] is one of the most widely used supervised features selection methods. The Fisher vector selects each feature independently according to its scores under the Fisher ratio, which leads to a suboptimal subset of features. The Fisher ratio is carried out in the feature domain to reject the noisy feature indexes and select the most informative combination from the remaining. The Fisher ratio is a measure of linear discriminating power of some variable [see Eq. (10)], with m_1 and m_2 being the means of class 1 (real) and class 2 (spoof) and v_1 and v_2 being the variances

$$\text{Fisherratio} = \frac{(m_1 - m_2)^2}{v_1 + v_2}. \tag{10}$$

4.5 *Classification*

In our algorithm, we used the Lib-SVM[51] to classify the feature vectors to real or fake faces. SVM performs the classification by finding the hyperplane that maximizes the margin between two classes. The vectors (cases) that define the hyperplane are called the support vectors.

5 Experiments and Results

To evaluate the performance of our approach, we used three challenge databases, which are CASIA-FAS, MSU-MFS, and Replay-Attack database. In this section, we will describe such databases in Sec. 5.1; then, we will present the setup of our approach in Sec. 5.2. Finally, in Sec. 5.3, we will discuss and analyze the results.

5.1 *Experimental Dataset*

To validate the performance of our proposed method, we used the three most challenging databases: CASIA-FASD, Replay-Attack database, and MSU-MFSD. The three databases contain video recording of real and fake attacks. A description of these databases is given below.

5.1.1 *CASIA Face Antispoofing*

CASIA-FAS database[52] contains 50 genuine subjects (see Fig. 7) and fake faces, which are recorded from genuine faces. Each subject has 12 videos (three real and nine fake faces), so the final database contains 600 videos. The CASIA database is constructed using three image quality descriptors (low, normal, and high) and three types of fake face attacks (warped photo attack, cut photo attack, and video attack). The protocol of CASIA-FAS has seven scenarios: low, normal, and high qualities, which evaluate the imaging quality, and warped photo, cut photo, and video attacks, which evaluate the type of media of the attack. The last scenario is the overall test, which has all types of qualities and attacks to evaluate the imaging quality and the media attack at the same time.

5.1.2 *Replay-Attack*

The Replay-Attack database consists of 1300 video clips of photo and video attacks, which were recorded from 50

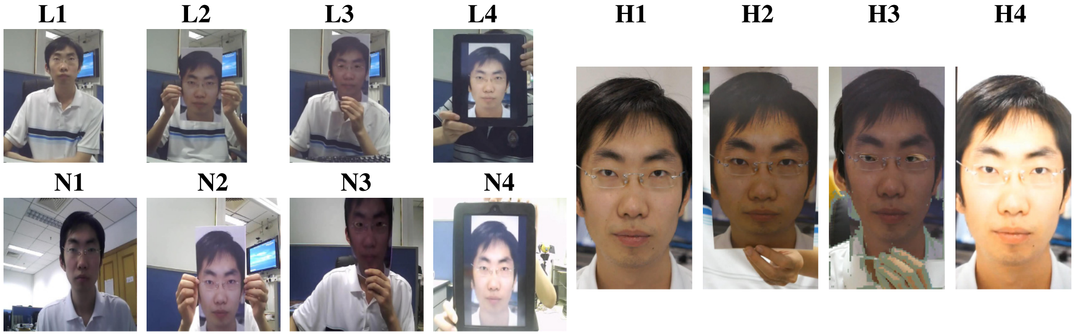

Fig. 7 Samples from the CASIA-FAS database. L, N, and H for low, normal, and high quality, respectively, and 1, 2, 3, and 4 for real face, warped photo, cut photo, and video attacks, respectively.

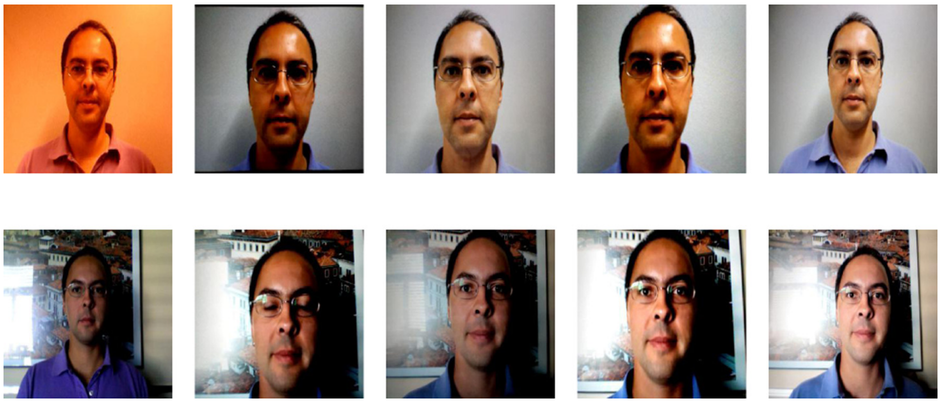

Fig. 8 Examples from the Replay-Attack database. The first row presents images taken from the controlled scenario, while the second row corresponds to the images from the adverse scenario. From the left to the right: real faces and hand video, hand photo, fixed video, and fixed photo.

subjects (see Fig. 8). These videos were recorded in two different conditions (controlled and adverse) using a built-in webcam of a Macbook laptop. To make the fake face attack for each person in high-resolution photos and videos, the image was taken using two cameras: a Canon PowerShot SX150 IS and an iPhone 3GS camera. There are two subsets for attacks, fixed attacks and hand attacks. On each subset, there are 10 videos: four mobile attacks using an iPhone 3GS screen with a resolution 480 × 320 pixels, four high-resolution screen attacks using an iPad first generation, with a screen resolution of 1024 × 768 pixels, and two hard-copy print attacks (produced on a Triumph-Adler DCC 2520 color laser printer) occupying the whole available printing surface on A4 paper. The database is split into three subgroups for train, development, and test.

5.1.3 *MSU Mobile Face Spoofing*

The publicly available MSU-MFSD for face spoof attacks was produced at the Michigan State University by the Patterns Recognition and Image Processing group. The database consists of 280 video clips of photo and video attack attempts on 35 clients. It was made by mobile phone to capture both genuine face and spoof attacks. They used two types of cameras: (1) a built-in camera in MacBook Air 13 in. (640 × 480) and (2) a front-facing camera of the Google Nexus 5 Android phone (720 × 480). Each subject had two video recordings; the first one is captured by a Laptop camera and the second one is captured using an Android camera (see Fig. 9). To generate the attacks, high-resolution video was captured for each subject using two devices: (1) a Canon PowerShot 550D SLR camera, recording 18.0M pixel photographs and 1080p high-definition video clips and (2) an iPhone 5S back-facing camera, recording 1080p video clips. There are three types of spoof attacks: (1) high-resolution replay video attacks using an iPad Air screen, with a resolution of 2048 × 1536, (2) mobile phone replay video attacks using an iPhone 5S screen, with a resolution of 1136 × 640, and (3) printed photo attacks using an A3 paper with fully occupied printed photo of the client's biometry, with a paper size of 11 × 17 (279 mm × 432 mm), printed by a HP Color Laserjet CP6015xh printer, with a printing resolution of 1200 × 600 dpi. To evaluate the

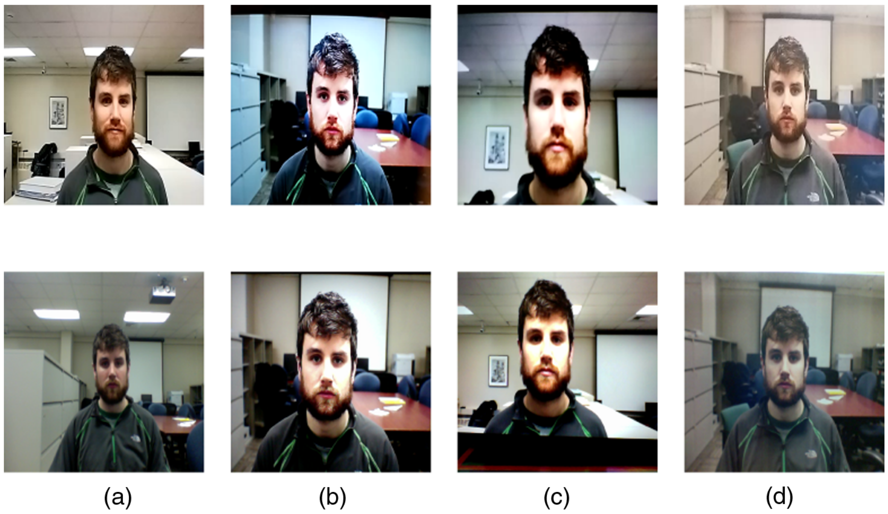

Fig. 9 Example images of genuine and spoof faces of one of the subjects in the MSU-MFSD captured using Google Nexus 5 smart phone camera (top row) and MacBook Air 13 in. laptop camera (bottom row). (a) Genuine faces, (b) spoof faces generated by iPad for video replay attack, (c) spoof faces generated by iPhone for video replay attack, and (d) spoof faces generated for printed photo attack.

performance, the 35 subjects of MSU-MFSD were divided into two subsets, 15 subjects for training and 20 subjects for testing.

5.2 *Experimental Setup*

In our experiments, after the face preprocessing step, we normalized the cropped faces to 128×128 pixels. Then the FD was computed before dividing the face into multiple blocks with ML representation using level 8. After that, on each block, we extracted features using three descriptors (LBP, LPQ, and BSIF). For the LBP, we used a uniform pattern descriptor with neighborhood $P = 8$ and a radius $R = 1$. For LPQ, we used a window of size (9×9) and a Gaussian derivative quadrature filter for local frequency estimation. For the last descriptor (BSIF), we used eight filters of size (11×11). The features extracted from all blocks of one image of FD were concatenated to get one feature vector. We computed the average of concatenated feature vectors of the first 149 frames. Once the enhanced histograms were computed on each video, we ranked them by FS. Finally, we used the Lib-SVM classifier[51] with fivefold cross-validation to determine whether the input video corresponds to a real person or not. The Lib-SVM classifier is trained using the training set of each database.

For CASIA and MSU databases, the performance is reported in terms of EER, while, for the Replay-Attack database, results are presented in terms of half-total error rate (HTER), which is the mean of the false acceptance rate and false rejection rate at the threshold, which corresponds to the EER of the development set.

5.3 *Experimental Results*

In this part, we will show and discuss the effectiveness of our proposed framework, focusing on the effect of the FDs, ML representation, and the FS on face antispoofing.

5.3.1 *Effectiveness and choosing hyperparameters*

In this section, we are discussing the effectiveness of choosing the FD associated with ML representation. In this context, first, we study the effect of the number of frames, and then we discover the superiority of ML compared with MB representation. Finally, we justify our choice of FD-ML by showing the results of the CASIA database.

In Table 4, we tested the performance on CASIA-FASD with respect to different time window sizes. Especially, we remarked that the average of 6 s (150 frames) gives a better result, knowing that the video sequences in the CASIA-FASD can reach 10 s. We observe that, when using a sufficient number of frames (up to 150), the motion in fake faces will be detected easily.

Figure 10 shows a comparison between representations (ML and MB), texture descriptors (LBP, LPQ, and BSIF), and FD using different levels. The EER is presented as a function using different levels. We observe from the figure that the performance of ML is better than MB when using the same descriptor. This is because the ML representation gives more detailed information of the image than the MB. Also, we observe in this figure that, when applying the ML or MB on the FD image, the performance is improved compared with using them on the gray image directly. We find that the performance of the combination FD + ML is the best among the other combinations. In our tests, we used three descriptors to compare their performances with the FD and the ML representation. We observe from the same figure that the LPQ feature extractor in our system gives the best result, which is perhaps a consequence of the blur-invariant property of LPQ. Also, we observe that there is variation in EER according to the number of the level. From level 2 to level 8, the performance is improved progressively; when we reach level 8, there is no great change in EER because each level takes the information of the previous levels and the actual level, so it represents more features. Based on the previous analyses, we choose to use FD with ML representation in 8 levels. We used also the FS to rank the obtained features, which improved the results, as is highlighted later. Next, we will compare our results with the state of the art on the CASIA-FASD. This is summarized in Table 5 and Fig. 11.

Table 4 Effect of different time window sizes on CASIA-FAS database.

Frames	EER (%)	Frames	EER (%)
5	38.59	100	26.60
10	29.23	125	25.08
15	31.04	150	**23.15**
25	29.18	175	23.37
50	25.91	200	23.77
75	25.91	225	23.70

Note: Bold font represents the best result.

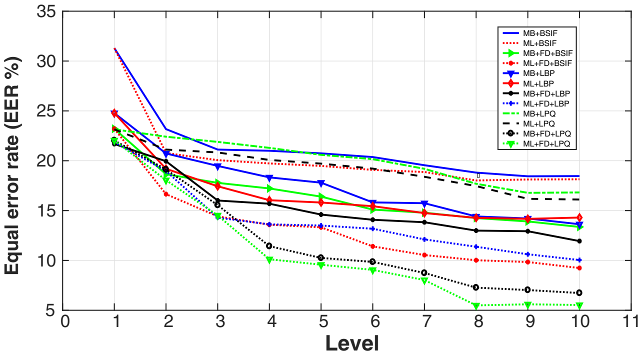

Fig. 10 Comparison number level of face representation and FD on CASIA-FAS.

To follow the official test protocol of CASIA-FAS, we computed the EERs for the seven scenarios, including different qualities and media. The included quality descriptors are low-, normal-, and high-quality image sequences, and the used media for spoofing attacks are warped photos, cut photos, and videos played on an iPad. The last scenario is the overall test. In this section, we will analyze the effects of image quality descriptors and spoofing media on the system performance. Furthermore, we observed that the proposed approach improves the results. The results of our approach using different descriptors are given in Fig. 11 and Table 5.

Table 5 Comparison between the proposed approach and the state-of-the-art methods on different scenarios on CASIA-FAS database.

	Scenarios						
Methods	Low	Normal	High	Warped	Cut	Video	Overall
IQA[20]	31.70	22.20	05.60	26.10	18.30	34.40	32.40
Difference of Gaussian (DoG) baseline [52]	13.00	13.00	26.00	16.00	06.00	24.00	17.00
Visual codebooks[28]	10.00	17.78	13.33	07.78	22.22	08.89	14.07
LBP-overlapping + Fisher[53]	07.20	08.80	14.40	12.00	10.00	14.70	13.10
CDD[1]	**01.50**	**05.00**	02.80	06.40	04.70	**00.30**	11.80
ML-LPQ Fisher[43]	12.49	08.96	05.22	13.62	09.66	10.10	11.39
LBP-TOP[2]	10.00	12.00	13.00	06.00	12.00	10.00	10.00
Kernel Fusion[30]	00.70	08.70	13.00	**01.40**	10.10	04.30	07.20
YCbCr + HSV-LBP[31]	07.80	10.10	06.40	07.50	05.40	08.10	06.20
FD-ML-LBP-FS (ours)	05.94	11.02	07.52	08.08	04.45	13.55	10.29
FD-ML-BSIF-FS (ours)	07.93	11.85	12.42	05.85	03.11	15.84	09.96
FD-ML-LPQ-FS (ours)	05.44	08.62	**01.62**	04.71	**01.93**	08.56	**04.62**

Note: Bold fonts represent the best result.

From Fig. 11(a), we observe that the LPQ descriptor combined with FD-ML gives the best results for the different images qualities (low, normal, and high), as well as with spoof media (warped photo, cut photo, and video attacks) [see Fig. 11(b)]. This can be explained by the fact that LPQ works well even in the presence of noise of motion on both real and fake faces compared with LBP and BSIF.

We see from Table 5 that our proposed approach gives better results in all scenario compared with CASIA baseline,[52] whom created the database. Also, we obtained the best results in high quality and cut photos in comparison with the state of the art in the same database. In the case of high quality, our approach can effectively detect the spoof attack because the FD in the case of real faces keeps more information than other qualities (see Fig. 2). Unlike[2] our approach, ML-FD can easily distinguish the cut and real photo because the eye region in the cut photo appears well when using FD. This proves the effectiveness of our approach in the face antispoofing CASIA database. In the following, we will compare our overall results with the state of the art on three challenge databases: Replay-Attack, MSU-MFS, and CASIA-FAS.

We present another experiment about the effectiveness of extracting the texture images using ML representation (8 levels). We see in Table 6 that dividing the whole image on ML improves the robustness of the three descriptors compared with using the whole image. We observe that the use of ML gives better results on CASIA-FAS, MSU-MFS, and Replay-Attack databases. Using ML representation, the EER on CASIA-FASD and MSU-MFSD has been reduced from

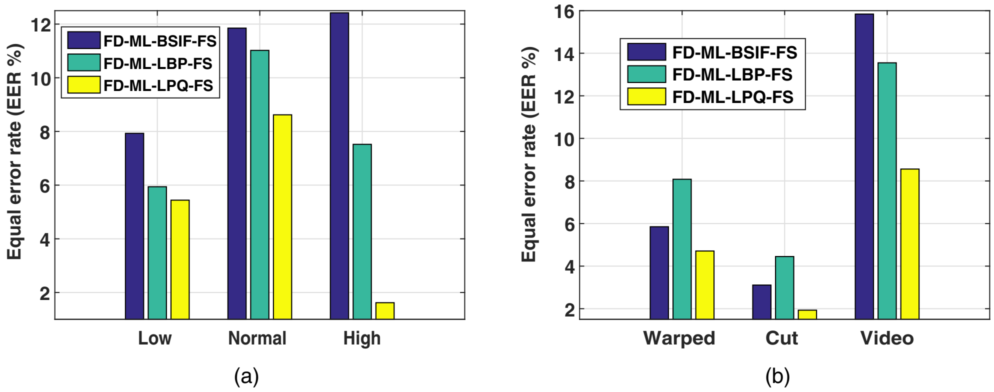

Fig. 11 Effect of quality and spoofing media on the performance on the CASIA-FASD: (a) quality and (b) spoofing media.

Table 6 Effect of the ML on the performance of CASIA, Replay-Attack, and MSU databases.

Method	CASIA (EER%)	MSU (EER%)	Replay (HTER%)
BSIF	31.30	30.33	23.00
ML-BSIF	18.02	21.85	20.25
FD-ML-BSIF	10.02	08.07	11.66
LBP	24.78	22.12	12.00
ML-LBP	14.27	20.04	09.62
FD-ML-LBP	11.37	07.15	09.70
LPQ	23.15	23.22	15.12
ML-LPQ	17.46	14.90	12.25
FD-ML-LPQ	**05.49**	**04.80**	**05.75**

Note: Bold fonts represent the best result.

23.15% to 17.46% and from 23.22% to 14.90%, respectively. The HTER on the Replay-Attack database has also been reduced from 15.12% to 12.25%.

Also, we conduct an experiment about the effect of the FDs on the performance of the ML approach. Table 6 shows that applying the ML approaches on the FDs improves the performance on the three databases. When using FD-ML-LPQ, the performance improvement on CASIA-FAS, MSU-MFS, and Replay-Attack databases are 68.55%, 67.78%, and 53.06%, respectively (see Table 6).

Table 7 shows the effect of features selection on the classification performances. We observe from this table that using the FS method with the FD-ML-LPQ method improves the performance on CASIA-FAS, MSU-MFS, and Replay-Attack databases with 15.84%, 47.91%, and 16.52%, respectively.

5.3.2 *Comparison with the state of the art*

Tables 5 and 8 present the comparison of our approach with the state of the art in face antispoofing techniques. In Table 5, we compared only on CASIA-FASD with different scenarios, which are low, normal, high, warped, cut, video, and overall test. As we see, our approach gives the best result on the high, cut, and overall scenarios. In Table 8, we observe that our proposed approach (FD-ML-LPQ-FS) gives good results compared with the state of the art on CASIA-FAS and MSU-MFS databases. The EER on CASIA, MSU, and REPLAY databases is 04.62%, 02.50%, and 5.62%, respectively [see detection error tradeoff (DET) curve in Fig. 12]. In the case of Replay-Attack database, our method shows interesting results compared with the other methods.

Table 7 Effect of the features selection on the performance of CASIA, Replay-Attack, and MSU databases.

Method	CASIA (EER%)	MSU (EER%)	Replay (HTER%)
FD + ML-LBP	11.37	07.15	09.70
FD + ML-BSIF	10.02	08.07	11.66
FD + ML-LPQ	05.49	04.80	05.75
FD-ML-LBP-FS	10.29	06.61	08.70
FD-ML-BSIF-FS	09.96	06.14	10.41
FD-ML-LPQ-FS	**04.62**	**02.50**	**04.80**

Note: Bold fonts represent the best result.

Table 8 Comparison between the proposed countermeasure and the state-of-the-art methods on the three benchmark datasets.

	CASIA	MSU	Replay-Attack	
Method	EER%	EER%	EER%	HTER%
IQA[20]	32.40	—	—	15.20
DMD[29]	21.75	—	05.30	03.75
LBP[15]	18.21	—	13.90	13.87
DoG baseline[52]	17.00	—	—	—
Spectral cubes[28]	14.07	—	—	02.75
LBP-overl + Fisher[53]	13.10	—	—	—
IDA[27]	12.90	08.58	—	07.41
CDD[1]	11.80	—	—	—
ML-LPQ Fisher[43]	11.39	—	—	—
LBP-TOP[2]	10.00	—	07.90	07.60
CNN[54]	07.40	—	06.10	**02.10**
Motion + LBP[19]	—	—	04.50	05.11
Color-LBP[31]	06.20	—	00.40	02.90
Bottleneck feature fusion + NN[55]	05.83	—	**00.83**	**00.00**
FD-ML-LPQ-FS (proposed)	**04.62**	**02.50**	05.62	04.80

Note: Bold fonts represent the best result.

5.3.3 *Cross-database analysis*

To gain insight into the generalization capabilities of our proposed method, we conducted a cross-database evaluation. To be clear, cross-database is a technique in which we trained and tuned on one database and tested on another database. There are different techniques for analysis, where training and testing occur in distinct databases. In our paper, we follow the cross-database used in these papers. [22,28,54,56] In these experiments, the countermeasure was trained and tuned with one database each time (CASIA-FAS, MSUMFS, or Replay-Attack) and then tested on the other databases. The results are reported in Table 9.

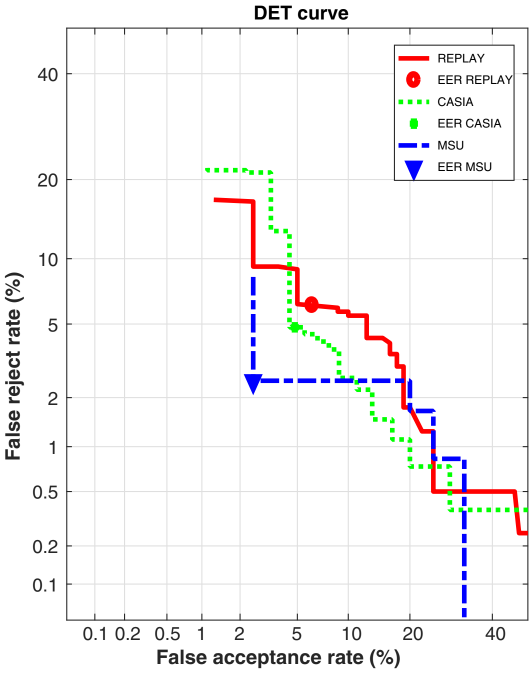

Fig. 12 DET curve of the proposed approach on REPLAY, CASIA, and MSU databases.

Table 9 The performance of the cross-database evaluation in terms of HTER(%) on the CASIA-FAS, MSU-MFS, and REPLAY-ATTACK.

Test on:	CASIA		MSU		Replay-Attack		
Train on:	Train	Test	Train	Test	Train	Dev	Test
CASIA	—	—	51.11	50.41	53.16	51.83	50.25
MSU	47.29	50.00	—	—	46.16	47.83	48.00
Replay-Attack	43.05	42.59	35.00	38.00	—	—	—

From Table 9, we can see that, in the case where the model is trained and tuned on CASIA database, then evaluated on the other databases, that the average of performance in terms of HTER for the train, development, and test sets on the Replay-Attack database is 51.74% and the average of performance for the train and the test sets on MSU database, is 50.76%. When the model is trained and tuned on the MSU database and evaluated on the other databases, the average of performance in terms of HTER for the train, development, and test sets on the Replay-Attack database is 47.33%; on the CASIA database, the average performance for train and test sets is 48.64%. Finally, the model is trained and tuned on Replay-Attack database then evaluated on the other databases; the average of performance in terms of HTER for train and test sets on the CASIA database is 42.82% and on the MSU database is 36.50%. As we observe in Table 9, the models trained on Replay-Attack and MSU-MFS databases are better than the model trained on CASIA-FASD. The reason why CASIA-FASD is not good as train set compared with the other databases on face antispoofing is because it has different qualities and attacks. In Table 10, we present the results of our proposed approach compared with the state-of-the-art techniques on cross-database. We observe in Table 10 that, when we use ML and FD, the performance is affected on face antispoofing methods, especially on cross-database compared with the state of the art.

Table 10 The results of the cross-database experiment on the CASIA-FAS, Replay-Attack, and MSU-MFS database compared with related studies.

Method	Train	Test	HTER %
Motion[56]	CASIA	Replay	50.20
	Replay	CASIA	47.90
LBP[56]	CASIA	Replay	45.90
	Replay	CASIA	57.60
LBP-TOP[56]	CASIA	Replay	49.70
	Replay	CASIA	60.60
Motion-Mag[22]	CASIA	Replay	50.10
	Replay	CASIA	47.00
Spectral cubes[28]	CASIA	Replay	**34.40**
	Replay	CASIA	50.00
CNN[54]	CASIA	Replay	48.50
	Replay	CASIA	45.50
Proposed	CASIA	Replay	50.25
		MSU	**50.41**
	Replay	CASIA	**42.59**
		MSU	**38.00**
	MSU	CASIA	**50.00**
		Replay	**48.00**

Note: Bold fonts represent the best result.

6 Conclusion and Future Work

In this paper, we proposed a face antispoofing approach to distinguish between real and fake faces. The proposed approach is based on extracting ML features from the FDs and then applying the FS for feature ranking. Three texture descriptors are used to extract the features from FD-ML: LBP, LPQ, and BSIF. We prove that the FD can play an important role in detecting spoofing attacks.

We evaluated our approach of face antispoofing on three challenging databases (CASIA-FAS, MSU-MFS, and Replay-Attack databases). Our experimental results show the impact of the ML, the FD, and the features ranking techniques on enhancing the performance of the face antispoofing method. The results obtained using the LPQ features yield excellent results on the CASIA and MSU databases and competitive performance on the Replay-Attack database.

This is due to the ability of LPQ to tolerate blurriness better than most of the previous texture features. Unlike the other proposed methods that show good performances on some databases and degraded results on others, our method was able to achieve a stable performance on the three databases. The performance improvement is particularly significant for the print and warped photo; moreover, our result when we used cross-databases showed that the performance of our system was more stable compared with the state of the art.

As future work, we will test our proposed methods using different color image representations instead of the grayscale images. We will use other descriptors, such as SIFT and SURF, to test the effectiveness of the combination FD ML in antispoofing detection. Moreover, we envision the improvement of the face alignment process.

References

1. J. Yang et al., "Face liveness detection with component dependent descriptor," in *2013 Int. Conf. on Biometrics (ICB)* (2013).
2. T. de Freitas Pereira et al., "Face liveness detection using dynamic texture," *EURASIP J. Image Video Process.* **2014**(1), 1–15 (2014).
3. J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: a survey in face recognition," *IEEE Access* **2**, 1530–1552 (2014).
4. G. Pan et al., "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *IEEE 11th Int. Conf. on Computer Vision (ICCV 2007)*, pp. 1–8, IEEE (2007).
5. K. Kollreider et al., "Real-time face detection and motion analysis with application in liveness assessment," *IEEE Trans. Inf. Forensics Secur.* **2**(3), 548–558 (2007).
6. Z. Zhang et al., "Face liveness detection by learning multispectral reflectance distributions," in *IEEE Int. Conf. on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pp. 436–441, IEEE (2011).
7. J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using micro-texture analysis," in *Int. Joint Conf. on Biometrics (IJCB)*, pp. 1–7, IEEE (2011).
8. P. Michelassi and A. Rocha, "Face liveness detection under bad illumination conditions," in *IEEE Int. Conf. on Image Processing* (2011).
9. X. Tan et al., "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conf. on Computer Vision (ECCV 2010)*, pp. 504–517, Springer (2010).
10. P. Massimo, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE Int. Conf. on*, Vol. **4**, pp. 3099–3104, IEEE (2004).
11. Z. Zhang, "CASIA face anti-spoofing database," 2012, http://www.cbsr.ia.ac.cn/english/FaceAntiSpoofDatabases.asp.
12. C. Ivana, A. Anjos, and S. Marcel, "The replay-attack database," 2012, https://www.idiap.ch/dataset/replayattack.
13. D. Wen, H. Han, and A. K. Jain, "The MSU mobile face spoofing database (MFSD)," 2015, http://biometrics.cse.msu.edu/Publications/Databases/MSUMobileFaceSpoofing/index.htm.
14. B. Biggio et al., "Security evaluation of biometric authentication systems under real spoofing attacks," *IET Biom.* **1**(1), 11–24 (2012).
15. I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. of the Int. Conf. of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–7, IEEE (2012).
16. J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using texture and local shape analysis," *IET Biom.* **1**(1), 3–10 (2012).
17. N. Kose and J.-L. Dugelay, "Classification of captured and recaptured images to detect photograph spoofing," in *Int. Conf. on Informatics, Electronics & Vision (ICIEV 2012)*, pp. 1027–1032, IEEE (2012).
18. N. Erdogmus and S. Marcel, "Spoofing face recognition with 3D masks," *IEEE Trans. Inf. Forensics Secur.* **9**(7), 1084–1097 (2014).
19. J. Komulainen et al., "Complementary countermeasures for detecting scenic face spoofing attacks," in *Int. Conf. on Biometrics (ICB)*, pp. 1–7, IEEE (2013).
20. J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *22nd Int. Conf. on Pattern Recognition (ICPR)*, pp. 1173–1178, IEEE (2014).
21. J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition," *IEEE Trans. Image Process.* **23**(2), 710–724 (2014).
22. S. Bharadwaj et al., "Face anti-spoofing via motion magnification and multifeature videolet aggregation," https://repository.iiitd.edu.in/jspui/handle/123456789/138 (03 June 2014).
23. D. Menotti et al., "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 864–879 (2015).
24. D. C. Garcia and R. L. de Queiroz, "Face-spoofing 2D-detection based on moiré-pattern analysis," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 778–786 (2015).
25. J. Yang et al., "Person-specific face antispoofing with subject domain adaptation," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 797–809 (2015).
26. I. Chingovska and A. R. dos Anjos, "On the use of client identity information for face antispoofing," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 787–796 (2015).
27. D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 746–761 (2015).
28. A. Pinto et al., "Face spoofing detection through visual codebooks of spectral temporal cubes," *IEEE Trans. Image Process.* **24**(12), 4726–4740 (2015).
29. S. Tirunagari et al., "Detection of face spoofing using visual dynamics," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 762–777 (2015).
30. S. R. Arashloo, J. Kittler, and W. Christmas, "Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features," *IEEE Trans. Inf. Forensics Secur.* **10**(11), 2396–2407 (2015).
31. Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face anti-spoofing based on color texture analysis," in *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 2636–2640, IEEE (2015).
32. K. Patel et al., "Live face video vs. spoof face video: use of moiré patterns to detect replay video attacks," in *Int. Conf. on Biometrics (ICB)*, pp. 98–105, IEEE (2015).
33. J. Galbally and R. Satta, "Three-dimensional and two-and-a-half-dimensional face recognition spoofing using three-dimensional printed models," *IET Biom.* **5**(2), 83–91 (2016).
34. E.-S. Ng and A. Y.-S. Chia, "Face verification using temporal affective cues," in *21st Int. Conf. on Pattern Recognition (ICPR)*, pp. 1249–1252, IEEE (2012).
35. I. Pavlidis and P. Symosek, "The imaging issue in an automatic face/disguise detection system," in *Proc. IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications*, pp. 15–24, IEEE (2000).
36. G. Chetty and M. Wagner, "Liveness verification in audio-video authentication," in *Proc. of the 10th Australian Int. Conf. on Speech Science and Technology (SST 2004)*, pp. 358–363 (2004).
37. N. Erdogmus and S. Marcel, "Spoofing attacks to 2D face recognition systems with 3D masks," in *Int. Conf. of the Biometrics Special Interest Group*, (EPFL-CONF-192407) (2013).
38. D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: reflections on replay attacks," *IEEE Trans. Inf. Forensics Secur.* **10**(4), 736–745 (2015).
39. T. Wang et al., "Face liveness detection using 3D structure recovered from a single camera," in *Int. Conf. on Biometrics (ICB)*, pp. 1–6, IEEE (2013).
40. D. Yi et al., "Face anti-spoofing: multi-spectral approach," in *Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks*, S. Marcel et al., Eds., pp. 83–102, Springer, London (2014).
41. Y. Kim et al., "Masked fake face detection using radiance measurements," *J. Opt. Soc. Am. A* **26**(4), 760–766 (2009).
42. F. J. Prokoski and R. B. Riedel, "Infrared identification of faces and body parts," in *Biometrics*, pp. 191–212, Springer (1996).
43. A. Benlamoudi et al., "Face spoofing detection using multi-level local phase quantization (ML-LPQ)," in *Proc. of the First Int. Conf. on Automatic Control, Telecommunication and signals ICATS15* (2015).
44. P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision* **57**(2), 137–154 (2004).
45. X. Tan et al., "Enhanced pictorial structures for precise eye localization under incontrolled conditions," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009)*, pp. 1621–1628, IEEE (2009).
46. S. Bekhouche et al., "Facial age estimation using BSIF and LBP," in *Proc. of the First Int. Conf. on Electrical Engineering (ICEEB 2014)* (2014).
47. S. E. Bekhouche et al., "Facial age estimation and gender classification using multi level local phase quantization," in *3rd Int. Conf. on Control, Engineering & Information Technology (CEIT 2015)*, pp. 1–4, IEEE (2015).
48. V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Image and Signal Processing*, pp. 236–243, Springer (2008).
49. J. Kannala and E. Rahtu, "BSIF: binarized statistical image features," in *21st Int. Conf. on Pattern Recognition (ICPR)*, pp. 1363–1366, IEEE (2012).
50. Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. Twenty-Seventh Conf. Uncertainty in Artificial Intelligence*, Barcelona, Spain, pp. 266–273, AUAI Press, Arlington, Virginia (2011).
51. C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.* **2**, 27 (2011).
52. Z. Zhang et al., "A face antispoofing database with diverse attacks," in *5th IAPR Int. Conf. on Biometrics (ICB)*, pp. 26–31, IEEE (2012).

53. A. Benlamoudi et al., "Face spoofing detection using local binary patterns and Fisher score," in *3rd Int. Conf. on Control, Engineering & Information Technology (CEIT)*, pp. 1–5, IEEE (2015).

54. J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," arXiv preprint arXiv:1408.5601 (2014).

55. L. Feng et al., "Integration of image quality and motion cues for face anti-spoofing: a neural network approach," *J. Visual Commun. Image Represent.* **38**, 451–460 (2016).

56. T. de Freitas Pereira et al., "Can face anti-spoofing countermeasures work in a real world scenario?" in *Int. Conf. on Biometrics (ICB)*, pp. 1–8, IEEE (2013).

Azeddine Benlamoudi received his BS and MSc degrees from the University Mohamed Khider of Biskra, Algeria, in 2010 and 2012, respectively. Currently, he is a PhD student in the Department of Electronics and Telecommunications, the University Kasdi Marbah of Ouargla. His research interests include computer vision, pattern recognition, signal and image processing, biometrics, and spoofing detection.

Kamal Eddine Aiadi received his BS degree in physics from Batna University, Algeria, in 1983 and his master's degree from Bridgeport University, USA, in 1986. He received his PhD from Batna University, Algeria, in 2006. Since 1987, he has been a teacher at the University of Ouargla and a researcher in the optoelectronic field at the same university.

Abdelkrim Ouafi received his BEng degree in electronic in 1997 and his magister degree in 2001 from Biskra University, Algeria. He received his PhD in electronic engineering image processing from Biskra University in 2012, where he has been an associate professor since 2012. His research interest includes image processing, image coding, motion capture, and biometrics.

Djamel Samai is an associate professor in the Department of Electronics and Telecommunications, the University of Ouargla, Algeria. He received his PhD in signal processing from the University of Annaba, Algeria. His research interests are in signal and image processing, image compression, pattern recognition, and biometrics.

Mourad Oussalah received a PhD in multisensor fusion in robotics from the University of Evry Val Essonnes, France, in 1998. He took several postdoctoral positions in KU Leuven, City University of London, before joining Academic Team of University from 2003 until 2016. Since April 2016, he has been a senior research fellow, professor, and head of Social Mining Research Group at Centre for Ubiquitous Computing, University of Oulu, Finland. He is a researcher in data mining, computer vision, and information fusion.