MSU Technical Report MSU-CSE-15-15, Oct 21, 2015

Secure Smartphone Unlock: Robust Face Spoof Detection on Mobile

Keyurkumar Patel, *Student Member, IEEE,* Hu Han, *Member, IEEE,* and Anil K. Jain, *Fellow, IEEE*

***Abstract*— With the wide deployment of face recognition systems in applications from de-duplication to mobile device unlocking, security against face spoofing attacks requires increased attention; such attacks can be easily launched via printed photos, video replays and 3D masks of a face. We address the problem of facial spoof detection against print (photo) and replay (photo or video) attacks based on the analysis of image aliasing (*e.g.,* surface reflection, moiré pattern, color distortion, and shape deformation) in spoof face images (or video frames). The application domain of interest is mobile phone unlock, given that growing number of phones have face unlock and mobile payment capabilities. We build a mobile spoof face database (MSU MSF) containing more than $1,000$ subjects, which is, to our knowledge, the largest spoof face database in terms of the number of subjects. Both print and replay attacks are captured using the front and rear cameras of a Nexus 5 phone. We analyze the aliasing of print and replay attacks using (i) different intensity channels (R, G, B and grayscale), (ii) different image regions (entire image, detected face, and facial component between the nose and chin), and (iii) different feature descriptors. We develop an efficient face spoof detection system on an Android smartphone. Experimental results on three public-domain face spoof databases (Idiap Print-Attack and Replay-Attack, and CASIA), and the MSU MSF show that the proposed approach is effective in face spoof detection for both cross-database and intra-database testing scenarios. User studies of our Android face spoof detection system involving 20 participants' show that the proposed approach works very well in real application scenarios.**

***Index Terms*— Face antispoofing, phone unlock, spoof detection on mobile, mobile spoof database, image aliasing**

I. Introduction

With the widespread use of smartphones, biometric authentication, such as face and fingerprint recognition, is becoming increasingly popular for confirming user identity. Two of the most popular mobile operating systems, Android and iOS, currently use face and fingerprint to authenticate users. With the release of Android 4.0 (Ice Cream Sandwich), Android allows users to unlock their smartphone via facial recognition (FR) technology; on all iPhones released after the iPhone 5c, iOS allows users to unlock their smartphone with their fingerprint (Touch ID). As the use of biometrics for smartphone unlocking and user authentication continues to increase, capabilities to detect spoof biometric attacks are needed to alleviate fraud and user concerns. Spoof biometric attacks launched against a smartphone's authentication system may allow malicious users to gain access to the smartphone, and therefore lead to the leakage of sensitive private data such as bank information via apps like Google Wallet and Apple Pay.

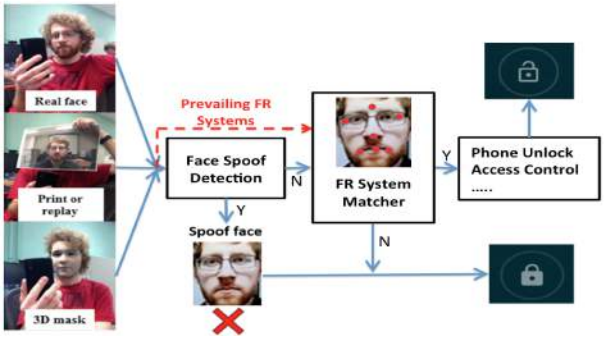


Fig. 1. A face recognition (FR) system with a spoofing detection module. Many FR systems either do not currently include this module or this module does not perform effectively.

Given the prevalence of high resolution face images shared, (often publicly) through social media, it is relatively easy to obtain a face image of a user and launch a spoof attack against FR systems (see Fig. 1). Compared to attacks against fingerprint, iris or speech recognition systems, the ubiquitous nature of image acquisition devices, such as cameras and smartphones, allows attackers to acquire facial images of a user easily and discretely [26]–[34].

A recent study of face recognition using a commercial off-the-shelf (COTS) matcher shows that the state-of-the-art face matchers are fragile against face spoof attacks [18], [35]. Spoof attacks against FR systems mainly consist of (i) print attacks, (ii) replay attacks, and (iii) 3D mask attacks. Print and replay attacks are 2D face spoof attacks, that can be launched using a smartphone to obtain a photograph or video of the target subject's face. By contrast, 3D face mask attacks require high resolution fabrication systems capturing the 3D shape and texture information of the target subject's face. Therefore, print and replay attacks can be more easily launched by malicious individuals than 3D mask attacks. For this reason, we focus on 2D face spoof attacks, such as printed photos, displayed photos, and video replays.

Significant progresses have been achieved in biometric spoofing detection technologies of individual modalities in recent years [36], [37]. As shown in Table I, the state-of-the-art face spoof detection methods were able to achieve less than 2% HTER on the public-domain Idiap Replay-attack database using an intra-database testing protocol. Despite recent progress [1], [7], [9], [12], [17], [18], [21], [25], [38], face spoof detection remains a difficult problem that requires continued efforts. A number of the published methods on face

K. Patel, H. Han and A. K. Jain are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA. E-mail: {patelke6, hhan, jain}@msu.edu.

An earlier version of this paper appeared in the Proceedings of the 8th IAPR International Conference on Biometrics (ICB), Phuket, 2015 [1].

TABLE I

A SUMMARY OF PUBLISHED METHODS ON 2D FACE SPOOF DETECTION.

Method	**Strength**	**Limitation**	**State of the art performance (HTER)†**
Face motion analysis [2]–[5]	Effective for print attack	Requires multiple frames	**ZJU Eyeblink** [3] (Intra-DB, Cross-DB) [3]: (95.7%, n/a)‡ **Idiap Replay-attack** [6] (Intra-DB, Cross-DB) [2]: (1.25%, n/a)
Face texture analysis [7]–[11]	Relatively low computational cost and fast response	Poor generalizability, Requires face and/or landmark detection	**Idiap Replay-attack** (Intra-DB, Cross-DB) (5.11% [12], 47.1% [10]) **CASIA** [13] (Intra-DB, Cross-DB) (11.8% (EER) [11], 48.3% [10])
Face 3D shape or depth analysis [8], [14]–[16]	Effective for 2D attacks	Requires multiple frames or additional devices	**Private database** [14] (Intra-DB, Cross-DB) [14]: (85%, 50.0%) **Idiap Replay-attack** [16] (Intra-DB, Cross-DB) [16]: (12.5%, n/a)
Image quality analysis [17]–[19]	Good generalizability, Low computational cost, Fast response time, Face and/or landmark detection not required	Image quality measures can be device dependent	**Idiap Replay-attack** (Intra-DB, Cross-DB) [17]: (15.2%, n/a) **CASIA** (Intra-DB, Cross-DB) [18]: (6.7% (EER), n/a) **MSU MFSD** [18] (Intra-DB, Cross-DB) [18]: (5.8% , 11.4%)
Frequency domain analysis [9], [20], [21]	Good generalization ability, Low computational cost	Spectral features can be device dependent	**Idiap Replay-attack** (Intra-DB, Cross-DB) [21]: (2.8%, 34.4%) **CASIA** (Intra-DB, Cross-DB) [21]: (14.0%, 38.5%) **3DMAD** [22] (Intra-DB, Cross-DB) [21]: (8.0%, 44.0%) **UVAD** [21] (Intra-DB, Cross-DB) [21]: (29.9%, 40.1%)
Active approach [23]	Good generalizability	Requires additional devices	**Private photo dataset** [23] (Intra-DB, Cross-DB) [23]: (92.2%, n/a)‡ **Private 3D mask dataset** [23] (Intra-DB, Cross-DB) [23]: (100.0%, n/a)‡
Multi-clue fusion [24], [25], [**Proposed**]	Good generalizability, Less sensitive to face and/or landmark detection errors, Whole image frame analysis	Moderate computational cost (0.47 sec. on desktop)	**Idiap Print-attack** (Intra-DB, Cross-DB) Proposed*: (0.5%, 50.0%) **Idiap Replay-attack** (Intra-DB, Cross-DB) Proposed*: (0.26%, 4.5%) **CASIA** (Intra-DB, Cross-DB) Proposed*: (0.0%, 2.5%) **RAFS** (Intra-DB, Cross-DB) Proposed*: (0.1%, 9.5%) **MSU MSF** (Intra-DB, Cross-DB) Proposed*: (6.25%, n/a)

†Half Total Error Rate (HTER) is defined as the average of false acceptance rate and false rejection rate; at the Equal Error Rate (EER) point where false acceptance rate equals false rejection rate, the HTER equals EER. The HTERs of the published methods in this table are from the original papers.
‡Classification accuracy was reported. *We used a five-fold, subject-exclusive cross validation protocol, and no reject option was used.

spoof detection are designed to safe guard the FR system [25] against one type of attack, and thus lack good generalizability to different face spoof attacks and application scenarios. Additionally, most of the published methods on face spoof detection are based on databases in which the spoof videos were captured using either low resolution (*e.g.,* webcam) or very high-resolution (*e.g.,* DLSR) cameras [3], [6] (*e.g.,* the CASIA and Idiap databases released in 2012). Therefore, these face spoof databases are not representative of mobile phone unlock scenarios. While face spoof detection under the mobile phone unlock scenarios was studied in [18], the mobile face spoof database used in [18] contained only 50 subjects (images of only 35 subjects are publicly available). Additionally, results based on a face spoof detection system running on a mobile platform were not reported.

In this paper, we study the problem of *face spoof detection on mobile phones* using a large mobile spoof face database, and provide a prototype face spoof detection system running on Android. This paper expands upon our preliminary work [1] in the following ways:

- Collection of a large mobile spoof face database (MSU MSF) with diverse 2D face spoof attacks (printed photos, displayed photos, and video replays) from more than 1,000 subjects to replicate the scenario of smartphones unlock.[1]
- A new feature representation method for face liveness detection by considering the complementarity between different feature clues, and study of possible reject options in face liveness detection.
- Verification of the conclusions drawn in [1] by using the MSU MSF database and the inclusion of several new experiments based on this new database.
- Leading edge spoof detection performance for cross-database testing scenarios.[2]
- Implementation of the proposed method on Android smartphones, and tests in real application scenarios.

[1] A 10k image portion of the MSF database (where subjects have given approval) will be made available to interested researchers.

[2] Cross-database testing involves, training on database A and testing on a different database B, collected in a different setting from database A and with different subjects. This is in contrast to the easier, but, not realistic protocol of intra-database testing where, cross-validation is used on a specific database.

The remainder of the paper is organized as follows. In Section II, we briefly review published methods, and face spoof databases for 2D face spoof detection. We detail the analysis of image aliasing in 2D spoof face images, and the proposed face spoof detection approach on mobile in Sections III and IV, respectively. Experimental setup, protocols, and results are given in Section V. Finally, we conclude this work in Section VI.

II. Related Work

A. Literature Review

As summarized in [37], studies of face spoofing detection can date back over 15 years. Since then, a number of methods have been proposed for face spoofing detection under print attacks [3], [5], [7], [9], replay attacks [2], [8], [13], and 3D mask attacks [39]. Since our focus is 2D face spoof attack detection (on mobile), we provide a brief summary and analysis of published 2D face spoof detection methods. Table I groups the published 2D face spoof detection methods into five categories: (i) face motion analysis based methods, (ii) face texture analysis based methods, (iii) face 3D depth analysis based methods, (iv) image quality analysis based methods, (v) frequency domain analysis based methods and (vi) active methods.

Face motion analysis based spoofing detection methods extract behavioral characteristics of the face, such as eye blink [3], and lip or head movement [2]. These methods require accurate face and landmark detection to localize the facial components. Additionally, multiple frames must be used in order to estimate the movement. These methods are designed to detect print attacks, and thus are not able to handle video replay attacks.

Face texture analysis based spoofing detection methods capture the texture differences (due to the different reflection properties of live face and spoof material) between face images captured from live faces and face images captured from various spoof medium (*e.g.,* paper and screen) [6], [7], [43]. These methods can perform spoof detection based on a single face image, and thus have relatively fast response. However, face texture analysis based methods may have poor generalizability when using small training sets with a limited number of subjects and spoofing scenarios.

Face 3D depth analysis based spoofing detection methods estimate the 3D depth of a face to discriminant between 3D live face and 2D spoof face [5], [8]. While live faces are 3D objects, spoof faces presented on 2D planar medium are 2D. Thus, these methods can be quite effective to identify 2D face spoof attacks if the 3D depth information of a face can be reliably estimated. Face 3D depth analysis based methods usually rely on multiple frames to estimate the depth or 3D shape information of face.

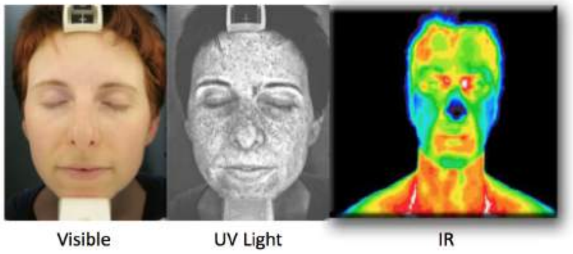


Fig. 2. Face images of a subject under the visible light (RGB) spectrum, ultraviolet light (UV) spectrum, and the infrared light (IR) spectrum.

Image quality analysis based spoofing detection methods analyze the image quality differences between live face images and spoof face images [17], [18], [44]. Since the spoof face images and videos are generated by recapturing live face images and videos in photographs or screens, there will be degradations of color, reflection, and blurriness in the spoof face images compared to the live face images and videos. These methods have been found to have good generalization ability to different scenarios [17]. However, studies on face spoofing detection based on image quality analysis are limited.

Frequency domain based spoofing methods analyze noise signals in recaptured video to distinguish between live and spoof face access [9], [20], [21]. During the recapture of printed photos or video replays, there is a decrease of low frequency components, and an increase of high frequency components. In order to quantize these signal changes, the input is usually transformed into the frequency domain using a Fourier Transform.

Active methods utilize additional sensors to capture modalities, such as near-infrared (NIR) and 3D depth of a face besides the 2D visual face image (see Fig. 2[3]) [22], [23]. These methods benefit from the information contained in the additional modalities, and provide better robustness against illumination and pose variations of the face. However, the use of additional sensors also limit the application scope of these approaches, particularly in mobile phone scenarios.

While many of the published methods belonging to the above five categories reported favorable results for intra-database testing, they did not show their method's effectiveness in cross-database testing scenarios, which are more representative of real applications. The few publications that do report cross-database testing tend to report poor results [10], [18], [21]. One plausible approach to improve the robustness of face spoof detection methods under cross-database testing scenarios, is to consider fusion of multiple physiological or behavioral clues [25].

B. 2D Spoof Face Databases

1) Public-domain Databases: In this section, we review the commonly used public-domain 2D face spoof databases in terms of their collection process and their limitations. Additionally, we discuss the database we collected that contains diverse 2D face spoof attacks from a large number of subjects.

The Print-attack and Replay-attack databases are both available from Idiap. While Print-Attack consists of only 2D face spoof attacks of printed photos from 50 subjects, Replay-Attack consists of photo and video replay attacks from 50 subjects. Live face videos of subjects were captured using the webcam on a MacBook. Replay attacks for each subject were captured using a Cannon PowerShot SX 150 IS camera

[3] www.cnet.com/news/uv-photography-reveals-our-sun-damaged-selves/

TABLE II

A SUMMARY OF PUBLIC-DOMAIN 2D FACE SPOOF DATABASES.

Database	# Subs.	# Images or Videos (Live, spoof)	Live face acqusition device	Spoof medium	Spoof face acquisition device	Subject race
NUAA [40]	15	(5105, 7509)	Webcam (640 × 480)	A4 paper	Webcam (640 × 480)	Asian
ZJU Eyeblink [41]	20	(80, 100)	Webcam (320 × 240)	High-quality photo	Webcam (320 × 240)	Asian
Idiap Print-attack [42]	50	(200, 200)	MacBook Webcam (320 × 240)	A4 paper	MacBook Webcam (320 × 240)	Mainly Caucasian
Idiap Replay-attack [6]	50	(200, 1000)	MacBook webcam (320 × 240)	iPad 1 (1024 × 768) iPhone 3GS (480 × 320)	Cannon PowerShot SX 150 IS (1280 × 720)	Caucasian 76%, Asian 22%, African 2%
CASIA [13]	50	(200, 450)	Sony NEX-5 (1280x720) USB camera (640x480)	iPad 1 (1024x768)	Sony NEX-5 (1280x720) Webcam (640 × 480)	Asian 100%
MSU RAFS† [1]	55	(55, 110)	Nexus 5 (frontal: 720 × 480) (rear: 1920 × 1080)	MacBook (1280 × 800)	iPhone 6 (rear: 1920 × 1080) Nexus 5 (rear: 1920 × 1080)	Caucasian 44%, Asian 53%, African 3%
UVAD [20], [21]	404	(808, 16, 268)	Six different cameras (no mobile phone) (1366 × 768)	Seven display devices	Six different cameras (no mobile phone) (1366 × 768)	Caucasian 44%, Asian 53%, African 3%
MSU MSF (this paper) To be made public	1,140	(1,140, 9,120)	Nexus 5 (frontal: 720 × 480) (rear: 3264 × 2448) Cameras used to capture celebrity photos	MacBook (2880 × 1800) Desktop (1280 × 800) Tablet (1920 × 1200) Printed photo on 11 × 8.5 in. paper	Nexus 5 (frontal: 1280 × 960) (rear: 3264 × 2448)	Diverse Set

†Contains an additional 200 spoof videos, 2 videos per subject from the Replay-Attack and CASIA databases.

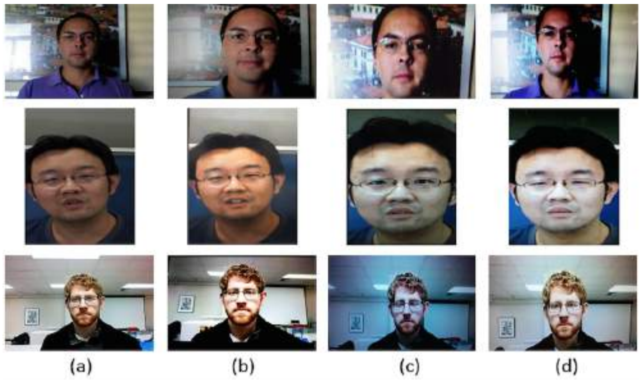


Fig. 3. Sample images of live and spoof faces from Idiap Replay-attack (top), CASIA (middle) and MSU RAFS (bottom) databases. (a) Live faces; (b) Original spoof faces; (c) Spoof faces generated by Google Nexus 5 using a MacBook for replay; (d) Spoof faces generated by iPhone 6 using a MacBook for replay.

that records 720p video clips. The high-resolution camera captured replay attacks displayed on an iPhone 3GS (480×320 resolution) and iPad I (1024 × 768 resolution).

The CASIA Face Anti-Spoofing Database consists of 600 video clips of 50 subjects [13]. Out of the 600 video clips, 150 clips represent video replay attacks. Compared to the Idiap database, the CASIA DB used a variety of cameras (Sony NEX-5-HD, two low quality USB) to capture replay attacks displayed on an iPad.

A key limitation of both the Idiap and CASIA databases is that they capture replay video attacks using either low-resolution cameras and spoof mediums that are now obsolete or DSLR cameras that are expensive. Low quality webcams often lack autofocus capability or have relatively slow autofocus speed. Because of these reasons, webcams often capture blurry images of a digital screen. Many DSLR cameras come equipped with anti-aliasing filters that sit immediately above the photo sensor (CCD array in most cameras) to reduce the occurrence of moiré patterns.[4] These filters reduce the sharpness of an image by smoothing the transitions between pixels, in turn reducing moiré patterns (but not completely eliminating them). These two types of cameras also do not replicate the real application scenarios of interest, namely user authentication on smartphones.

Smartphones that are equipped with FR systems will capture replay attacks using their built-in cameras instead of an external camera. In [18], a database named Mobile Face Spoofing Database (MFSD) was collected to study the effects of using such videos or images for spoof attacks against smartphones. However, MFSD contains only 280 video clips of photo and video attacks from 35 subjects.

2) MSU Mobile Spoof Face (MSF) Database: In [1], we collected a replay attack database for smartphones with 465 videos from 155 subjects. Of these 465 videos, 155 videos were live face videos, and the remaining 310 videos were spoof face videos which were captured by showing the live face videos on a MacBook screen (1280 × 800), and recapturing the face videos using the built-in rear camera of Google Nexus 5 and built-in rear camera of iPhone 6[5], respectively.[6] The average standoff of the smartphone camera from the screen of the MacBook was 15 cm to ensure that replay videos did not contain the bezels (edges) of the MacBook screen.

In this work, we have significantly increased the number of subjects (1,000+ subjects) as well as the number of live face and spoof images (13,000) in the MSU MSF database. Current public-domain spoof databases often lack diversity in terms of background, illumination, and image quality. The MSU MSF database was specifically created to ensure that it contains a mixture of environments, image qualities, image capture devices and subject diversity. This is essential to obtain generalizable and robust antispoofing methods. Additionally, running evaluations on a large database of this size will provide statistically significant results for predicting real world performance.

[4] www.lifepixel.com/blog/anti-aliasing-low-pass-filter-removal

[5] Nexus 5 spec.: https://en.wikipedia.org/wiki/Nexus_5, iPhone 6 spec.: https://en.wikipedia.org/wiki/IPhone_6

[6] Videos were not deliberately captured to include moiré patterns; only a single attempt was made to capture the video.

Fig. 4. Sample images of live faces included in the MSU MSF database.

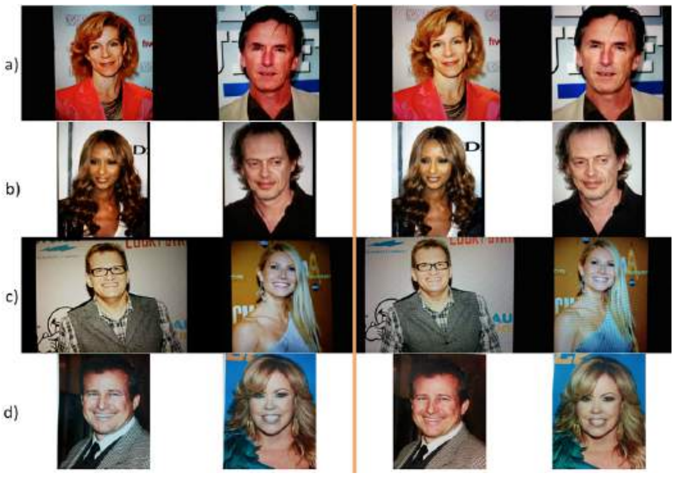


Fig. 5. Sample images of spoof faces from the MSU MSF database. Images on the left were captured using the front camera, and images on the right were captured using the rear camera on the Nexus 5. a) Replay attack on MacBook, b) Replay attack on Nexus 5, c) Reply attack on Nvidia shield tablet, and d) Printed photo attacks.

Two versions of the MSU MSF database were created, a 10K and 13K datasets. The 13K dataset contains images from subjects who withheld consent to allow us to share their face images with other researchers as well images from a private database that we used to supplement the live face images (2,818 additional images in which users withheld consent). However, both the 10K and 13K datasets contain the same number of spoof images (9,120 images). We will report performance on both datasets; the 10K MSU MSF database will allow interested researchers to verify our findings.

To create the MSU MSF database, we used a subset (1,000 subjects) of the web faces database collect in [45] to construct the new large spoof database. This database contains images of celebrities taken under a variety of backgrounds, illumination conditions and resolutions. We filtered the images to only contain a single fontal facing face (for mobile face unlock applications, it is reasonable to expect cooperative user scenario). The other 140 subjects are from the Idiap (50), CASIA (50) and the MSU RAFS (40) public databases. Thus, the new database contains color face images of 1,140 subjects, where the average resolution of the live subject images is 705×865.

We simulated spoof attacks for both replay attacks as well as printed photo attacks as these two types of attacks are relatively inexpensive to launch. In order to capture the spoof attacks, we used both the front (1280×960) and rear (3264×2448) facing cameras on the Google Nexus 5. This allows researchers to study how the quality of the spoof images affects spoof detection performance. Moreover, it allows researchers to examine the images to understand how camera quality affects image quality which in turn affects the presence of artifacts (*i.e.,* moiré patterns, reflections).

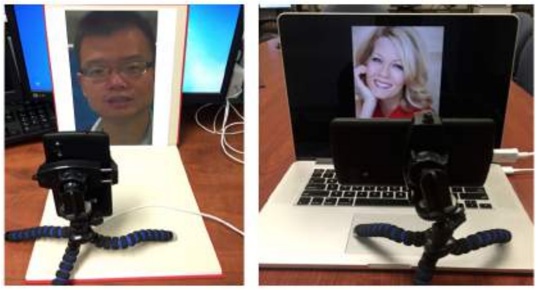


Fig. 6. Demonstration of how samples of (a) print attacks, and (b) display and replay attacks were collected, using paper and laptop screen as the spoof medium and a smartphone as an acquisition device. This simulates how a user may launch an attack against a FR system.

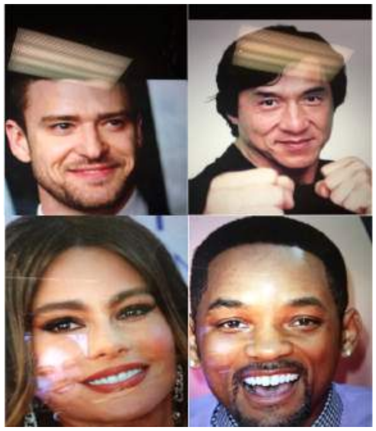


Fig. 7. Examples of spoof attacks launched using a digital screen show evidence of surface refection. The top row shows bright indoor lighting reflecting off a digital screen. The bottom row shows the screen of a mobile device reflecting the image it's capturing.

Given that most people have access to either a laptop, a mobile device or a tablet, we captured replay attacks on all three spoof mediums. The spoof attacks are captured by showing the live face image on the screen of one of the spoof mediums and using both the front and rear facing cameras of the Google Nexus 5 to capture the simulated attack. Therefore, the MSU MSF database contains 6,840 images of replay attacks captured using different camera quality and spoof mediums.

To capture printed photo attacks, we printed images of all 1,140 subjects using a HP Color Laserjet CP6015xh printer (1200×600dpi) on a 8.5×11 inch white paper. The live subject images were scaled to ensure the image covered as much of the computer paper as possible while maintaining the original image aspect ratio to minimize distortions. Additionally, we placed the photos in a manner to minimize reflection from ambient lighting inside our laboratory. Then we used both the cameras on the Google Nexus 5 to simulate printed photo attacks to a FR system. Thus, the MSU MSF contains 2,280 images of printed photo attacks. Figure 6 shows the setup used to capture both printed photo attacks and replay attacks for the MSU MSF database.

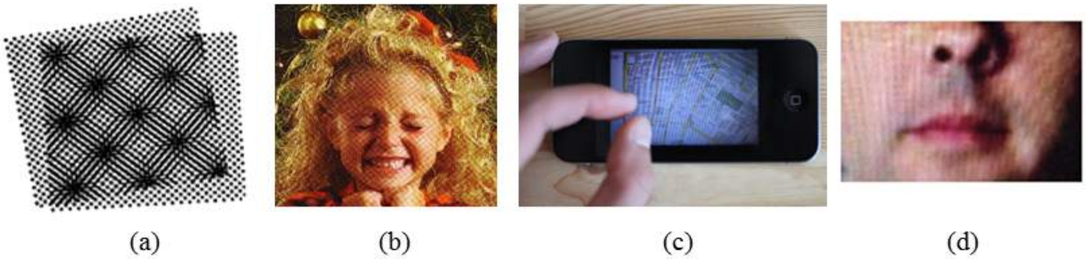


Fig. 8. Examples of moiré patterns. (a) an overlay of two patterns generates moiré patterns, (b) moiré patterns exist in color printing with halftoning, (c) moiré patterns appear while capturing the screen of digital devices, and (d) moiré patterns appeared in video replay attacks in the MSU MSF database we collected (We magnify the bottom portion of a face to show the moiré patterns more clearly).

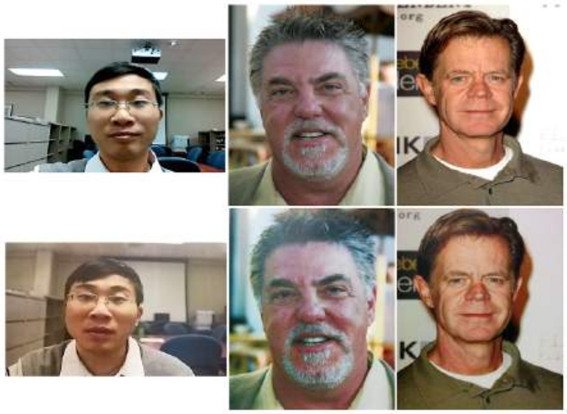


Fig. 9. Examples of spoof attacks reducing color diversity due to improper printing or rendering of live face images. Top row shows live face images whereas bottom row shows the corresponding spoof face images.

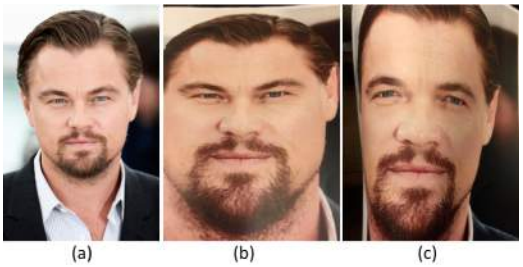


Fig. 10. a) Normal face image, b) skewed image captured by holding the camera closer to the bottom portion of the image than the top of the image, c) skewed image caused by the bending of the sides of an image.

III. Image Aliasing Analysis For 2D Spoof Face Images

Different types of image aliasing appear during the recapture of a face image or video, which generally include (1) surface reflection by the spoof medium, (2) moiré patterns, (3) color distortions, and (4) shape deformations.

A. Spoof Medium Surface Reflection

2D face spoofing attacks are mainly lunched by printing a face image or displaying a digital face image or video on a screen. Glossy photo papers and digital screens often generate specular reflections of the light, and lead to reflection aliasing in the spoof face images (see Fig. 7). Additionally, both paper and digital screens have different reflective properties than the skin of a face [19], which leads to reflectance differences between live and spoof face images.

B. Color Distortion

Color distribution may change during the recapturing of a face image, which leads to either reduced color diversity or color cast. For example, while the color distortion of printed attacks is due to the quality of the printer and photo paper, the color distortion of replay attacks is mainly caused by the fidelity and resolution of the screen [18]. Figure 9 shows the color distortion in spoof face images from three subjects in the MSU MSF database.

C. Moiré Pattern

Moiré patterns are an undesired aliasing of images caused by a overlap of digital grids [46]. Moiré patterns appear when two or more patterns are overlaid on top of each other, resulting in a third new pattern (Fig. 8 (a)).[7] The display of digital devices (laptops, mobile devices, and tablets) exhibit a naturally occurring fixed repetitive pattern created by the geometry of color elements that are used for color displays. Therefore, whenever a image of a digital screen is recorded, moiré patterns will naturally present themselves due to the grid overlap between the digital screen and the digital camera. In color printing with CMYK (cyan, yellow, magenta, and black) halftoning model, moiré patterns are often inevitable (Fig. 8 (b)).[8] Moiré patterns are also observed in screen shooting photography (Fig. 8 (c)).[9] The fundamental reason for moiré patterns in screen shooting photography is because of the spatial frequency differences between the display and the acquisition devices. For example, when the scene (on the display of a replay device) contains repetitive details that exceed the camera resolution, moiré patterns are observed. Therefore, moiré patterns can be quite useful in face spoof detection of displayed photo and video replay attacks [44].

Color distribution may change during the recapturing of a face image, which leads to either reduced color diversity or color cast. For example, while the color distortion of printed attacks is due to the quality of the printer and photo paper, the color distortion of replay attacks is mainly caused by the fidelity and resolution of the screen [18]. Figure 9 shows the color distortion in spoof face images from three subjects in the MSU MSF database.

[7] www.ishootshows.com/2012/04/09/understanding-moire-patterns-in-digital-photography/

[8] users.ecs.soton.ac.uk/km/imaging/course/moire.html

[9] blog.ishback.com/?cat=132

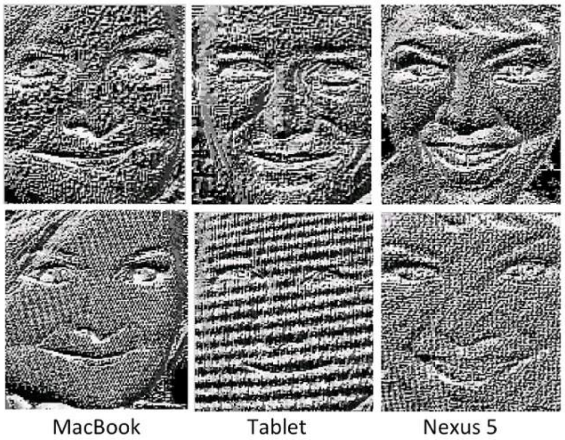


Fig. 11. The top row shows the LBP image of live subjects and the bottom row shows the LBP image of a live subjects displayed on digital screens (spoof faces).

D. Face Shape Deformation

In print attacks, the bending of the photo paper may lead to skewed face shape in the spoof face images. Additionally, the viewing directions of the camera will also lead to the deformation of the face shape in the spoof face images. Figure 10 shows the face shape distortion in spoof face images of print attacks from one subject in the MSU MSF database.

TABLE III

FEATURE DIMENSIONALITY, COMPUTATIONAL COST FOR FEATURE EXTRACTION, AND PERFORMANCE OF FEATURE REPRESENTATION METHODS EXTRACTED FROM THE IMAGES IN THE MSU MSF DATABASE. A 5-FOLD CROSS-VALIDATION TESTING PROTOCOL IS USED.

Method	Feature dimension	Avg. time per image (s)	HTER on MSU MSF
LBP Whole‡ Frame	4248	.014	4.95%
LBP* [7]	4248	.014	7.36%
LBP + Color Hist.*	4349	.044	7.08%
LBP + Color Moment*	4263	.021	7.80%
MLBP* [47]	11328	.072	8.38%
LUCID* [48]	51840	.021	19.49%
SIFT* [49]	34560	.303	15.32%
Color Hist.*	101	.031	38.66%
Specularity*	3	.112	41.34%
Blurriness*	1	.007	49.40%
Color Moment*	15	.008	24.12%
Image Quality† Analysis* [18]	121	.159	19.81%

‡The whole frame is resized to the same size as the cropped face image (144 × 120). *Denotes facial region used for feature extraction. †Feature level fusion of color histogram, specularity, blurriness, and color moment as used in [18]. The time is profiled on the same desktop (Intel Core 2 quad 3.0 GHz CPU and 8GB RAM).

IV. 2D SPOOF FACE DETECTION ON A MOBILE

A. Face Detection and Normalization

To detect faces on a mobile phone from the camera input, we used the built-in Android face detector. This detector only returns values for the inter-pupillary distance (IPD) and the mid-point of the face for all faces it detects in a given input image and therefore does not provide extract coordinates of the left and right eyes. Thus, we used a scale factor between the IPD and the mid-point of a detected face to normalize the face image. We then normalize the detected face into 144 × 120 pixel resolution. Based on our experience, the values that the Android face detector returns can vary greatly leading to different face cropping even in image frames captured only milliseconds apart. This variability in face detection results sometimes leads to inaccurate face spoof detection results. For our experiments on a desktop, we use the face detector that comes in the PittPatt face recognition SDK.[10]

B. Feature Representation

One popular and simple image descriptors for face images is Local Binary Patterns (LBP) [7]. This was later generalized to multi-scale LBP (MLBP) which has been shown to perform better than LBP, for example in [47] when matching composite sketches to face photos. This motivated us to conduct experiments to test if MLBP performs better than LBP when detecting for face liveness. Additionally, [48] introduced a new low complexity, effective image descriptor called Locally Uniform Comparison Image Descriptor (LUCID), which gives comparable results to the well known SURF descriptor. Since no results have been published on the effectiveness of LUCID on face liveness detection, we conducted experiments to analyze its potential. We also analyzed the SIFT (scale invariant feature transform) feature descriptor as this descriptor is largely invariant to scale, illumination, and local affine distortions [49].

Given the strengths and limitations of individual feature representation methods (Table I), we design our feature representation method by considering the complementarity between different clues. For example, the color histogram (top 100 colors in an image) based method depends upon how the image was presented to the FR system when conducting a spoof attack. A digital screen such as a laptop or a mobile phone can display millions of colors, thus the color diversity in a spoof image might be very similar to a live face image. However, in printed photo attacks the color diversity is greatly reduced, therefore this feature might be better suited to handle printed photo attacks. The same could be said for blurriness, as high resolution digital screens will display photos with high definition, however again when printing an image, the quality could be degraded. Thus, we hypothesized that a feature level fusion of texture features and image quality features would provide robust performance.

Since we are focusing on 2D face spoof detection which contains printed photo, displayed photo, and replayed video attacks, we chose to use the feature representation methods that work for both single face image and multiple video frames. We summarize the feature representation methods we considered in Table III. Given the performance of individual features and requirement of fast response in spoof detection on mobile phones, we choose to use a fusion of LBP (effective for face texture analysis) and color histogram (effective for

[10] PittPatt was acquired by Google in 2011, and the SDK is no longer publicly available.

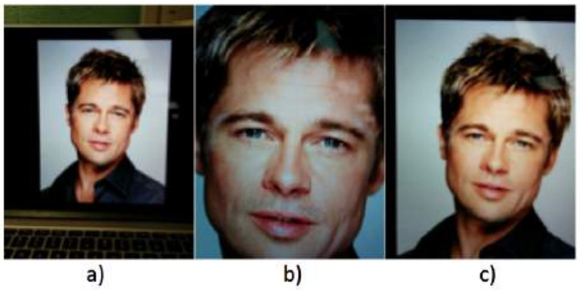


Fig. 12. Examples of inputs that were rejected using the proposed reject option: (a) IPD value below threshold, (b) IPD value above threshold, and (c) detected bezels along the top and left side of input image.

image quality analysis). Specifically, given an input face image (can be the whole frame or a face region), we calculate the LBP features by dividing the image into 32×32 patches with 16 pixels overlap. The LBP features from individual patches are concatenated together to construct a feature vector. As shown in Fig. 11, the LBP feature descriptor can capture patterns that appear in spoof imagery quite effectively. To extract the color histogram features, we used the entire face image based on the method described in [18]. We find that such a complementary feature representation method is effective to detect individual image aliasing artifacts in spoof face images (summarized in Section III), particularly under cross-database testing scenarios. Additionally, such a feature vector can be computed very efficiently, 0.044 sec. per face image on average, or about 20 fps. All the times are profiled with a Matlab implementation on a Windows 7 platform with Intel Core 2 quad 3.0 GHz CPU and 8GB RAM.

C. Multi-frame Voting

We perform face spoof detection on mobile by capturing a sequence of three face image frames. We utilize a 200 millisecond separation between the successive image captures to allow for the motion of a subject's hand holding the device to introduce subtle changes in the images captured.

Given the feature vectors extracted from the training images, we train a SVM classifier with an RBF kernel (using optimized parameters) to distinguish between live and spoof faces.[11] If two or more frames within the three frames in a session are classified as live faces then a given session will be classified as live, otherwise a spoof (majority voting). Using input from multiple frames allowed us to stabilize the decision for a session. While more frames may further improve the performance, we use three frames to ensure the proposed approach can run efficiently on mobile.[12]

D. Rejection Option

We observed that most malicious users tend to hold the spoof medium (smartphone or printed photo) at a certain distance to a smartphone camera when they are trying to spoof FR systems. They do this as they believe this will lead to higher quality face images being captured. Additionally, to hide the evidence of a spoof attack (bezels of a digital device and boundary of a printed photo) a malicious user may need to hold the spoof medium as close as possible to the FR system. An experiment conducted on 10 subjects shows that malicious users indeed tend to hold the spoof medium as close as possible to the FR system. This motivated us to utilize a threshold on IPD to reject an image.

In-order to find an acceptable range of IPD, we conducted experiments using 20 subjects, where we asked the users to take 10 pictures of themselves using a Google Nexus 5. Subjects that were used for this study had arms of varying lengths. The subjects were instructed to hold a smartphone as they would during normal usage and to capture a number of selfie pictures. Using these 200 images, we determined the typical IPD values under normal smartphone use. The average IPD of live faces (captured by the front camera of a mobile) is $\mu = 25.9\%$ of the image width (120 pixels), and the standard deviation of IPD is $\sigma = 4.1\%$ of image width. Based on these statistics, we reject faces that are too small (faces are very far from the smartphone camera) or too large (faces are very close to the smartphone camera) by using a threshold range of $[\mu - a\sigma, \mu + a\sigma]$. By setting $a = 2$, about 95% of the input face images are accepted and submitted to the spoof detection system.

Additionally, we define another reject option based on the detection of bezels of the spoof medium being used. This is done by detecting black stripes along the left and right sides (bezels) of the image as shown in Fig. 5, when the whole image is used. These stripes quickly allow us to detect spoof attacks, as these stripes will only appear on digital screens such as on laptops, smartphones and tablets.

Using the two rejection options described above greatly helps in detecting spoof attacks using minimal processing time. The combination of restricting the IPD of a subject and detecting the bezels of an input image reduces the number of images that are processed using our spoof detection method. This is due to the fact that when replay attacks are manufactured, the restriction on the IPD leads to capture of images that often contain the bezels of the spoof medium. Thus, the reject options help in reducing the number of false accepts in our system. Moreover, due to the update to Face Unlock with the release of Android 5.0, users no longer can view what the FR system is capturing. Therefore, malicious users no longer can ensure input to a FR system is free of any bezels. Figure 12 shows a couple examples of input face images that are rejected by the proposed method.

E. Prototype System on Mobile

We implement a prototype system of the proposed approach on a Nexus 5 with API level-21 support from Android v5.1. A minor change of the proposed method in the prototype system on desktop is that we now use Android face detector instead of the PittPatt face detector. Therefore, we retrained our face spoof detection model for the prototype system by utilizing the Android face detector to detect individual faces on the training dataset, and retrain our face spoof detection using the same method described in Sections IV.B-IV.D.

[11] LIBSVM is used: www.csie.ntu.edu.tw/~cjlin/libsvm.

[12] We also tried the score level fusion of all the frames, but it gives worse performance than the proposed voting scheme. A possible reason is the present of abrupt changes in decision scores between successive frames.

TABLE IV

PERFORMANCE OF FACE SPOOF DETECTION USING FACE IMAGES CAPTURED WITH FRONT AND REAR CAMERAS OF THE GOOGLE NEXUS 5.[†]

Training Set	Testing Set	FRR	FAR	HTER
Rear Camera	Front Camera	6.36%	44.13%	25.23%
Front Camera	Rear Camera	4.07%	37.14%	20.61%
Rear Camera	Rear Camera	4.72%	5.60%	5.16%
Front Camera	Front Camera	4.07%	5.19%	4.63%

[†] A five-fold cross-validation protocol is used. Rear Camera signifies spoof image captured by the rear facing camera on the Nexus 5 and Front Camera signifies spoof images captured by the front facing camera. FRR is the false rejection rate of live face images, and FAR is the false acceptance rate of spoof face images.

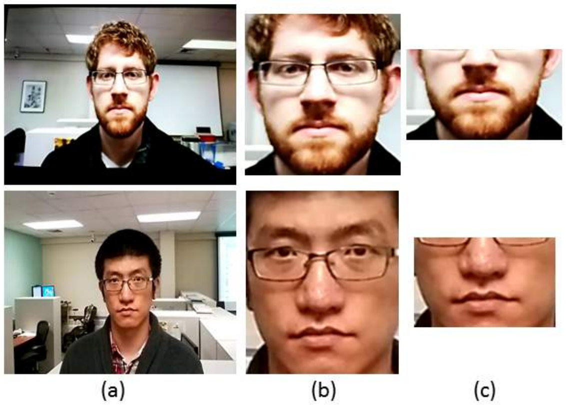


Fig. 13. Examples of three different image regions (of two different subjects) that are used for face spoof detection analysis: (a) the whole video frame, (b) the detected face image, and (c) the bottom half of the face image.

V. EXPERIMENTAL RESULTS

We perform face spoof detection experiments using the collected MSU MSF database, and the public-domain Replay-Attack, CASIA, and RAFS face databases. We study the influences of a number of factors (*e.g.,* image acquisition device, image region, IPD, and database size) to the proposed face spoof detection approach. The proposed approach is compared with the state of the art methods in both cross-database and intra-database testing scenarios. Unless otherwise stated, we perform each experiment using a five-fold, subject-exclusive cross validation protocol. We report performance in terms of HTER (see definition in Table I).

A. Influence of Image Acquisition Device

Table IV shows the effects on face spoof detection when cameras of different specifications are used to capture the training and testing face images. When face images from the training and testing sets are captured using cameras of different specifications, the HTER is larger than the HTER when the training and testing face images are captured using the same camera. See Table IV. To close this performance gap, the training set used to learn face spoof detection models should include a wide variety of image acquisition devices for both live and spoof face images. Therefore, the MSU MSF database should help to learn better face spoof detection models, as it contains live and spoof face images captured using several different cameras.

TABLE V

FACE SPOOF DETECTION PERFORMANCE ON THE MSU MSF DATABASE USING DIFFERENT REGIONS OF THE FACE IMAGE.

Image Region	HTER	Standard Dev.
Whole Frame	4.95%	0.58%
Whole Face	7.08%	0.59%
Bottom Face	10.17%	0.73%

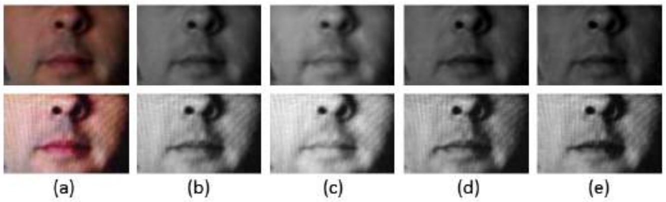


Fig. 14. Examples of live face images (top row) and spoof face images (bottom row) for one subject in the Idiap database. Video frames are shown using the (a) RGB image, (b) grayscale image, (c) red channel, (d) green channel, and (e) blue channel, respectively.

B. Influence of Different Image Regions

We study the effect of different image regions (i.e. whole image, detected face image, and bottom half of a face) on spoof detection performance using the MSU MSF database. Table V shows that when using the whole image to train face spoof detection models, the HTER is smaller than the HTER when using the detected facial region. This result seems to be counter to the prevailing wisdom that the background area of a face contains noise which may degrade performance. However, with further examination, we realize that the spoof detection model is tuned to detect the black stripes along the left and right sides (bezels) of the image, when the whole image is used as mentioned in Section IV.D. Thus when we consider the whole image, only images that did not contain any black strips along the edges were misclassified. Therefore, face spoof detection models specifically trained with whole images can efficiently detect printed photo and replayed video attacks, which often have black stripes due to the limited sizes of the photograph paper and screen. However, in more challenging scenarios, *e.g.,* when no black stripes appear in spoof face images (particularly when the malicious users intentionally prevent the paper or screen boundary appearing in the camera), experiments in [1] showed that using the detected facial region provides better performance than using the whole image.

C. Influence of Color Channel

We analyze the performance of the proposed face spoof detection method by using features extracted from the grayscale, red, green and blue channels of the detected face images from the MSU MSF database (see Fig. 14). Table VI shows that different color channels highlight varying amounts of texture in an image. The red channel gives better performance than the other color channels. Apparently the texture that can distinguish between spoof and live faces has higher contrast in the red channel of a face image.

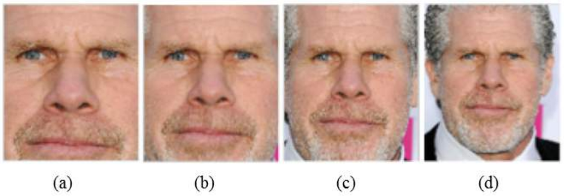


Fig. 15. Example of normalized face images with different IPDs: (a) 70 pixels, (b) 60 pixels, (c) 50 pixels, and (d) 40 pixels.

TABLE VI

PERFORMANCE OF FACE SPOOF DETECTION USING DIFFERENT COLOR CHANNELS (GRAYSCALE, RED, GREEN AND BLUE) ON THE MSU MSF DATABASE.

Color Channel	HTER	Standard Dev.
Grayscale	7.15%	0.68%
Red	7.08%	0.59%
Blue	7.52%	0.95%
Green	7.57%	0.74%

D. *Influence of IPD*

Given that most published methods extract features from the detected facial region, we analyze how the IPD affects the face spoof detection performance on the MSU MSF database. Given the cropped face images of fixed size (144×120), we vary the cropping of the facial region by altering the IPD (i.e. 50, 60, 70 and 80 pixels). Figure 15 shows the normalized face image of a subject when cropping using different IPD. As shown by the ROC curves in Fig. 16, using an IPD of 60 pixels when cropping a face leads to the best performance. This is due to the fact that cropping a face to have an IPD of 60 pixels removes most of the background area while retaining as large of the facial region as possible. Removing the background eliminates the background clutter from a normalized face image while a large facial region retains more distinctive features for classification of live and spoof face images.

E. *Influence of Database Size*

Most of the available public domain face spoof databases contain no more than 50 subjects. Therefore we study how the number of subjects in the training set affects spoof detection performance using the MSU MSF database. Figure 18 shows that using a larger training set significantly improves the cross-database performance on the Replay-Attack, CASIA and RAFS databases. Additionally, using 13K face images to train a face spoof detection model returns better performance than using 10K training face images when conducting intra-database testing on the MSU MSF database (see Fig. 17). The above results show that increasing the number of training face images to cover more diversities, from individual subjects to image acquisition devices, helps to learn more robust classifiers. Thus, larger databases such as the MSU MSF database will be very helpful in advancing solutions to the face spoof detection problem.

F. *Cross-database Testing*

It is now generally accepted that intra-database testing (training and test images, while distinct, are captured in the same environment and possibly of the same subjects) does not represent real world scenarios, as it lacks generalization ability [10]. We first evaluate the proposed approach under cross-database testing scenarios. We report the HTER of the proposed approach when conducting cross-database testing on the Replay-attack, CASIA, and the RAFS databases. The MSF MSU database is used to train the face spoof detection models. To avoid bias, we removed the overlapping subjects (50 from Replay-Attack, 50 from CASIA, 40 from RAFS) that appear in both the MSU MSF database and the testing databases. As shown in Table I, the proposed approach achieves 4.5%, 2.5%, and 9.5% HTERs on the Idiap, CASIA and RAFS database, respectively (see ROC curves from Fig. 18).

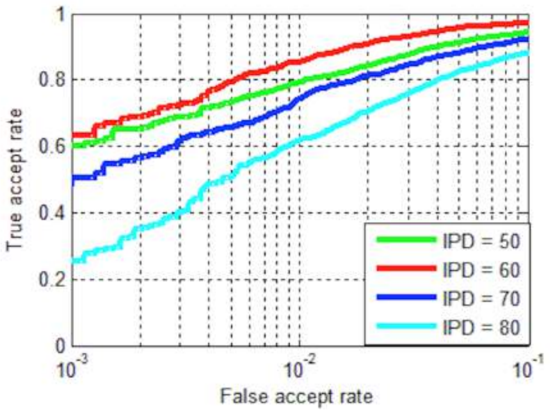


Fig. 16. Performance of face spoof detection on the MSU MSF database using face images with different IPD values (50, 60, 70, and 80 pixels).

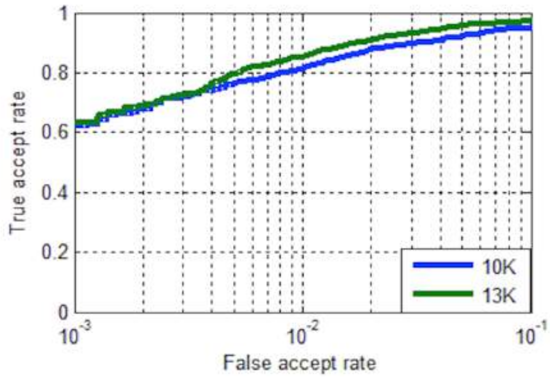
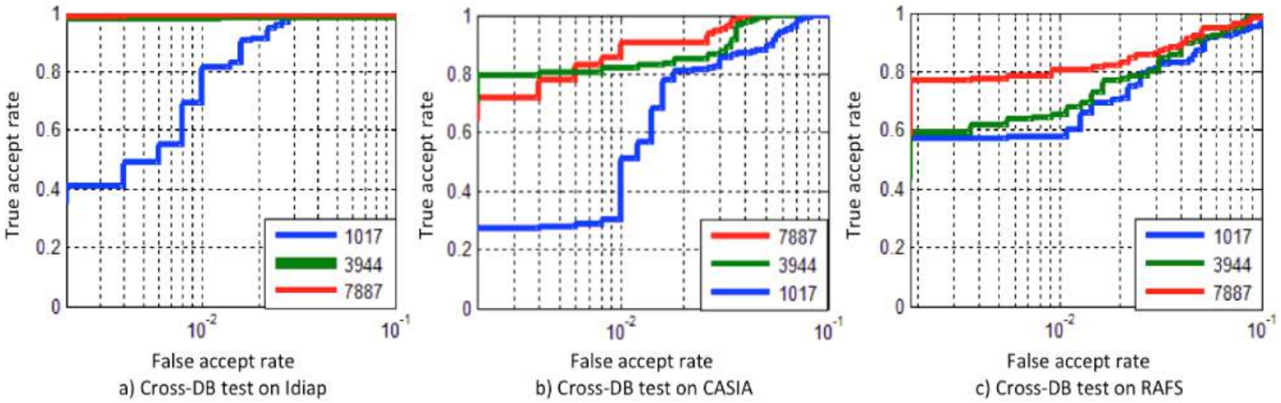


Fig. 17. Performance of the proposed face spoof detection approach on the MSU MSF database using 10K and 13K training face images.

Examples of correct classifications and misclassifications by the proposed approach on cross-database testing are shown in Fig. 19. No examples of false reject are reported by the proposed approach because in all three experiments, the false reject rate is 0.

G. *Intra-database Testing*

We also evaluate the proposed approach under the intra-database testing scenarios on the Replay-Attack, CASIA, and RAFS databases. Example images of subjects from these databases are shown Fig. 3. Table I shows that the proposed approach achieves 0.26%, 0.0%, and 0.1% HTERs on the Replay-Attack, CASIA and RAFS database, respectively.

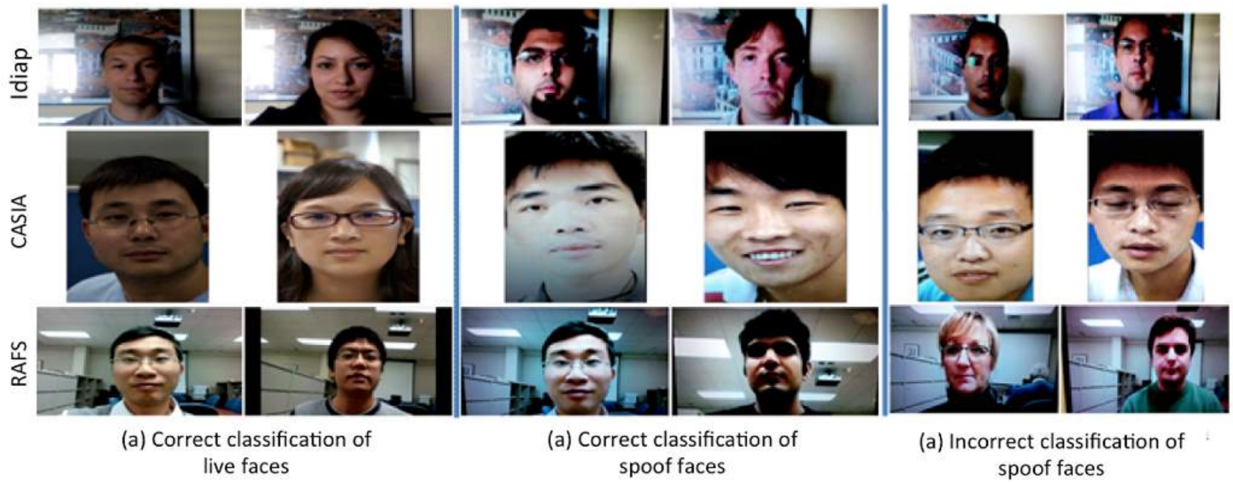


a) Cross-DB test on Idiap

b) Cross-DB test on CASIA

c) Cross-DB test on RAFS

Fig. 18. Performance of cross-database testing on the Replay-Attack, CASIA and RAFS databases using the MSU MSF database for training. The three ROC curves show how the varying training set size of the MSU MSF database (7887, 3944, 1017 face images) affects face spoof detection performance.



(a) Correct classification of live faces

(a) Correct classification of spoof faces

(a) Incorrect classification of spoof faces

Fig. 19. Examples of correct (a, b) and incorrect (c) classifications by the proposed approach in cross-database testing on the Replay-Attack (top row), CASIA (middle row), and MSU MSF (bottom row) databases.

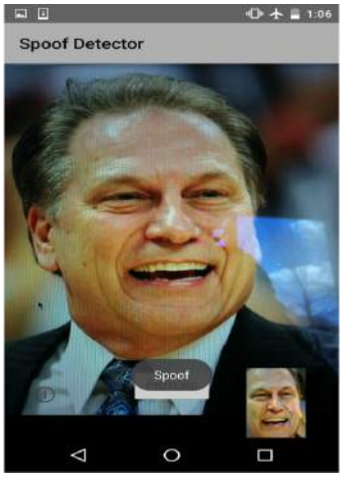


Fig. 20. The GUI of our android application. The figure shows the recaptured image of a face replay attack on a MacBook Pro screen; as shown on the Nexus 5 screen, the application successfully detected the input as a spoof access. The face image in the bottom-right corner displays the detected face.

When using subjects from the Idiap database, our approach gives slightly smaller HTER (0.26%) than the state of the art method (1.3% reported in [2]), but no cross-database testing result was reported in [2]. Using subjects from the CASIA database, our approach achieves much smaller EER (0.0%) than the state of the art (11.8% EER reported in [11]). Again, no cross-database testing result was reported in [2], [11]. As shown in Table I, intra-database testing scenarios are trivial compared to cross-database testing scenarios, which are more representative of real world scenarios.

H. *Performance Evaluation on Mobile Phones*

We evaluate the performance of our Android application by asking 20 subjects to use it in real world situations. The spoof detector application was loaded onto a Google Nexus 5 and a HTC Desire Eye (see GUI in Fig. 20). These subjects were chosen to make sure that the test set included a diverse set of subjects in terms of race, age, sex and facial hair style. The face spoof detecting models were trained on a desktop using the MSU MFS database.

One set of experiments was designed to determine whether our application could successfully detect live faces. These tests were conducted in various illumination conditions such as a dark hallway, sunny outside environment, and an indoor apartment setting with a large window. The users were instructed to hold the phone at different arm lengths and to move around in their environment to introduce variations. They were then instructed to periodically press the ”verify” button on the application and the result of face liveness detection were automatically recorded. For each subject, five verification tests were conducted. Among the 100 live face attempts (5 per subject), our Android application successfully accepted 96 faces (96.0% accuracy) on the Google Nexus 5 and 94 faces (94.0% accuracy) on the HTC Desire Eye.

Additionally, we conducted experiments to determine whether the application could effectively detect spoof face accesses. We asked the participating subjects in the live face detection experiment to capture selfie images of themselves (using both the rear and front facing cameras), which we would use afterwards to launch spoof face attacks. For spoof attacks, the subject's selfie images were displayed on an iPhone 6 and an Apple MacBook Pro laptop with retina display. Again, we did five tests per spoof medium. Among the 200 spoof face accesses, our Android application on the Google Nexus 5 correctly rejected 155 spoof faces (77.5% accuracy) and 157 spoof faces (78.5% accuracy) when the MacBook Pro laptop and iPhone 6 were used as the spoof medium, respectively. On the HTC Desire Eye, our application correctly rejected 136 spoof faces (68.0% accuracy) and 162 spoof faces (81.0% accuracy) when the MacBook Pro laptop and iPhone 6 were used as the spoof medium, respectively. The above spoof face detection results were recorded by turning off our rejection option. If we use the rejection option, numerous input to the FR system was rejected due to the detection of a bezel and the IPD constraint, and the accuracies of our application on both Google Nexus 5 and HTC Desire Eye were significantly higher.

The above results show that although our system was trained on the MSU MFS database in which the spoof face images were captured using a Nexus 5 camera, it generalizes well to different image acquisition devices under real application scenarios. The proposed method can be integrated into an operational mobile environment to detect most of the 2D face spoofing attacks while retaining a high true acceptance rate of live faces. For the incorrect classifications in live face access test, we notice that poor illumination condition is the main reason, particularly the dim light and yellow light. For the false acceptances of spoof face accesses, we notice that the main reason is the lack of moiré patterns which are caused by the occasional slow autofocus capability of the smartphone cameras.

I. Moiré Pattern Detection On Mobile

Given an input face image, our method will classify the input as a spoof access if moiré patterns are detected. As we discussed in Sec. III.B and our earlier paper [1], the presence of moiré patterns is the evident of displayed photo and video replay attacks lunched using a digital screen. To verify that our Android application is effective in detecting moiré patterns, we tested it on non-face images such as solid color images, nature images, and car wallpapers (see Fig. 21). For each image, five verification tests were conducted. Among the 75 spoof attempts our Android application correctly rejected 65 (86.7% accuracy) when using a MacBook Pro laptop to display the non-face images. Thus, our system is effective in detecting the presence of moiré patterns when capturing non-face images displayed on a digital screen. This experiment also shows that the proposed method still performs well for detecting displayed photo and video replay attacks, even if the face detection module does not give accurate face detection results.

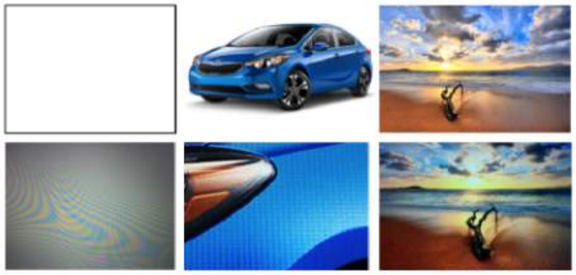


Fig. 21. Detection of the presence of moiré patterns from non-face images. The top and bottom rows shows three non-face images and three recaptured images from a MacBook screen, respectively.

J. Running Time on Mobile

The Android spoof detection application must provide fast response to the users. The current implementation takes .02 seconds for classification and 1.65 seconds to extract features from a single image frame (144×120) for a total time of 1.67 seconds. However, using three frames to make a decision leads to only a marginal increase in the total time to 1.95 seconds because of our multithreaded implementation on Android. All the times are profiled on a Google Nexus 5 smartphone with 2GB of ram and Quad-core 2.3 GHz Krait 400 CPU running native Android 5.0 ROM. As a comparison, the proposed approach takes 0.47 seconds on a desktop (see Section IV.B for desktop specification) for feature extraction and classification of a single frame.

VI. Summary and Conclusions

Spoofing attacks are a menace to biometric systems in terms of public perception and adoption. Face recognition systems can be easily targeted due to the low cost in launching face spoofing attacks such as printed photos or video replays. In order to address the problem of face spoofing detection on mobile phones, we propose an efficient face spoof detection approach based on the analysis of image aliasing in 2D spoof face images and the complementarity of individual clues. We also collected a large database, called the MSU Mobile Face Spoof (MSU MSF) database that contains replay and printed photo attacks captured by different smartphone cameras. Experimental evaluations using the MSU MSF database show that a large database is essential to learn robust face spoofing detection models, particularly under cross-database testing scenarios. Additionally, we propose a simple but efficient rejection option for face images based on IPD. We also study the influences of the image acquisition device, image color channel, and facial cropping region to the face spoof detection system. Our prototype system on two Android smartphones (Google Nexus 5 and HTC Desire Eye) shows that the proposed approach can perform face spoof detection efficiently using smartphones.

For future work, we plan to extend the MSU MSF database to include 3D facial mask attacks, and additional replay and printed photo attacks captured using various smartphones to increase the database diversity of face spoof attacks. We will also make use of the temporal and contextual information included in multiple video frames to build more robust face spoof models. Additionally, we will analyze whether the proposed method in combination with movement clues (*e.g.,* eyeblink) can improve spoof detection performance.

References

[1] K. Patel, H. Han, A. K. Jain, and G. Ott, "Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks," in *Proc. ICB*, 2015, pp. 1–8.

[2] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proc. CVPR Workshops*, 2013, pp. 105–110.

[3] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proc. ICCV*, 2007, pp. 1–8.

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

[5] W. Bao, H. Li, N. Li, and W. Jiang, "A liveness detection method for face recognition based on optical flow field," in *Proc. IASP*, 2009, pp. 233–236.

[6] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. IEEE BIOSIG*, 2012, pp. 1–7.

[7] J. Määtta, A. Hadid, and M. Pietikäinen, "Face spoofing detection from single images using micro-texture analysis," in *Proc. IJCB*, 2011, pp. 1–7.

[8] M. De Marsico, M. Nappi, D. Riccio, and J. Dugelay, "Moving face spoofing detection via 3d projective invariants," in *Proc. ICB*, 2012, pp. 73–78.

[9] J. Li, Y. Wang, T. Tan, and A. K. Jain, "Live face detection based on the analysis of fourier spectra," in *Proc. SPIE: Biometric Technology for Human Identification*, 2004, pp. 296–303.

[10] T. F. Pereira, A. Anjos, J. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *Proc. ICB*, 2013, pp. 1–8.

[11] J. Yang, Z. Lei, S. Liao, and S. Li, "Face liveness detection with component dependent descriptor," in *Proc. ICB*, 2013, pp. 1–6.

[12] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 864–879, Apr. 2015.

[13] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *Proc. ICB*, 2012, pp. 26–31.

[14] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection using 3D structure recovered from a single camera," in *Proc. ICB*, 2013, pp. 1–6.

[15] A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes, and S. Sridharan, "Liveness detection based on 3d face shape analysis," in *Proc. IWBF*, 2013, pp. 1–4.

[16] W. Kim, S. Suh, and J.-J. Han, "Face liveness detection from a single image via diffusion speed model," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2456–2465, Aug. 2015.

[17] 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.*, vol. 23, no. 2, pp. 710–724, Feb. 2014.

[18] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 746–761, Apr. 2015.

[19] H. Yu, T.-T. Ng, and Q. Sun, "Recaptured photo detection using specularity distribution," in *Proc. ICIP*, 2008, pp. 3140–3143.

[20] A. Pinto, W. Robson Schwartz, H. Pedrini, and A. De Rezende Rocha, "Using visual rhythms for detecting video-based facial spoof attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 1025–1038, May 2015.

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

[22] N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *Proc. BTAS*, 2013, pp. 1–6.

[23] Z. Zhang, D. Yi, Z. Lei, and S. Z. Li, "Face liveness detection by learning multispectral reflectance distributions," in *Proc. FG*, 2011, pp. 436–441.

[24] R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, and F. Roli, "Fusion of multiple clues for photo-attack detection in face recognition systems," in *Proc. IJCB*, Oct. 2011, pp. 1–6.

[25] J. Komulainen, A. Hadid, M. Pietikäinen, A. Anjos, and S. Marcel, "Complementary countermeasures for detecting scenic face spoofing attacks," in *Proc. ICB*, 2013.

[26] R. Raghavendra and C. Busch, "Robust scheme for iris presentation attack detection using multiscale binarized statistical image features," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 703–715, Apr. 2015.

[27] O. Komogortsev, A. Karpov, and C. Holland, "Attack of mechanical replicas: Liveness detection with eye movements," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 716–725, Apr. 2015.

[28] A. Czajka, "Pupil dynamics for iris liveness detection," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 726–735, Apr. 2015.

[29] I. Chingovska and A. Rabello dos Anjos, "On the use of client identity information for face antispoofing," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 787–796, Apr. 2015.

[30] J. Yang, Z. Lei, D. Yi, and S. Li, "Person-specific face antispoofing with subject domain adaptation," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 797–809, Apr. 2015.

[31] J. Sanchez, I. Saratxaga, I. Hernaez, E. Navas, D. Erro, and T. Raitio, "Toward a universal synthetic speech spoofing detection using phase information," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 810–820, Apr. 2015.

[32] A. Sizov, E. Khoury, T. Kinnunen, Z. Wu, and S. Marcel, "Joint speaker verification and antispoofing in the i -vector space," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 821–832, Apr. 2015.

[33] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "An investigation of local descriptors for biometric spoofing detection," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 849–863, Apr. 2015.

[34] M. Hildebrandt and J. Dittmann, "Stirtracev2.0: Enhanced benchmarking and tuning of printed fingerprint detection," *IEEE Trans. Inf. Forensics and Security*, vol. 10, no. 4, pp. 833–848, Apr. 2015.

[35] D. Smith, A. Wiliem, and B. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 736–745, Apr. 2015.

[36] N. Evans, S. Z. Li, S. Marcel, and A. Ross, "Guest editorial: Special issue on biometric spoofing and countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 699–702, Apr. 2015.

[37] J. Galbally, S. Marcel, and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access*, vol. 2, no. 1530-1552, 2014.

[38] M. Chakka, A. Anjos, S. Marcel, R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, F. Roli, J. Yan, D. Yi, Z. Lei, Z. Zhang, S. Li, W. Schwartz, A. Rocha, H. Pedrini, J. Lorenzo-Navarro, M. Castrillon-Santana, J. Maatta, A. Hadid, and M. Pietikäinen, "Competition on counter measures to 2-D facial spoofing attacks," in *Proc. IJCB*, 2011, pp. 1–6.

[39] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3d masks," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1084–1097, Jul. 2014.

[40] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Proc. ECCV*, 2010, pp. 504–517.

[41] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *In Proc. ICCV*, 2007, pp. 1–8.

[42] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: A public database and a baseline," in *Proc. IJCB*, 2011, pp. 1–7.

[43] J. Bai, T.-T. Ng, X. Gao, and Y.-Q. Shi, "Is physics-based liveness detection truly possible with a single image?" in *Proc. ISCAS*, 2010, pp. 3425–3428.

[44] D. C. Garcia and R. L. de Queiroz and, "Face-spoofing 2d-detection based on moiré-pattern analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 778–786, Apr. 2015.

[45] D. Wang, S. Hoi, Y. He, J. Zhu, T. Mei, and J. Luo, "Retrieval-based face annotation by weak label regularized local coordinate coding," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 550–563, Mar. 2014.

[46] I. Amidror, *The Theory of the Moiré Phenomenon Volume I: Periodic Layers, 2nd ed.* Springer, 2009.

[47] H. Han, B. Klare, K. Bonnen, and A. Jain, "Matching composite sketches to face photos: A component-based approach," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 191–204, Jan. 2013.

[48] A. Ziegler, E. Christiansen, D. Kriegman, and S. Belongie, "Locally uniform comparison image descriptor," in *Proc. NIPS*, 2012, pp. 1–8.

[49] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, 1999, pp. 1150–1157.