A Closer Look Tells More: A Facial Distortion Based Liveness Detection for Face Authentication

Yan Li
Xidian University
Advanced Digital Science Center
li.yan@adsc-create.edu.sg

Zilong Wang
Xidian University
zlwang@xidian.edu.cn

Yingjiu Li
Singapore Management University
yjli@smu.edu.sg

Robert Deng
Singapore Management University
robertdeng@smu.edu.sg

Binbin Chen
Advanced Digital Science Center
binbin.chen@adsc-create.edu.sg

Weizhi Meng
Technical University of Denmark
weme@dtu.dk

Hui Li
Xidian University
lihui@xidian.edu.cn

ABSTRACT

Face authentication is vulnerable to media-based virtual face forgery (MVFF) where adversaries display photos/videos or 3D virtual face models of victims to spoof face authentication systems. In this paper, we propose a liveness detection mechanism, called FaceCloseup, to protect the face authentication on mobile devices. FaceCloseup detects MVFF-based attacks by analyzing the distortion of face regions in a user's closeup facial videos captured by built-in camera on mobile device. It can detect MVFF-based attacks with an accuracy of 99.48%.

CCS CONCEPTS

• **Security and privacy** → **Biometrics**; *Usability in security and privacy*.

KEYWORDS

Liveness detection, perspective distortion, face authentication

ACM Reference Format:
Yan Li, Zilong Wang, Yingjiu Li, Robert Deng, Binbin Chen, Weizhi Meng, and Hui Li. 2019. A Closer Look Tells More: A Facial Distortion Based Liveness Detection for Face Authentication. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3321705.3329850

1 INTRODUCTION

Most of existing face authentication systems are vulnerable to *media-based virtual face forgery* (MVFF) where an adversary displays a photo/video or a 3D virtual face model of a victim. Liveness detection has been proposed to counter MVFF-based attacks. Some liveness detection thwarts *photo-based attacks* based on users' facial motions or expressions, such as eye blink and head rotation [6]. But such liveness detection approaches are still vulnerable to *video-based attacks* where an adversary replays a pre-recorded face video.

Two recent liveness detection approaches were proposed to defeat MVFF-based attacks, which are FaceLive and Face Flashing. FaceLive performs liveness detection by examining the consistency between a captured face video and device movement data [8]. FaceLive can detect photo-based attacks and video-based attacks but is still subject to 3D virtual face model-based attacks. Face Flashing [12] analyzes reflection light from a face to detect MVFF-based attacks. However, Face Flashing incurs significant network traffic and raise privacy concern because it requires cloud computing.

In this work, we propose FaceCloseup, a facial distortion-based liveness detection mechanism to protect face authentication on mobile devices against MVFF-based attacks. FaceCloseup can detect not only photo/video based attacks, but also 3D virtual face model-based attacks. FaceCloseup only requires a generic front-facing camera on mobile devices but no specific usage settings such as controlled lighting and sending facial videos to remote server. FaceCloseup is thus suitable for on-device liveness detection and can be deployed on commodity mobile phones. Empowered with a CNN-based classification algorithm, FaceCloseup determines the liveness of a face based on facial distortion changes in a facial video.

To thwart MVFF-based attacks, FaceCloseup detects 3D characteristics of a live user's face by analyzing the changes of distortion in facial video frames. The distortion of the user's face in the video is a common phenomenon in photography especially when the camera is close to the face. The distortion is mainly caused by the uneven 3D surface of the face. Facial regions in the video frames are displayed in different scales.

We collect real-world facial photo and video data from legitimate authentication requests and MVFF-based attacks. We mimic the 3D virtual face model-based attack using the

AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand

ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3329850

state-of-the-art 3D face reconstruction technique [5] which synthesizes facial photos with facial distortion. Our results show that FaceCloseup can detect MVFF-based attacks with an accuracy of 99.48%.

2 THREAT MODEL

The media-based virtual face forgery (MVFF) enables an adversary to forge users' face biometrics based on their facial photos or videos. The adversary may display the forged face to spoof face authentication and therefore pose a serious threat against face authentication systems.

In the photo-based attack and video-based attack, an adversary replays a user's pre-recorded facial photos and videos which the adversary may obtain from online, such as online social networks. The 3D virtual face model-based attack is more complicated and powerful where an adversary builds up a 3D virtual face model for a user based on the user's facial photos and videos. The adversary can synthesize facial videos with facial motions and/or expressions so as to spoof face authentication system.

The effectiveness of the photo/video based attacks usually depends on the quality and availability of the victim's facial photos/videos, which may be mitigated by extra facial motions and expressions. The 3D virtual face model-based attack poses significant risks to face authentication systems because the adversary can display a 3D virtual face model of the victim and synthesize the required facial motions and expressions in real time. The 3D virtual face model can be estimated by the adversary based on the victim's face photos and videos regardless of facial movements and expressions [1]. It is important for liveness detection to defend against the 3D virtual face model-based attack.

Our proposed liveness detection mechanism, FaceCloseup, aims to prevent MVFF-based attacks including the photo/video based attacks and the 3D virtual face model-based attacks. In MVFF-based attacks, it is assumed that *an adversary cannot obtain a user's pre-recorded facial photos/videos taken within 30cm from the user's face*. It is difficult for an adversary to directly capture the users closeup facial photos/videos without the users' awareness. In comparison, the user is more likely to leak his/her facial photos and videos taken no shorter than 30cm from the face by online sharing such as sharing selfie photos and videos in online social networks and video calls such as video chat or video conference. The adversary may access these facial photos and videos.

3 DESIGN

FaceCloseup includes three modules which are Video Frame Selector (VFS), Distortion Feature Extractor (DFE), and Liveness Classifier (LC). The VFS module takes facial video as input and selects multiple frames from the facial video based on the size of the face in the frames. With the extracted frames, the DFE module detects a number of facial landmarks in each frame and calculate features about the facial distortion changes among different frames. At last, the LC module utilizes a classification algorithm to distinguish a real face from a forged face in MVFF-based attacks.

As a mobile device moves towards or away from a user's face, the camera on the device firstly captures a video which includes a number of frames about the user's face taken at different distances between the camera and the face. The size of the faces in the video frames changes due to the movement. Using Viola-Jones face detection algorithm [13], the Video Frame Selector (VFS) extracts and selects a sequence of K frames $(f_1, f_2, ..., f_K)$ in the video based on the detected face size $(sz_1, sz_2, ..., sz_K)$ where $sz_i \in (np_{il}, np_{iu})$.

Secondly, with $(f_1, f_2, ..., f_K)$ as input, DFE calculates the geometric distances between different facial landmarks in each frame and uses them as features for detecting distortion changes in the facial video. We use the supervised descent method (SDM) to detect 66 facial landmarks from each frame [15]. The 66 facial landmarks are located at various regions of a face, including chin (17), eyebrows (10), nose stem (4), below nose (5), eyes (12), and lips (18), which are shown in Figure 1. The facial landmarks are denoted as $(p_1, p_2, ..., p_{66})$ where $p_i = (x_i, y_i)$ is the coordinate.

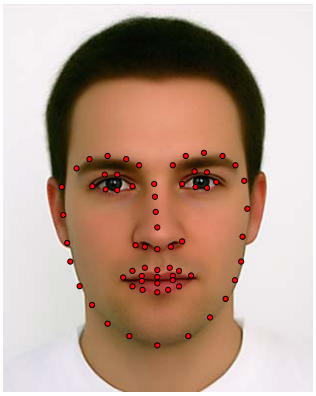

Figure 1: 66 facial landmarks

In order to capture facial distortion, we calculate the distance between any two facial landmarks p_s and p_t as $d = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2}$, where $s, t \in \{1, 2, ..., 66\}$ and $s \neq t$. The 66 facial landmarks in each frame yield 2145 pairwise distances $d_1, d_2, ..., d_{2145}$. Assuming the size of a detected face in a frame is w in width and h in height, a geometric vector about the detected face is formed as $geo = (d_1, d_2, ..., d_{2145}, w, h)$. Then we calculate relative distances by normalizing the geometric vector of each frame according to a base facial image, which is registered by a user in a registration phase. The geometric vector for the base image is calculated as $geo_b = (d_{b1}, d_{b2}, ..., d_{b2145}, w_b, h_b)$. For each selected frame f_i, we calculate a relative geometric vector $rio_i = (r_{i,1}, r_{i,2}, ..., r_{i,2145}, r_{i,w}, r_{i,h})$, where $r_{i,j} = d_{i,j}/d_{bj}$ for $j = 1, 2, ..., 2145$, $r_{i,w} = w_i/w_b$, and $r_{i,h} = h_i/h_b$. The facial distortion in K selected frames is represented by a $K \times 2147$ matrix FD.

Thirdly, LC module takes FD as input and uses a classification algorithm to determine whether FD is taken from a real face or a forged face from MVFF-based attacks. Due to high dimension of matrix FD, convolutional neural network (CNN) is customized in the LC module including 2 convolution layers, 2 pooling layers, 2 fully connected layers, and 1 output layer. Given a $K \times 2147$ feature matrix FD, the convolution layer $Conv_1$ computes a tensor matrix TM_1'. In order to achieve nonlinear properties without affecting the receptive fields in the convolution layer $Conv_1$, a rectified linear unit (ReLU) is used as activation function over TM_1' and outputs a tensor matrix TM_1. The ReLU is formed as $f(x) = \max(0, x)$. The pooling layer $Pool_1$ performs a nonlinear downsampling on TM_1. The convolution layer $Conv_2$ and the pooling layer $Pool_2$ perform the same operations as $Conv_1$ and $Pool_1$ in the third and fourth steps, respectively. Next, the fully connected layers FC_1 and FC_2 perform high-level reasoning. Assuming FC_2 consists of M neurons, a vector $fc = (e_1, e_2, ..., e_M)^T$ is produced by FC_2 and it is passed to the output layer OUT. The output layer estimates the probabilities for C classes. The probability of each class c is estimated using the following multinomial distribution

$$P(y = c) = S_c = \frac{\exp(V_c^y \cdot fc + b_c^y)}{\sum_{c=1}^{C} \exp(V_c^y \cdot fc + b_c^y)} \quad (1)$$

where C is number of classes, V_c^y is the c-th row of a learnable weighting matrix V^y, and b_c^y is a bias. Since liveness detection is used to distinguish between a real face and a forged face, C is set to 2.

4 DATA COLLECTION AND DATASET GENERATION

An IRB-approved user study is conducted to collect users' data for both legitimate requests and MVFF-based attacks which include the photo-based attacks, video-based attacks, and 3D virtual face model-based attacks.

4.1 Data Collection

Our user study consists of two parts and involves 43 males and 28 females with the age range between 18 and 35.

In the first part, we collect the participants' multiple selfie facial videos at controlled device positions. Each participant is asked to hold a mobile phone and to take 3 selfie frontal facial video clips over a controlled distance D_{FD} between his/her face and mobile phone. The mobile phone in our experiments is a Google Nexus 6P smartphone with an 8-megapixel front-facing camera and Android 7.1.1 operating system. The front-facing camera is used to take 1080p HD video recording at 30 fps. Each resulting video clip lasts for 3 seconds where each frame is 1920×1080 pixels in size with the face in the middle of the frame. The range of the controlled distance D_{FD} between the face and smartphone includes 20cm, 30cm, 40cm, and 50cm. We collected 12 selfie frontal facial video clips from each participant.

In the second part, participants are asked to perform trials of FaceCloseup with the controlled device movement distances using the provided smartphone. Each participant holds and moves the smartphone away from his/her face from the distance $20cm$ to the distance $50cm$, from the distance $30cm$ to the distance $50cm$, and from the distance $40cm$ to the distance $50cm$, respectively. 10 trials are performed by each participant under each controlled movement setting. In total, facial video data from 30 trials by each participant.

4.2 Dataset Generation

To mimic legitimate requests and MVFF-based attacks, we generate a legitimate dataset, a photo-based attack dataset, a video-based attack dataset, and a 3D virtual face model-based attack dataset based on the collected facial videos.

4.2.1 Legitimate Dataset. The legitimate dataset includes the closeup facial videos taken during trials of FaceCloseup with the smartphone movement from the distance $D_{FD} = 20cm$ to the distance $D_{FD} = 50cm$ which is presented in Section 4.1. Thus the legitimate dataset includes 710 trials.

4.2.2 Photo-Based Attack Dataset. We firstly manually extract 10 facial frames from the selfie frontal facial video clips taken by each participant at each fixed distance D_{FD} including 30cm, 40cm, and 50cm. The majority of the participants never reveal selfie facial photos/videos taken at the distance shorter than 30cm due to the obvious facial distortion.

Secondly, we display each extracted facial frame on an iPad Retina screen to mimic the photo-based attacks. The scale of face region in the frame is adjusted to be displayed in full screen so that the size of the face displayed on the screen is close to the size of a real face. While the smartphone is fixed on the table with the front-facing camera always shooting at the iPad screen, we move the iPad away from the smartphone from the distance $D_{FD} = 20cm$ to the distance $D_{FD} = 50cm$. During the movement, the front-facing camera on the smartphone records video about the face on the screen. In total, the photo-based attack dataset includes 1420 videos.

4.2.3 Video-Based Attack Dataset. We use the videos of the trials taken by each participant who moves the smartphone from the distance $D_{FD} = 30cm$ to the distance $D_{FD} = 50cm$ and from the distance $D_{FD} = 40cm$ to the distance $D_{FD} = 50cm$. Each of the video is displayed on the iPad screen while the smartphone is fixed on the table with the front-facing camera always recording the screen. The scales of the video frames are adjusted accordingly so that the face region in the first frame of the video is displayed in full screen. The distance between the iPad and the smartphone is fixed to 20cm since the displayed video includes the movements similar to the legitimate request. Thus we have 1420 attacking videos in the video-based attacks in total.

4.2.4 3D Virtual Face Model-Based Attack Dataset. There exists a variety of 3D face reconstruction algorithms [2, 10, 11, 16]. Most of the 3D face reconstruction algorithms take a single or multiple regular facial photos of the victim as input. Based on the detected facial landmarks, a 3D virtual face model for the victim is estimated by optimizing the

geometry of a 3D morphable face model in order to match the observed 2D landmarks. Note that the optimization of the 3D morphable face model is based on an important assumption that a virtual camera is shooting at the face with a pre-defined distance between the virtual camera and the face (usually assumed to be infinite). Then, image-based texturing and gaze correction techniques are applied to adjust the 3D face model. At last, the textured 3D face model of the victim can be used to produce different facial expressions and head rotation in real time.

FaceCloseup determines the liveness of a face based on the change of facial distortion as the distance between the camera and the face changes. The above 3D virtual face model cannot defeat FaceCloseup because the 3D virtual face model cannot generate the facial distortion when the camera is close to the face.

In order to simulate a powerful adversary in the 3D virtual face model-based attack, we use the state-of-the-art perspective-aware 3D face reconstruction algorithm, by Fried et al. [5] published in SIGGRAPH'2016, which can generate both the facial distortion according to the changes of a virtual camera position and any changes of facial expressions and head poses. To reconstruct a perspective-aware 3D face model for a victim, the perspective-aware 3D face reconstruction algorithm [5] firstly extracts 69 facial landmarks from a given facial photo of the victim. Among the extracted facial landmarks, 66 facial landmarks are automatically detected by the SDM-based landmark detection algorithm [15]. The other 3 facial landmarks on top of head and ears are manually labelled for higher accuracy. Because the 3D face model is correlated with an identity vector $\beta \in \mathbb{R}_{1\times 50}$, an expression vector $\gamma \in \mathbb{R}_{1\times 25}$, an upper-triangular intrinsic matrix $U \in \mathbb{R}_{3\times 3}$, a rotation matrix $R \in \mathbb{R}_{3\times 3}$, and a translation matrix $T \in \mathbb{R}_{3\times 4}$, the facial photo and the 69 facial landmark locations are used to fit a 3D head model by finding the best parameters β, γ, U, R, T such that the Euclidean distance between the facial landmarks and the projection of the landmarks on the 3D head model is minimized. After a good fit is made between the input facial photo and the 3D head model, the 3D head model is manipulated for a new projected head shape by changing the virtual camera distance and head poses. In particular, one can move the virtual camera towards/away from the face by adjusting the translation T and rotate head by adjusting both the translation T and the rotation R. At last, the manipulated 3D head model produces a 2D facial photo with the distortion corresponding to the changes of camera position. Due to the space limit, we refer readers to [5] for the details of this perspective-aware 3D face reconstruction algorithm.

To perform the 3D virtual face model-based attack, for each participant, we extract 10 facial photos from the selfie facial video taken at each controlled distance D_{FD} between the participant's face and smartphone which are collected in the first part of the user study as explained in Section 4.1. The range of D_{FD} includes 30cm, 40cm, and 50cm. Given each facial photo as an input, we use the perspective-aware 3D face reconstruction algorithm to generate the facial photos with facial distortion by manually changing the virtual camera distance in the algorithm. The values of the virtual camera distance include 20cm, 25cm, 30cm, 35cm, 40cm, 45cm, and 50cm. We manually adjust the scale of the resulting manipulated facial photos in compliance with the size of face region in the original facial photos extracted from the selfie facial video taken over the same/similar distances. Therefore, we generate a sequence of 7 manipulated facial photos from each extracted facial photo. In total, the 3D virtual face model-based attack dataset consists of 2130 sequences of manipulated facial photos.

5 EVALUATION AND EXPERIMENTAL RESULTS

In this section, we present the settings of our experiments. Then we evaluate the performance of FaceCloseup in terms of security, effectiveness and practicality.

5.1 Experiment Settings

To determine the liveness of a face, the VFS of FaceCloseup firstly selects K frames from an input facial video based on the size ranges $(sz_1, sz_2, ..., sz_K)$ of the face region detected in the video frames as presented in Section 3. Since the size of the face region detected in the frame mainly depends on the distance between a participant's face and the smartphone, the size ranges $(sz_1, sz_2, ..., sz_K)$ are determined based on the distribution of the size of the detected faces in the facial videos taken at the distance $D_{FD} = 20cm, 30cm, 40cm, 50cm$. These facial videos with the different distance D_{FD} are collected in the user study as described in Section 4.1. We set $K = 7$ and choose a base facial photo of each user with the size of face region in sz_1 for best performance and better coverage of the video frames taken at different distances in our experiments. The size ranges of the detected faces are shown in Table 1. For a given facial video, we select a sequence of the frames by randomly choosing a frame among the frames containing a face with size in sz_i where $i \in \{1, 2, ..., 7\}$. We repeat the selection for 20 times and extract 20 sequences of frames as samples from the given facial video. Therefore, $20 \times 710 = 14200$ samples are generated based on the legitimate dataset. $20 \times 1420 = 28400$ samples are generated based on the photo-based attack dataset and the video-based attack dataset, respectively. And 2130 samples are generated based on the 3D virtual face model-based attack dataset.

Table 1: Size ranges of the face region for frame selection

	Size range in mega-pixels
sz_1	$(0.75, 0.85)$
sz_2	$(0.65, 0.75)$
sz_3	$(0.55, 0.65)$
sz_4	$(0.45, 0.55)$
sz_5	$(0.35, 0.45)$
sz_6	$(0.25, 0.35)$
sz_7	$(0.15, 0.25)$

The LC of FaceCloseup is empowered with a CNN-based classification algorithm. The structure and parameters of the CNN model are shown in Table 2. We use 5-fold cross validation method to evaluate FaceCloseup. Thus 80% of the samples are used to train the CNN model on a desktop equipped with a 12GB TITAN X graphics card, 60GB memory, and 20 Intel Core-i7 CPUs. The learning rate is set to 0.1, weight decay is 0.0001, and the max iteration is 1000 accordingly.

Table 2: The structure and parameters of the CNN model

Layer	Size	Stride	Padding
$Conv_1$	32 5×5 filters	1	1
$Pool_1$	3×3	2	1
$Conv_2$	32 3×3 filters	1	1
$Pool_2$	3×3	2	1
FC_1	1×1024	0	0
FC_2	1×192	0	0
OUT	1×2	0	0

5.2 Experimental Results

5.2.1 Detecting MVFF-based Attacks. FaceCloseup is accurate in detecting MVFF-based attacks, including photo/video based attacks and 3D virtual face model-based attacks.

In the photo-based attacks, the facial photos taken at the distance $D_{FD} = 30cm, 40cm, 50m$ are displayed to generate the attack videos as explained in Section 4.2. Figure 2 shows that FaceCloseup can effectively detect the photo-based attacks. In particular, FaceCloseup achieves the accuracy of 99.23%, 99.28%, and 99.31% against photo-based attacks with photos taken at 30cm, 40cm, and 50cm away, respectively. Because FaceCloseup determine the liveness of a face by analyzing the facial distortion changes correlated to the 3D depth information of the real 3D face and the changes of the camera distance, it is difficult for the adversary to generate the correct facial distortion by displaying a 2D facial photo on a 2D surface and moving the facial photo.

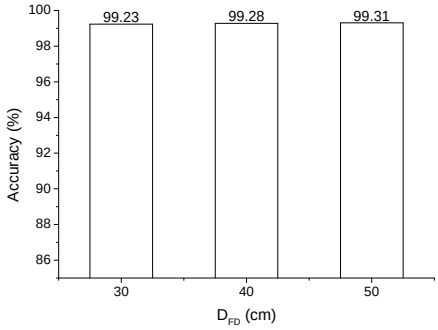

Figure 2: Accuracy of FaceCloseup against the photo-based attacks

In the video-based attacks, the facial videos are displayed to FaceCloseup. The attacking facial videos are taken when the smartphone moves from the distance $D_{FD} = 30cm$ to the distance $D_{FD} = 50cm$ and from the distance $D_{FD} = 40cm$ to the distance $D_{FD} = 50cm$, respectively. FaceCloseup achieves the accuracy of 99.24% and 99.27% against the two types of attack videos, as shown in Figure 3. FaceCloseup requires closeup videos containing obvious changes of the facial distortion while the rate of the facial distortion changes become lower as the camera move away from the face. Therefore, the two types of the attacking facial videos do not include obvious and sufficient changes of the facial distortion because the camera is not close to the face enough.

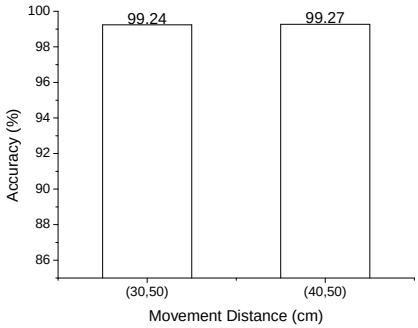

Figure 3: Accuracy of FaceCloseup against the video-based attacks

The 3D virtual face model-based attack is a powerful attack. The synthesized facial photos are produced which include the estimated facial distortion based on the changes of the camera distance. FaceCloseup successfully detects this attack with an accuracy of 99.48%. An important reason is that it is still challenging for the existing 3D virtual face model to synthesize the closeup facial photos with significant facial distortion due to the complex and uneven 3D surface of the faces and occlusion of partial facial regions [5].

6 RELATED WORK

We summarize the related work based on the liveness indicators they use, including 3D face, texture pattern, real-time response, and multimodal.

The 3D face liveness indicator is based on the clue that a real face has 3D depth characteristics. The 3D face characteristics detection is usually associated with optical flow analysis and changes of face views. A 3D face has the characteristic of optical flow that the motion speed of the central part of face is higher than the outer face region [6]. Along this line, Kollreider et al. proposed a liveness detection algorithm to analyze the optical flow based on ears, nose, and mouth [7]. However, the optical flow based methods usually require high-quality input videos with ideal lighting conditions, which may be difficult to achieve in practice. Compared to these works, FaceCloseup takes input video from a generic camera which can be easily achieved in practice.

On the other hand, the 3D characteristics about a real face can be detected in face movements. Chen et al. examined

the 3D characteristics of nose in the liveness detection [4] which compares the the direction changes of the mobile phone measured by the accelerometer and the changes of nose edge in the video. However, to produce the clear nose edge, a controlled lighting is required. The controlled lighting may not be possible in practice. Li et al. proposed FaceLive which requires a user to move the mobile device in front of his/her face and analyzes the consistency between the motion data of the mobile device and head rotation in the facial video [8]. Unfortunately, although the above two liveness detection algorithms can detect the photo-based attacks and the video-based attacks, they are vulnerable to the 3D virtual face model-based attacks as an adversary can synthesize the correct nose changes and head rotation video according to the device movements in real time [16].FaceCloseup can detect all MVFF-based attacks including the photo-based attack, video-based attack, and 3D virtual face model-based attack.

The texture pattern based techniques examines detectable texture patterns due to the printing process and the material printed on. IDIAP team took a facial video as input and the local binary patterns from each extracted frame in the video in order to build a global histogram for the video. The liveness of face is determined based on the global histogram [3]. Tang et al. proposed Face Flashing which captures face videos with strong enough random screen light shooting at a user's face and sends the video to a remote server such as cloud services for analysis of the light reflection in the face videos for liveness detection [12]. The texture pattern based techniques usually require high-quality photos/videos captured in ideal lighting conditions and significant computation power in the analysis, which may be hard to achieve on mobile devices in practice. Using computation power from remote server or cloud services could incur the cost of significant network traffic and privacy issues. In contrast, FaceCloseup takes closeup facial videos as input and analyzes the input videos locally on mobile devices.

The real-time response based approaches require interaction with users in real time. Pan et al. required users to blink their eyes in order to detect the liveness [9]. Unfortunately, these approaches are subject to the video-based attacks and the 3D virtual face model-based attacks. FaceCloseup can detect such video-based attacks effectively.

Finally, multimodal based liveness detection approaches take face biometrics and other biometrics into account in user authentication. Wilder et al. took facial thermogram from an inferred camera and face biometrics from a generic camera in authentication process [14]. Unlike the above approaches, which rely on the hardware sensors rarely deployed on mobile devices, our approach requires a front-facing camera which is pervasively available on most mobile devices.

7 CONCLUSION

In this paper, we proposed an effective and practical liveness detection mechanism, FaceCloseup, for face authentication to prevent MVFF-based attacks. FaceCloseup does not require any additional hardware but a front-facing optical camera which is widely available on mobile devices. FaceCloseup detects 3D characteristics of a real face by using deep learning techniques to analyze and identify the changes of facial distortion in a closeup facial video. FaceCloseup can detect all MVFF-based attacks with an accuracy as high as 99.48%.

8 ACKNOWLEDGMENTS

The work was supported in part by NSFC under Grant 61802289, 61671013, 61672410. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Yingjiu Li was supported in part by the Singapore National Research Foundation under NCR Award Number NRF2014NCR-NCR001-012. Robert Deng was supported in part by AXA Research Fund.

REFERENCES

[1] Andrea F Abate, Michele Nappi, Daniel Riccio, and Gabriele Sabatino. 2007. 2D and 3D face recognition: A survey. *Pattern Recognition Letters* 28, 14 (2007), 1885–1906.
[2] Christian Baumberger, Mauricio Reyes, Mihai Constantinescu, Radu Olariu, Edilson de Aguiar, and Thiago Oliveira Santos. 2014. 3D face reconstruction from video using 3d morphable model and silhouette. In *SIBGRAPI*. IEEE, 1–8.
[3] Murali Mohan Chakka, Andre Anjos, Sebastien Marcel, Roberto Tronci, Daniele Muntoni, Gianluca Fadda, Maurizio Pili, Nicola Sirena, Gabriele Murgia, Marco Ristori, et al. 2011. Competition on counter measures to 2-d facial spoofing attacks. In *IJCB 2011*. IEEE, 1–6.
[4] Shaxun Chen, Amit Pande, and Prasant Mohapatra. 2014. Sensor-assisted Facial Recognition: An Enhanced Biometric Authentication System for Smartphones. In *MobiSys 2014*. 109–122.
[5] Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. 2016. Perspective-aware manipulation of portrait photos. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 128.
[6] O. Kahm and N. Damer. 2012. 2D face liveness detection: An overview. In *BIOSIG 2012*. 1–12.
[7] Klaus Kollreider, Hartwig Fronthaler, and Josef Bigun. 2009. Non-intrusive liveness detection by face images. *Image and Vision Computing* 27, 3 (2009), 233–244.
[8] Yan Li, Yingjiu Li, Qiang Yan, Hancong Kong, and Robert H Deng. 2015. Seeing your face is not enough: An inertial sensor-based liveness detection for face authentication. In *CCS 2015*. ACM, 1558–1569.
[9] Gang Pan, Lin Sun, Zhaohui Wu, and Shihong Lao. 2007. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *ICCV 2007*. 1–8.
[10] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M Seitz. 2014. Total moving face reconstruction. In *European Conference on Computer Vision*. Springer, 796–812.
[11] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2015. What makes tom hanks look like tom hanks. In *ICCV*. 3952–3960.
[12] Di Tang, Zhe Zhou, Yinqian Zhang, and Kehuan Zhang. 2018. Face Flashing: a Secure Liveness Detection Protocol based on Light Reflections. In *NDSS 2018*. Internet Society.
[13] Paul Viola and Michael J Jones. 2004. Robust real-time face detection. *International journal of computer vision* 57, 2 (2004), 137–154.
[14] Joseph Wilder, P Jonathon Phillips, Cunhong Jiang, and Stephen Wiener. 1996. Comparison of visible and infra-red imagery for face recognition. In *FG 1996*. IEEE, 182–187.
[15] Xuehan Xiong and Fernando De la Torre. 2013. Supervised descent method and its applications to face alignment. In *CVPR 2013*. IEEE, 532–539.
[16] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos. In *USENIX security symposium*. 497–512.