Computers and Electrical Engineering 72 (2018) 39–52

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

A Survey Of mobile face biometrics

Ajita Rattani*, Reza Derakhshani

Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, USA

ARTICLE INFO

Article history:
Received 21 March 2017
Revised 4 September 2018
Accepted 5 September 2018

Keywords:
Biometrics
Face detection
Face recognition
Face spoof attacks
Mobile devices

ABSTRACT

Face biometrics have attracted significant attention as a technology for secure access to mobile devices. This is because almost all smartphones have RGB cameras suitable for capturing faces, and the required user interaction is acceptable given the popularity of 'selfies'. Most of the traditional methods for face biometrics may not be amenable to native execution on mobile hardware due to their limited memory and computing power. Consequently, a number of algorithms specifically designed or adapted to the mobile environment have been proposed for face biometrics. However, the state-of-the- art related to face biometrics in a mobile environment is not well known. This paper thoroughly and critically surveys face biometrics in terms of face detection and normalization, recognition, and anti-spoofing methods proposed for mobile devices. The overall aim is to improve understanding and discuss the advantages and limitations of the existing methods. Further, challenges and future research directions are identified for further research and development.

1. Introduction

Biometrics is the science of establishing human identity by using physical traits (such as face, fingerprint, and iris) and behavioral traits (such as keystroke and gait) [1]. With the unprecedented mobile technology revolution, an increasing number of individuals are using their smartphone for sensitive applications and transactions. However, mobile phones are misplaced, lost, and stolen more than other computing devices. Therefore, efforts have been directed at development of biometrically secured mobile access and transactions. The use of biometric technology in mobile devices has been referred to as *mobile biometrics*, encompassing both the sensors that acquire biometric signals, and software algorithms for their verification [2–4].

According to Acuity Market Intelligence Forecast[1], mobile biometric revenue is expected to surpass 33 billion dollars by 2020 not just for unlocking devices but to approve payments and as a part of multi-factor authentication services. It is estimated that by then 2.4 billion mobile biometric users will be utilizing the technology for about 825 billion transactions.[2] Along with advancement in computing and sensor technology, the rapidly expanding market for biometric authentication is exemplified by mobile wallets and other payment systems such as Apple Pay and Android Pay, along with major players like MasterCard[3] that utilize biometric authentication via smartphones.

* Corresponding author.
E-mail addresses: rattania@umkc.edu (A. Rattani), derakhshanir@umkc.edu (R. Derakhshani).

[1] http://www.acuity-mi.com/GBMR_Report.php.

[2] http://www.npr.org/blogs/alltechconsidered/2015/04/23/401466507/biometrics-may-ditchthe-password-but-not-the-hackers.

[3] http://newsroom.mastercard.com/eu/press-releases/mastercard-makes-fingerprint-and-selfie-paymenttechnology-a-reality/.

https://doi.org/10.1016/j.compeleceng.2018.09.005
0045-7906/© 2018 Elsevier Ltd. All rights reserved.

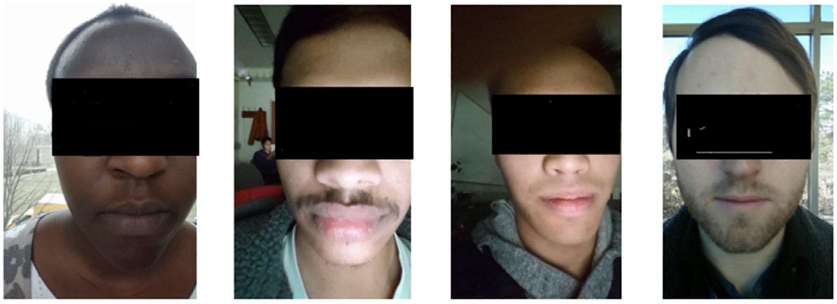


Fig. 1. Example face images acquired using the front facing camera of iPhone 5 s. The eye regions have been masked in order to preserve the privacy of users.

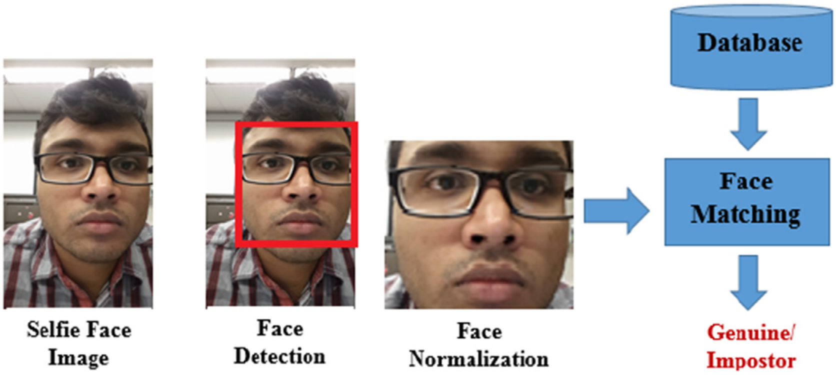


Fig. 2. Complete face recognition pipeline in mobile device.

Popular mobile biometric modalities include fingerprint, face, and ocular region consisting of iris, conjunctival vasculature, and periocular area [2–6]. Among other reasons, face biometrics has become especially attractive because of the popularity of ‘selfies’ and the fact that the face image can be acquired using only smartphone cameras instead of requiring a dedicated sensor such as in the case of fingerprint and iris. In 2016 Mobile World Congress, MasterCard[4] introduced selfie authentication based on face biometrics. Some versions of the Android mobile operating system have also used face biometrics to log in users (Google has developed “face unlocks” for Android 4.0).[5] E-commerce giant Alibaba has introduced a facial recognition service in Alipay Wallet.[6]

The general face recognition pipeline consist of the following steps: (1) face image acquisition, (2) face detection (determining if there exist a face in an image and if so segmenting it), (3) face normalization and feature extraction, and (4) identity verification by matching features from two face images: a prior enrollment vs. the presented test query. The performance of a typical face recognition system is effected by intra-class variations attributed to factors such as pose, occlusion due to prescription glasses, facial expression, make-up and illumination variations. However, existing methods for face detection and recognition may not be readily adaptable to mobile environment because of the following factors:

- Due to device mobility and operation in an uncontrolled environment, face images acquired using mobile phone’s front facing cameras are usually degraded due to factors such as specular reflection, motion-blur, illumination variation and background lighting. Therefore, more efficient and robust methods may be required for integration in the mobile devices. Fig. 1 shows sample face images acquired using the front facing camera of an iPhone 5 s.
- Although the computational power of mobile devices is growing rapidly, it still may not be sufficient for real time operation of highly accurate and computationally costly face biometrics. A good user experience requires the whole face recognition process to take under one second. Given that about half of that time is spent by the camera module to initialize, meter, and capture the image, an ideal face recognition module should take less than half a second.

Therefore, most of the proposed studies on mobile face biometrics have emphasized developing computationally efficient methods (low memory and CPU impact) for face detection and recognition [7,8]. Fig. 2 show the complete face recognition pipeline consisting of selfie face acquisition, face detection, normalization and matching with the stored template in the

[4] http://www.bbc.com/news/technology-35631456.

[5] https://www.technologyreview.com/s/425805/new-google-smart-phone-recognizes-your-face/.

[6] https://www.computerworld.com/article/2897117/alibaba-uses-facial-recognition-tech-for-online-payments.html.

dataset for authentication in mobile devices. Face normalization reduce the effect of intra-class variations such as lighting and pose variation through pre-processing and registration routines.

Further, as the use of face biometrics for smartphone user authentication continues to increase, capabilities to detect spoof attacks are needed to alleviate user concerns. A spoof attack occurs when an adversary mimics the biometric trait of another individual in order to circumvent the system for illegitimate access and advantages [9]. These attacks pose a serious threat because they can be executed at the sensor (camera) level without requiring any technical knowledge of the functioning of the biometric system. The US National Institute of Standards and Technology (NIST) now lists the vulnerability of biometrics to spoofing attacks in its National Vulnerability Database. For face biometrics, spoof attacks usually involve presenting a high quality photograph or video of the claimant's facial image [10]. The ubiquitous nature of phones with high quality cameras plus online resources such as social media has allowed attackers to acquire facial images (or even videos) of unsuspecting users easily and discretely. Lack of efficient antispoofing and liveness detection methods may create a formidable psychological barrier in the mass adoption of face biometrics in mobile applications. Therefore, there is a pressing need for development of robust countermeasures against spoof attacks for mobile face biometrics.

The goal of this survey is to advance the state-of-the-art in mobile face detection, recognition and spoof detection by improving the understanding of the existing methods. To this aim, specific contributions of this paper are as follows:

- analysis of the existing methods on mobile face pre-processing, detection, and recognition through key attributes
- tabulation of the reported results from existing literature in terms of normalization time and reported accuracy
- discussion of the countermeasures against spoof attacks for face biometrics in mobile devices

The rest of this paper is organized as follows: Section 2 discusses the proposed methods for face detection, registration, and normalization in mobile environment. Section 3 discusses the methods for face matching in mobile environment. Anti-spoofing in the context of mobile face spoof attacks are discussed in Section 4. Challenges and future research directions are discussed in Section 5. The final conclusions are drawn in Section 6.

2. Mobile face detection and normalization methods

Face detection is the first important step in face recognition. This is often implemented as a binary classification task in which a classifier trained on example faces decide whether a particular region of the image contains a face or not. In the context of mobile face detection, algorithms can be broadly classified into (a) skin-tone based [11–14], (b) machine learning-based [15–20], and (c) combination of skin-tone and machine learning-based [21–23]. Next, we discuss existing mobile face detection algorithms categorized on the basis of these attributes.

2.1. Mobile face detection methods

1. *Skin-tone based:*
   Rahman et al. [11] proposed an optimized real-time skin-color based face detection by binarizing the image with a Gaussian Mixture Model (GMM) based skin color detector. The algorithm is based on the assumption that the distribution of human skin color within an ethnicity forms a tight cluster in any chrominance color space. Therefore, it is possible to represent this cluster by a GMM. The binarized image is then passed through a fast subblock shape processing which uses face size, aspect ratio and probability scoring to determine face locations in the image. The optimization was done by using a key frame to synchronize face detection and tracking in videos, reducing the detection area by narrowing the search region for next frames to be in a neighborhood area around the detected face, using fixedpoint processing, accommodations for different lighting conditions using various look-up tables, and sub image-block processing.
   Suzuki et al. [12] proposed another face detection method based on the relationship between skin color in RGB and its brightness expressed by Y in YCbCr color space. Then, the relationship was represented with a linear function by the recursive least square algorithm. Authors calculated Y values of an image to get appropriate RGB values of skin colors in some lighting conditions. By using the RGB threshold functions (linear functions), candidate facial areas were found.
   Che and Chang [13] used YCbCr space using look-up table indexed by Cr and stored three 16-bit data fitting for the Cb range. Each bit represented whether the pixel is a skin color pixel or not under the corresponding Cb and Cr. This was followed by thresholding and integral projection of the generated binary image.
   Yu [14] proposed face detection method by converting RGB image to LAB color space. Then two global thresholds were selected based on A and B space. Pixels with A and B values above specific thresholds were determined as skin pixels.
2. *Machine learning-based:* Most surveyed literature in mobile face detection [15–20,24–27] report standard Adaboost based Viola-Jones face detector by exhaustively scanning face images at different scales. The sequence of scanned windows, typically sized between 16 × 16 and 48 × 48 pixels, have been pre-processed using histogram equalization, brightness, and contrast adjustment followed by feature extraction. A set of Haar-wavelets computed from integral images are input as features to Adaboost classifier followed by classifier cascading to boost detection rate. In fact, many embedded computers

and mobile devices can run this detector efficiently and in real time. Worth mentioning, open source software packages such as openCV[7] and Dlib[8] provide implementation of Viola-Jones based face detector.

Studies in [7,8,28] proposed optimizations to standard Viola- Jones face detector for mobile platforms to enable frame rate speeds. The optimizations introduced were in terms of data reduction (by sub-sampling and fixing step, scale, and minimum face size), search reduction by using key frames, and fixed-point Q format computations. It was shown that by reducing data search, through fixing the minimum size of the face image and through sub-sampling, and by performing computations in fixed-point, realtime throughput could be achieved without using any dedicated hardware.

Ren et al. [3] proposed a subspace face/eye detector by studying the eigenspectrum of the covariance matrices of face/non-face class and eye/non-eye class. The proposed algorithm performed Asymmetric PCA (APCA) to remove unreliable dimensions. This was followed by performing Asymmetric Discriminant Analysis (ADA) in the APCA subspace to choose generalized eigenvectors corresponding to large eigenvalues. Then, the first d_i dimensions were used to construct the face detector at ith-stage. Lastly, the scanning window with the maximum score was detected as the face window.

Farrugia et al. [29] proposed Convolutional Face Finder (CFF) that is an image-based neural network approach which allows robust detection of multiple semi-frontal faces of variable sizes and appearances in real-world images. The CFF is mainly based on convolution and sub-sampling operations.

Studies in [30–32] used combination of Modified Census Transform (MCT) along with Adaboost classifier for face detection in mobile environment. MCT presents the structural information of the local regions in an image with binary patterns providing information about edges, contours and intersections.

3. *Combination of skin-tone and machine learning-based:* Tresadern et al. [21], Darwaish et al. [22] and Hu et al. [23] used LBP extracted from local image regions in combination with Adaboost for face detection. Ciaramello and Hemami [33] and Rahman et al. [34] proposed a combination of skin color, texture, and Viola-Jones for face detection in mobile video conferencing. Skin-color segmentation was implemented by thresholding the Mahalanobis distance computed between a given pixel's chrominance value and the skin pixel distribution modeled as a bivariate Gaussian distribution. Face detection using texture analysis was performed using Chi-square distance between LBP-histogram and the average face histogram.

Using skin-color as a feature for face detection has an advantage that color processing is much faster than processing facial features. However, face detection using skin-color as a feature is highly impacted by factors such as illumination variation, specular-reflection, and motion-blur. Further, different cameras produce significantly different color values even for the same person under the same lighting condition and skin color differs from person to person mostly due to white balance inaccuracies. The advantages of machine learning-based over skin-color based face detection methods are efficient feature selection, and less sensitivity to illumination variations and affine transformations. The disadvantages include higher computational complexity, proneness to over-fitting, and ineffectiveness to pose variations. Skin-tone combined with learning-based methods has the advantage of skin-color and machine learning-based methods for better detection accuracy at the cost of higher computational complexity.

Table 1 categorizes existing mobile face detection methods into skin-tone based, machine-learning based and their combination and tabulates reported results in terms of required processing time and the obtained accuracy. In this table, NA entries indicate nonavailability of the information in the associated research papers. Skin-color based methods used pixel thresholding for face detection and Viola-Jones face detector is based on a combination of Haar Wavelet and Adaboost classifier. As can be seen in Table 1, maximum face detection accuracy of 99.7% and fastest processing time of 0.006 s is reported for machine-learning based Modified Census Transform along with Adaboost classifier [30] on an in-house database. The average reported face detection accuracy is 89% and a processing time of about 5.56 s.

2.2. Mobile face image registration and normalization methods

Face image normalization routine aims to mitigate the effect of illumination and pose variation through pre-processing and registration routines. The registration and normalization routine is usually performed after face detection.

Study in [19] used Self Quotient Image (SQI) algorithm for face image pre-processing against illumination normalization.

In [8,18,23], histogram equalization was performed on the gray scale image to compensate for under-and over exposure for better edge detection. Chen et al. [8] performed face registration using location of the eyes. Histogram equalization was applied to mitigate adverse impact of illumination.

In [7], after face detection, the face image was aligned based on facial features extracted using Viola-Jones detector. In [22], face normalization consist of face alignment, background removal and illumination normalization. The face alignment was done using the eye positions detected using the trained Viola-Jones detector. Aligned faces were preprocessed using gamma correction, difference of gaussian blur and contrast equalization.

Tresadern et al. [21] used Active Appearance Model (AAM) for face registration. This was followed by background information removal, and face warping to correct unwanted distortions (e.g., due to expression). Finally, the image was

[7] http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html.

[8] http://dlib.net/face_detector.py.html.

Table 1
Reported results of mobile face detection methods.

Reference	Categorization	Database (total images × no. of subjects)	Face Detection Method	Processing time (in seconds (s))	Accuracy [%]
[29]	Machine Learning-based	In-house (FPGA implementation)	Multi-layer Perceptron	0.03 s	87.0
[11]	Skin-tone based	In-house (100 × 5) OMAP3430 mobile platform	Skin-tone + GMM	52.9 s	88.5
[13]	Skin-tone based	In-house (FPGA implementation)	Skin-tone	0.096 s	NA
[30]	Machine Learning-based	In-house (FPGA implementation)	MCT + Adaboost	0.006 s	99.7
[33]	Skin-tone + Machine Learning-based	In-house(10 × 3) HTC Apache PocketPC	Skin-tone + LBP + Viola-Jones	15.08 s	90.0
[27]	Machine Learning-based	In-house Verilog HDL and FPGA implementation	Viola-Jones	0.23 s	NA
[14]	Skin-tone based	In-house (80 × 4)	Skin-tone	0.88 s	80.0
[31]	Machine Learning-based	In-house Google Android platform	MCT + Adaboost	0.9 s	87.0
[34]	Skin-tone + Machine Learning-based	In-house (500 face and non-face) OMAP3430 mobile platform	Haar + LBP + Adaboost	0.05 *s*	93.0
[3]	Machine Learning-based	O2FN (2000 × 50) AR (1050 × 75) CAS-PEAL (30863 × 1040)	Asymmetric PCA + LDA	0.11 s	NA
[7]	Machine Learning-based	YaleB (576 × 10) Eten M600 PDA (1800 × 20)	Optimized Viola-Jones	NA	85.0
[2]	Machine Learning-based	In-house (126 × 7) Google x92 s Android	Eye coordinates+ Skin-tone	1.2 s	96.0
[14]	Skin-tone based	In-house (80 × 4)	Skin-tone	0.88 s	80.0
[16]	Machine Learning-based	In-house (114 × 38) Samsung Nexus	Viola-Jones	NA	90.5
[24]	Machine Learning-based	In-house HTC Desire, Samsung Galaxy Tab	Viola-Jones	0.03 s	NA

normalized to standardized brightness and contrast level. Cheng and Wang [17] registered and normalized the detected face using facial landmarks such as eyes, nose, and mouth.

Different from other works, Wasnik et al. [35] proposed a method for face image quality assessment in smartphone devices. The commercial VeriLook Face SDK[9] was used to compute the initial quality scores from face images to classify the images as good or bad quality. International Standard Organization (ISO) image quality metric [36] was used as a feature vector from the classified images to train a random forest classifier.

3. Mobile face recognition methods

After face detection and normalization, the identity of the subject is verified by classifying a test image as belonging to the same (genuine) or different identity (impostor).

The existing literature on mobile face recognition can be broadly categorized into (a) *client-server based* or (b) *device-based*. In the client-server approach, face acquisition, face detection and sometimes feature extraction are performed on the device side. The remaining computationally intensive tasks such as classifier training and recognition are performed on the server. In the device-based approach, all of the operations are performed within the device.

These proposed face recognition methods can also be further classified based on the type of representation used i.e., global feature-based and local feature-based methods [37–40]. Next we review the existing literature based on this criteria.

1. *Client-server based*:
   Global feature-based: Studies reported in [14–16,20,26] used PCA for face recognition in mobile environment using a client-server architecture. In most of these studies [15,20], face detection was performed on the mobile device using the Viola-Jones face detector. After face detection, compressed version of the image was transmitted to the server. Eigenspace decomposition and recognition was thus performed at the server side. Finally, authentication results were transmitted to the mobile device. Yu et al. [14] used skin color based face detection instead of Viola-Jones. In addition, Kremic et al. [26] also used public key cryptography to ensure security of the user data (face images) over the network while being

[9] http://www.neurotechnology.com/verilook.html.

transmitted to the server. The public keys were stored in public key repository at the server side. The final classification was done using Euclidean-distance. Hu et al. [41] proposed a 3 factor authentication system based on a password, a universal subscriber identity module (usim) and a face authentication method using a PCA-based client-server architecture. The user's face was captured via the front camera of the mobile device, which was then sent to the server. Face detection, feature extraction and matching with the stored template was performed at the server side. On successful matching of the features, a one-time password was generated and sent to user's mobile number via SMS.

Imaizumi and Moshnyaga [25] used combination of PCA and LDA for face recognition using the client-server model. PCA was used for dimensionality reduction and LDA was used for face recognition. The mobile device was used only as an input and output interface. Face detection and recognition was performed on the server. Face detection was accomplished using Viola-Jones based face detector. The final classification was done using Euclidean-distance classifier. The efficacy of PCA and LDA are dependent on the total amount of information contained in the number of principal components chosen to represent the subspace.

Local feature-based: In [42], a prototype model for face recognition was built for android devices. The images captured by mobile phones were matched with those stored on a remote server using LBP features. The database was stored on a remote server and a mobile device was used for image capture, feature extraction, and matching. The performance of LBP is impacted by the size of the local window used for feature extraction.

2. *Device-oriented based*

Global feature-based: Ren et al. [3] evaluated various subspace methods such as PCA, Bayesian-based, Enhanced Maximum Likelihood and Dual-space LDA for face recognition at the device side. The face detection was performed using the Viola-Jones method. The final classification was done using Euclidean-distance. In [43] a three factor authentication prototype model consisting of PCA based face recognition, personal identification number (PIN) and international mobile equipment identity (IMEI) number, was proposed for a mobile banking application in android based smartphones.

In [31], a combination of standard PCA and LDA was used for face recognition. The face detection was performed using Modified Census Transform (MCT) along with Adaboost classifier. Further, comparative analysis of PCA, LDA and HMM was performed for face recognition in mobile environments. The final classification was done using Euclidean-distance classifier.

Local feature-based: Studies in [17,28] used Gabor features for face matching. Using GPU acceleration, both the face image and the Gabor kernels were multiplied in Fourier domain for faster convolution in this study. The final classification was done using a k-nearest neighbor classifier.

Oh et al. [44] extracted Gabor features from landmark points located in facial images. The computational overhead in extracting Gabor features was reduced by generating the Gabor features only at some specific facial landmarks on the facial image. LDA was performed on the extracted Gabor features to obtain the final feature vector. The final classification was performed using Euclidean-distance between feature vectors from two face images.

Marsico et al. [19] used spatial correlation between sub-regions in two face images for face recognition in mobile environment. For each sub-region in a face image, the region that maximizes the correlation coefficient around corresponding position in another face image was selected. The global correlation between two face images, computed as a sum of these local correlations, was used for matching two face images.

Tao and Veldhuis [7,8,22–24,45] used LBP extracted from local sub-regions in the face image by comparing each pixel to its neighbors and accordingly assigning binary values. The histogram was computed over each sub-region and finally, histograms of all the sub-regions were concatenated. The impact of the size of the local window on the performance of the face recognition system was not discussed by the authors. The similarity between two images was established using Chi-square based histogram matching.

In [46], Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and BSIF feature representations were used. The final classification was performed using a nearest-neighbor classifier and histogram matching. In another work by the same author [46], face biometric was fused with periocular region for performance enhancement using the same feature representation for smartphones.

The performance of geometry-based methods is dependent on accurate localization of the landmark points. Further, the efficacy of spatial correlation-based methods is dependent on efficient registration and alignment of the face images to be matched.

Off late, deep learning CNN solutions have been successfully ported into mobile phones and they are working with very high accuracy and speed both on device- and server side [47–50] applications. One widely deployed commercial example is Face++(https://www.faceplusplus.com/). OpenFace [51] is an open source Python and Torch implementation of face recognition with deep neural networks for mobile devices.

In 2013, evaluation of eight face recognition methods was conducted in mobile environment using MOBIO dataset [47] acquired using NOKIA N93*i* and 2008 MacBook. As a baseline, face recognition based on combination of PCA and LDA along with Cosine similarity measure was used for comparison. The performance of this baseline was compared against eight face recognition methods, namely, (a) LBP with Chi-square distance, (b) CNN with dot product, (c) Local Gabor Binary Pattern (LGBP) with Partial Least Square, (d) Gabor Jets using Elastic Bunch Graph Matching with dot product, (e) fusion of Gabor Wavelet and Local Phase Quantization along with histogram matching, (f) Patterns of Oriented Edge Magnitudes (POEM) and Gabor Jets with correlation measure, (g) variant of LBPs namely, Directional and Multispectral LBP, and (h) temporal

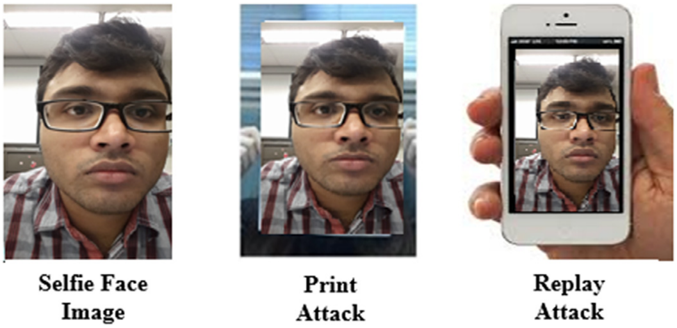


Fig. 3. Example of print and replay attacks for face biometrics in mobile device.

fusion of intensity values, Gabor features and LBP, along with PCA dimensionality reduction and cosine similarity measure for recognition. The method based on temporal fusion of intensity values, Gabor features and LBP, along with PCA-based dimensionality reduction and cosine similarity measure outperformed all other methods.

The benefits of the client-server based models, a.k.a. cloud-based biometrics, include relieving the handset from heavy computations that are required for the most accurate solutions. This also enables service providers to support a larger swath of lower end devices which is the hallmark of massive cost-conscious markets. High-security sectors such as fintech companies may also prefer having at least the latter part of the face recognition pipeline to be executed on their secure servers to minimize the chances of foul play on the device side. The downside of server-side solutions include long and, at times, unpredictable delays due to network congestion, or worse, complete lack of service due to data connection issues. Other issues include privacy concerns, especially when the image itself is uploaded to the server. Local-feature based methods are applicable even for a single template and are cost-efficient. Therefore, they have been used mainly for device side face recognition methods. Nowadays, flagship handsets are offered with much better CPUs and GPUs, as well as hardware-based secure computing capabilities, making device-based solutions a more attractive option. With advancement in mobile technology, hybrid and CNN-based face recognition have become feasible solutions for device-side applications as well. The industry has taken the lead in deep learning based mobile face recognition with both Apple[10] and many Android mobile devices[11] already taking advantage of hardware-accelerated deep learning for fast, accurate, and private on-device face recognition.

Table 2 categorizes the existing mobile face recognition methods into client-server and device-oriented based and tabulates the reported accuracies. The maximum reported face recognition accuracy of 96.0% with processing time of 0.1 s was obtained using combination of PCA and LDA along with Euclidean distance [25]. However, the proposed method was evaluated on in-house database captured using Sony Tablet-S consisting of 60 samples from 6 subjects. The average reported accuracy is 83.6% with 1.64 s of processing time. EER is the rate at which False Accept Rate (FAR) is equal to False Reject Rate (FRR) of the classifier. In an independent evaluation [47] in 2016, the method based on temporal fusion of intensity values with Gabor and LBP outperformed all other methods by obtaining an EER of 2.2% with processing time of 0.0014 s. The combination of PCA and LDA was used as a baseline and obtained EER of 14.8% with processing time of 0.0055 s on the same dataset.

4. Counter-measures for mobile face spoof attacks

Vulnerability of biometric system to spoof attacks is well-known [54–57]. Despite progress in anti-spoofing methods, face spoofing still pose a major threat to face recognition system. Compared to attacks against fingerprint, iris or speech recognition systems, the ubiquitous nature of image acquisition devices such as cameras and social medias, such as facebook, allow attackers to acquire facial images of a user easily and discretely [58]. A study conducted using commercial off-the-shelf matcher demonstrated that state-of-the-art face recognition methods are fragile against spoofing attacks [59].

With the growing popularity of face recognition for access control, the topic of spoof detection has attracted significant attention over the past five years [54–57], including the EU funded FP7 project TABULA RASA[12] (trusted biometrics under spoofing attacks) [60]. Spoof attacks against face recognition system mainly consist of (i) *print attacks*, (ii) *replay attacks*, and (iii) *3D mask attacks*. Print attacks can be executed using a simple photograph of the enrolled person's face, which may be displayed in hard-copy or on a mobile screen. The video replay attacks are performed by displaying a video on a mobile screen. Fig. 3 show example of print and replay attacks for face biometrics in mobile device. In contrast, 3D face mask attacks require high resolution fabrication system capturing the 3D shape and texture information of the target subject's face. Therefore, print and replay attacks can be launched more easily by malicious users than 3D mask attacks. For this reason, existing literature mainly focus on 2D face spoof attacks (print and replay attacks) over 3D spoof attacks.

[10] https://www.apple.com/.

[11] https://developer.qualcomm.com/software/snapdragon-neural-processing-engine-ai.

[12] http://www.tabularasa-euproject.org/.

Table 2
Results of mobile face recognition methods.

Reference	Database (total images × no. of subjects)	Client-server/Device -oriented	Categorization	Face recognition method	Processing time (in seconds (s))	Accuracy or EER [%]
[6]	AR (4000 × 126) ATT (400 × 40) EYFD (16128 × 38) ETRI (1100 × 55) BIOID (1521 × 23)	Device-oriented	Hybrid	Local Random Basis + Neural Network	0.06 s	70.0 87.0 74.0 80.0 82.0
[3]	O2FN (2000 × 50) AR (1050 × 14) CAS-PEAL (30863 × 1040)	Device-oriented	Global	Subspace Classifiers (PCA and Bayesian) + Euclidean Distance	0.77 s	3.8 EER 1.9 EER 5.6 EER
[19]	In-house (98 × 49) Samsung Galaxy Tab 2.0	Device-oriented	Local	Spatial Correlation	NA	84.0
[7]	YaleB (576 × 10) Eten M600 PDA (1800 × 20)	Device-oriented	Local	LBP + Chi-square	0.13 s	5.0 EER
[24]	In-house Samsung Galaxy	Device-oriented	Local	LBP + Chi-square	0.13 s	6.9 EER
[25]	In-house (60 × 6) (Sony Tablet-S)	Device-oriented	Global	PCA+ LDA Euclidean Distance	0.1 s	96.0
[2]	In-house (126 × 7)	Client-server oriented	Global	PCA + Euclidean Distance	NA	80.0
[45]	MOBIO (150 × 61)	Device-oriented	Local	LBP + Chi-square	NA	11.0 HTER
[16]	In-house (114 × 38) Samsung Nexus	Client-server oriented	Global	PCA + Euclidean Distance	NA	78.5
[21]	MOBIO (150 × 61)	Device-oriented	Local	LBP + Chi-square	NA	4.5 EER
[44]	ORL (8586 × 159) Web images(100 × 10)	Device-oriented	Local	Gabor Features + LDA + Nearest Neighbor	0.01 s	62.0
[17]	In-house	Device-oriented	Local	Gabor Feature + Nearest Neighbor	4.6 s	93.8
[52]	In-house	Device-oriented	Local	Gabor Feature + Nearest Neighbor	8.5 s	92.8
[8]	In-house (1000 × 10) Nokia 5230 mobile phone	Device-oriented	Local	LBP + Chi-square	0.5 s	87.0
[26]	In-house (680 × 34) Samsung Galaxy	Client-server oriented	Global	PCA + Euclidean distance	NA	88.8
[8]	In-house (1000 × 10) Nokia 5230	Device-oriented	Local	LBP + Chi-square	NA	87.0
[53]	In-house Google Nexus (17 × 6)	Client-server or device-oriented	Global	Sparse Representation Classifier	NA	82.4
[46]	In-house (78 × 15) Samsung Galaxy S5 Samsung Galaxy Note 10.1	Device-oriented	Local	SIFT, SURF and BSIF	NA	3.6EER
[23]	In-house (15 × 10) Google Nexus 5	Client-server oriented	Global	PCA + Euclidean Distance	NA	96.0
[47]	MOBIO (152 × 72) (NOKIA N93i+ 2008 MacBook)	Device-oriented	Local	PCA+ LDA + Cosine measure LBP+ Chi-square CNN + Dot Product LGBP + Partial Least Square Gabor Jets + Dot Product Gabor Wavelets +LPQ + Histogram Matching POEM + Gabor Jets + Correlation Measure LBP+ DLBP + MSLBP + SVM Intensity Values+ Gabor Features + LBP with PCA and Cosine Similarity	0.0055 0.0009 0.015 0.0039 0.013 0.0012 0.0021 0.0026 0.0014	14.8 EER 9.2 EER 4.1 EER 7.9 EER 6.2 EER 6.7 EER 4.2 EER 5.4 EER 2.2 EER

Face spoofing countermeasures aim to disambiguate live and real face captures from spoof counterparts in order to avoid spoof attacks. These countermeasures are either based on detecting life-like signals (blood pressure and motion) or the artifacts introduced due to fake fabrication process (presence of noise and change in texture). The countermeasures based on detecting life-like signals are often called liveness detection methods and those based on detecting fake artifacts are called spoof detection methods.

The existing countermeasures can be coarsely classified into motion analysis-based [61–63], texture-based [54,64–68], image-quality based [58,66,69,70], and deep learning-based [71]. Motion analysis based methods are liveness detection, and texture, image-quality and deep learning-based methods are spoof detection methods. Recently in 2017, an international competition was organized to compare the generalization performance of mobile face anti-spoofing countermeasures on OULU-NPU face spoof attack dataset [71]. Various teams from academic and industrial institutions across the world participated in this competition by submitting texture-, image quality and deep learning-based methods as countermeasures against print and replay attacks on mobile device. Next, we discuss the existing methods based on the above mentioned categorization.

1. *Motion-based:* Motion analysis based methods detect visible movements in a live face, such as eye blink [72], mouth movement [56], and head rotation [56].
   Patel et al. [72] proposed non-learning based eye-blink detection for liveness detection of face images. The proposed method was based on calculating the difference image between successive frames of eye region. The eye state was determined by binarizing the difference image and counting the percentage of white pixels. The proposed eye-blink based liveness detection method was further combined with deep texture features for performance enhancement in cross-dataset scenario.
   Siddiqui et al. [61] proposed Histogram of Oriented Optical Flow (HOOF) features coupled with SVM for motion estimation from face videos. The proposed motion estimation method was combined with LBP-based texture features for further performance enhancement.
   Tirunagari et al. [62] proposed a method based on Dynamic Mode Decomposition (DMD) along with SVM for motion estimation from face videos. Authors have combined DMD with LBP for further performance enhancement.
   Pinto et al. [63] used time-spectral feature descriptors from the videos that gather temporal and spectral information across the biometric sample and used visual codebook concept to find mid-level feature descriptors computed from these low-level ones. These features were coupled with SVM for liveness detection.
   Further, Face++[13] (face identification technology) use lip reading verification for liveness detection of face videos. Given that detecting motion usually do not require learning computational complex solutions, these methods are expected to generalize better. One of the major limitations of the motion-based methods are that they can be circumvented or confused by other kind of motions such as background motion, camera motion, image warping, shaking, and, most importantly, video replay attacks.
2. *Texture-based*: Texture-based methods aim to detect the artifacts introduced in the images due to the fake fabrication process. For instance, some characteristics of recaptured 2D prints such as sharpness may be lost. Thus, texture and frequency components can be used to detect spoofing artifacts from live face images. Researchers have shown steady progress in developing texture-based anti-spoofing methods for 2*D* face spoof attacks [54–57,73]. Next, we review the texture-based face spoof detection methods for smartphones.
   Akhtar et al. [64] used the combination of textural features to detect spoofing attacks. Specifically, authors presented a multi-biometric approach that detects face, iris, and fingerprint spoof attacks in mobile applications by employing texture descriptors such as Locally Uniform Comparison Image Descriptor (LUCID), Census Transform Histogram (CENTRIST) and POEM that do not require floating point operations. The efficacy of the proposed method was evaluated on heterogeneous dataset.
   LBP texture descriptor along with SVM was used as a baseline anti-spoofing countermeasure against video replay attacks in Replay-Attack dataset [54]. LBP texture descriptor along with SVM was also used as a baseline anti-spoofing countermeasure against print and video replay attacks in OULU-NPU dataset [65].
   Costa-Pazo et al. [66] computed Gabor Jets over a regular 10×10 grid using 40 Gabor Wavelets with default parametrization. The feature vectors were extracted from face region detected using Viola-Jones face detector. The SVM classifier trained on these features was used to classify print and video attacks against live face images.
   Boulkenafet et al. [65] analyzed the joint color-texture information from the luminance and the chrominance channels in an image using a color LBP descriptor. Experimental investigation on Replay-Attack [54] database suggest 14.9% reduction in EER using color LBP on YCbCr and HSV space in comparison to gray-scale LBP.
   Patel et al. [58] used combination of multi-scale LBP and Dense Scale Invariant Feature Transform (DSIFT) to represent moire patterns that differentiate replayed spoof face from a live face on MSU Face Spoof database. Gan et al. [67] and Atoum et al. [68] used feature representation extracted with CNN along with SVM evaluated on Replay-Attack database and MSU Spoof Face database, respectively.
   In an international competition in 2017 [71], color LBP extracted and concatenated from HSV and YCbCr color space and fed into a softmax classifier was used as a baseline method. Other texture-based methods include Local Phase

[13] https://www.faceplusplus.com/.

Quantization (LPQ) along with Softmax classifier, Pyramid multi-level representation with LPQ and SVM, Binary Statistical Image Features (BSIF) with Softmax classifier.

3. *Image-quality based:* These methods classify input image as live or fake based on the quality of the image [74]. The underlying concept is that the quality of the fake images are usually lower than the live counterpart due to the degradations introduced during the fake fabrication process such as reduced sharpness.
   Wen et al. [69] proposed face spoof detection algorithm based on Image Distortion Analysis (IDA) for mobile devices. Four different features (specular-reflection, blurriness, chromatic moment, and color diversity) were extracted to form the IDA feature vector. An ensemble classifier, consisting of multi-class SVM trained for printed photo and replayed video attacks, was used to distinguish live face images from spoof attacks on MSU Mobile Spoof Database. The proposed approach was extended to multi-frame face spoof detection for videos using a voting-based scheme.
   Costa-Pazo et al. [66] proposed a set of 18 image-quality measures such as Mean squared Error, Peak Signal to Noise Ratio, Average Difference, Structural Content, Cross-correlation, Signal-to-Noise Ratio, Mean Angle Similarity, Spectral Magnitude Error, and Structural Similarity Index along with SVM for face spoof detection evaluated on Replay-Mobile dataset.
   Galbally and Marcel [70] used set of 14 image-quality measures such as Mean-squared Error, Peak Signal to Noise Ratio, Structural Content, and Maximum Difference along with SVM for face spoof detection evaluated on Replay-Attack dataset.
4. *Deep-learning based:* Various CNN-based mobile face spoof detection methods were submitted in a recent international competition in [71]. These methods are discussed in this section as follows. Two CNN models consisting of five convolution layers along with fully connected layers were used for face spoof attack detection. The output of the last fully connected layer was used as an input to a Softmax classifier. One of the submitted methods used CNN model pre-trained on face and non-face images as feature extractor. The extracted features are fed into multi-layer perceptron (MLP) for classification. One of the methods used end-to-end deep learning model that was trained on the provided training set using back propagation algorithm and the error terms of the models were generated by the LBP features extracted from the convolutional layers. In testing stage, the obtained LBP features were fed into an SVM classifier. Other submitted methods used transfer learning where SqueezeNet and InceptionNet were fine-tuned for the task of mobile face spoof detection.
5. *Combination-based:* A combination of image-quality [66] based, multi-scale LBP and deep features fused using multiple kernel learning was submitted in international competition in [71]. In another method, combination of motion, LBP, and quality-based methods [70] along with Gaussian Mixture Model (GMM) was used for mobile face spoof detection.

MSU Mobile Spoof [69], OULU-NPU [65], Replay-attack [54] and Replay-mobile attack [66] are the publicly available spoof databases consisting of print and video replay attacks generated using mobile devices for face biometrics. These datasets are explained as follows:

- *MSU Mobile Spoof* [69] dataset consists of 280 video clips of photo and video attack attempts from 35 subjects at Michigan State University. Real access videos (on an average 12 s long) have been captured using two devices: a 13 MacBook Air (using it’s built-in camera) and a Google Nexus 5 (Android 4.4) phone. Videos captured using the laptop camera have a resolution of 640 × 480, and those captured using the Android camera have a resolution of 720 × 480. Three kind of spoof-attacks included in the database were: print attacks, video replays on a smartphone (iPhone 5s), and high-definition (HD) video replays (captured on a Canon 550 DSLR and played back on an iPad Air).
- *OULU-NPU* [65] dataset include short video sequences of real access and attack attempts corresponding to 55 subjects. The data was collected using six smartphones (Samsung Galaxy S6, HTC Desire EYE, MEIZU X5, ASUS Zenfone Selfie, Sony XPERIA and Oppo N3). The attack types considered in this database are print and video-replay attacks. The high resolution photos were printed on A3 glossy paper using two different printers (a Canon imagePRESS C6011 and a Canon PIXMA) to generate print attacks. The high-resolution videos were replayed on two different display devices (a 19 Dell Ultra Sharp 1905 FP display with 1280 × 1024 resolution and an early 2015 Macbook 13 laptop with retina display of 2560 × 1600 resolution) to generate replay attacks.
- *Replay-Attack and Replay-Mobile* Replay-Attack face dataset [54] consisting of short video recordings of both real-access and attack attempts to 50 different identities was introduced. Replay attacks using mobile device was performed by acquiring high resolution videos with an iPhone 3GS (3.1 megapixel camera) and displaying them using the iPhone screen. Replay- Mobile [66] dataset is a publicly available dataset that include both real-access and attack attempts from 40 different subjects. For photo-based attacks, a Nikon Coolpix P520 camera was used to capture high resolution images. Video-based attacks were recorded by using the back camera of the LG-G4 smartphone which records 1080 p full HD video clips.

Table 3 summarizes performance of proposed countermeasures against face spoof attacks in mobile devices. It can be seen that methods based on image quality analysis for replay-attacks obtained EER of 6.8% [69] on MSU Mobile Face Spoofing Database. Texture-based methods obtained EER of 2.8% and 5.5% for print and replay attacks, respectively, on the assembled dataset. Evaluation of texture-and image-quality based methods on Replay-Mobile dataset suggest superiority of the former for both print and replay attacks. This suggest that replay attacks are difficult to circumvent than print attacks. The average obtained EER of motion-based, texture-based and image-quality based methods are 4.6%, 8.4% and 10.8%, respectively. In an international competition in 2017 [71], a method based on feature extraction using pre-trained CNN with MLP

Table 3
Reported results of countermeasures against mobile face spoof attacks.

Reference	Anti-spoof/ Liveness detection method	Dataset	Method	Type of Attack	EER [%]
[72]		Replay-Attack	Eye blink + SVM	Replay Attack	12.4
		MSU Mobile Spoof	Eye blink + SVM	Replay Attack	9.1
[61]	Motion-based	Replay-Attack	HOOF+LBP+SVM	Replay Attack	0.0
[62]		Replay-Attack	DMD+LBP+SVM	Replay Attack	3.7
[63]		Replay-Attack	Time-spectral+Visual Codebook	Replay Attack	2.6
[63]		Print Attack	LUCID+LBP+SVM	Print Attack	2.8
		Replay-Attack	POEM+CENTRIST	Replay Attack	5.5
		NUAA	POEM+CENTRIST	Replay Attack	1.5
	Texture-based	NUAA	POEM+CENTRIST	Print Attack	7.1
[65]		OULU-NPU	LBP+SVM	Replay Attack	8.0
		OULU-NPU	LBP+SVM	Print Attack	23.2
		OULU-NPU	LBP+SVM	Replay Attack	24.2
[54]		Replay-Attack	LBP+SVM	Replay Attack	15.0
[58]	Texture-based	MSU Mobile Spoof	MLBP+DSIF+SVM	Replay Attack	70.0 (Accuracy)
[67]		Replay-Attack	CNN-features + SVM	Replay Attack	0.2
[68]		Replay-Attack	CNN-features + SVM	Replay Attack	0.4
[69]	Image quality-based	MSU Mobile Spoof	Specular Reflection +Blurriness Chromatic Moment + Color Diversity + SVM	Replay Attack	6.8
[70]		Replay-Attack	Image Quality Measure + SVM	Replay Attack	15.2
	Image quality-based	Replay-Attack	Image Quality Measure + SVM	Print Attack	7.7
[66]		Replay-Mobile	18 Image Quality Measure + SVM	Replay Attack	13.6
			Gabor Jets + SVM	Print Attack	8.6
	Texture-based	Replay-Mobile	Gabor Jets + SVM	Replay Attack	9.5
			LBP + SVM	Print Attack	3.8
			LBP + SVM	Replay Attack	7.1
			Pyramid LPQ + Softmax	Print Attack	11.3
			Pyramid LPQ + Softmax	Replay Attack	7.5
	Texture-based	OULU-NPU	Color LBP + Softmax	Print Attack	3.3
			Color LBP + Softmax	Replay Attack	5.4
			BSIF + Softmax	Print Attack	0.4
			BSIF + Softmax	Replay Attack	3.3
			LPQ + YCbCr	Print Attack	44.2
			LPQ + YCbCr	Replay Attack	31.7
[71]			SqueezeNet	Print Attack	0.8
			SqueezeNet	Replay Attack	3.3
			CNN features+ MLP	Print Attack	0.0
	Deep Learning-based	OULU-NPU	CNN features+ MLP	Replay Attack	3.0
			CNN + LBP	Print Attack	7.5
			CNN + LBP	Replay Attack	8.8
			Motion + Texture+Quality	Print Attack	13.3
	Combination-based	OULU-NPU	Motion + Texture+Quality	Replay Attack	9.6
			Deep features+ Texture+Quality	Print Attack	7.1
			Deep features+ Texture+Quality	Replay Attack	9.6

classifier outperformed other methods for print and replay mobile spoof attacks. Large-scale evaluation on a common test set should be conducted for statistical significance of the obtained results.

5. Challenges and future research directions

One of the main challenges involve developing accurate and computationally efficient methods for face biometrics in mobile environment. Table 1, suggest maximum face detection accuracy of 99.7% and fastest processing time of 0.006 s. The average reported face detection accuracy is 89.0% and a processing time of about 5.56 s. Table 2 suggest maximum reported face recognition accuracy of 96.0% with processing time of 0.1 s. The average reported accuracy is 83.6% with 1.64 s of processing time. However, as most of the existing methods are either evaluated on in-house mobile dataset of very small size, relevance of the reported results cannot be established. Large scale evaluation of existing methods on a common benchmark mobile face dataset should be conducted for statistical significance of the reported results.

Reported error rates in Table 3 regarding performance of proposed countermeasures against face spoof attacks in mobile devices are usually high, especially for replay attacks. This suggest need for advanced and accurate methods for facial liveness and spoof detection. Further, continuous advancement in spoofing techniques will lead to the use of novel methods to launch spoof attacks. Thus, there is an immediate need for designing a liveness detection/ anti-spoof method that is robust across novel spoof attacks [75]. Therefore, development of advanced and open-set liveness/ anti-spoof detection methods for known and novel facial spoof attacks should be the path forward.

With the advancement in the mobile technology, deep learning-based solutions became viable for client-oriented and device-side mobile applications. Consequently, deep learning based solutions for facial detection, recognition and anti-spoofing should be developed. Advanced loss functions such as triplet- [48] and center-loss [49] should be utilized for the task. Efforts should also be directed towards large-scale database collection for selfie face images in order to evaluate and compare deep learning solutions on common test set.

6. Conclusion

This survey suggests that existing mobile face detection methods can be categorized into skin-tone, machine-learning and those based on the combination of both. Similarly, mobile face recognition methods can be broadly categorized into client-server or device-oriented. The average reported mobile face detection and recognition accuracies of 89% and 83%, respectively, are quite low for mobile-based user authentication. Current countermeasures to face spoof attacks use motion, texture, image quality, and deep-learning based methods for print and replay attacks on a mobile device. However, the reported error rates are high, suggesting the need for further advancement. Development of advanced deep-learning based solutions for face biometrics along with evaluation on large-scale publicly available mobile datasets is the path forward.

Acknowledgement

This work was supported in part by a gift from ZOLOZ (www.zoloz.com). Authors would like to thank Brendan O'Connor for proof-reading the manuscript.

References

[1] Jain AK, Flynn P, Ross A. Handbook of biometrics. Springer; 2007.

[2] Doukas C, Maglogiannis I. A fast mobile face recognition system for android os based on eigenfaces decomposition. In: Papadopoulos H, Andreou A, Bramer M, editors. Artificial intelligence applications and innovations. IFIP Advances in Information and Communication Technology, 339. Springer Berlin Heidelberg; 2010. p. 295–302.

[3] Ren J, Jiang X, Yuan J. A complete and fully automated face verification system on mobile devices. Pattern Recognit 2013;46(1):45–56.

[4] Rattani A, Derakhshani R, Saripalle SK, Gottemukkula V. ICIP 2016 competition on mobile ocular biometric recognition. In: Proceedings of the IEEE international conference on image processing, challenge session on mobile ocular biometric recognition. Phoenix, AZ; 2016. p. 320–4.

[5] Rattani A, Derakhshani R. Ocular biometrics in the visible spectrum: a survey. Image Vis Comput 2017;59:1–16.

[6] Choi K, Toh K, Byun H. Realtime training on mobile devices for face recognition applications. Pattern Recognit 2011;44(2):386–400.

[7] Tao Q, Veldhuis R. Biometric authentication system on mobile personal devices. IEEE Trans Instrum Meas 2010;59(4):763–73.

[8] Chen B, Shen J, Sun H. A fast face recognition system on mobile phone. In: Proceedings of the international conference on systems and informatics. Yantai; 2012. p. 1783–6.

[9] Chingovska I, d Anjos AR, Marcel S. Biometrics evaluation under spoofing attacks. IEEE Trans Inf Forensics Secur 2014;9(12):2264–76.

[10] Galbally J, Marcel S, Fierrez J. Biometric antispoofing methods: a survey in face recognition. IEEE Access 2014;2:1530–52.

[11] Rahman M, Ren J, Kehtarnavaz N. Real-time implementation of robust face detection on mobile platforms. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing. Taipei; 2009. p. 1353–6.

[12] Suzuki S, Mitsukura Y, Takimoto H, Tanabata T, Kimura N, Moriya T. A human tracking mobile-robot with face detection. In: Proceedings of the annual conference of IEEE industrial electronics. Porto; 2009. p. 4217–22.

[13] Che M, Chang Y. A hardware/software co-design of a face detection algorithm based on FPGA. In: Proceedings of the international conference on measuring technology and mechatronics automation; vol. 1. Washington, DC, USA; 2010, p. 109–112.

[14] Yu H. Face recognition for mobile phone using eigenfaces. Technical Report. University of Michigan; 2010.

[15] Kumar S, Singh P, Kumar V. Architecture for mobile based face detection/ recognition. Int J Comput Sci Eng 2010;2(3):889–94.

[16] Findling RD, Mayrhofer R. Towards face unlock: on the difficulty of reliably detecting faces on mobile phones. In: Proceedings of the international conference on advances in mobile computing and multimedia. Bali, Indonesia; 2012. p. 275–80.

[17] Cheng KT, Wang Y. Using mobile GPU for general-purpose computing a case study of face recognition on smartphones. In: Proceedings of the international symposium on VLSI design, automation and test. Hsinchu; 2011. p. 1–4.

[18] Fathy ME, Patel VM, Chellappa R. Face-based active authentication on mobile devices. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing. South Brisbane, QLD; 2015. p. 1687–91.

[19] Marsico MD, Galdi C, Nappi M, Riccio D. FIRME: face and iris recognition for mobile engagement. Image Vis Comput 2014;32(12):1161–72.

[20] Walgamage T, Farook C. A real-time hybrid approach for mobile face recognition. In: Proceedings of the international conference on intelligent systems, modelling and simulation; 2014. p. 1–6.

[21] Tresadern P, McCool C, Poh N, Matejka P. Mobile biometrics (mobio):combined face and voice verification for a mobile platform. Pervasive Comput 2013;12(1):79–87.

[22] Darwaish SF, Moradian E, Rahmani T, Knauer M. Biometric identification on android smartphones. Procedia Comput Sci 2014;35:832–41.

[23] Hu J, Peng L, Zheng L. XFace: A face recognition system for android mobile phones. In: Proceedings of the IEEE international conference on cyber–physical systems, networks, and applications. Hong Kong; 2015. p. 13–18.

[24] Vazquez-Fernandez E, Garcia-Pardo H, Gonzalez-Jimenez D, Perez-Freire L. Built-in face recognition for smart photo sharing in mobile devices. In: Proceedings of the IEEE international conference on multimedia and expo. Barcelona; 2011. p. 1–4.

[25] Imaizumi K, Moshnyaga VG. Network-based system for face recognition on mobile wireless devices. In: Proceedings of the IEEE international conference consumer electronics (ICCE-Berlin). Berlin; 2013. p. 406–9.

[26] Kremic E, Subasi A, Hajdarevic K. Face recognition implementation for client server mobile architecture. In: Proceedings of the international conference on information technology interfaces. Dubrovnik, Croatia; 2012. p. 435–40.

[27] Cho J, Mirzaei S, Oberg J, Kastner R. FPGA-based face detection system using haar classifiers. In: Proceedings of the international symposium on field programmable gate arrays. NY, USA; 2009. p. 103–12.

[28] Wang Y, Donyanavard B, Cheng K. Energy-aware real-time face recognition system on mobile CPU-GPU platform. In: Proceedings of the european conference on trends and topics in computer vision - volume part II. Berlin, Heidelberg. Springer-Verlag; 2012. p. 411–22.

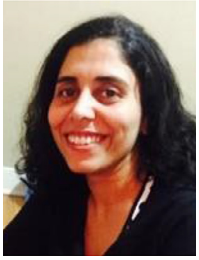
[29] Farrugia N, Mamalet F, Roux S, Yang F, Paindavoine M. Fast and robust face detection on a parallel optimized architecture implemented on FPGA. IEEE Trans Cir and Sys Video Technol 2009;19(4):597–602.

[30] Han D, Choi J, Cho J, Kwak D. Design and VLSI implementation of high-performance face-detection engine for mobile applications. In: Proceedings of the IEEE International conference on consumer electronics. Las Vegas, NV; 2011. p. 705–6.

[31] Jeong K, Han D, Moon H. Optimization of face recognition algorithms for smartphone environment. Int J Sec Appl 2013;7(6):303–8.
[32] Suguna T, Mahesh Y. Hdl based illumination invariant high performance face detection system for mobile applications. Int J Eng Res Appl 2013;3(4):147–51.
[33] Ciaramello FM, Hemami SS. Real-time face and hand detection for videoconferencing on a mobile device. In: Proceedings of the workshop on video processing and quality metrics for consumer electronics; 2009. p. 1–6.
[34] Rahman M, Kehtarnavaz N, Ren J. A hybrid face detection approach for real-time deployment on mobile devices. In: Proceedings of the international conference on image processing. Cairo; 2009. p. 3233–6.
[35] Wasnik P, Raja KB, Ramachandra R, Busch C. Assessing face image quality for smartphone based face recognition system. In: Proceedings of the international workshop on biometrics and forensics. Coventry; 2017. p. 1–6.
[36] Wueller D.. Standardization of image quality analysis ISO 19264. https://www.image-engineering.de/content/library/conference_papers/2016_04/Standardization_of_Image_Quality_Analysis%E2%80%93ISO_19264.pdf(2016); 2016.
[37] Zhao W, Chellappa R, Rosenfeld A. Face recognition: a literature survey. ACM Comput Surv 2003;35:399–458.
[38] Yang J, Chen X, Kunz W. A PDA-based face recognition system. In: Proceedings of the sixth ieee workshop on applications of computer vision; 2002. p. 19–23.
[39] Abate AF, Nappi M, Riccio D, Sabatino G. 2D And 3D face recognition: a survey. Pattern Recognit Lett 2007;28(14):1885–906.
[40] Tan X, Chen S, Zhou Z, Zhang F. Face recognition from a single image per person: a survey. Pattern Recognit 2006;39(9):1725–45.
[41] Hu JY, Sueng CC, Liao WH, Ho CC. Android-based mobile payment service protected by 3-factor authentication and virtual private adhoc networking. In: Proceedings of the IEEE computing, communications and applications conference. Hong Kong, China; 2012. p. 111–16.
[42] Rana A, Ciarulli A. Face recognition on android smartphones. Int J Intell Comput Res 2014;5(1).
[43] Gode P, Nakhate ST, Mane SS. Authentication for mobile banking by using android based smart phones. Imp J Interdiscip Res 2017;3(3).
[44] Oh J, Choi S, Kim C, Cho J, Choi C. Selective generation of gabor features for fast face recognition on mobile devices. Pattern Recognit Lett 2013;34(13):1540–7.
[45] Khoury E, Shafey LE, McCool C, Gnther M, Marcel S. Bi-modal biometric authentication on mobile phones in challenging conditions. Image Vis Comput 2014;32(12):1147–60.
[46] Raja KB, Raghavendra R, Stokkenes M, Busch C. Multi-modal authentication system for smartphones using face, iris and periocular. In: Proceedings of theinternational conference on biometrics. Phuket; 2015. p. 143–50.
[47] Gnther M, Costa-Pazo A, Ding C, Boutellaa E, Chiachia G, Zhang H, et al. The 2013 face recognition evaluation in mobile environment. In: Proceedings of the international conference on biometrics. Madrid; 2013. p. 1–7.
[48] Schroff F, Kalenichenko D, Philbin J. Facenet: a unified embedding for face recognition and clustering. CoRR 2015;abs/1503.03832:815–23.
[49] Wen Y, Zhang K, Li Z, Qiao Y. A discriminative feature learning approach for deep face recognition. In: Leibe B, Matas J, Sebe N, Welling M, editors. Proceedings of the european conference on computer vision. Cham; 2016. p. 499–515.
[50] Hu G, Yang Y, Yi D, Kittler J, Christmas WJ, Li SZ, et al. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. CoRR 2015;abs/1504.02351:384–92.
[51] Amos B, Ludwiczuk B, Satyanarayanan M. OpenFace: A general-purpose face recognition library with mobile applications. Technical Report. CMU-CS-16-118, CMU School of Computer Science; 2016.
[52] Wang YC, Cheng KT. Energy-optimized mapping of application to smartphone platform- a case study of mobile face recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition workshops. Colorado Springs, CO; 2011. p. 84–9.
[53] Raghavendra R, Raja KB, Pflug A, Yang B, Busch C. 3D face reconstruction and multimodal person identification from video captured using smartphone camera. In: Proceedings of the IEEE international conference on technologies for homeland security; 2013. p. 552–7.
[54] Chingovska I, Anjos A, Marcel S. On the effectiveness of local binary patterns in face anti-spoofing. In: Proceedings of the international conference of biometrics special interest group (BIOSIG). Germany; 2012. p. 1–7.
[55] Zhang Z, Yan J, Liu S, Lei Z, Yi D, Li SZ. A face anti-spoofing database with diverse attacks. In: Proceedings of the IEEE international conference on biometrics. New Delhi; 2012. p. 26–31.
[56] Bao W, Li H, Li N, Jiang W. A liveness detection method for face recognition based on optical flow field. In: Proceedings of the international conference on image analysis and signal processing. Taizhou; 2009. p. 233–6.
[57] Bharadwaj S, Dhamecha TI, Vatsa M, Singh R. Computationally efficient face spoofing detection with motion magnification. In: Proceedings of the computer vision and pattern recognition workshop. Portland, OR; 2013. p. 105–10.
[58] Patel K, Han H, Jain AK, Ott G. Live face video vs. spoof face video: Use of moire patterns to detect replay video attacks. In: Proceedings of the international conference on biometrics. Phuket; 2015. p. 98–105.
[59] Anjos A, Marcel S. Counter-measures to photo attacks in face recognition: A public database and a baseline. In: Proceedings of the international joint conference on biometrics. Washington, DC; 2011. p. 1–7.
[60] Chakka MM, Anjos A, Marcel S. Competition on counter measures to 2D facial spoofing attacks. In: Proceedings of the international joint conference on biometrics. Washington, DC; 2011. p. 1–6.
[61] Siddiqui TA, Bharadwaj S, Dhamecha TI, Agarwal A, Vatsa M, Singh R, et al. Face anti-spoofing with multifeature videolet aggregation. In: Proceedings of the international conference on pattern recognition. Cancun; 2016. p. 1035–40.
[62] Tirunagari S, Poh N, Windridge D, Iorliam A, Suki N, Ho ATS. Detection of face spoofing using visual dynamics. IEEE Trans Inf Forensics Secur 2015;10(4):762–77.
[63] Pinto A, Pedrini H, Schwartz WR, Rocha A. Face spoofing detection through visual codebooks of spectral temporal cubes. IEEE Trans Image Process 2015;24(12):4726–40.
[64] Akhtar Z, Michelon C, Foresti GL. Liveness detection for biometric authentication in mobile applications. In: Proceedings of the international carnahan conference on security technology. Rome; 2014. p. 1–6.
[65] Boulkenafet Z, Komulainen J, Li L, Feng X, Hadid A. OULU-NPU: A mobile face presentation attack database with real-world variations. In: Proceedings of the IEEE international conference on automatic face gesture recognition. Washington, DC; 2017. p. 612–18.
[66] Costa-Pazo A, Bhattacharjee S, Vazquez-Fernandez E, Marcel S. The replay-mobile face presentation-attack database. In: Proceedings of the international conference of the biometrics special interest group. Germany; 2016. p. 1–7.
[67] Gan J, Li S, Zhai Y, Liu C. 3D convolutional neural network based on face anti-spoofing. In: Proceedings of the international conference on multimedia and image processing. Wuhan; 2017. p. 1–5.
[68] Atoum Y, Liu Y, Jourabloo A, Liu X. Face anti-spoofing using patch and depth-based CNNs. In: Proceedings of the IEEE international joint conference on biometrics. Denver, CO; 2017. p. 319–28.
[69] Wen D, Han H, Jain AK. Face spoof detection with image distortion analysis. IEEE Trans Inf Forensics Secur 2015;10(4):746–61.
[70] Galbally J, Marcel S. Face anti-spoofing based on general image quality assessment. In: Proceedings of the international conference on pattern recognition. Stockholm; 2014. p. 1173–8.
[71] Boulkenafet Z, Komulainen J, Akhtar Z, Benlamoudi A, Samai D, Bekhouche SE, et al. A competition on generalized software-based face presentation attack detection in mobile scenarios. In: Proceedings of the IEEE international joint conference on biometrics. Denver, CO; 2017. p. 688–96.
[72] Patel K, Han H, Jain AK. Cross-database face antispoofing with robust feature representation. In: You Z, Zhou J, Wang Y, Sun Z, Shan S, Zheng W, et al., editors. Biometric Recognition. Cham: Springer International Publishing; 2016. p. 611–19.
[73] Khm O, Damer N. 2D face liveness detection: An overview. In: Proceedings of the international conference of the biometrics special interest group. Darmstadt, Germany; 2012. p. 1–12.

[74] Galbally J, Marcel S, Fierrez J. Image quality assessment for fake biometric detection: application to iris, fingerprint, and face recognition. IEEE Trans Image Process 2014;23(2):710–24.

[75] Rattani A, Scheirer WJ, Ross A. Open set fingerprint spoof detection across novel fabrication materials. IEEE Trans Inf Forensics Secur 2015;10(11):2447–60.



Ajita Rattani is an Adjunct Graduate Faculty with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City. Prior to joining the University of Missouri, she was a Postdoctoral Fellow with Michigan State University. She received the Ph.D. degree in computer engineering from the University of Cagliari, Italy. Her research interests include Pattern Recognition, Machine Learning, Biometrics, and Information Fusion.

Reza R. Derakhshani is an Associate Professor with the Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City. He received the Ph.D. degree in computer engineering from West Virginia University, Morgantown, WV. His research focus is Computational Intelligence with applications in Biometrics and Biomedical Signal and Image Processing.