Your Face Your Heart: Secure Mobile Face Authentication with Photoplethysmograms

Yimin Chen*, Jingchao Sun*, Xiaocong Jin*, Tao Li*, Rui Zhang†, and Yanchao Zhang*
*School of Electrical, Computer and Energy Engineering (ECEE), Arizona State University
†Department of Computer and Information Sciences, University of Delaware
{ymchen, jcsun, xiaocong.jin, tli}@asu.edu, ruizhang@udel.edu, yczhang@asu.edu

Abstract—**Face authentication emerges as a powerful method for preventing unauthorized access to mobile devices. It is, however, vulnerable to photo-based forgery attacks (PFA) and video-based forgery attacks (VFA), in which the adversary exploits a photo or video containing the user's frontal face. Effective defenses against PFA and VFA often rely on liveness detection, which seeks to find a live indicator that the submitted face photo or video of the legitimate user is indeed captured in real time. In this paper, we propose FaceHeart, a novel and practical face authentication system for mobile devices. FaceHeart simultaneously takes a face video with the front camera and a fingertip video with the rear camera on COTS mobile devices. It then achieves liveness detection by comparing the two photoplethysmograms independently extracted from the face and fingertip videos, which should be highly consistent if the two videos are for the same live person and taken at the same time. As photoplethysmograms are closely tied to human cardiac activity and almost impossible to forge or control, FaceHeart is strongly resilient to PFA and VFA. Extensive user experiments on Samsung Galaxy S5 have confirmed the high efficacy and efficiency of FaceHeart.**

I. Introduction

Protecting mobile devices from unauthorized access is becoming more than indispensable in these days. In particular, mobile devices such as smartphones and tablets are pervasive in personal life and business world. They are storing increasingly more highly sensitive information such as personal contacts and multimedia information, usernames and passwords, emails, browsing histories, business secrets, and health conditions. At the same time, mobile devices may be lost, stolen, or hacked. For example, 70 million smartphones are lost every year, with only 7% recovered, and 4.3% of company-issued smartphones are lost/stolen every year [1]. In addition, the malware infection rate on mobile devices rose to 0.75% in Q2 2015 from 0.68% in December 2014, and there were as many Android devices infected with malware as Windows laptops in the second half of 2014 alone [2].

Mobile authentication is widely adopted to protect mobile devices from unauthorized access and has two forms. First, a user is authenticated to unlock a device. Second, many mobile apps such as bank apps and password managers authenticate the user before s/he can use these apps. Mobile authentication traditionally follow a password approach based on PINs, alphanumeric passwords, or pattern locks. As functionalities of mobile devices keep improving, people have recently developed more secure and/or usable mobile authentication techniques based on behavioral biometrics such as inputting habits [3]–[6] and physiological biometrics such as fingerprints and deauthentication techniques based on proximity [7].

In this paper, we focus on improving the security of face authentication on mobile devices. As the name suggests, face authentication verifies or identifies a person by validating selected facial features from a digital image or a video frame. The facial features of a person are quite unique and difficult to forge. So face authentication has been very popular in various traditional application scenarios, e.g., gate and automated border control systems. It has also been introduced into mobile devices as a strong authentication method since Android 4.0, as well as many apps such as BioID and MobileID. Although we aim at face authentication on mobile devices, our work can be generalized to other scenarios involving face authentication without much modification.

Face authentication is vulnerable to both photo-based forgery attacks (PFA) and video-based forgery attacks (VFA). In PFA (or VFA), the adversary uses a photo (or video) containing the user's frontal face to bypass the otherwise highly-secure face authentication system. Both PFA and VFA are fairly easy to conduct, as the victim's photo or video usually can be easily found online, e.g., on popular social network sites. The adversary may also capture the victim's photo or video without being noticed, e.g., in crowded public places or through a high-definition camcorder from a long distance.

The prior defenses against PFA and/or VFA aim at *liveness detection*, which seeks to find a live indicator that the submitted face photo or video of the legitimate user is indeed captured in real time. The user's eye blink, lip movement, or head rotation in a video have been proposed as live indicators [8], [9]. These schemes are effective against PFA but invalid for VFA. The countermeasures against both PFA and VFA either use an infrared camera to obtain the thermogram of the user's face [10], or utilize texture analysis to detect the existence of a printed photo [11], or explore motion analysis to detect the existence of 2D images [12]. Besides very high computation complexity, these methods [10]–[12] require additional sensors or advanced cameras unavailable in COTS mobile devices.

The accelerometer in almost all COTS devices has recently been explored for liveness detection against PFA and VFA. In [13], Chen *et al.* proposed to compare the small motions extracted from the recorded video of the user's frontal face and those from the accelerometer to see if the motions are consistent. Similarly, Li *et al.* compared two motion vectors independently extracted from the video and the accelerometer of the mobile device for liveness detection [14]. Although these schemes [13], [14] are very effective against PFA and VFA, they require the legitimate user to move the mobile device in front of him/herself in some predefined manner, which can be inconvenient or even socially awkward. In addition, the randomness of the user-generated device movement may be too limited so that the adversary may have a good chance to successfully imitate the user after careful observations.

In this paper, we propose FaceHeart, a novel and practical liveness detection scheme for securing face authentication on mobile devices. FaceHeart targets mobile devices with both front and rear cameras that are available on most recently shipped mobile devices. The key idea of FaceHeart is to check the consistency of two concurrent and independently extracted photoplethysmograms of the user as the live indicator. For this purpose, FaceHeart records a video of the user's face by the front camera and a video of the user's fingertip by the rear camera at the same time. Then FaceHeart applies photoplethysmography (PPG) to extract two underlying photoplethysmograms from the face and fingertip videos. If the two photoplethysmograms are from the same live person and measured at the same time, they must be highly consistent and vice versa. As photoplethysmograms are closely tied to human cardiac activity and almost impossible for the adversary to forge or control, the consistency level of two extracted photoplethysmograms can well indicate the confidence level in the liveness of a face authentication request.

We design a complete set of tools to check the consistency of two photoplethysmograms for liveness detection. Specifically, given the face or fingertip video, the corresponding photoplethysmogram is extracted as a time series according to the principle of PPG. As a result, two time series can be obtained by using similar computer vision tools. After that, a set of features such as estimated heart rates and cross correlation of the two photoplethysmograms can be calculated by combining the two time series. Finally, lightweight machine learning algorithms are used for classifier training and subsequent testing. In this paper, we adopt and compare three machine learning algorithms, i.e., Bayesian network (BN), logistic regression (LR), and multilayer perceptron (MLP), to demonstrate the feasibility of FaceHeart.

We also conduct extensive experiments to evaluate FaceHeart. 18 users from diverse background are involved in our experiments. In typical settings, FaceHeart achieves a true positive rate (TPR) as high as 97.5%, a false negative rate (FNR) as low as 5.2%, and an equal error rate (EER) as low as 5.98%. Furthermore, we study the impact of various factors on FaceHeart, such as the head pose, background illumination, and location. Overall, the experimental results confirm that FaceHeart can effectively and reliably defend against PFA and VFA and thus secure face authentication on mobile devices.

The rest of the paper is organized as follows. Section II introduces the background of camera-based PPG. Section III details the FaceHeart design. Section IV presents the experimental evaluation. Section V discusses the limitations and security of FaceHeart. Section VI concludes this paper.

II. Background of Camera-Based PPG

In PPG, a photoplethysmogram is an optically obtained plethysmogram, which is a volumetric measurement of cardiovascular shock and sedation [15]. With each cardiac cycle, the heart pumps blood to the periphery, which generates pressure pulse that distends arteries and arterioles in the subcutaneous tissue. The corresponding volume change generated by the pressure pulse can be detected by measuring the amount of light either transmitted through or reflected from the skin. The evolvement of such volume changes across time carries exactly the user's heart beat signal.

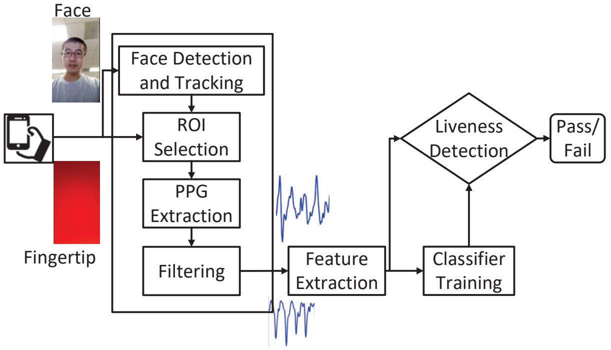


Fig. 1: A system overview of FaceHeart.

We adopt the model in [16] for camera-based PPG-based heart rate measurements. When the incident light arrives at the user's skin, a major part gets reflected back by the skin surface and does not interact with the tissue underneath the skin. The remaining (minor) part of the incident light first penetrates underneath the skin surface, then is absorbed by the tissue and the chromophores in blood inside arteries and capillaries, and finally gets reflected back to the camera. These two parts are usually referred to as surface reflectance and subsurface reflectance, respectively. The former dominates the overall light received by the camera but does not carry any information of human cardiac activity, while the latter is much smaller but bears the heart beat signal.

Given a skin region-of-interest (ROI) R in the video, the average pixel value at time t can be modeled as

$$y(t) = I(\alpha p(t) + b) + n(t), \tag{1}$$

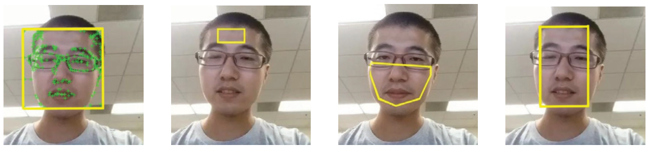
in which $y(t)$ is the average pixel value, I is the incident light intensity in R, α is the strength of blood perfusion, $p(t)$ is the blood volume change pulse, b is surface reflectance from the skin in R, and $n(t)$ is the quantization noise of the camera. $\alpha p(t)$ denotes subsurface reflectance and is much smaller compared to b (i.e., $\alpha p(t) \ll b$). Normally, I can vary across R and may change significantly across time if the illumination source or the environment change across time. In this paper, we assume I to be constant as the duration of the entire authentication process is usually less than five seconds and can be considered very short. Meanwhile, the user is asked to keep as still as possible, and we try to keep the environment, such as the illumination, as stable as possible. α and b are also assumed to be constants for the same ROI and the same user. On the contrary, $n(t)$ is a random variable, and a large variance of $n(t)$ may mask the small heart beat signal exhibited in $p(t)$. Equivalently, if noise is not considered, $y(t)$ can be viewed as the combination of a large DC part and a small AC part. The latter carries the information of human cardiac activity and can be extracted through a set of signal processing tools.

III. FaceHeart

FaceHeart can be used as a standalone mobile authentication module in the mobile OS or integrated in any app desiring face authentication. In this section, we give an overview of FaceHeart and then detail its design.

A. Overview

FaceHeart works as follows. First, the user uses his/her fingertip to cover the rear camera and also flashlight without



(a) Detected face and "good features" (b) R_1, forehead [17] (c) R_2, lower face [18] (d) R_3, central face [19]

Fig. 2: Camera-based PPG.

applying any pressure. Then FaceHeart uses the front and rear cameras simultaneously to record the face and fingertip videos, respectively. The user needs to stay as still as possible while the recording is ongoing. Next, FaceHeart extracts two photoplethysmograms from the two videos and compares them for liveness detection. In the meantime, one frame of the face video (for instance, any frame after the first second of recording) is sent to the conventional face authentication module to decide whether the person in the frame is the legitimate user. Only when liveness detection and conventional face authentication both succeed is the user considered authentic.

Fig. 1 depicts the flow chart of FaceHeart. Given a pair of face and fingertip videos, FaceHeart uses the following modules to accomplish liveness detection. The Signal Processing module is first invoked to obtain two photoplethysmograms independently from the two videos. Then the output is fed into the Feature Extraction module to generate a feature vector which characterizes the consistency level of the two photoplethysmograms. In the next Classifier Training module, machine learning algorithms are used to train a classifier based on a library of feature vectors. Finally, the classifier is used in the Liveness Detection module to determine whether a new pair of face and fingertip videos can pass liveness detection.

B. Signal processing

As shown in Fig. 1, the Signal Processing module comprises four submodules: face detection and tracking, ROI (region-of-interest) selection, photoplethysmogram extraction, and filtering. The face video requires all four submodules, while the fingertip video just needs the last three.

1) Face detection and tracking: In this step, we first detect the user's face in the first frame of the face video using the classical Viola-Jones detection algorithm [20]. This algorithm can work in real time and is highly accurate.

Next, instead of applying relatively costly face detection to every frame, we use the Kanade-Lucas-Tomasi (KLT) feature tracker to track the identified features from frame to frame [21], [22]. More specifically, the KLT feature tracker identifies multiple local feature points, commonly known as "good features to track" [23]. Then it tries to search as many as possible of the identified feature points in the previous frame. Given two sets of features points in the current and previous frame, the KLT feature tracker can estimate the translation, rotation, and scale between the two consecutive frames and then compute an affine function for face tracking. Since the duration of the face video is short, the established feature tracker is still valid for the last frame.

Finally, we can obtain the coordinates of the user's face in each frame. As depicted in Fig. 2(a), we obtain four coordinates forming a rectangular box in each frame, which approximates the whole face region. The green cross markers depict the "good features to track" of the shown frame.

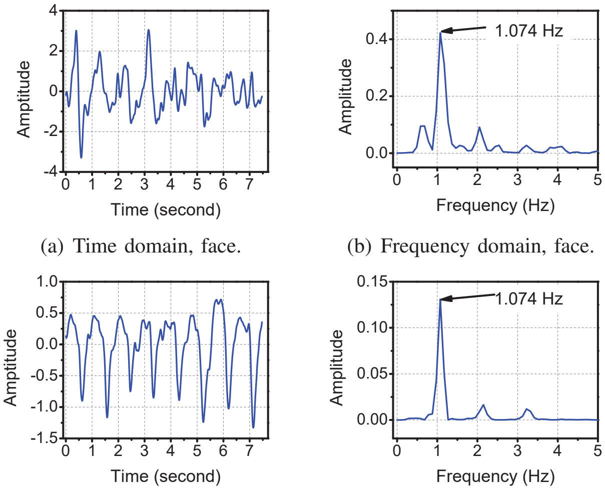
2) ROI selection: Different types of ROIs have been used in the literature. Fig. 2(b), Fig. 2(c), and Fig. 2(d) illustrate three most frequently used ROIs, denoted by R_1 [17], R_2 [18], and R_3 [19], respectively. Some schemes use random selection while some others assign weights to every segmented unit of the face. Intuitively, the amount of photoplethysmogram information extracted from a specific ROI is closely related to where the ROI is. The reason is that the extracted photoplethysmogram is proportional to $p(t)$ in Eq. (1), i.e., the amount of blood volume change underneath the ROI. Meanwhile, the distribution of blood carrying capillaries differs from region to region, further resulting in different amount of extractable photoplethysmogram information. The size of the selected ROI may also have influence on the extracted photoplethysmogram. On the one hand, a smaller size requires a highly accurate face tracker to avoid too much noise in the extracted photoplethysmogram. On the other hand, a larger size averages the contribution across the entire region and therefore may shrink the strength of the photoplethysmogram.

In this paper, we choose R_3 as the ROI for extracting photoplethysmogram, which is the central part of the whole face and encompasses 60% of the width and the full height of the detected face region. In contrast to R_1 and R_2 that require a resource-demanding feature detector [24], R_3 only requires the basic computationally efficient Viola-Jones detector. In addition, our experimental evaluations in Section IV-D show that R_1 and R_2 do not show much performance improvement over R_3 mainly because the required face tracker has limited accuracy in constrained mobile environments. It is possible to have a weighted combination of multiple ROIs as in [16], which nevertheless requires multiple iterations and thus incurs larger computation overhead. How to use multiple ROIs more efficiently in FaceHeart is part of our future work.

3) Photoplethysmogram extraction: We extract the photoplethysmogram from an ROI by averaging all pixel values therein. A recorded video has three channels: red, green, and blue. In the literature [16], [18], [19], [25], [26], it is widely accepted that the three channels carry different amount of photoplethysmogram information. The green channel carries the strongest photoplethysmogram, as the green light is easier to absorb by hemoglobin in the blood and thus penetrates deeper into the skin [16]. It is tempting to use all three channels to enhance the SNR of the extracted photoplethysmogram, but the recent studies [16], [18], [25] show that this approach is not necessarily beneficial because the three channels do not yield statistically mutually independent information. So we follow the suggestion in [16], [18], [25] to obtain the photoplethysmogram only from the green channel.

4) Filtering: This step applies two filters to the extracted photoplethysmogram. First, we use a Normalized Least Mean Square (NLMS) adaptive filter to alleviate the illumination interference [27]. The motivation is that small environment changes—such as a person passing by or small camera movements—may induce overall illumination shifting in the video. This undesirable effect can be mitigated by estimating the amount of interference and then subtracting it from the overall measurement. In Section II, we use $y(t)$ to denote the photoplethysmogram of a selective ROI R. Given the illumination interference, $y(t)$ can be divided into two parts:

$$y(t) = y_c(t) + n_i(t), \tag{2}$$



(a) Time domain, face.

(b) Frequency domain, face.

(c) Time domain, fingertip.

(d) Frequency domain, fingertip.

Fig. 3: Illustration of extracted photoplethysmograms.

where $y_c(t)$ is due to human cardiac activity, and $n_i(t)$ is due to illumination interference. $n_i(t)$ can be assumed to be proportional to the average pixel value of the background regions other than the face region. We thus have

$$n_i(t) = hy_{bg}(t), \tag{3}$$

where $y_{bg}(t)$ is the average pixel value of a selective background region, and h is a linear coefficient. In our implementation, we simply select a pixel block of 20×20 in the top-right corner in each frame as the background region. h can be estimated by the NLMS adaptive filter as

$$h(j+1) = h(j) + \mu \frac{y_c(j)}{y_{bg}(j)}, j = 0, 1, 2, \ldots, N-1. \tag{4}$$

Here μ is the step size equal to 1, and N is the length of $y(t)$ (or $y_c(t)$, equivalently). We also set $h(0) = 0$ in the implementation. After the final $h = h(N)$ is obtained, $n_i(t)$ can be subtracted from $y(t)$ according to Eq. (2) to finally reveal $y_c(t)$.

Next, we use a bandpass FIR filter (second-order Butterworth filter) with a passband of $[0.7, 4]$ Hz to reduce the interference of out-of-band noise. The signal after filtering is the final photoplethysmogram for liveness detection.

5) Processing fingertip video: Extracting the photoplethysmogram from a fingertip video is much easier. Specifically, no face detection or tracking is needed, and the entire frame is used as the ROI. Meanwhile, since the rear camera is fully covered by the user's fingertip, there is no illumination interference so that the NLMS adaptive filter is not needed.

C. *Feature extraction*

In this module, we use the two extracted photoplethysmograms to calculate a feature vector for classifier training and liveness detection. Denote the photoplethysmograms from the face and fingertip videos by P_{face} and P_{ftip}, respectively. P_{face} and P_{ftip} are two time series of the same length N, from which the following features are calculated.

- **Heart rate difference**. The heart rate difference is the absolute difference between the heart rates from the face and the fingertip. We denote them by h_{face} and h_{ftip}, respectively. To obtain h_{face}, we first multiply P_{face} with an N-point Hanning window such that the two endpoints of P_{face} can meet rather than having a sharp transition between them. Then we apply fast fourier transform (FFT) on windowed P_{face}, select the highest peak within $[0.7, 4]$ Hz, multiply it by 60, and obtain h_{face}. We can also obtain h_{ftip} in the same way. Then heart rate difference is calculated as

$$\Delta h = |h_{\text{face}} - h_{\text{ftip}}| \tag{5}$$

- **Maximum cross correlation**. We obtain the maximum cross correlation between P_{face} and P_{ftip} by searching the optimal alignment between them. Specifically, we first obtain the optimal alignment $\hat{k}$ by the following equation.

$$\hat{k} = \arg\min \sum_{i=1}^{N-k+1} \frac{P_{\text{face}}(i)P_{\text{ftip}}}{N-k}, \tag{6}$$
$$\text{subject to } 0 \leq k < N_{\text{ftip}}.$$

Here N_{ftip} is the approximate length of a period of P_{ftip} and equals $\lceil \frac{60F_s}{h_{\text{ftip}}} \rceil$, where F_s is the frame rate of the fingertip video (and equivalently that of the face video). After $\hat{k}$ is found, we truncate P_{face} and P_{ftip} into two shorter vectors of the same length as

$$\tilde{P}_{\text{face}} = P_{\text{face}}(1 : N - \hat{k}), \tilde{P}_{\text{ftip}} = P_{\text{ftip}}(\hat{k} + 1 : N). \tag{7}$$

Then the maximum ratio is calculated as

$$\rho_{\max} = \sum_{i=1}^{\tilde{N}} \frac{\tilde{P}_{\text{face}}(i)\tilde{P}_{\text{ftip}}(i)}{\tilde{N}}, \tag{8}$$

where $\tilde{N} = N - \hat{k}$.

- **Mean, min, max, and standard deviation of amplitude ratio**. Given the aligned $\tilde{P}_{\text{face}}$ and $\tilde{P}_{\text{ftip}}$, we first calculate amplitude ratio as $R(i) = \frac{\tilde{P}_{\text{face}}(i)}{\tilde{P}_{\text{ftip}}(i)}, i = 1, 2, \ldots, \tilde{N}$. Then we further calculate the mean, min, max, and standard deviation of R as features, denoted by $R_{\text{mean}}, R_{\min}, R_{\max}$, and R_{SD}, respectively.

D. *Classifier training*

Our training set contains two classes of instances. Each instance consists of a feature vector in the form of $v = [\Delta h, \rho_{\max}, R_{\text{mean}}, R_{\min}, R_{\max}, R_{\text{SD}}]$. The feature vectors of the instances in Class I (labelled as $l = 1$) are computed from a pair of simultaneously recorded face and fingertip videos. On the contrary, those of the instances in Class II (labelled as $l = 0$) are computed from a pair of face and fingertip videos recorded separately. Ideally, the classifier should be able to label the instances in both classes as accurately as possible. As in [14], we use and compare three supervised machine learning techniques in the Weka toolkit [28] for classifier training and testing: Bayesian network (BN), logistic regression (LR), and multilayer perceptron (MLP). In particular, BN is based on constructing a probabilistic graphic model representing a set of random variables and their conditional dependencies via a directly acyclic graph [29]. The constructed probabilistic model is used to infer the label of unlabeled instances. LR uses the sigmoid function as the hypothesis to estimate the relationship between the features and corresponding labels

[30]. MLP is a feedforward artifical neural network model that maps the sets of input data onto a set of appropriate output [31]. One important advantage of MLP is that it can be used to distinguish data that are not linearly separable.

The classifier training is neither user-specific nor device-specific. It is exclusively done by the FaceHeart developer who can easily maintain and update a large number of instances for Classes I and II. The trained classifier is preloaded into the mobile device when FaceHeart is installed.

E. Liveness detection

Given a new pair of face and fingertip videos for authentication, FaceHeart computes the corresponding feature vector and then inputs into the classifier. If the output label is 1, the new pair passes liveness detection and fails otherwise. In the former case, if the face image additionally passes conventional face authentication, the user is deemed legitimate.

IV. Performance Evaluation

This section evaluates the performance of FaceHeart.

A. Adversary model

We consider a typical adversary model in this paper. The adversary possesses the victim's mobile device and seeks to pass the face authentication employed by the device itself or some sensitive apps. Since VFA can be considered an advanced version of PFA, we focus on evaluating the resilience of FaceHeart to VFA. The adversary can surreptitiously obtain the videos containing the legitimate user's frontal face, e.g., by online searches or realtime capturing through a high-definition camcorder from a long distance. In contrast, fingertip videos are very rare online or almost impossible to capture in real time, so the adversary can only use the fingertip video of himself or a random user. In addition, the adversary is fully aware of FaceHeart. We consider two types of VFA as follows.

Type-I VFA. This attack does not involve any realtime video recording and serves as a "stress test" for FaceHeart. In particular the adversary directly feeds his fingertip video and the victim's face video into FaceHeart. Each participant in our experiments is assumed as the adversary once, in which case the other participants are used as the victims.

Type-II VFA. This attack resembles the practical attack scenario. The adversary first replays the victim's face video on the screen of his/her own device such as an iPad. The distance between the victim device and the adversary's device screen is properly adjusted such that the victim device's front camera can well capture the victim's face in the replayed video. While the face video is replayed and recorded, the adversary let the victim device's rear camera take his/her fingertip video simultaneously. Two random participants are chosen as the adversary for the Type-II VFA. When either is chosen, each other participant serves as a victim.

B. Experiment setup

We used a Samsung Galaxy S5 in the experiments. In particular, we utilized the dual-camera mode of the Camera app on Galaxy S5, which can record a video with both the front and rear cameras simultaneously. The frame size of the recorded video is 720×1280, which can be equally divided into the upper and lower parts, corresponding to the face and fingertip videos, respectively. After the useless black region on left and right sides is removed, the frame size of both face and fingertip videos becomes 480×640. Since almost all recently shipped mobile devices have both front and rear cameras, it is rather straightforward to obtain the simultaneously-recorded face and fingertip videos on other device models.

We recruited 18 participants in the experiments, including two females and 16 males. The participants are graduate students in Arizona State University, whose ages range between 20 and 35. All the participants were given the following instructions. First, each participant tries to sit as still as possible. The distance between the user and the front camera varies between 30 to 45 cm, which has been proved to be a convenient distance for the users and that the captured user face is reliably detected. Then s/he activates the dual-camera mode of the Camera app on Galaxy S5 and ensures that the front camera properly captures her/his frontal face. Subsequently, s/he rests any of her/his fingertip on the rear camera without applying any pressure. Finally, s/he proceeds to record a video of approximately ten seconds by tapping the video recorder icon.

As cardiac activity highly depends on current user conditions, the videos were recorded when the participant was under different conditions to fully evaluate the performance of FaceHeart. In particular, we investigated three user conditions. Under the rest condition, each participant was asked to sit quietly without her/his legs crossed for five minutes. After that, s/he recorded videos for 15 times. Under the reading condition, each participant was asked to read recent news on a smartphone for five minutes. After that, s/he recorded videos for 15 times. Under the gaming condition, each participant was asked to play the video game "No Limits" or "Strikers 1945-3" on a smartphone for five minutes. After that, s/he recorded videos for 15 times. For the same participant, cardiac activities are expected to be different under these three conditions [32]. Particularly, the heart rate of the same user in the gaming condition is usually higher than those in the rest and reading conditions, which was also confirmed in the experiments.

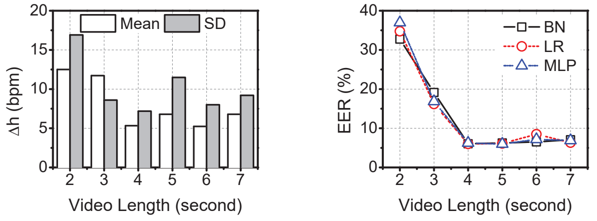
The following default settings were used unless stated otherwise. Participants were asked to maintain the front head pose during video recording. Videos were recorded under normal illumination in a typical research lab (e.g., 500 lux). During the recording process, other persons may leave/enter the lab.

Our main dataset, denoted by $\mathcal{S}$, consists of $\mathcal{S}_p$ for positive (Class I) instances and $\mathcal{S}_n$ for negative (Class II) instances. The instances in $\mathcal{S}_p$ come from legitimate users, while those in $\mathcal{S}_n$ are from Type-I adversary. Given 18 participants with each recording 15 videos under each of the three user conditions, there are $18 \times 3 \times 15 = 810$ instances in $\mathcal{S}_p$. To generate $\mathcal{S}_n$, we first randomly selected two pairs of face and fingertip videos for each participant. Each participant acted as the adversary once, in which case each other participant acted as the victim. So $\mathcal{S}_n$ contains $2 \times 2 \times 17 = 68$ instances per participant and $68 \times 18 = 1224$ instances in total. For the following evaluations, we repeated the generation process of $\mathcal{S}_n$ for 40 times and obtained the average results.

C. Performance metrics

We use the following performance metrics.

Receiver operating characteristic (ROC) curve. An ROC curve can be used to illustrate the performance of a binary classifier as its discrimination threshold changes. According to the definition in [33], we can obtain an ROC curve by plotting TPR (true-positive rate) with respect to FPR (false-positive



(a) On difference between h_{face} and h_{ftip}

(b) On EER

Fig. 4: Impact of video length on Δh and EER.

rate) in various threshold settings.

Acceptance rate. We define the acceptance rate as the ratio between the number of correctly-classified positive (legitimate) instances and that of all positive instances in a testing dataset. A higher acceptance rate means that the system is more likely to admit legitimate users.

Detection rate. We define the detection rate as the ratio between the number of correctly-classified negative (adversarial) instances and that of all negative instances in a testing dataset. A higher detection rate means that the system can more effectively detect VFA.

Computation time. We define the computation time as the time FaceHeart takes to determine whether a given pair of face and fingertip videos can pass liveness detection. Intuitively, the computation time should be as short as possible.

D. Experimental results

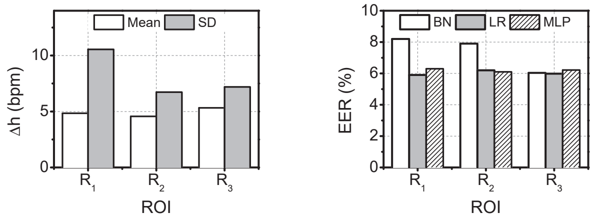
1) Video length: Here we show the impact of video length on FaceHeart.

Fig. 4(a) shows the mean and standard deviation (SD) of Δh in $\mathcal{S}_p$, which is the absolute difference between h_{face} and h_{ftip} in the same authentication session. Since the SNR of the photoplethysmogram from the fingertip video is usually high, h_{ftip} can be treated as the reference heart rate. As we can see, the mean and SD of Δh decrease from around 12 and 17 bpm to around 5 and 7 bpm when the video length increases from two to four seconds. This means that the accuracy of h_{face} increases along with the video length. When the video length is larger than four seconds, the mean and SD of Δh do not change much.

Fig. 4(b) shows the EER (equal error rate) of FaceHeart under the Type-I attack using $\mathcal{S}$. We can see that FaceHeart exhibits similar EER performance with BN, LR, and MLP. Therefore, we believe that FaceHeart works well along with mainstream machine learning algorithms. Meanwhile, the EER decreases quickly when the video length increases from two to four seconds and then stays relatively the same as the video length further increases. Such results are consistent with those in Fig. 4(a) because a smaller Δh indicates that the two corresponding photoplethysmograms in the same authentication session are more consistent. Consequently, this makes it easier for the classifier to distinguish between positive and negative instances, leading to a lower EER.

As a shorter video length means that the legitimate user can record a shorter video for authentication, the required minimum video length of FaceHeart is preferably as short as possible. Based on the above results, the default video length is set to four seconds hereafter unless specified otherwise.

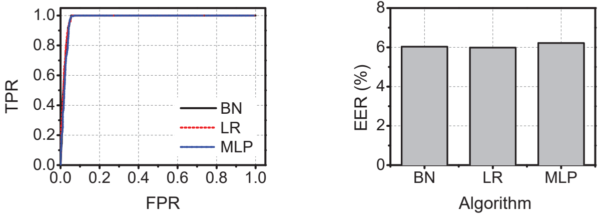
2) ROI: Now we demonstrate the impact of ROI on FaceHeart using $\mathcal{S}$.



(a) On difference between h_{face} and h_{ftip}

(b) On EER

Fig. 5: Impact of ROI on Δh and EER.



(a) ROC curve

(b) EER

Fig. 6: ROC and EER performance of FaceHeart under Type-I attacks.

Fig. 2(b), Fig. 2(c), and Fig. 2(d) illustrate the three ROIs to study. Fig. 5(a) shows the mean and SD of Δh in $\mathcal{S}_p$. As we can see, the means of Δh using R_1, R_2, and R_3 are 4.84, 4.56, and 5.32 bpm, respectively, and the SDs are 10.55, 6.73, and 7.19 bpm, respectively. Fig. 5(b) shows the corresponding EERs when R_1, R_2, and R_3 are used as the selected ROI, respectively. The EERs with R_1 using BN, LR, and MLP are 8.2%, 5.9%, and 6.3%, respectively, those with R_2 are 7.9%, 6.2%, and 6.1%, respectively, and those with R_3 are 6.0%, 6.0%, and 6.2%, respectively.

The results above show that the three ROIs lead to similar EER performance while the EERs with R_3 are slightly better than those with R_1 or R_2. More importantly, the computation time of FaceHeart using R_3 as the selected ROI is much shorter than that using R_1 or R_2, as shown soon in Section IV-D6. Therefore, we select R_3 as the ROI for photoplethysmogram extraction by default.

3) Type-I attack: Here we show the resilience of FaceHeart to the Type-I attack.

Fig. 6(a) and Fig. 6(b) show the ROC curve and EER of FaceHeart, respectively. The TPRs using BN, LR, and MLP are 90.2%, 97.5%, and 94.6%, respectively, the FPRs are 3.8%, 5.2%, and 4.6%, respectively, and the EERs are 6.03%, 5.98%, and 6.21%, respectively. The results show that the performance of FaceHeart is similar to those of the state-of-the-art systems, such as FaceLive in [14]. To sum up, FaceHeart can achieve very high TPR and very low FPR at the same time, meaning that it can correctly distinguish between legitimate requests and VPAs with high probability.

Fig. 6(b) shows the EERs of FaceHeart in different user conditions. The EERs using BN, LR, and MLP under the rest condition are 7.70%, 5.57%, and 5.40%, respectively, those under the reading condition are 8.77%, 5.53%, 5.73%, respectively, and those under the gaming condition are 8.27%, 8.54%, and 5.65%, respectively. Overall, the EERs in the three user conditions are low, so FaceHeart can be used even when the user's cardiac activity changes. In addition, the EERs in the gaming condition are slightly higher than those under the

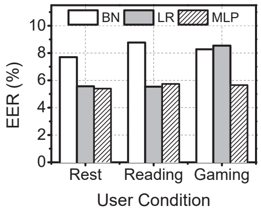


Fig. 7: EER performance of FaceHeart under Type-I attacks in different user conditions.

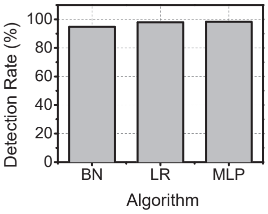


Fig. 8: EER performance of FaceHeart under Type-II attacks.

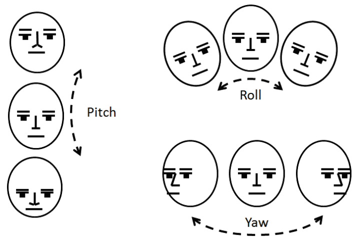
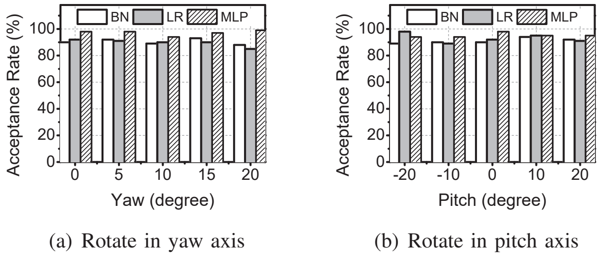


Fig. 9: Illustration of head pose in yaw, pitch, and roll axes.



(a) Rotate in yaw axis

(b) Rotate in pitch axis

Fig. 10: Impact of head pose on acceptance rate.

other conditions. This is anticipated because the heart rate in the gaming condition is usually higher than others so that the SNR of the extracted photoplethysmogram usually decreases due to the increased noise level in the higher frequency range. Therefore, the consistency between the two photoplethysmograms from a pair of face and fingertip videos in the same authentication session drops, leading to a higher EER. Based on $\mathcal{S}$, we obtain the corresponding classifiers with BN, LR, and MLP, respectively, by using 10-fold cross validation for training. Then we use the trained classifier models for testing in the following.

4) Type-II attack: Now we show the detection rate of FaceHeart under the Type-II attack. We first obtained the negative (adversarial) instances for the Type-II attack as follows. Two of the 18 participants acted as the adversaries. For each adversary, the other 17 participants were regarded as her/his victims. For each victim, we randomly selected 10 face videos from her/his recordings. Then the two adversaries launched the Type-II attack, resulting in $2 \times 10 \times 17 = 340$ negative instances. After that, we applied the trained classifiers in Section IV-D3 to the collected negative instances and obtained the detection rate. As shown in Fig. 8, the detection rates using BN, LR, and MLP are 94.71%, 97.94%, and 98.24%, respectively, indicating that FaceHeart can detect VFA with overwhelming probability.

5) Robustness of FaceHeart: In the following, we study the robustness of FaceHeart against different factors including head pose, illumination, and location.

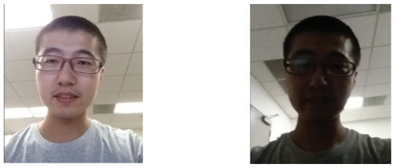
Head pose. We first study the impact of head pose on the acceptance rate of FaceHeart. As illustrated in Fig. 9 [34], the relative rotation of a user's head to the front head pose can be described by rotation angles in three independent axes, which are yaw, pitch, and roll, respectively. Hereafter we also refer to the rotation angles in yaw, pitch, and roll axes as yaw, pitch, and roll, respectively. For the front head pose, yaw, pitch, and roll are equal to zero. Roll is easier to adjust by the user, and a zero roll also benefits face detection. So participants were asked to adjust their head poses such that the rolls are as near to zero as possible. As a result, we only focus on the other two types of head rotation angles, i.e., yaw and pitch.

Data collection worked as follows. First, we asked two participants to record videos for authentication with different yaws or pitches. Specifically, they recorded videos when the yaws changed and the pitches remained near to zero and continued when the pitches changed and the yaws remained near to zero. After that, we applied the trained classifiers in Section IV-D3 to the collected dataset and obtained the acceptance rate. Each participant recorded 50 videos for the same yaw or pitch, resulting in 1,000 videos in total.

Fig. 10(a) and Fig. 10(b) show the acceptance rates of FaceHeart with different yaws and pitches, respectively. The acceptance rate is almost always higher than 90% and changes only slightly when the yaw of user head pose changes from zero to 20 degrees, or the pitch changes from -20 to 20 degrees. The results are as expected because FaceHeart is based on comparing two photoplethysmograms extracted from a pair of face and fingertip videos, and a small yaw or pitch (less than ±20 degrees) does not affect photoplethysmogram extraction much. Assuming that users tend to record videos with small yaws or pitches (less than ±10 degrees) in practice, we believe that FaceHeart is robust to head pose changes.

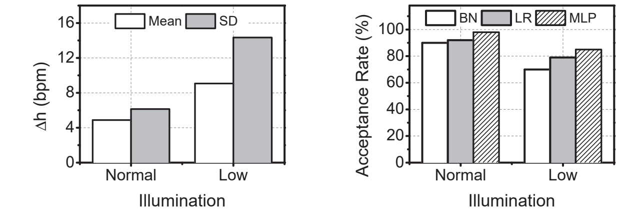
Illumination. Here we study the impact of illumination on the acceptance rate of FaceHeart. For this experiment, we asked two participants to record videos for authentication under two different illuminations, i.e., normal (in the range of hundreds lux) and low illuminations (less than 20 lux). Fig. 11 illustrates the clear influence of normal and low illuminations on video recording. The illumination was adjusted by turning off part of the lights in our office. After that, we applied the trained classifiers in Section IV-D3 to the collected dataset and obtained the acceptance rate. Each participant recorded 50 videos for the same illumination, resulting in 200 videos in total for this experiment.

Fig. 12(a) and Fig. 12(b) show the mean and SD of Δh and acceptance rate of FaceHeart, respectively. The mean and SD of Δh increase from 4.88 and 6.14 bpm to 9.07 and 14.34 bpm, respectively, when the illumination switches from normal to low. Correspondingly, the acceptance rates using BN, LR, and MLP drop from 90%, 92%, and 98% to 70%, 79%, and 85%, respectively. The results indicate that FaceHeart is greatly affected by illumination in the environment, which can be explained as follows. FaceHeart relies on comparing the photoplethysmograms extracted from a pair of face and fingertip videos, and low illumination leads to a low SNR



(a) Normal illumination (b) Low illumination

Fig. 11: Captured images under different illuminations.



(a) On difference between h_{face} and h_{ftip} (b) On acceptance rate

Fig. 12: Impact of illumination on Δh and acceptance rate.

of the extracted photoplethysmogram. Hence, the consistency between the face and fingertip photoplethysmograms reduces (partially illustrated by the increased Δh), leading to the decreased acceptance rate.

Location. We also study the impact of locations on the acceptance rate of FaceHeart. First, we asked two participants to record videos for authentication in four different locations, i.e., our office, the apartments (APTs) of the participants, the university library (LIB), and an outdoor bench on our campus. After that, we applied the trained classifiers in Section IV-D3 to the collected dataset and obtained the acceptance rate. Each participant recorded 50 videos for the same location, resulting in a dataset of 400 videos in total.

Fig. 13 shows the acceptance rate of FaceHeart with different locations. The acceptance rates are always higher than 90% and do not change much when the location changes. The results indicate that FaceHeart is robust to location changes and thus can be used in different locations. The reason is that locations have little impact on photoplethysmogram extraction and consequently little impact on the classification results.

6) Computation time: Here we study the computation time of FaceHeart for different ROIs. For this experiment, we randomly select 100 pairs of face and fingertip videos from our collected data. Each pair of videos were both chopped to a length of four seconds. Then we run FaceHeart with the given video pairs and obtained the average computation time. To use R_1 or R_2, we first used the face tracker in [24] to track the facial landmarks in each frame and then calculated the coordinates of R_1 or R_2. Fig. 14(a) depicts the tracked 49 landmarks on the user face which are used for the calculation of R_1 and R_2.

Fig. 14(b) shows the computation time using R_1 or R_2 or R_3 as the selected ROI. The average computation time using R_1, R_2, and R_3 are 18.05, 18.19, and 0.96 seconds, respectively. Therefore, selecting R_3 as the ROI for photoplethysmogram extraction is much faster than selecting R_1 or R_2. Such results are as expected because R_1 and R_2 require much more computationally-expensive face trackers than that used by R_3.

The computation time of FaceHeart is comparable to the state of art. In particular, Li *et al.* reported an average time

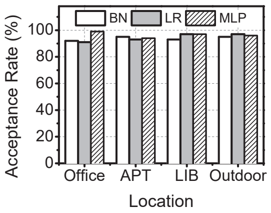
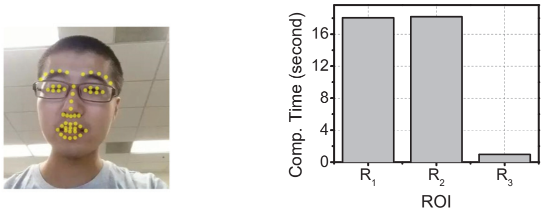


Fig. 13: Impact of location on acceptance rate.



(a) Detected landmarks for ROI calculation (b) Average computation time using different ROIs

Fig. 14: Impact of ROI on computation time.

of 3.3 seconds for device movement (equivalent to the video length in FaceHeart) in [14] and did not explicitly evaluate the computation time for liveness detection. In [13], the authors mentioned that the average authentication time for video recording and also liveness detection is 2.8 seconds when successful and failed authentications are combined and 4.9 seconds when only successful authentications are considered. Given the video length of four seconds used in our evaluations, we believe that the computation time of FaceHeart is similar to the state-of-the-art, but FaceHeart is more secure and user-friendly.

V. Discussion

As the first system exploring photoplethysmogram for secure face authentication on mobile devices, FaceHeart certainly has limitations. In this section, we outline the possible ways to further improve FaceHeart.

A. Camera-based PPG

As the camera-based PPG method in [19] is adopted to extract photoplethysmograms, FaceHeart naturally inherits its limitations related to user movement and the environment illumination. More specifically, the user is required to keep her/his head as still as possible in order to extract more accurate photoplethysmograms. Meanwhile, as shown in Section IV-D5, the performance of FaceHeart depends greatly on the illumination in the environment. Hence, there should be sufficient and stable illumination in the environment to guarantee the high performance of FaceHeart.

Advanced schemes have been explored to alleviate the requirements on user movement and the environment illumination. For example, researchers have proposed schemes to improve the estimation accuracy of the heart rate under adverse situations, such as when the user spontaneously moves his head a little bit [16] or the illumination in the environment is below normal [35]. Although such schemes are not directly applicable to FaceHeart, they indicate a promising direction worth exploring. Other minor issues inherited from camera-based PPG methods include the impact of facial occlusion, facial expression, and user skin tone, which we plan to fully investigate in our future work.

B. Authentication time

In FaceHeart, the authentication time for liveness detection can be broken into two parts, i.e., video length and computation time. Given the video length of four seconds and the computation time of 0.96 seconds with R_3 as the ROI, the total authentication time of FaceHeart is around 4.96 seconds. In [13], the authors reported that the authentication time of their liveness detection scheme is around 4.9 seconds, which is comparable to 4.3 seconds of credential-based authentication schemes. In this regard, the authentication time of FaceHeart is acceptable and also comparable to the state-of-the-art.

Similar to [13], [14], the authentication time of FaceHeart is dominated by the required video length, which is four seconds in this paper. A shorter video length may be adopted, however, at the cost of higher EERs. One possible way to shorten the required video length is to extract new features from extracted photoplethysmograms. For example, heart rate variability and the absolute delay between the two photoplethysmograms from face and fingertip videos are very promising candidates. These two features can be useful only when the SNRs of the two photoplethysmograms are sufficiently high, which we plan to explore in the future.

VI. Conclusion

In this paper, we presented the design and evaluation of FaceHeart, a novel and practical scheme for liveness detection to secure face authentication on COTS mobile devices. FaceHeart relies on the non-forgeability of the photoplethysmograms extracted from two videos simultaneously taken through the front and rear cameras on a mobile device. Extensive user experiments confirmed that FaceHeart can effectively thwart photo-based and video-based forgery attacks on mobile face authentication systems.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-1421999, CNS-1320906, CNS-1514381, CNS-1619251, CNS-1700032, and CNS-1700039. We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

References

[1] [Online]. Available: http://www.channelpronetwork.com/article/mobile-device-security-startling-statistics-data-loss-and-data-breachesl

[2] http://resources.alcatel-lucent.com/asset/189669.

[3] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *NDSS'13*, San Diego, USA, Feb. 2013.

[4] M. Shahzad, A. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *ACM MobiCom'13*, Miami, USA, Sep. 2013.

[5] J. Sun, X. Chen, J. Zhang, Y. Zhang, and J. Zhang, "TouchIn: Sightless two-factor authentication on multi-touch mobile devices," in *IEEE CNS'14*, San Francisco, CA, Oct. 2014.

[6] Y. Chen, J. Sun, R. Zhang, and Y. Zhang, "Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices," in *IEEE INFOCOM'15*, Hong Kong, China, Apr./May 2015.

[7] T. Li, Y. Chen, J. Sun, X. Jin, and Y. Zhang, "ilock: Immediate and automatic locking of mobile devices against data theft," in *ACM CCS'16*, Vienna, Austria, Oct. 2016.

[8] O. Kähm and N. Damer, "2d face liveness detection: An overview," in *IEEE BIOSIG'12*, Darmstadt, German, Sep. 2012.

[9] K. Kollreider, H. Fronthaler, and J. Bigun, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, no. 3, pp. 233–244, Feb. 2009.

[10] R. Ghiass, O. Arandjelovic, H. Bendada, and X. Maldague, "Infrared face recognition: a literature review," in *IEEE IJCNN'13*, Dallas, TX, Aug. 2013.

[11] J. Määttä, A. Hadid, and M. Pietikainen, "Face spoofing detection from single images using micro-texture analysis," in *IEEE IJCB'11*, Washington, DC, Oct. 2011.

[12] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *ECCV'10*, Crete, Greece, Sep. 2010.

[13] S. Chen, A. Pande, and P. Mohapatra, "Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones," in *ACM MobiSys'14*, Bretton Woods, NH, Jun. 2014.

[14] Y. Li, Y. Li, Q. Yan, H. Kong, and R. Deng, "Seeing your face is not enough: An inertial sensor-based liveness detection for face authentication," in *ACM CCS'15*, Denver, CO, Oct. 2015.

[15] K. Shelley and S. Shelley, "Pulse oximeter waveform: photoelectric plethysmography," *Clinical Monitoring, Carol Lake, R. Hines, and C. Blitt, Eds.: WB Saunders Company*, pp. 420–428, 2001.

[16] M. Kumar, A. Veeraraghavan, and A. Sabharwal, "Distanceppg: Robust non-contact vital signs monitoring using a camera," *Biomedical optics express*, vol. 6, no. 5, pp. 1565–1588, May 2015.

[17] M. Lewandowska, J. Rumiński, T. Kocejko, and J. Nowak, "Measuring pulse rate with a webcama non-contact method for evaluating cardiac activity," in *IEEE FedCSIS'11*, Szczecin, Poland, Sep. 2011.

[18] X. Li, J. Chen, G. Zhao, and M. Pietikainen, "Remote heart rate measurement from face videos under realistic situations," in *IEEE CVPR'14*, Columbus, OH, Jun. 2014.

[19] M. Poh, D. McDuff, and R. Picard, "Non-contact, automated cardiac pulse measurements using video imaging and blind source separation." *Optics express*, vol. 18, no. 10, pp. 10 762–10 774, May 2010.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE CVPR'01*, Kauai, HI, Dec. 2001.

[21] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, 1981, pp. 674–679.

[22] C. Tomasi and T. Kanade, *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[23] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR'94*, Seattle, WA, Jun. 1994.

[24] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *IEEE CVPR'14*, Columbus, OH, Jun. 2014.

[25] W. Verkruysse, L. Svaasand, and J. Nelson, "Remote plethysmographic imaging using ambient light." *Optics express*, vol. 16, no. 26, pp. 21 434–21 445, 2008.

[26] A. Lam and Y. Kuno, "Robust heart rate measurement from video using select random patches," in *IEEE CVPR'15*, Santiago, Chile, Dec. 2015.

[27] S. Haykin and B. Widrow, *Least-mean-square adaptive filters*. John Wiley & Sons, 2003, vol. 31.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[29] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[30] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[31] S. Haykin and N. Network, "A comprehensive foundation," *Neural Networks*, vol. 2, no. 2004, 2004.

[32] M. Gregoski, M. Mueller, A. Vertegel, A. Shaporev, B. Jackson, R. Frenzel, S. Sprehn, and F. Treiber, "Development and validation of a smartphone heart rate acquisition application for health promotion and wellness telehealth applications," *International journal of telemedicine and applications*, vol. 2012, p. 1, 2012.

[33] J. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[34] https://i-msdn.sec.s-msft.com/dynimg/IC584331.png.

[35] S. Xu, L. Sun, and G. Rohde, "Robust efficient estimation of heart rate pulse from video," *Biomedical optics express*, vol. 5, no. 4, pp. 1124–1135, March 2014.