Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 3240 – 3258
© TÜBİTAK
doi:10.3906/elk-2010-93

Presentation attack detection for face recognition using remote photoplethysmography and cascaded fusion

Mehmet Fatih GÜNDOĞAR[1,2,*], **Çiğdem EROĞLU ERDEM**[2]
[1]Research and Development Center, Architecht Bilisim Sistemleri ve Pazarlama Ticaret A.S., Istanbul, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Marmara University, Istanbul, Turkey

Received: 21.10.2020 • **Accepted/Published Online:** 03.09.2021 • **Final Version:** 30.11.2021

Abstract: Spoofing (presentation) attacks are important threats for face recognition and authentication systems, which try to deceive them by presenting an image or video of a different subject, or by using a 3D mask. Remote (non-contact) photoplethysmography (rPPG) is useful for liveness detection using a facial video by estimating the heart-rate of the subject. In this paper, we first compare the presentation attack detection performance of three different rPPG-based heart rate estimation methods on four datasets (3DMAD, Replay-Attack, Replay-Mobile, and MSU-MFSD). We also present a cascaded fusion system, which utilizes a multistage ensemble of classifiers using rPPG, motion-based (including head-pose, eye-gaze and eye-blink), and texture-based features. Experimental results show that the proposed method outperforms several other presentation attack detection methods in the literature, which utilize rPPG.

Key words: Face recognition, presentation attack detection, non-contact photoplethysmography, rPPG

1. Introduction

Biometric authentication is needed in a wide variety of applications including security, finance, law enforcement, and education. Biometric authentication systems utilize physiological or behavioral characteristics of a person such as fingerprint, iris, voice, face, gait, etc. which are prone to spoofing attacks. Face recognition systems, which utilize images or videos have become prevalent recently. However, these systems may be spoofed using hyper-realistic three-dimensional face masks or high-resolution photo and video recordings, which are also known as presentation attacks. It is a challenging task to propose a generalizable solution for face recognition, which is able to distinguish between a fake (i.e. presentation attack) and a real face.

Presentation attack detection (PAD) using features containing vitality data such as the heart rate estimated using rPPG has gained importance recently [1], which uses a facial video to estimate the heart rate of a person. The light absorption/reflection property of the capillary layer of the skin changes proportional to the volume of blood each time the heart beats. This causes subtle changes in the red, green and blue color channels of the skin pixels on the face, where green channel has been shown to carry more information. The color changes have some structural differences due to the absence of a live face in photograph or mask attacks. Based on these temporal color variations due to pulses, some PAD methods in the literature use features extracted by an rPPG process for the classification of the face video of a subject as genuine or fake [2, 3].

There are other motion and texture-based features used for presentation attack detection. Recently,

[*]Correspondence: mehmet.gundogar@architecht.com

learning-based features have also been used. In a recent survey, [4] hardware and software-based PAD methods in the literature are summarized. However, this survey does not include recent rPPG-based and deep learning-based PAD methods.

The contributions of this paper are as follows:

- We give a short literature survey of most recent PAD methods, focusing on recent rPPG and deep learning-based methods.
- We compare the performance of three rPPG-based feature extraction methods for presentation attack detection.
- We present a novel cascaded PAD system, which fuses rPPG, texture, head, and facial motion features at feature level and/or decision level, using a multi-stage ensemble of classifiers. To the best of our knowledge, this is the first work that fuses rPPG, texture, and motion-based features in a PAD method.

The rest of the paper is organized as follows. In Section 2, we present a brief literature survey on recent PAD methods, focusing more on deep learning-based and rPPG-based methods. In Section 3, we give the details of the proposed cascaded fusion method. In Section 4, we provide our experimental results including the comparison of three rPPG methods for PAD. Finally, in Section 5, concluding remarks are given.

2. Literature survey of presentation attack detection

In the following subsections, we briefly summarize the literature on PAD which uses 2D images and videos.

2.1. rPPG-based methods

There are studies for PAD, which combine texture-based and rPPG-based features. In [2] heart rate and local binary pattern (LBP) [5] features are combined to achieve better results for photo and 3D mask attacks since they do not contain heartbeat related rPPG signals. The same expectation does not apply for video replay attacks since attack videos still contain rPPG signals, hence only LBP features are employed to detect video replay attacks in [2]. They identify 68 facial landmarks by the discriminative response map fitting (DRMF) method [6]. Then, detrending, moving average, and Hamming window-based FIR band-pass filters are applied to the temporal rPPG signals. Finally, fast fourier transform (FFT) is used to obtain the power spectrum. The resulting feature vector is used to train a support vector machine (SVM) classifier for the two-class classification task (genuine/fake).

Hernandez-Ortega et al. [7] collected a new heart rate (HR) dataset that included near-infrared (NIR) spectral samples as well as RGB images and applied a PAD approach similar to [2]. It was observed that the NIR spectrum is more resistant to light variations than the RGB spectrum. They also evaluated equal error rate (EER) metrics using different video length schemes for both RGB and NIR spectrum bands. Heusch et al. [3] distinguish presentation attacks from genuine accesses using the long-term statistical spectral feature (LTSS) of the PPG signal and an SVM classifier. Nowara et al. [8] extract the power spectrums separately from the pixels in three different face regions, which are the forehead, left cheek, and right cheek as well as two different regions of the background outside the face area and use them as a feature vector for classification. They train SVM and random forest classifiers and observe that the using spectral features of the background region positively affect the performance of classification.

Detection of fake face images and videos produced by generative adversarial networks (GAN) is also possible using rPPG features [9]. In [9] a novel PPG map feature is introduced and fed into a convolutional neural network (CNN) with the video frames. They also test the effectiveness of different segment durations for the probabilistic video classification process.

In general, rPPG-based methods could provide a robust solution for PAD, but they have a difficulty in the detection of video replay attacks that still contain rPPG signals. Due to this fact, an rPPG-based PAD solution needs to be supported with additional features for the detection of video replay attacks, which is the main concern of this paper.

2.2. Deep learning-based methods

It has been observed that combining facial motion and texture using deep learning methods produces promising results for PAD. Shao et al. [10] train a 5-layer convolutional neural network (CNN) and extract an average optical flow to realize genuine and fake face classification.

Li et al. [11] train a 3D CNN by replicating spatial and temporal information with a specially designed data augmentation method. They also apply a spatial augmentation method to detect background variations in the video and capture the clues of the mediums used in the presentation attack, such as the bezels of the mobile phone or tablet device used in the attack. This process also considers the segments of the face and uses them as additional training data for the CNN. As a different data augmentation method, they apply contrast degree correction flow to the video frames.

Song et al. [12] develop the spatial pyramid coding micro-texture (SPMT) and template face matched binocular depth (TFBD) features to characterize local appearance information. Along with these features, they also utilize a single shot detector (SSD) to detect context cues and configure an end-to-end PAD system. TFBD features require two separate camera images taken from different angles at the same time.

Mohammadi et al. [13] mention the domain-shift problem of PAD methods and propose a novel method to improve cross-dataset performance by modeling the nuisance factors like lighting conditions and different camera devices using a face recognition dataset containing millions of bona fide images. Li et al. [14] approach the generalization problem of PAD as a supervised anomaly detection problem and utilize a hypersphere loss function.

Recently, a PAD challenge was organized to introduce the multimodal CASIA-SURF dataset [15], which contains RGB, depth, and infrared (IR) information about the faces. This challenge revealed that ensemble learning has an exceptional advantage in deep learning. Another observation was that rPPG signals provide essential differences between genuine and spoof faces. Yang et al. [16] also present a solution to collect a large amount of live face data and synthesize spoofing data with reflection artifacts. They propose the spatio-temporal antispoof network (STASN) deep network and report significant performance increase in inter and cross-testing experiments.

Deep learning-based PAD methods require a large amount of training data. However, most PAD datasets contain a limited set of data, which is a common impediment of deep learning-based methods. If the dataset is small, workaround methods such as data augmentation or the use of a different dataset for training may be utilized. However, if the training and test datasets are different, it will become harder to compare the results of a PAD method with other PAD methods in the literature.

2.3. Other methods

Although rPPG-based methods achieve successful results for photo and mask attacks, they are insufficient for video replay attacks when used alone. Patel et al. [17] use multi-scale LBP (MLBP) and scale-invariant feature transform (SIFT) methods for the detection of cascading textures occurring in video replay attacks without using the rPPG signal. The calculated histograms are classified using an SVM classifier. Chingovska et al. [18] propose the use of client-specific information for anti-spoofing. The proposed solution directly rejects a subject if the subject is not enrolled in the system (even if the subject is real). Since the solution is dependent on client data, it is not possible to use it at enrollment time immediately. Hao et al. [19] introduce a solution based on Siamese CNNs. They train the Siamese neural network with the paired input of enrolled subjects' images. The suggested solution first performs face recognition, then retrieves the real image of the identified subject and uses a client-specific Siamese neural network for genuine or fake classification.

2.4. Datasets

The **Replay-Attack** [20] dataset contains a total of 1300 face videos collected from 50 subjects. The videos were recorded using a MacBook camera at a resolution of 320 x 480 pixels. Some of the videos have uniform background and lighting conditions, and others were recorded in a complex environment with natural lighting and reflections. The presentation attack tools used are iPad 1, iPhone 3GS, and A4 printed paper. The **Replay-Mobile** [21] dataset contains 1190 videos from 40 subjects. It contains video recordings of high quality and resolution to reveal more compelling attack scenarios. Recordings were done with iPad Mini2 tablet and LG-G4 smartphone devices using a resolution of 1280×720 pixels. The **MSU-MFSD** [22] dataset contains 280 video recordings collected from 35 subjects of different ethnic origins under different lighting conditions. Videos were recorded with the built-in camera of a MacBook Air laptop using a resolution of 640×480 pixels and with the frontal camera of a Google Nexus 5 smartphone using a resolution of 720×480 pixels. The **3DMAD** [23] dataset contains 255 videos from 17 subjects. There are ten real videos for each subject and five videos captured using the physically obtained mask of the subject. Each of the videos has a resolution of 640×480 pixels and is recorded with the Kinect device for 10 s. The **CASIA FASD** [24] dataset consists of 50 genuine subjects from Asian ethnic regions. It contains three types of face spoofing attacks: warped photo attacks, cut photo attacks, and video attacks. For each subject, the dataset contains three real and nine fake videos. The **Oulu-NPU** [25] dataset contains 4950 videos recorded by the front cameras of six different mobile phones. It contains print attack clips obtained from two printers and video replay attack clips obtained from two display devices. The **Facebook DeepFake detection challenge** [26] contains 10K videos in the blackbox dataset consisting of real face videos and fake videos generated by deep neural networks. The subjects present diversity in terms of age, skin tone, and gender attributes.

Table 1 provides an overview of the popular PAD datasets in the literature. Figure 1 shows example images for genuine images and presentation attacks in four different datasets.

2.5. Evaluation Metrics

In the literature, rPPG-based PAD algorithms generally use the half total error rate (HTER) (4) metric for performance evaluation, which is calculated using a threshold value (θ) that minimizes the EER metric. The EER corresponds to the point that the false acceptance rate (FAR) is equal to the false rejection rate (FRR). The HTER metric is calculated using the FAR and FRR metrics with the corresponding EER threshold value

Table 1. Brief overview of the PAD datasets in the literature.

Dataset	**Resolution**	**Attack Types**	**# of Subjects**	**# of Genuine Videos**	**# of Attack Videos**	**Fold Structure**
Replay-Attack [20]	320 × 240	printed photo, replay video	50	300	1000	Training, Test Development, Enrollment
Replay-Mobile [21]	1280 × 720	printed photo, replay video	40	550	640	Training, Test Development, Enrollment
MSU-MFSD [22]	640 × 480 720 × 480	mobile photo, mobile video	35	70	210	Training, Test
3DMAD [23]	640 × 480	3D mask	17	170	85	17-fold LOOCV
CASIA-FASD [24]	640 × 480 1920 × 1080	warped photo, cut photo, replay video	50	150	450	Training, Test
OULU-NPU [25]	Six different resolutions	printed photo, replay video	55	990	3960	Training, Test Development

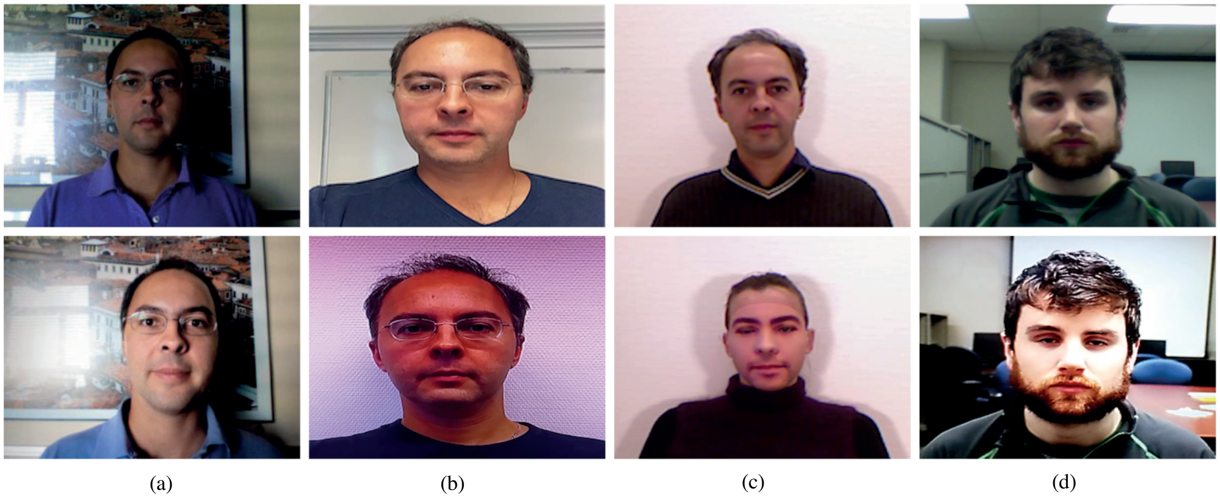

Figure 1. The top row shows images of real faces and the bottom row shows images of presentation attacks from the (a) Replay-Attack, (b) Replay-Mobile, (c) 3DMAD, and (d) MSU-MFSD datasets.

(θ) as follows:

$$FAR = \frac{\#\text{ of false acceptance}}{\#\text{ of identification attempts}} \tag{1}$$

$$FRR = \frac{\text{\# of false rejection}}{\text{\# of identification attempts}} \tag{2}$$

$$EER = \text{the point in the ROC curve where } FAR = FRR \tag{3}$$

$$HTER(\theta) = \frac{FAR(\theta) + FRR(\theta)}{2} \tag{4}$$

HTER takes values in the range [0, 1] and values closer to the 0 is desired for the experiments for this metric.

3. Cascaded fusion for presentation attack detection

The overall view of the proposed cascaded fusion system PAD is shown in Figure 2. The face video is the input, which may contain a mask, image or video replay attack. The goal is to determine whether the input video is a genuine access or a presentation attack. Before feature extraction, we extract the face of interest (ROI) by employing the Viola–Jones [27] algorithm, which provides a rough alignment of the face. The feature extraction block consists of the extraction of three types of features: rPPG features, motion-based features (consisting of head-pose, eye-gaze, eye-blinks), and texture-based features. These blocks will be explained in more detail in the following sections.

After extraction of the above features, the cascaded fusion block is used for classification. The features may be fused using different combinations giving longer feature group (FG) vectors, which are then passed through a feature selection step and classified using specific SVMs for each feature group. Thus, we obtain an ensemble of classifiers trained on different feature groups. Each SVM in the ensemble gives a probability vector output for a test video, indicating the probability of it belonging to a certain class. These vectors are then fused using decision-level fusion approaches. Finally, the fused probability vector is used as a feature vector for the second phase of the SVM classification. The details of each block will be described in the following sections.

3.1. rPPG Features

Most of the rPPG-based heart rate estimation algorithms in the literature [28–32] analyze changes in the mean color of the facial region of a subject using these steps:

- The face is detected at each frame and a region of interest (ROI) is specified.
- Skin pixels are determined in the ROI, and the color values of the skin pixels are averaged at each frame, resulting in temporal signals. This eliminates the background or irrelevant pixels, which may introduce noise to the rPPG signal.
- The temporal signals are transformed or decomposed into components, one of which is expected to represent to the heart signal.
- The power spectrum of the component signals are analyzed to estimate the heart rate.

rPPG algorithms are sensitive to changes in illumination and motion of the subject. Some algorithms implement additional pre-processing steps to improve their performance. In this work, we compared three

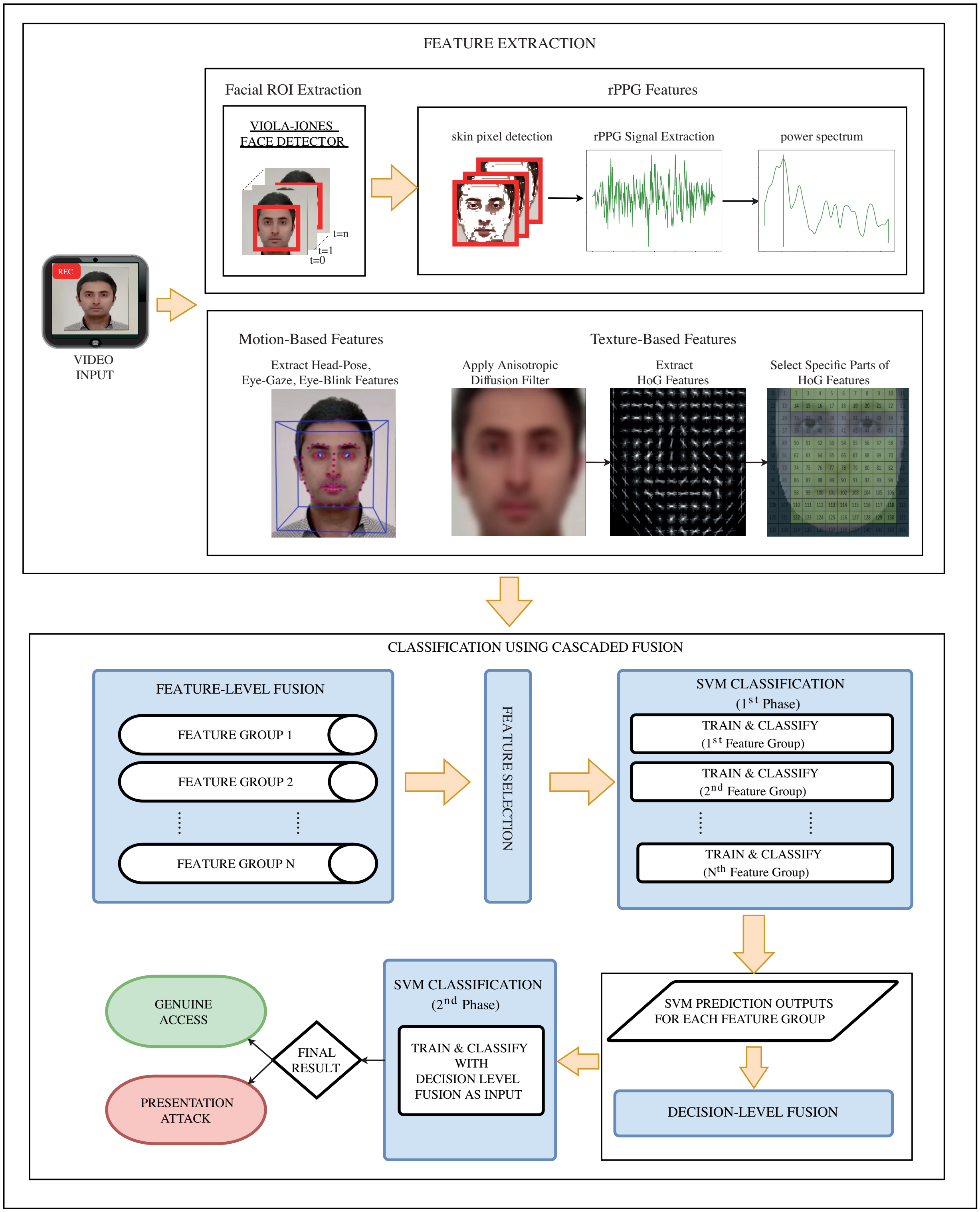
FEATURE EXTRACTION
Facial ROI Extraction
VIOLA-JONES FACE DETECTOR
t=n
t=1
t=0
rPPG Features
skin pixel detection
rPPG Signal Extraction
power spectrum
REC
VIDEO INPUT
Motion-Based Features
Extract Head-Pose, Eye-Gaze, Eye-Blink Features
Texture-Based Features
Apply Anisotropic Diffusion Filter
Extract HoG Features
Select Specific Parts of HoG Features
CLASSIFICATION USING CASCADED FUSION
FEATURE-LEVEL FUSION
FEATURE GROUP 1
FEATURE GROUP 2
FEATURE GROUP N
FEATURE SELECTION
SVM CLASSIFICATION (1st Phase)
TRAIN & CLASSIFY (1st Feature Group)
TRAIN & CLASSIFY (2nd Feature Group)
TRAIN & CLASSIFY (Nth Feature Group)
SVM PREDICTION OUTPUTS FOR EACH FEATURE GROUP
DECISION-LEVEL FUSION
SVM CLASSIFICATION (2nd Phase)
TRAIN & CLASSIFY WITH DECISION LEVEL FUSION AS INPUT
FINAL RESULT
GENUINE ACCESS
PRESENTATION ATTACK

Figure 2. Overall view of the proposed presentation attack detection for face recognition method using cascaded fusion.

methods for extraction of rPPG features, which are briefly described below. For further details, the reader is referred to the respective papers.

Spatial subspace rotation (2SR) [28]: The 2SR algorithm requires facial ROI detection and skin pixels extraction from the facial ROI of a subject. After detection of face ROI, the algorithm estimates the Gaussian parameters of the skin color distribution for the cropped face image. This step computes the mean and covariance matrix of the skin pixels in the normalized red-green color space to obtain a skin mask for given threshold input. After that, the algorithm calculates the eigenvalues and eigenvectors from the skin color correlation matrix. At each frame, the rPPG signal building process begins with the use of these eigenvalues and eigenvectors.

Let C denote the skin color correlation matrix calculated for the face at the k^{th} frame and $U^k = \{u_i^k\}$ and $\Lambda^k = \{\lambda_i^k\}$ are the corresponding eigenvectors and eigenvalues. Subspace rotation SR vector in a single stride (temporal window) is calculated as [28]:

$$SR = \sqrt{\frac{\lambda_1^k}{\lambda_2^\tau}} {u_1^k}^T u_2^\tau {u_2^\tau}^T + \sqrt{\frac{\lambda_1^k}{\lambda_3^\tau}} {u_1^k}^T u_3^\tau {u_3^\tau}^T \tag{5}$$

where λ_i^τ and u_i^τ, $i = 2, 3$ are the eigenvalues and eigenvectors for the reference frame τ.

Also in the current stride, multiple SR vectors are calculated between the reference and successive frames, and they are concatenated in a trace SR. Then, the pulse signal for that stride is built using the first two traces SR_1, SR_2 in the trace SR as [28]:

$$p = SR_1 - \frac{\sigma(SR_1)}{\sigma(SR_2)} SR_2 \tag{6}$$

where σ is the standard deviation operator.

The final pulse signal is estimated with accumulating the $(p - \mu(p))$ value calculated in each stride as [28]:

$$P = P + (p - \mu(p)) \tag{7}$$

where $\mu(p)$ is the mean signal calculated for the current stride.

Chrominance-Based rPPG (CHROM) [31]: The CHROM algorithm obtains the skin color signals from a subject's facial ROI at each frame of a video sequence. Similar to the 2SR algorithm, estimation of the Gaussian parameters of the skin color distribution is performed to gather a skin mask using the provided threshold. The algorithm projects the mean skin color calculated from the RGB color space to the defined chrominance XY space to achieve robustness to illumination variations as follows [31]:

$$\begin{aligned} X &= (3.0 * r) - (2.0 * g) \\ Y &= (1.5 * r) + g - (1.5 * b) \end{aligned} \tag{8}$$

where $r, g,$ and b corresponds to the values of the red, green, and blue color channels, respectively. Later on, the algorithm applies a bandpass filter in the chrominance space.

Li's CVPR14 [32]: In this algorithm, first, skin color signals from the facial ROI of each video frame are estimated. The color signals of the background are also used to reduce the impact of global illumination variations. Let g_{face} denote the mean green value of pixels obtained from the face ROI and let g_{bg} denote the mean green value of pixels of the background ROI extracted from the top-left part of the video frame. The relation between them is:

$$\begin{aligned} g_{face} &= s + y \\ y &\approx h g_{bg} \end{aligned} \tag{9}$$

where s denotes the green value variations due to the cardiac pulse and y denotes the green value variations for the illumination changes. The illumination rectified signal g_{IR} is defined as [32]:

$$g_{IR} = s + (y - h g_{bg}) \tag{10}$$

Then, a normalized least mean square (NLMS) filter is applied to calculate the optimal h value. After obtaining the illumination rectified signal, it is divided into temporal segments and segments having a high standard deviation are removed to eliminate the effects of motion. In the final step, the algorithm applies a detrending filter, a moving average filter, and a Hamming window-based finite impulse response bandpass filter to build the rPPG signal.

We first estimate the rPPG signal using one of the three algorithms mentioned above and then estimate the power spectrum density (PSD) of the rPPG signal $P(f)$. Then, the peak frequency of the PSD with the maximum power value is detected in the frequency range [0.7Hz, 4Hz]. Finally, similar to the methods in [2], [33], the frequency corresponding to the highest peak is multiplied by 60 to obtain a heart rate estimate between 42 beats-per-minute (bpm) and 240 bpm.

We used the implementations of the three algorithms given in Bob's toolbox [34]. We represented the rPPG-based feature vector in two different ways:

- Let f_{max} denote the frequency, which has the highest power in $P(f)$, i.e. $f_{max} = \arg\max_f P(f)$. Then, we calculate the following ratio [2]:

  $$\Gamma = \frac{P(f_{max})}{\sum_{\forall f \in [0.7,4]} P(f)} \tag{11}$$

  The two dimensional rPPG feature vector is then formed as:

  $$PPG_1 = [P(f_{max}), \Gamma]. \tag{12}$$

- We also used the magnitude of the power spectrum of the rPPG signal ($P(f)$) as a feature vector after band-pass filtering with a passband [0.7Hz, 4Hz]:

  $$PPG_2 = |P(f)|, \tag{13}$$

  which gives us a 1×541 dimensional feature vector.

3.2. Motion-based features

Although rPPG features are beneficial for liveness detection, they are not sufficient to distinguish all kinds of presentation attacks as the pulse signal still exists in a video replay attack. Illumination variations in a video can also impact the rPPG feature extraction process negatively. Hence, we used motion-based features of the head and face for PAD. We utilized the OpenFace 2.0 [35–39] toolbox for the extraction of motion features. For the motion-based features, face alignment is done by Open Face 2.0 toolbox with the utilization of estimated landmark points on the face. We estimate the eye-blink, eye-gaze angle, and head-pose features at each frame and calculate their mean to use as features for a video. These motion features are expected to be useful for detecting photo attacks since photo attacks have no eye-blinks and no significant eye-gaze and head-pose changes.

The blink feature will be denoted by $BLINK$, which is a scalar. If there is no eye-blink in a video, the value of the blink feature will be 0, otherwise it will be 1. Eye-blinks are detected based on the openness and closeness states of both eyes. OpenFace 2.0 uses the HOG features of each frame to detect eye-blinks.

Gaze angle (α) corresponds to the angle between the $\vec{u}$ and $\vec{v}$ vectors, where $\vec{u}$ is the vector between the pupil location (point P) and the point of interest (I - which is the attention point of the subject), and $\vec{v}$ is the vector between point P and the camera center (point C) [40], as illustrated in Figure 3. Let's denote the gaze angle feature vector as $EYEGAZE = [\mu(\angle_x), \mu(\angle_y)]$, where $\angle_x$ and $\angle_y$ indicate the eye-gaze direction in radians in world coordinates averaged for both eyes, which depend on the vector $\vec{u}$ in Figure 3 and $\mu(\cdot)$ is the mean operator over video frames.

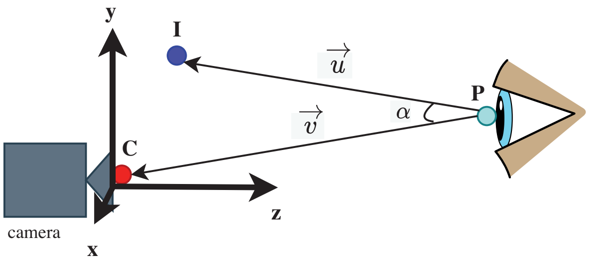

Figure 3. Illustration of computation of gaze angle.

The head-pose feature vector is six-dimensional:

$$HEADPOSE = \mu(LOC_x), \mu(LOC_y), \mu(LOC_z), \mu(ROT_x), \mu(ROT_y), \mu(ROT_z)] \tag{14}$$

where $LOC_x^k, LOC_y^k, LOC_z^k$ represents the location of the head with respect to the camera, and $ROT_x^k, ROT_y^k, ROT_z^k$ represents the rotation angles around the x, y, z axes for the k^{th} frame.

We first performed preliminary experiments on the Replay-Attack and Replay-Mobile datasets to see the impact of motion-based features. We anticipated that these features could provide high classification accuracy, especially for the photo attack presentations since there is lack of face and head motion in photo attacks. Consequently, extracted eye-blink, eye-gaze, and head-pose features from photo attack presentations could exhibit similar patterns. According to our initial experimental results utilizing the motion-based feature set $\{BLINK, EYEGAZE, HEADPOSE\}$, the classification accuracy on the photo attack labeled videos of the Replay-Attack and the Replay-Mobile datasets were 94.16% and 94.44%, respectively. Thus, in order to improve the performance of the proposed PAD system, we also decided to utilize the motion-based features together with other features.

3.3. Texture-based features

Texture-based features are useful for discriminating flat patterns such as the surface of the masks used in 3D mask attacks. We apply an anisotropic diffusion filter on facial ROI to smooth the frames and reduce noise as a preprocessing step since the fake presentations have been shown to be affected more from this filter [41]. After the application of anisotropic diffusion filter, we extract the HoG feature descriptors for each of the 81 face regions indicated by numbered yellow squares in Figure 4 that consists of 31 bins corresponding to the angles from 0 to 360 in increments of 12 °. The magnitude values are added to the corresponding bins representing their direction.

If we concatenate the HoG feature descriptors for the selected 81 face regions, the dimension of the HOG feature vector becomes 1×2511 (81 face regions × 31 bins). We will denote the texture feature vector as HOG.

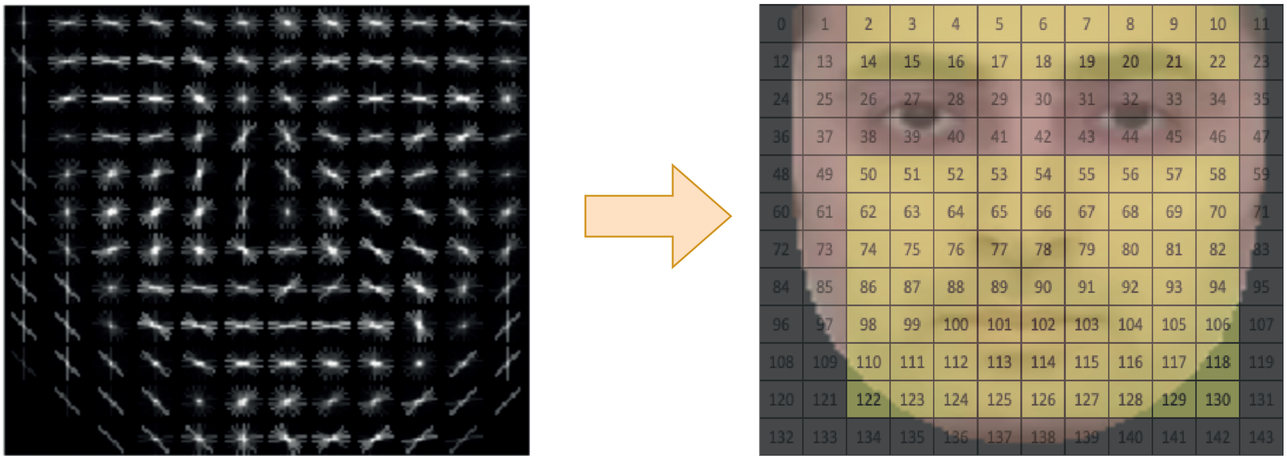

Figure 4. (left) HoG features of each cell (right) Corresponding facial region cells. The HoG feature descriptors of the yellow squares are concatenated to obtain the HOG feature vector to represent the face.

3.4. Feature-level fusion and feature selection

In order to test the effectiveness of various feature combinations, we introduce a feature-level fusion step for combining different feature vectors in our feature set $\{PPG_1, PPG_2, BLINK, EYEGAZE, HEADPOSE, HOG\}$ in order to form different feature groups, which will be denoted as FGi, where $i = 1, \dots, N$.

We tested all semantic combinations of feature groups and selected the ones giving the best experimental results, which resulted in the feature groups shown in the second column of Table 2. In order to improve the classification accuracy and reduce overfitting, we apply feature selection by utilizing the extra-trees classifier [42], which provides feature importances. Each decision tree uses the original training samples. For each tree, k randomly selected features are provided, and the tree selects the best feature to split the data using a criterion such as the Gini index [43] or entropy [44]. This random sampling method creates multiple decorrelated decision trees. Each feature's importance is calculated using the Gini index, and the features are listed in descending order. It is expected that the Gini index is equal to 0 for a split which gives perfect classification. When using the Gini criterion, the probability of each class is calculated for each feature in each branch of a built tree. Then the sum of the squared probabilities is subtracted from 1, which gives us the value of the Gini index. The sum of the calculated Gini indexes of each split is used to calculate feature importances.

After feature ranking, we eliminate the less important features according to a certain threshold (for our experiments we eliminated 10% of the features considering their importance order) to improve the performance of SVM classifiers (1^{st} phase) trained on the N different feature groups. Hence, we obtain an ensemble of N classifiers, the outputs of which are 1×2 dimensional probability vectors for two-class classification (genuine/attack) as $[p^i_{genuine}, p^i_{attack}]$ for the i^{th} SVM classifier. We also employed three-class classification (genuine/photo attack/video-replay attack), the outputs of which are 1×3 dimensional probability vectors as $[p^i_{genuine}, p^i_{photoattack}, p^i_{videoattack}]$ for the i^{th} SVM classifier.

3.5. Decision-level fusion

We introduce a decision-level fusion step to combine the results of N feature groups (if $N > 1$ in an experiment). In the first phase (see Figure 2), we train N SVM classifiers in an ensemble using each feature group. Each of these SVM classifiers will give a probability vector as output containing the genuine class and presentation attack class probabilities. In two-class classification, we will get a probability vector of dimensions 1×2 for each SVM. Sum of the probability values in this 1×2 dimensional probability vector is equal to 1, that is $p^i_{genuine} + p^i_{attack} = 1$ for the SVM output of the i^{th} feature group. Similarly, for three-class classification, $p^i_{genuine} + p^i_{photoattack} + p^i_{videoattack} = 1$.

We concatenate the N genuine class probabilities $p^i_{genuine}$, which are produced by the N SVM classifiers in the 1^{st} phase corresponding to the N feature groups. Feature vector obtained after this concatenation is denoted as F, which is $1 \times N$ dimensional:

$$F = [p^1_{genuine}, p^2_{genuine}, ..., p^N_{genuine}] \tag{15}$$

Then, in the second phase, we train another single SVM to classify the input video as an attack or genuine presentation (see Figure 2). If the value of $p^i_{genuine}$ is closer to 1 for most of the selected feature groups, it is expected that the classification will result in a genuine access decision.

4. Experimental results

In this section we first experimentally compare the three rPPG estimation methods (2SR, CHROM, Li's CVPR14) in terms of their PAD performance on four datasets (3DMAD, Replay Attack, Replay Mobile, MSU-MFSD), which include different types of spoofing attacks as mask, photo and video-replay attacks. Then, we give the results of the proposed cascaded fusion method.

4.1. Experimental setup

For the experiments on the 3DMAD dataset videos, we adopted a leave-one-subject-out-cross-validation (LOOCV) protocol similar to [2, 45]. Since there are 17 subjects in this dataset, we have 17 folds in our experiments. At each fold, we leave one subject for testing and the remaining subjects are split equally into training and development sets. For the Replay-Attack and the Replay-Mobile datasets, we use the given train, development, and test partitions provided in the grandtest protocol similar to the studies in [3, 8]. For the MSU-MFSD dataset, we use the given train and test partitions, which were also used in the experiments in [2]. For the 3DMAD, Replay-Attack, and Replay-Mobile datasets, we report the results with the HTER metric. Since the MSU-MFSD dataset does not contain any development fold for the experiments, we report the results for this dataset with the EER metric.

4.2. Experimental comparison of rPPG methods for PAD

The experimental results in terms of HTER and EER using the 2D PPG (PPG_1 defined in (12)) and magnitude features (PPG_2 defined in (13)) extracted using three different rPPG algorithms are given in the first six rows in Table 2 as configurations [config-1 - config-6]. We can observe that the magnitude features extracted with the CHROM algorithm give the smallest HTER for the 3DMAD (8.82%) and the Replay-Attack (11.25%) datasets. On the other hand, the 2SR algorithm performs better on the Replay-Mobile and the MSU-MFSD datasets as compared to the CHROM algorithm. The magnitude features extracted by the Li's CVPR14 algorithm also improves the classification performance on the Replay-Mobile dataset.

4.3. Experiments for evaluating cascaded fusion

In our experiments, we focused on the impact of the feature-level and decision-level fusion phases of our PAD system. The experimental results for different configurations are presented in Table 2 as experimental configurations config-7 - config-11. In config-7 setup, we fuse all types of features (rPPG, texture-based, and motion-based) in a single feature group. Since config-7 uses a single feature group, the decision-level fusion property is not active in this setup. In config-8 setup, we only use rPPG-based features extracted using CHROM algorithm in a single feature group without utilizing decision-level fusion. In the config-9 setup, we only use rPPG features obtained by the CHROM algorithm and apply a decision level fusion with two feature groups. In the config-10 setup, we place each type of feature in a single feature group and try to see the impact of decision-level fusion. In the last configuration (config-11), we try to see the impact of a semantic grouping of features as separate feature groups. In this setup, the first feature group contains only PPG features, the second feature group contains texture-based features, and the third feature group contains motion-based features.

Experiments are performed on a MacBook Pro 13-inch laptop with a 2.4 GHz Quad-Core Intel Core i5 processor. It takes on average 24 s to detect the face region and extract PPG features for a 10 s long video with a resolution of 320×240. Using the OpenFace 2.0 toolbox, the average elapsed time is about 9 s to extract $HOG, BLINK, EYEGAZE$, and $HEADPOSE$ features. Feature selection takes approximately 5 s and SVM classification takes less than a second.

4.4. Discussion and comparison of the results

As it can be observed from Table 2, fusing all types of features in a single group (config-7) only gives the best result for the 3DMAD dataset (0.58% HTER), which has only mask attacks. Besides, for the 3DMAD dataset the HTER values are very low as 0.58%, 1.47%, and 0.88% in config-7, config-10, and config-11, respectively, where all these configurations contain HOG features in common. Based on these results, we can make an inference that HOG features perform well on the mask attacks. For the other datasets, which contain different types of attacks, concatenating all features does not give the best results. Using only rPPG features (config-8) provides the best HTER (4.62%) for the Replay-Attack dataset. The same setup does not work on the Replay-Mobile dataset because the Replay-Mobile dataset contains more variety of presentation attacks including high-resolution matte-screen videos and photo attacks printed on matte-papers. This variety impacts the performance of PAD with rPPG features negatively (config-8 and config-9) since a video replay attack generated with a matte-screen video still contains a high-quality pulse signal and it is too hard to discriminate it from a genuine video. The MSU-MFSD dataset differs from other datasets in terms of the varying ethnicity of the subjects and challenging illumination conditions. This case directly impacts the performance of the rPPG process and as a consequence, only the utilization of rPPG features (config-8 and config-9) or using all features in a single feature

Table 2. The tested configurations for feature-level and decision-level fusion phases.

Experimental configuration Id	**Feature-level fusion details***	**Decision level fusion status****	**3DMAD (HTER)**	**Replay Attack (HTER)**	**Replay Mobile (HTER)**	**MSU MFSD (EER)**
config-1	FG1:$\{PPG_{1-chrom}\}$ - dim:1×2	Passive	21.76%	33.75%	45.42%	37.61%
config-2	FG1:$\{PPG_{2-chrom}\}$ - dim:1×541	Passive	8.82%	11.25%	54.8%	60.15%
config-3	FG1:$\{PPG_{1-2sr}\}$ - dim:1×2	Passive	22.64%	45.62%	29.26%	40.0%
config-4	FG1:$\{PPG_{2-2sr}\}$ - dim:1×541	Passive	11.76%	49.37%	47.27%	35.83%
config-5	FG1:$\{PPG_{1-cvpr14}\}$ - dim:1×2	Passive	37.35%	42.47%	42.37%	43.49%
config-6	FG1:$\{PPG_{2-cvpr14}\}$ - dim:1×541	Passive	41.02%	31.12%	24.43%	50.0%
config-7	FG1:$\{PPG_{2-chrom},$ $BLINK, EYEGAZE,$ $HEADPOSE, HOG\}$ dim:1×(541+1+2+6+2511)=1×3061	Passive	**0.58**%	50.0%	24.92%	48.33%
config-8	FG1:$\{PPG_{1-chrom}, PPG_{2-chrom}\}$ dim:1×(2+541)=1×543	Passive	41.17%	**4.62%**	54.64%	51.66%
config-9	FG1:$\{PPG_{1-chrom}\}$ dim:1×2	Active	36.47%	5.87%	54.19%	45.0%
	FG2:$\{PPG_{2-chrom}\}$ dim:1×541					
config-10	FG1:$\{PPG_{1-chrom}\}$ dim:1×2	Active	1.47%	10.25%	14.23%	**27.5**%
	FG2:$\{PPG_{2-chrom}\}$ dim:1×541					
	FG3:$\{PPG_{1-2sr}\}$ dim:1×2					
	FG4:$\{PPG_{2-2sr}\}$ dim:1×541					
	FG5:$\{BLINK\}$ dim:1×1					
	FG6:$\{EYEGAZE\}$ dim:1×2					
	FG7:$\{HEADPOSE\}$ dim:1×6					
	FG8:$\{HOG\}$ dim:1×2511					
config-11	FG1:$\{PPG_{1-chrom}, PPG_{2-chrom}\}$ dim:1×543	Active	0.88%	17.0%	**11.69**%	35.0%
	FG2:$\{HOG\}$ dim:1×2511					
	FG3:$\{BLINK, EYEGAZE,$ $HEADPOSE\}$ dim:1×9					

*The acronym FG represents a feature group. In each experimental configuration, different feature groups are tested. For each FG a separate SVM classifier is trained. Dimensions (dim) of each feature group is also indicated in the second column.
**Decision-level fusion does not exist when there is a single feature group in the corresponding experimental configuration. If there are more than one feature groups in an experimental configuration, a second phase SVM is utilized.

group (config-7) does not provide acceptable results for the MSU-MFSD dataset. After using the decision-level fusion setups (config-10 and config-11), we see the real contribution of the decision-level fusion step of our PAD system on the Replay Mobile and MSU-MFSD datasets as compared to using only the feature-level fusion or using only rPPG features, since the HTER are smaller.

The best results for each dataset are obtained using different feature groups, which can be considered as a domain-shift problem related to the structural differences of the datasets [13]. As listed in Table 1, properties of all PAD datasets differ from each other in terms of the resolution of the videos and the variety of attack types. Besides, spoofing mediums, different types of camera devices used for video recordings, illumination conditions, ethnicity of the subjects, and other environmental conditions are all important factors that impact the results of experiments on PAD. Since we performed our experiments on datasets with different properties, best performing configurations are different for each dataset. Figure 5 illustrates the t-distributed stochastic neighbor embedding (t-SNE) [46] plot obtained with the feature vectors of config-7 in Table 2, extracted from the Replay-Attack, Replay-Mobile, 3DMAD, and MSU-MFSD datasets. In t-SNE, high-dimensional feature data is projected onto a two-dimensional space for the visualization of a dataset. As we can see from the t-SNE plot, the datasets form separate clusters with different distributions, which shows us the domain-shift problem caused by various factors mentioned in the above paragraph.

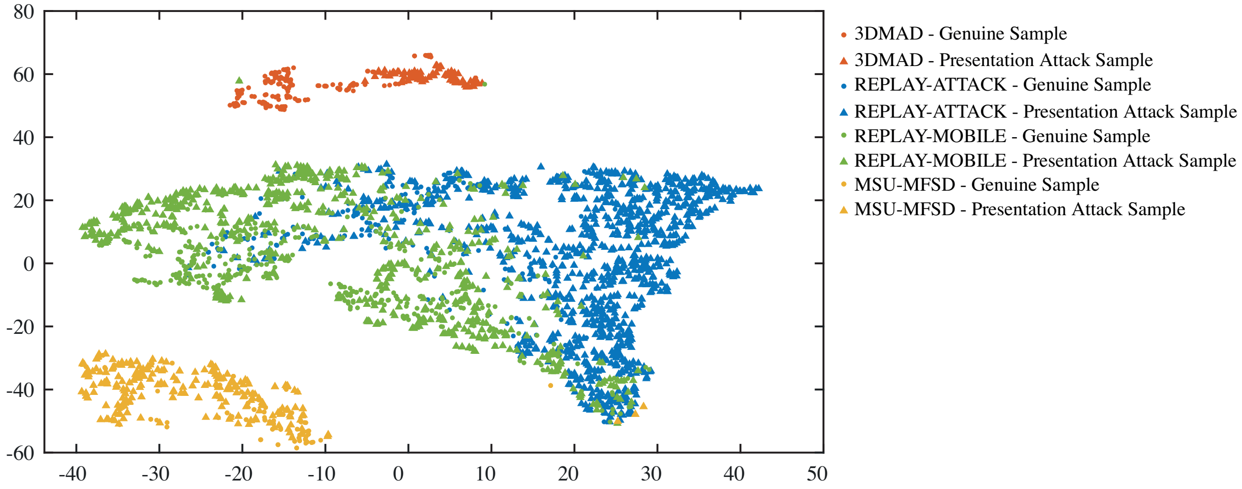

Figure 5. The t-SNE plot of the features used in config-7 in Table 2 for the Replay-Attack, Replay-Mobile, 3DMAD, and MSU-MFSD datasets. Genuine and presentation attack samples of each dataset are shown with circles and triangles, respectively.

However, grouping similar features and adopting decision-level fusion provides promising results on challenging datasets. For instance grouping similar features and using three different feature groups as in config-11 with decision-level fusion provide a dramatic decrease in the value of HTER from 24.92% to 11.69% for the Replay-Mobile dataset, which is much better than using the same types of features in a single feature group as in config-7. Since the final step of decision-level fusion process is based on the genuine class probability output of each separate feature group, we can infer that the second phase SVM is a process that performs a consultation among the separate decisions of each feature group to give its final decision regarding how genuine a subject is. On the other hand, using a single feature group is solitary decision which could not produce better results as compared to using decision-level fusion.

The experimental results compared with other studies in the literature are given in Table 3. Our cascaded fusion system which utilizes rPPG features outperforms some of the rPPG-based PAD methods [2, 3, 8] as shown in Table 3. Our results are comparable with the results reported in [2, 3, 8] since we use the same test protocols for the datasets in our experiments as mentioned in the experimental setup section. However, some of the deep learning-based and texture-based methods in the literature give lower HTER as compared to our study, but most of them apply specific data augmentation methods or specific techniques adopted to a single dataset without any generalization property to obtain better results. For instance, in [2], the LBP-based method gives perfect results for the 3DMAD dataset with the achievement of 0% HTER value, but the same method fails on another REAL-F dataset with varying HTER values between 26% and 47%. In the same study [2], the pulse-based method gives more stable results for both 3DMAD and REAL-F datasets.

Table 3. Comparison of rPPG-based pad results using HTER (%) for four datasets.

Methodology	**3DMAD**	**Replay-Attack**	**Replay-Mobile**	**MSU-MFSD**
Li CVPR + LTSS [3]	17.0	16.1	32.5	35.0
PPG [8]	43.0	25.5	35.9	31.7
CHROM + LTSS [3]	29.0	20.9	38.1	50.6
2SR + LTSS [3]	13.0	5.9	37.7	43.3
PPG + LBP [2]	7.94	-	-	36.67
Proposed method (config-7)	**0.58**	50.0	24.92	48.33
Proposed method (config-8)	41.17	**4.62**	54.64	52.08
Proposed method (config-10)	1.47	10.25	14.23	**27.5**
Proposed method (config-11)	0.88	17.0	**11.69**	35.0

5. Conclusion and future work

In this paper, we investigated the use of rPPG features together with motion and texture features for presentation attack detection for face recognition. Liveness detection with the utilization of rPPG features is an essential factor for obtaining successful results on PAD, especially on mask and photo attacks. However, rPPG-based PAD has an impediment that it is not possible to extract rPPG features on short length videos. In addition, the experimental results reveal that rPPG features are not sufficient for every type of attack scenario, especially video attacks. After augmentation of rPPG features with additional discriminative motion and texture-based features, we obtained lower HTER. The proposed method is supported by different feature types and does not rely on very large training data as deep learning-based methods require. Therefore, it provides a solution for the presentation attack scenarios, which can not be handled by rPPG-based or deep learning-based solutions mentioned in Section 2.

The experimental results show that the approach of concatenating a limited set of features into a single feature group may not give the optimum results. Since the availability of features and datasets for PAD are limited, decision-level fusion is also an essential factor for improving PAD performance when dealing with different types and groups of features. The results of our experiments demonstrate that the contribution of the same features to the performance of a PAD system using decision-level fusion is higher as compared to using them only in feature-level fusion.

Each PAD dataset has different characteristics due to the included attack types, used spoofing mediums, diversity of the recording device, and different illumination conditions of the video records. These are the primary reasons for the domain-shift problem as shown in Figure 5, which also prevents us from finding a seamless working cascaded fusion configuration for all datasets. There is no guarantee for any PAD solution to work perfect for an unseen blackbox PAD dataset. As future work, we will try to adopt our cascaded fusion approach to deep learning-based methods by utilizing more than one deep learning models to achieve a more generalizable PAD framework.

Acknowledgment

In the experiments we used the Replay-Attack, Replay-Mobile and 3DMAD Corpus made available by the Idiap Research Institute, Martigny, Switzerland. Replay-Mobile Corpus was captured in collaboration with the Galician R&D Center on Advanced Telecommunications (GRADIANT), Vigo, Spain.

References

[1] Chen X, Cheng J, Song R, Liu Y, Ward R et al. Video-based heart rate measurement: recent advances and future prospects. IEEE Transactions on Instrumentation and Measurement 2019; 68 (10): 3600-3615. doi: 10.1109/TIM.2018.2879706

[2] Li X, Komulainen J, Zhao G, Yuen PC, Pietikäinen M. Generalized face anti-spoofing by detecting pulse from face videos. In: 2016 23rd International Conference on Pattern Recognition (ICPR); Cancun; 2016. pp. 4244-4249. doi: 10.1109/ICPR.2016.7900300

[3] Heusch G, Marcel S. Pulse-based features for face presentation attack detection. In: 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS); Redondo Beach, CA, USA; 2018. pp. 1-8. doi: 10.1109/BTAS.2018.8698579

[4] Ramachandra R, Busch C. Presentation attack detection methods for face recognition systems: a comprehensive survey. ACM Computing Surveys 2017; 50 (1): 8. doi: 10.1145/3038924

[5] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 2002; 24 (7): 971–987. doi: 10.1109/TPAMI.2002.1017623

[6] Asthana A, Zafeiriou S, Cheng S, Pantic M. Robust discriminative response map fitting with constrained local models. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition; Portland, OR; 2013. pp. 3444-3451. doi: 10.1109/CVPR.2013.442

[7] Hernandez-Ortega J, Fierrez J, Morales A, Tome P. Time analysis of pulse-based face anti-spoofing in visible and NIR. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Salt Lake City, UT; 2018. pp. 657-6578. doi: 10.1109/CVPRW.2018.00096

[8] Nowara EM, Sabharwal A, Veeraraghavan A. PPGSecure: biometric presentation attack detection using photopletysmograms. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017); Washington, DC; 2017. pp. 56-62. doi: 10.1109/FG.2017.16

[9] Ciftci UA, Demir I, Yin L. FakeCatcher: detection of synthetic portrait videos using biological signals. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020; pp. 1-1. doi: 10.1109/TPAMI.2020.3009287

[10] Shao R, Lan X, Yuen PC. Joint discriminative learning of deep dynamic textures for 3D mask face anti-spoofing. IEEE Transactions on Information Forensics and Security 2019; 14 (4): 923-938. doi: 10.1109/TIFS.2018.2868230

[11] Li H, He P, Wang S, Rocha A, Jiang X et al. Learning generalized deep feature representation for face anti-spoofing. IEEE Transactions on Information Forensics and Security 2018; 13 (10): 2639-2652. doi: 10.1109/TIFS.2018.2825949

[12] Song X, Zhao X, Fang L, Lin T. Discriminative representation combinations for accurate face spoofing detection. Pattern Recognition 2019; 85: 220-231. doi: 10.1016/j.patcog.2018.08.019

[13] Mohammadi A, Bhattacharjee S, Marcel S. Improving cross-dataset performance of face presentation attack detection systems using face recognition datasets. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Barcelona, Spain; 2020. pp. 2947-2951. doi: 10.1109/ICASSP40776.2020.9053922

[14] Li Z, Li H, Lam K, Kot AC. Unseen face presentation attack detection with hypersphere loss. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Barcelona, Spain; 2020. pp. 2852-2856. doi: 10.1109/ICASSP40776.2020.9054420

[15] Liu A, Wan J, Escalera S, Escalante HJ, Tan Z et al. Multi-modal face anti-spoofing attack detection challenge at CVPR2019. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); Long Beach, CA, USA; 2019. pp. 1601-1610. doi: 10.1109/CVPRW.2019.00202

[16] Yang X, Luo W, Bao L, Gao Y, Gong D et al. Face anti-spoofing: model matters, so does data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); Long Beach, CA, USA; 2019. pp. 3502-3511. doi: 10.1109/CVPR.2019.00362

[17] Patel K, Han H, Jain A, Ott G. Live face video vs. spoof face video: use of moiré patterns to detect replay video attacks. In: 2015 International Conference on Biometrics (ICB); Phuket; 2015. pp. 98-105. doi: 10.1109/ICB.2015.7139082

[18] Chingovska I, Anjos A. On the use of client identity information for face antispoofing. IEEE Transactions on Information Forensics and Security 2015; 10 (4): 787-796. doi: 10.1109/TIFS.2015.2400392

[19] Hao H, Pei M, Zhao M. Face liveness detection based on client identity using Siamese network. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV); 2019. pp. 172-180.

[20] Chingovska I, Anjos A, Marcel S. On the effectiveness of local binary patterns in face anti-spoofing. In: 2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG); Darmstadt; 2012. pp. 1-7.

[21] Costa-Pazo A, Bhattacharjee S, Vazquez-Fernandez E, Marcel S. The Replay-Mobile face presentation-attack database. In: 2016 International Conference of the Biometrics Special Interest Group (BIOSIG); Darmstadt; 2016. pp. 1-7. doi: 10.1109/BIOSIG.2016.7736936

[22] Wen D, Han H, Jain AK. Face spoof detection with image distortion analysis. IEEE Transactions on Information Forensics and Security 2015. 10 (4): 746-761. doi: 10.1109/TIFS.2015.2400395.

[23] Erdogmus N, Marcel S. Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect. In: 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS); Arlington, VA; 2013. pp. 1-6. doi: 10.1109/BTAS.2013.6712688

[24] Zhang Z, Yan J, Liu S, Lei Z, Yi D et al. A face antispoofing database with diverse attacks. In: 2012 5th IAPR International Conference on Biometrics (ICB); New Delhi; 2012. pp. 26-31. doi: 10.1109/ICB.2012.6199754

[25] Boulkenafet Z, Komulainen J, Li L, Feng X, Hadid A. OULU-NPU: a mobile face presentation attack database with real-world variations. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017); Washington, DC; 2017. pp. 612-618. doi: 10.1109/FG.2017.77

[26] Dolhansky B, Howes R, Pflaum B, Baram N, Ferrer CC. The Deepfake Detection Challenge (DFDC) preview dataset. arXiv preprint 2019, arXiv:191008854.

[27] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001); Kauai, HI, USA; 2001. pp. I-I. doi: 10.1109/CVPR.2001.990517

[28] Wang W, Stuijk S, de Haan G. A novel algorithm for remote photoplethysmography: spatial subspace rotation. IEEE Transactions on Biomedical Engineering 2016; 63 (9): 1974-1984. doi: 10.1109/TBME.2015.2508602

[29] Demirezen H, Erdem CE. Remote photoplethysmography using nonlinear mode decomposition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Calgary, AB; 2018. pp. 1060-1064. doi: 10.1109/ICASSP.2018.8462538

[30] Demirezen H, Erdem CE. Heart rate estimation from facial videos using nonlinear mode decomposition and improved consistency check. Signal, Image and Video Processing (SIVP) 2021; doi:10.1007/s11760-021-01873-x

[31] de Haan G, Jeanne V. Robust pulse rate from chrominance-based rPPG. IEEE Transactions on Biomedical Engineering 2013; 60 (10): 2878-2886. doi: 10.1109/TBME.2013.2266196

[32] Li X, Chen J, Zhao G, Pietikäinen M. Remote heart rate measurement from face videos under realistic situations. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition; Columbus, OH; 2014. pp. 4264-4271. doi: 10.1109/CVPR.2014.543

[33] Rouast PV, Adam MTP, Chiong R, Cornforth D, Lux E. Remote heart rate measurement using low-cost RGB face video: a technical literature review. Frontiers of Computer Science 2018; 12: 858-872. doi: 10.1007/s11704-016-6243-6

[34] Anjos A, Shafey E, Wallace R, Günther M, Mccool C et al. Bob: a free signal processing and machine learning toolbox for researchers. In: Proceedings of the 20th ACM international conference on Multimedia; New York, NY, USA; 2012. pp. 1449–1452. doi: 10.1145/2393347.2396517

[35] Baltrusaitis T, Zadeh A, Lim YC, Morency L. OpenFace 2.0: facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018); Xi'an; 2018. pp. 59-66. doi: 10.1109/FG.2018.00019

[36] Zadeh A, Lim YC, Baltrušaitis T, Morency L. Convolutional experts constrained local model for 3D facial landmark detection. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW); Venice; 2017. pp. 2519-2528. doi: 10.1109/ICCVW.2017.296

[37] Baltrusaitis T, Robinson P, Morency L. Constrained local neural fields for robust facial landmark detection in the wild. In: 2013 IEEE International Conference on Computer Vision Workshops; Sydney, NSW; 2013. pp. 354-361. doi: 10.1109/ICCVW.2013.54

[38] Wood E, Baltruaitis T, Zhang X, Sugano Y, Robinson P et al. Rendering of eyes for eye-shape registration and gaze estimation. In: 2015 IEEE International Conference on Computer Vision (ICCV); Santiago; 2015. pp. 3756-3764. doi: 10.1109/ICCV.2015.428

[39] Baltrušaitis T, Mahmoud M, Robinson P. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In: 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG); Ljubljana; 2015. pp. 1-6. doi: 10.1109/FG.2015.7284869

[40] Banzhaf C. Extracting facial data using feature-based image processing and correlating it with alternative biosensors metrics. MSc, University of Stuttgart, Stuttgart, Germany, 2017.

[41] Alotaibi A, Mahmood A. Deep face liveness detection based on nonlinear diffusion using convolution neural network. Signal, Image and Video Processing (SIVP) 2017; 11: 713-720. doi: 10.1007/s11760-016-1014-2

[42] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B et al. Scikit-learn: Machine learning in python. The Journal of Machine Learning Research 2011; 12: 2825-2830.

[43] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Machine Learning 2006; 63 (1): 3–42. doi: 10.1007/s10994-006-6226-1

[44] Quinlan JR. Induction of decision trees. Machine Learning 1986; 1: 81–106. doi: 10.1007/BF00116251

[45] Erdogmus N, Marcel S. Spoofing face recognition with 3D masks. IEEE Transactions on Information Forensics and Security 2014; 9 (7): 1084-1097. doi: 10.1109/TIFS.2014.2322255

[46] Maaten L, Hinton G. Visualizing Data using t-SNE. The Journal of Machine Learning Research 2008; 9 (86): 2579-2605.