MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

DETECTION OF PRESENTATION ATTACKS FOR FACE RECOGNITION SYSTEMS

MEHMET FATİH GÜNDOĞAR

MASTER THESIS
Department of Computer Engineering

Thesis Supervisor
Prof. Çiğdem Eroğlu Erdem

ISTANBUL, 2020

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

DETECTION OF PRESENTATION ATTACKS FOR FACE RECOGNITION SYSTEMS

MEHMET FATİH GÜNDOĞAR

MASTER THESIS
Department of Computer Engineering

Thesis Supervisor
Prof. Çiğdem Eroğlu Erdem

ISTANBUL, 2020

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Mehmet Fatih GÜNDOĞAR, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended his thesis entitled "**Detection of Presentation Attacks for Face Recognition Systems**", on July 16, 2020 and has been found to be satisfactory by the jury members.

Jury Members

Prof. Dr. Çiğdem Eroğlu ERDEM (Advisor)
Marmara University .....

Prof. Dr. Songül VARLI (Jury Member)
Yıldız Technical University .....

Assist. Prof. Mehmet BARAN (Jury Member)
Marmara University.....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Mehmet Fatih GÜNDOĞAR be granted the degree of Master of Science in Department of Computer Engineering, Computer Engineering Program on 22.07.2020 (Resolution no: 2020/17-07).

Director of the Institute
Prof. Dr. Bülent EKİCİ

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Çiğdem Eroğlu Erdem for her continuous support during my M.Sc. studies and research; for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me both during the time of research and writing of this master thesis.

I also would like to thank Prof. Songül Varlı, Assoc. Prof. Borahan Tümer and Asst. Prof. Mehmet Baran for participating in my thesis committee and for their insightful comments.

I would like to thank my dear family: my wife Songül Gündoğar and my son Yiğit Kerem Gündoğar for supporting me spiritually throughout my life.

July, 2020 **Mehmet Fatih GÜNDOĞAR**

TABLE OF CONTENTS

ÖZET

YÜZ TANIMA SİSTEMLERİNDE YANILTMA ATAKLARININ TESPİTİ

Yüz tanıma ve doğrulama sistemlerini başka bir kişiye ait görüntü veya video göstererek ya da üç boyutlu maske kullanarak aşmaya çalışan yanıltma atakları önemli tehditlerdir. Yanıltma ataklarının tespiti yüz tanıma tabanlı biyometrik kimlik doğrulama sistemleri için zorlu bir görevdir. Yanıltma ataklarının tespiti hareket tabanlı ve doku tabanlı özelliklerin yardımıyla yapılabileceği gibi canlılık ipuçlarını elde etmeye yarayan temassız fotopletismografi özellikleri kullanılarak da yapılabilmektedir. Temassız fotopletismografi yöntemi ek bir donanıma ihtiyaç duyulmadan sadece kamera ile kayıt edilen videodan kalp atım hızı, solunum oranı ve kalp hızı değişkenliği gibi canlılık özellikleri içeren fizyolojik verilerin çıkarımını sağlayabilmektedir. Bu fizyolojik veriler bir makine öğrenimi sınıflayıcısının sahte ve gerçek yüzleri ayırt edebilmesi amacıyla eğitimi için özellikler olarak kullanılabilir. Bu tezde ilk olarak temassız fotopletismografi verileriyle yanıltma ataklarının tespitine dair uygulanabilecek yöntemler hakkında literatür özeti sunulmuştur. Ayrıca literatürdeki farklı temassız fotopletismografi yöntemlerinden elde edilen özelliklerle yanıltma atağı tespiti deneyleri yapılmış ve 3DMAD, Replay-Attack, Replay-Mobile ve MSU-MFSD veri setleri üzerinde performansları karşılaştırılmıştır. Fotopletismografi tabanlı özelliklerle elde edilen yanıltma atağı tespitine dair sınıflandırma sonuçlarını iyileştirmek için sınıflandırma sürecine hareket tabanlı (kafa pozu, göz bakış açısı ve göz kırpma) ve doku tabanlı özellikler de dahil edilmiştir. Tez kapsamında hem fotopletismografi özelliklerini hem de hareket tabanlı ve doku tabanlı özellikleri kullanarak yanıltma atağı tespiti yapan yeni bir sıralı füzyon yöntemi sunulmuştur. Tez kapsamında önerilen yöntemin ve literatürdeki diğer çalışmaların yarı toplam hata oranı metrikleri üzerinden veri setleri üzerindeki deneysel sonuç karşılaştırmaları sunulmuştur. Deney sonuçları önerilen yöntemin fotopletismografi tabanlı diğer yöntemlerden daha iyi sonuç verdiğini göstermektedir.

July, 2020 **Mehmet Fatih GÜNDOĞAR**

ABSTRACT

DETECTION OF PRESENTATION ATTACKS FOR FACE RECOGNITION SYSTEMS

Spoofing (presentation) attacks are significant threats for face recognition and authentication systems, which try to deceive them by presenting an image or video of a different subject or using a three-dimensional mask. Presentation attack detection is a challenging task for biometric identity verification systems based on face recognition. As the spoofing detection could be done using motion-based, and texture-based features, it could also be done utilizing remote (non-contact) photoplethysmography features that could help to obtain vitality clues. Remote photoplethysmography is useful for liveness detection from a facial video by estimating physiological signals such as the heart rate, respiratory rate, and heart rate variability that indicate liveness features using only a camera recording without any additional hardware. These physiological signals could be used as features to train a machine learning classifier to differentiate fake faces from genuine faces. In this thesis, we first provide a literature survey of presentation attack detection methods which use non-contact photoplethysmography. Then, we perform presentation attack detection experiments with the features obtained from different non-contact photoplethysmography methods in the literature and compare their performances on 3DMAD, Replay-Attack, Replay-Mobile, and MSU-MFSD datasets. To improve the presentation attack detection results obtained with photoplethysmography-based features, we introduce additional motion-based (head pose, eye gaze, and blink) and texture-based features to the classification process. We also present a novel cascaded fusion system that utilizes an ensemble of classifiers using non-contact photoplethysmography, motion, and texture-based features. We compare the data set evaluation results of the proposed method with the other studies in the literature using the half total error rate metric. Experimental results show that the proposed method outperforms several other PAD methods in the literature, which utilize non-contact photoplethysmography.

July, 2020 **Mehmet Fatih GÜNDOĞAR**

ABBREVIATIONS

2SR : Spatial Subspace Rotation

ACER : Average Classification Error Rate

APCER : Attack Presentation Classification Error Rate

AUC : Area Under Curve

CHROM : Chrominance

CNN : Convolutional Neural Network

CSP : Common Spatial Patterns

DRMF : Discriminative Response Map Fitting

DSIFT : Densely Sampled Scale-Invariant Feature Transform

EER : Equal Error Rate

FAR : False Acceptance Rate

FIR : Finite Impulse Response

FRR : False Rejection Rate

GAN : Generative Adversarial Network

HoG : Histogram of Gradients

HR : Heart Rate

HTER : Half Total Error Rate

IR : Infra-Red

LBP : Local Binary Pattern

LTSS : Long-Term Statistical Spectral

MLBP : Multi-Scale Local Binary Pattern

MMD : Maximum Mean Discrepancy

NIR : Near Infra-Red

NPCER : Normal Presentation Classification Error Rate

PAD : Presentation Attack Detection

PCA : Principal Component Analysis

PPG : Photoplethysmography

RAM : Region Attention Module

RBF : Radial Basis Function

RGB : Red, Green, Blue

RMSE : Root Mean Square Error

ROC : Receiver Operating Characteristics

ROI : Region of Interest

rPPG : Remote Photoplethysmography

SASM : Spatial Anti-Spoofing Module

SIFT : Scale-Invariant Feature Transform

SPMT : Spatial Pyramid Coding Micro-Texture

SSD : Single Shot Detector

STASN : Spatio-Temporal Anti-Spoof Network

SVM : Support Vector Machine

TASM : Temporal Anti-Spoofing Module

TFBD : Template Face Matched Binocular Depth

VGG : Visual Geometry Group

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Problem Definition and Motivation

Presentation attack detection (PAD) for face recognition systems gains importance day by day due to various face recognition applications (such as unlocking the smartphone with the use of the face, safe access controls). Since the hyperrealistic three-dimensional face masks, high-resolution photos, and video recordings are easily accessible (Fig. 1.1); it is challenging to apply a generalizable solution to distinguish a fake from a real face. For the detection of presentation attacks, the subject's face in a video needs to be classified as genuine or fake. In the literature, research for presentation attack detection with the use of features containing vitality data such as heart rate and heart rate variation obtained by the rPPG methods has gained importance in recent years.

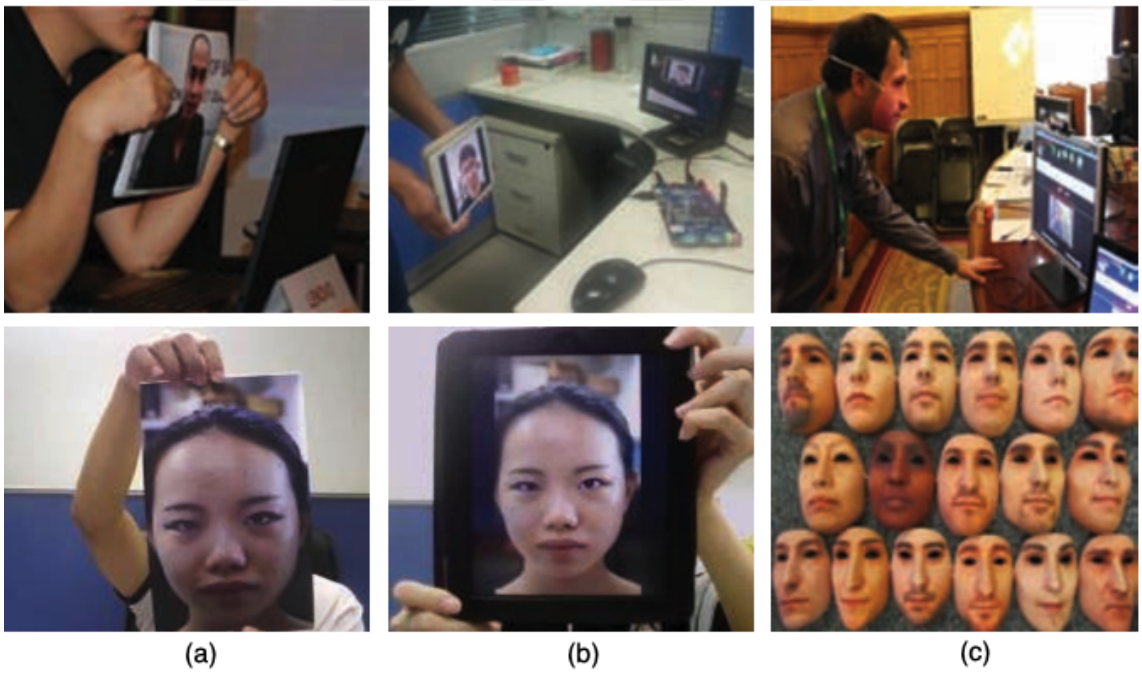


Figure 1.1: Several types of spoofing mediums [1]. (a)Photo attack (b)Video replay attack (c)3D mask attack

In general, rPPG methods require a detection of rectangular region of interest (ROI) on the face with a face detection algorithm in every frame of a video [2]. After the detection of ROI, the skin pixels are detected. Subsequently, a time-dependent signal generation takes place by calculating an average value from the components in the

Red-Green-Blue (RGB) color space of all pixels with skin color, or only through the green channel component. In the final step, the rPPG algorithm generates the power spectrum of the signal and estimates the heart rate, considering the highest frequency in the power spectrum.

The power spectrum estimated by an rPPG method shows some structural differences due to the absence of a live and real face image in the photograph or mask attack types. Based on these differences, some PAD methods in the literature uses features extracted by an rPPG process for the classification of the subjects included in the videos of a data set as genuine or fake. The comprehensive literature survey study of Ramachandra et al. [3] examines hardware and software-based PAD methods in detail. However, this work does not include rPPG based and deep learning-based PAD methods.

1.2. Contributions of the Thesis

The contributions of this thesis are as follows:

- A literature survey of the most recent rPPG-based PAD methods is given.
- A literature survey of recent deep learning-based PAD methods is given.
- Experimental comparison of several open-source rPPG signal extraction methods in the literature is done for heart rate estimation and their performances for PAD are compared.
- A new cascaded PAD method with the use of rPPG, texture-based, and motion-based features is proposed.
- The proposed PAD method is experimentally evaluated on the 3DMAD, Replay-Attack, Replay-Mobile, and MSU-MFSD datasets.

1.3. Outline of the Thesis

The rest of the thesis is structured as follows. Section 2 presents a comprehensive literature survey on PAD methods, including deep learning-based and rPPG based methods. Also, this section provides information about data sets for PAD experiments and related evaluation metrics for performance evaluation of PAD algorithms. Section 3 covers three different rPPG methods used for biometric feature extraction. Section 4 presents experimental results that uses extracted rPPG features for PAD. In Section 5, a

novel cascaded fusion system that introduces additional motion-based and texture-based feature extraction methods and utilizes an ensemble of SVM classifiers is proposed for the improvement of PAD classification results. The success ratio and performance evaluation of PAD experiments of the new proposed system is discussed in Section 6. Finally, Section 7 concludes the thesis and presents future research areas on the PAD problem.

2. LITERATURE SURVEY OF PRESENTATION ATTACK DETECTION

This section summarizes the methods suggested in the literature for PAD under three main groups as deep learning-based methods, rPPG based methods, and other methods. Besides, data sets and evaluation criteria used for PAD are also mentioned.

2.1. Deep Learning-Based Presentation Attack Detection Methods

It is observable that a hybrid combination of movement and texture-based features with deep learning methods produces promising results. In a real face region, the eyes, mouth, and cheek areas contain more facial movements as compared to the face region of a mask image. Shao et al. [4] try to distinguish between real and false faces using deep patterns of the face. They build a 5-layer convolutional neural network (CNN) and extract an average optical flow [5] from 256 channels separately to detect subtle facial movements in each layer. They calculate the channel-specific spatial discrimination value for each channel, keep the weight ratio of the informative channels higher considering prediction loss values, and treat the non-informative channels as noise, and they are lowly weighted. Ultimately, that method applies a joint learning model over an optimization problem that will minimize a function that calculates the channel-shared spatial discriminability value and realizes genuine and fake face classification flows.

Li et al. [6] create a 3D CNN by replicating spatial and temporal information with a specially designed data augmentation method. Besides, they make a generalization arrangement by trying to minimize the maximum mean discrepancy (MMD) distance value on different activity areas. 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 right, left, up, and below parts of the face ROI as separate frames and includes them as additional training data into the CNN. As a different data augmentation method, they apply contrast degree correction flow to the video images and obtain additional images from it to include them in CNN's training process.

Song et al. [7] develop the original 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) deep learning structure to detect context cues and configure an end-to-end PAD system. While it is possible to use the SSD structure directly in any data set with the SPMT feature, the only usage area of TFBD features is the data set prepared in that study due to the extraction of that feature requires two separate camera images taken from different angles at the same time. Therefore, this feature does not provide generalizable use.

Mohammadi et al. [8] mention the domain-shift problem of PAD methods and propose a novel method to improve cross-dataset performance on the PAD process by modeling the nuisance factors like lighting conditions, different camera devices with using a face recognition dataset containing millions of bona fide images. Li et al. [9] approach the generalization problem of PAD as a supervised anomaly detection problem that utilizes a hypersphere loss function in their proposed method, and they provide experimental results on EER, ACER, and AUC metrics.

Recently, a PAD challenge was organized to introduce the multimodal CASIA-SURF data set [10], 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. [11] also present a solution to collect a large amount of live face data and synthesize spoofing data with reflection artifacts. They propose the STASN deep network and report significant performance increase in inter and cross-testing experiments.

Table 2.1, 2.2, and 2.3 present an overview of deep learning based PAD methods.

2.2. Remote Photoplethysmography Based Presentation Attack Detection Methods

Li et al. [12] propose a cascaded PAD system that uses a combined heart rate and local binary pattern (LBP) features. The expectation is that rPPG based methods will give better results in photo and 3D mask attacks. The same expectation does not apply to video replay attacks since these types of attack videos still contain a PPG signal. In addition to the rPPG features, they adapt the LBP method to handle video replay attacks successfully.

Table 2.1: Brief Overview of Deep Learning-Based PAD Methods (I)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Shao et al. [4] (2019)	VGG net pretrained on ImageNet dataset - five convolutional layers Average optical flow calculation, Deep dynamic texture joint learning	3DMAD (HTER: 1.76%), SUP (HTER: 13.44%), SMAD (HTER: 11.7%)	Proposed methods gives acceptable results for mask attacks.
Li et al. [6] (2018)	3D CNN with Torch library Spatial data augmentation, Gamma correction based data augmentation Generalization loss as regularization term for increasing generalization capability	Replay-Attack (HTER: 1.2%), CASIA-FASD (EER: 1.4%), MSU (EER: 0.0%), Rose-Youtu (EER: 7.0%)	Cross-dataset setting of proposed method is measured as HTER: 28.7% which is far from intra-dataset performance.
Song et al. [7] (2018)	VGG16-BaseNet, Template Face Matched Binocular Depth (TFBD), Multi-scale Local Binary Pattern (MLBP) and Spatial Pyramid Coding Micro-Texture (SPMT) features	Replay-Attack (HTER: 0.06%), CASIA-FASD (EER: 0.04) Self collected dataset (EER: 3.53%)	It is not possible to extract TFBD features on most of the PAD datasets due to the requirement of recording same subject from two different angles. MLBP and SPMT features could improve PAD results

Table 2.2: Brief Overview of Deep Learning-Based PAD Methods (II)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Mohammedi et al. [8] (2020)	Train a CNN with reconstructed images generated by an auto-encoder resembling low-pass filtered versions of the original images	*Results are not reported according to a standard metric. Only AUC graphic of log-scale ROC is reported in a graphic.*	Modeling nuisance factors with an auto-encoder by utilizing bona fide face images from large public face recognition datasets could improve cross-dataset performance
Li et al. Li et al. [9] (2020)	ResNet18 based CNN hypersphere loss function for deep anomaly detection Training CNN with face images in both RGB and HSV colorspace	SiW (EER: 15.2%) Rose-Youtu (AUC: 91.3%) CASIA-FASD, Replay-Attack, MSU-MFSD (AUC: 96.2%)	Proposed loss function could contribute to the performance of a PAD system positively

Table 2.3: Brief Overview of Deep Learning-Based PAD Methods (III)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Yang et al. [11] (2019)	Spatio-Temporal Anti-Spoof Network (STASN): Temporal Anti-Spoofing Module (TASM) - *(Conv-LSTM structure)* Region Attention Module (RAM) - (applies depth wise and conventional convolution) Spatial Anti-Spoofing Module (SASM) - (multibranch network)	Oulu-NPU (ACER: 1.9%) SiW (ACER: 0.3%)	Training of a PAD system with a large amunt of data is more practical for real-world applications
Chalearn Challenge [10] (2019)	Different CNN-based methods of 13 teams: Resnet-18, Resnet-34, Resnet-50, SEresnext, Fishnet, MobileNetv2, Resnext, ShuffleNet-V2, Vgg16, LightCNN, Densenet	CASIA-SURF (ACER of the best result: 0.08%)	Some of the misclassified videos by all teams in the dataset is easily distinguishable by human eye. This shows that use of physiological signals is essential for a PAD system

In the preprocessing stage, they use the Viola-Jones [13] algorithm to identify the face ROI and identify 66 facial points by discriminative response map fitting (DRMF) [14] method. For the temporal filtering and power spectrum distribution stages, they apply the detrending filter, moving average filter, the Hamming window-based finite impulse response (FIR) bandpass filter and the fast Fourier transform. They give the resulting multidimensional feature vector to a support vector machine (SVM) classifier as input for the two-class classification task. Hernandez-Ortega et al. [15] applied a PAD approach similar to [12] in their study and collected a new heart rate (HR) dataset that included near-infrared (NIR) spectral samples in addition to components in the RGB color space. As a result of experimental studies on the newly collected HR data set, it is observable that the NIR spectrum is more resistant to light variations than the RGB spectrum. Another contribution is that they evaluate equal error rate (EER) metrics in different video length schemes in both RGB and NIR spectrum bands.

Heusch et al. [16] attempt to distinguish presentation attacks from genuine access attempts using the long-term statistical spectral (LTSS) feature of the PPG signal. They do presentation attack or genuine access classification feeding LTSS feature vectors into an SVM classifier. During the experimental study, they do performance evaluations and comparisons according to the results obtained through the LTSS features gathered with three different PPG signal extraction algorithms (Li's CVPR14 [17], CHROM [18] and 2SR [19]).

Nowara et al. [20] extract the power spectrums separately from the pixels in three different face regions as the forehead, left cheek, and right cheek and two different regions of the background outside the face area and use them as a feature vector for the classification process. They train SVM and random forest classifiers and use them as classifiers. As a result of the experimental studies, it is observable that the spectral properties added from the background regions outside the face region affect the performance of the classification process regarding the accuracy metric positively.

Ciftci et al. [21] propose a solution for the detection of deep fakes produced by generative adversarial networks (GAN). They train a CNN with the input of Green channel PPG and CHROM PPG inputs obtained from three different facial regions, six PPG signals in total. They introduce a novel PPG map feature and feed into a CNN

with the video frames, cropped face frames, and other features. They use Matlab is used for signal processing, Open Face [22] library for face detection, libSVM [23] for radial basis function (RBF) kernel-based SVM classification processes. They also test with the effectiveness of different segment durations for the probabilistic video classification process. For the dimensionality reduction process, they practice principal component analysis (PCA) and common spatial patterns (CSP).

Table 2.4 and 2.5 presents an overview of rPPG-based methods.

2.3. Other Presentation Attack Detection Methods

Although rPPG based methods achieve successful results in photo and mask attacks, they are insufficient for video replay attacks alone. Patel et al. [24] 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 rPPG signal. They calculate some histograms with these methods and give them as feature inputs to an SVM classifier to obtain classification results. This approach, which is promising for video replay attacks, can be developed for the detection of photo and mask attacks by including different additional features such as rPPG signals.

Chingovska et al. [25] propose the use of client-specific information for the anti-spoofing process. This approach requires enrollment samples for each possible subject to perform well, and the proposed solution directly rejects a subject if he/she is not enrolled in the system, even for a real subject. The discriminative client-specific method in that study requires training a separate SVM classifier for each client using LBP, LBP-TOP, and motion features. Since the solution depends on client data, it is not possible to immediately use it at enrollment time.

Hao et al. [26] introduce a solution based on twin CNNs. They train the Siamese neural network with the paired input of enrolled subjects' images. The suggested solution first applies a face recognition flow, then gets the real image of the identified subject and uses the Siamese neural network for genuine or fake decisions.

Table 2.6 and 2.7 presents an overview of other PAD methods.

Table 2.4: Brief Overview of rPPG-Based PAD Methods (I)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Li et al. [12] (2016)	6-dimensional features vector obtained from the pulse signal different local binary pattern features (*multi-scale, block-wise, color, grayscale)*	3DMAD - pulse method (HTER: 7.94%) 3DMAD - LBP method (HTER: 0%) REAL-F - pulse method (HTER: 4.29%) REAL-F - LBP method (HTER: 25.92%) MSU-MFSD EER(7.50%)	LBP method fails on REAL-F dataset since it does not provide a generalizable solution. Combination of LBP features with pulse provides better results
Hernandez-Ortega et al. [15] (2018)	6-dimensional features vector obtained from pulse signal for RGB videos 2-dimensional features vector obtained from pulse signal for NIR videos	3DMAD (EER: 25%) Self collected HR dataset (EER: 0%)	It is possible to extract PPG signals on NIR spectrum band and use it in videos for PAD process. However proposed solution does not provide results close to state-of-the-art for short duration videos as confirmed from the results of 3DMAD dataset

Table 2.5: Brief Overview of rPPG-Based PAD Methods (II)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Heusch et al. [16] (2018)	Implementation of CHROM, SSR, Li CVPR algorithms Utilization of long-term spectral~statistics features	3DMAD (HTER: 13.0%) Replay-Attack (HTER: 5.9) Replay-Mobile (HTER: 32.5) MSU-MFSD (HTER:43.3)	When motions and illumination variations are present, the PPG process is impacted and this case reduces the performance of the PAD process
Nowara et al. [20] (2017)	Extraction of PPG signals from the forehead and the cheeks of the face, and from the background regions of the subject	3DMAD (HTER: 43.0%) Replay-Attack (HTER: 25.5) Replay-Mobile (HTER: 35.9) MSU-MFSD (HTER:31.7)	Unexpected intensity changes in the facial region reduces the quality of PPG signal. Weak PPG signals does not exhibit enough temporal clues for PAD process
Ciftci et al. [21] (2019)	Extraction of PPG signals from the cheeks and mid-region of the face Implementation of novel PPG map features for CNN input	Face Forensics (ACC: 99.39%) Self collected Deep Fakes dataset (ACC: 91.07%)	PPG map features used in detection of deep fakes produced by GANs could be adopted for a PAD process

Table 2.6: Brief Overview of Other PAD Methods (I)

Author and Year	**Used Methods and Features**	**Dataset Performance**	**Main Results**
Patel et al. [24] (2015)	utilization of multi-scale LBP (MLBP) and scale-invariant feature transform (SIFT) for capturing moire patterns in video replay attacks	Replay-Attack (HTER: 3.3%) CASIA-FASD (HTER: 0%) Self collected RAFS dataset (HTER: 11.3%)	DSIFT and MLBP features are helpful in the detection of moire patterns usually present in video replay attacks
Chingovska et al. [25] (2015)	PAD based on client id information (requires enrollment data during system initialization) training of a separate SVM classifier for each client with use of LBP, LBP-TOP, and motion features	Replay-Attack (HTER: 3.95%)	Using client id information (which is dependent on enrollment data) in the proposed PAD method improves the results on Replay-Attack dataset relatively about %50

Table 2.7: Brief Overview of Other PAD Methods (II)

Author and Year	Used Methods and Features	Dataset Performance	Main Results
Hao et al. [26] (2019)	Face liveness detection based on client ID information (requires enrollment data) training of a Siamese network with positive (two real image) and negative (one real, one fake image) pairs	NUAA (HTER: 1.96%) Replay-Attack (HTER: 0.86)	Training a CNN with client ID information significantly improves performance. However, this solution can not be applied without enrollment data

2.4. Datasets

This section covers the structure of the datasets commonly used in the literature for PAD experiments. Table 2.8 and 2.9 provides an overview of the PAD datasets.

2.4.1. Replay-Attack

Replay-Attack dataset contains a total of 1300 face videos collected from 50 subjects [27]. The videos are recorded using a MacBook camera at a resolution of 320 x 480 pixels under different lighting conditions. The frame rate of the videos is 25 Hz. Some of the videos have a 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 dataset contains training, testing, and development fold separation for an experimental setup. Besides, an enrollment data containing 100 videos is provided for the experiments using client id information for the PAD process.

2.4.2. Replay-Mobile

Replay-Mobile dataset [28] contains 1030 videos from 40 subjects. Videos are recorded in 1280 x 720 resolution under different lighting conditions. Recordings are done with an iPad Mini2 tablet device and LG-G4 smartphone device. The dataset contains video recordings of high quality and resolution to reveal more compelling attack scenarios. Like the Replay-Attack dataset folding structure, this dataset also contains training, testing, development, and enrollment data folds.

2.4.3. MSU-MFSD

MSU-MFSD [29] data set contains 280 video recordings collected from 35 subjects of different ethnic origins under different lighting conditions. Videos are recorded with a MacBook Air built-in camera with 640x480 resolution and with the frontal camera of a Google Nexus 5 smartphone with 720x480 resolution. The frame rate of the videos in this dataset is about 30 Hz. As different from the Replay-Attack and Replay-Mobile datasets, only training and testing subjects are provided. There is no additional folding structure for development or validation purposes. Fig. 2.1 illustrates example images data of genuine and fake subjects published in the MSU-MFSD dataset.

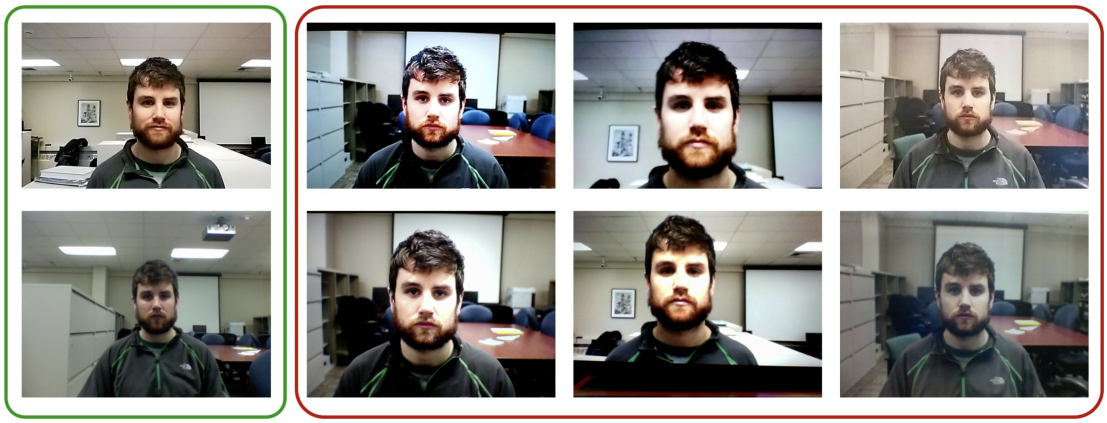


Figure 2.1: Example images of genuine and spoof faces of one of the subjects in the MSU-MFSD database captured using Google Nexus 5 smart phone camera (top row) and MacBook Air 13" laptop camera (bottom row).(a) Genuine faces; (b) Spoof faces generated by iPad for video replay attack; (c) Spoof faces generated by iPhone for video replay attack; (d) Spoof faces generated for printed photo attack [29].

2.4.4. 3DMAD

3DMAD [30] data set contains 255 video images 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 x 480 and recorded with the Kinect device for 10 seconds duration. There is no separate folding structure for the experimental setups. Most of the experiments on this dataset use a 17-fold leave-one-subject-out cross-validation protocol. There are 170 genuine videos and 85 mask attack videos in the dataset.

2.4.5. CASIA-FASD

CASIA FASD [31] dataset consists 600 videos of 50 different subjects from Asian ethnic regions. 150 videos of the dataset contain genuine subject videos; the rest of the 450 videos are prepared for attack scenarios. It contains three types of face spoofing attacks: a warped photo attack, a cut photo attack (with the removal of eye regions in printed photo), and a video attack. For each subject, the dataset contains three real and nine fake videos. Videos are recorded in three different resolutions. Separation of the dataset is done as a training and a testing set. There is no development or validation set is provided.

2.4.6. OULU-NPU

Oulu-NPU [32] 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 dataset is partitioned as training, development, and test folds. Fig. 2.2, contains a sample of genuine and attack subject data from the Oulu-NPU dataset.

2.4.7. SiW

SiW [33] dataset contains 4478 videos from 165 subjects at 1920x1080 resolution. The video frame rate is about 30 Hz, and each video duration is 15 seconds. Each subject has 8 real access and 20 fake videos. Distance, pose, illumination, and facial expression are the main diversification criteria of the subject videos. The dataset is partitioned as training and test folds.

Figure 2.2: Sample images of real and attack videos captured with Samsung Galaxy S6 edge phone [32].

2.4.8. MSU-USSA

MSU-USSA [34] dataset contains photo attacks and replay attacks of 1000 subjects. A laptop, tablet, and smartphone are used to capture replay attacks. The suggested experimental protocol is five-fold subject exclusive cross-validation according to a given fivefold subject list file. Each fold contains one genuine image for a subject and four spoof images for that subject. Training is done with the other 4 folds, and testing is done on the reserved single fold.

2.4.9. CASIA-SURF

CASIA-SURF [35] data set contains 21000 videos of 1000 subjects recorded in three modalities: RGB, depth, and infrared. Videos are recorded with the Intel RealSense SR300 camera. RGB videos are recorded in 1280x720 resolution, and depth and IR videos are recorded in 640x480 resolution. The dataset contains 3000 genuine videos of 1000 subjects, 1000 videos for each modality. There are also 6000 attack videos for each modality. The dataset is partitioned as train, test, and validation sets. The dataset contains six different types of photo attacks, including cuts from eye, nose, and mouth regions.

2.4.10. FaceBook DeepFake Detection Challenge

FaceBook DeepFake detection challenge [36] data set contains videos generated by GANs and subjects present diversity in terms of age, skin tone, and gender attributes. This dataset is provided for a deepfake detection challenge with a public dataset

partitioned as a train and a test data group. A black-box dataset not shared with competition attendees is used for the determination of the winner.

2.5. Evaluation Metrics

In the literature, rPPG based PAD algorithms generally use half total error rate (HTER) (3.2) metric for performance evaluation, which is generally calculated over a threshold value (τ) that minimizes the equal error rate (EER) metric. The EER corresponds to the point that the false acceptance rate (FAR) equals to the false rejection rate (FRR). HTER metric is calculated accoring to the FAR and FRR metrics with the corresponding EER threshold value (τ).

$$FAR = \frac{\text{\# of false acceptance}}{\text{\# of identification attempts}} \tag{2.1}$$

$$FRR = \frac{\text{\# of false rejection}}{\text{\# of identification attempts}} \tag{2.2}$$

$$EER = \text{the point in the ROC curve where } FAR = FRR \tag{2.3}$$

$$HTER(\tau) = \frac{FAR(\tau) + FRR(\tau)}{2} \tag{2.4}$$

Moreover, there are Attack Presentation Classification Error Rate (APCER), Normal Presentation Classification Error Rate (NPCER), and Average Classification Error Rate (ACER) metrics, which also have common usages in the literature:

$$APCER = \frac{\text{\# of false positives}}{(\text{\# of true negatives} + \text{\# of false positives})} \tag{2.5}$$

$$NPCER = \frac{\text{\# of false negatives}}{(\text{\# of false negatives} + \text{\# of true positives})} \tag{2.6}$$

Table 2.8: Brief Overview of Face-Antispoofing Datasets (I)

Dataset	**Resolution**	**Attack Types**	**# of Subjects**	**# of Genuine Videos**	**# of Attack Videos**	**Fold Structure**
Replay-Attack [27]	320 x 240	print photo replay video	50	300	1000	Training, Test Development, Enrollment
Replay-Mobile [28]	1280 x 720	print photo replay video	40	550	640	Training, Test Development, Enrollment
3DMAD [30]	640 x 480	3D mask	17	170	85	17-fold LOOCV
MSU-MFSD [29]	640 x 480 720 x 480	mobile photo mobile video	35	70	210	Training, Test
CASIA-FASD [31]	640 x 480 1920 x 1080	warp photo cut photo replay video	50	150	450	Training, Test
OULU-NPU [32]	six resolutions	print photo replay video	55	990	3960	Training, Test Development
SiW [33]	1920 x 1080	print photo replay video	165	1320	3158	Training, Test
MSU-USSA [34]	1280 x 960 3264 x 2448	print photo replay video	1000	1000	8000	Five-Fold Subject Exclusive CV

Table 2.9: Brief Overview of Face-Antispoofing Datasets (II)

Dataset	**Resolution**	**Attack Types**	**# of Subjects**	**# of Genuine Videos**	**# of Attack Videos**	**Fold Structure**
FaceBook Deep Fake [36]	-	deep fake	3500	-	10000	public dataset (train / test) blackbox dataset
CASIA-SURF [35]	1280 x 720 640 x 480	print photo cut (eye region) photo	1000	1000x3	6000x3	train, test, validation

$$ACER = \frac{APCER + NPCER}{2} \tag{2.7}$$

According to the ISO/IEC 30107-3 standard, APCER and NPCER metrics need to be calculated separately for each different attack type.

3. REMOTE PHOTOPLETHYSMOGRAPHY METHODS FOR HEART RATE ESTIMATION

Remote photoplethysmography (rPPG) is a non-contact technique for extracting biometrical information like pulse signal, heart rate, and respiratory rate from a video sequence recorded by a camera. In general, most of the rPPG algorithms focus on the extraction of pulse signals analyzing the changes in the spatial mean color of the facial region of a subject and considering the variation rate of this value in the temporal domain [37]. As illustrated in Fig. 3.1, rPPG algorithms execute the following steps to obtain an rPPG signal from a video sequence input:

- First, an rPPG algorithm detects and crops the facial ROI of a subject in video frames.
- Second, the algorithm determines the skin pixels in the facial ROI that is required for the extraction of spatial information and calculates the average value of the pixels in one or all of the RGB color space.
- Finally, the algorithm evaluates the temporal differences in spatial mean color information of each frame and extracts a pulse signal.

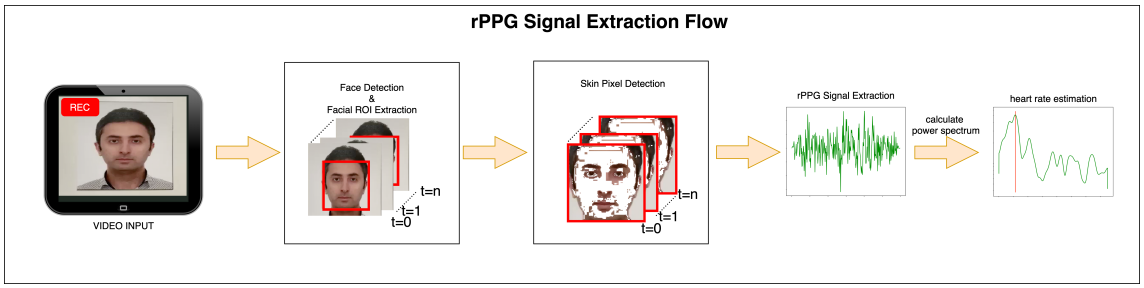


Figure 3.1: rPPG Based Heart Rate Estimation

Each algorithm follows different techniques to achieve results in the steps mentioned above. The performance of each algorithm differs from each other. rPPG algorithms are mostly sensitive to the changes in illumination and motion of the subject. Some algorithms implement additional preprocessing steps to improve their performance.

In this section, we briefly discuss the steps of three rPPG algorithms and their performances on PURE [38] and UBFC-RPPG [39] datasets.

3.1. Spatial Subspace Rotation (2SR) Method

2SR [19] 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 colorspace to obtain a skin mask for given threshold input, as illustrated in Fig. 3.2.

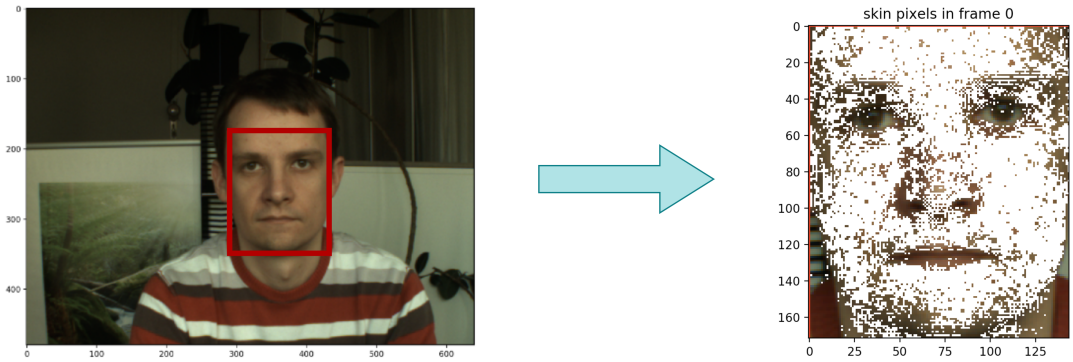


Figure 3.2: The calculation of skin pixel mask for the cropped face region.

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. Fig. 3.3 presents a sample plot of pulse signals obtained from 600 video frames of an input subject.

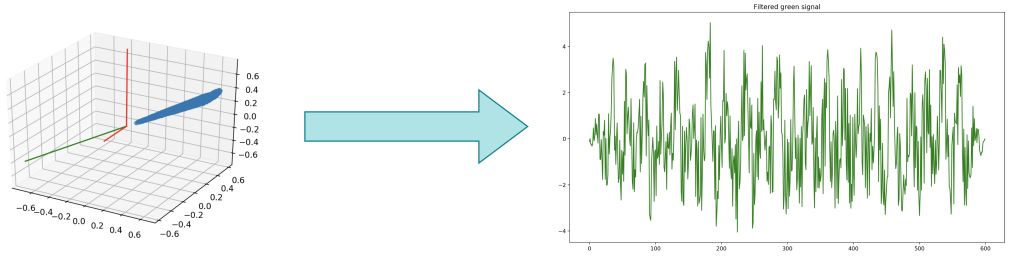


Figure 3.3: The eigenvector calculation and derivation of a pulse signal for 600 video frames.

3.2. Chrominance-Based rPPG (CHROM) Method

The CHROM [18] algorithm obtains the skin color signals from a subject's facial ROI at each frame of a video sequence. Like 2SR algorithm implementation, estimation of the Gaussian parameters of the skin color distribution takes place to gather a skin

mask according to the provided threshold input. The algorithm projects the mean skin color calculated from the RGB color space to the defined chrominance space 3.1. Fig. 3.4 illustrates the plot of X and Y values in the chrominance subspace.

$$\begin{aligned} X &= (3.0 * r) - (2.0 * g) \\ Y &= (1.5 * r) + g - (1.5 * b) \end{aligned} \tag{3.1}$$

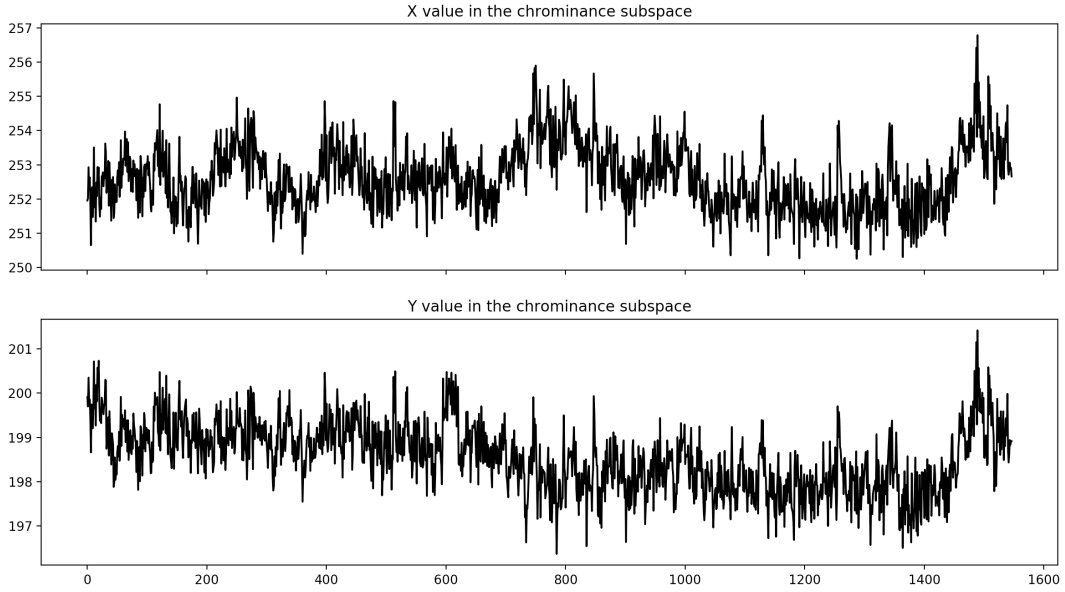


Figure 3.4: The X and Y values in the chrominance subspace.

Later on, the algorithm applies a bandpass filter in the chrominance space. Fig. 3.5 illustrates the band passed X and Y values. In the final step, the algorithm builds the rPPG signal.

3.3. Li's CVPR14 Method

As usual in other rPPG algorithms, Li's CVPR14 [40] algorithm extracts skin color signals from the facial ROI of each video frame. The algorithm uses the Dlib landmark detector as an implementation of [41] to detect 68 face landmarks. Then, it builds a mask on the lower part of the face region using selected face landmark points as it can be observable in Fig. 3.6.

The algorithm also extracts the background signals to reduce the impacts of variations in global illumination. Fig. 3.7 shows the overlaid masks for both the face

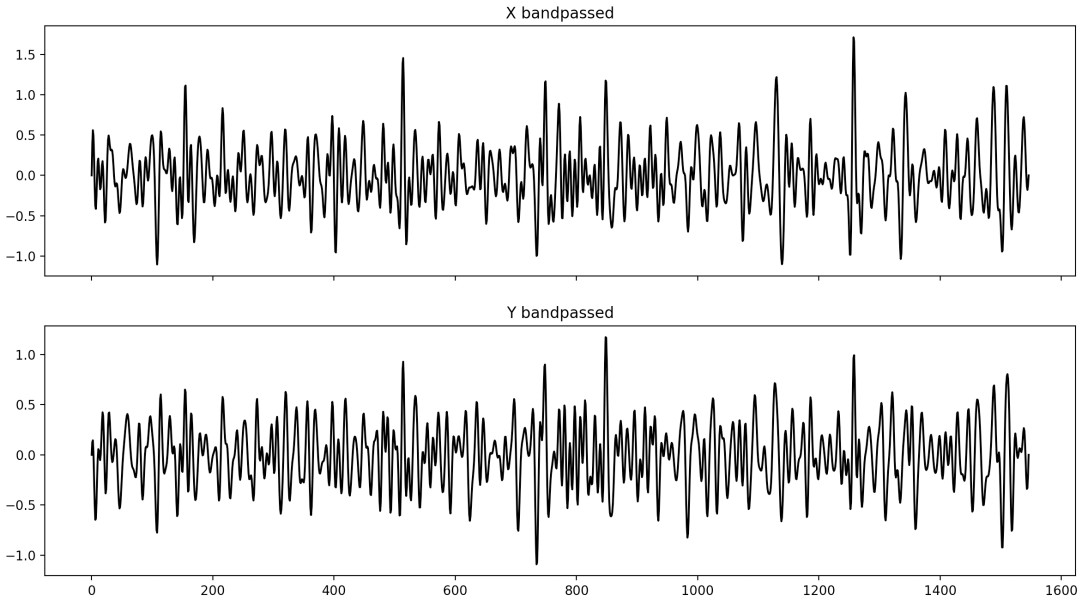


Figure 3.5: The band passed X and Y values in the chrominance subspace.

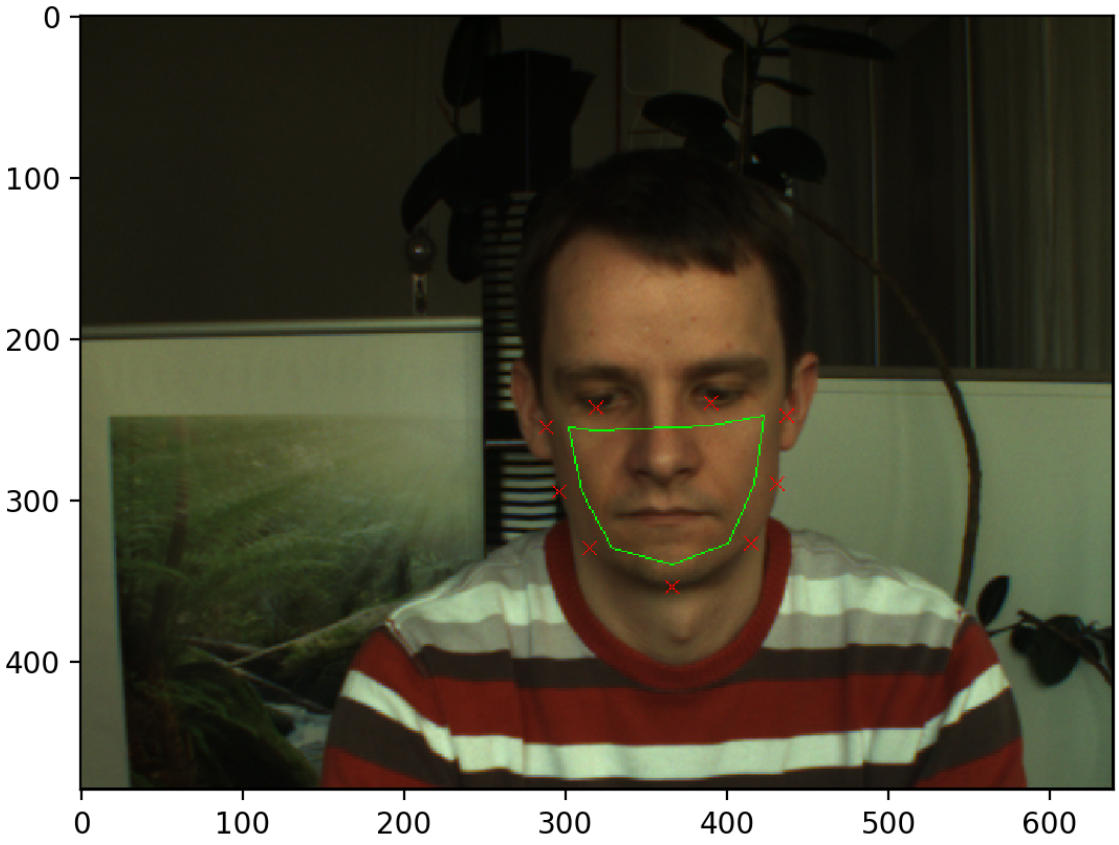
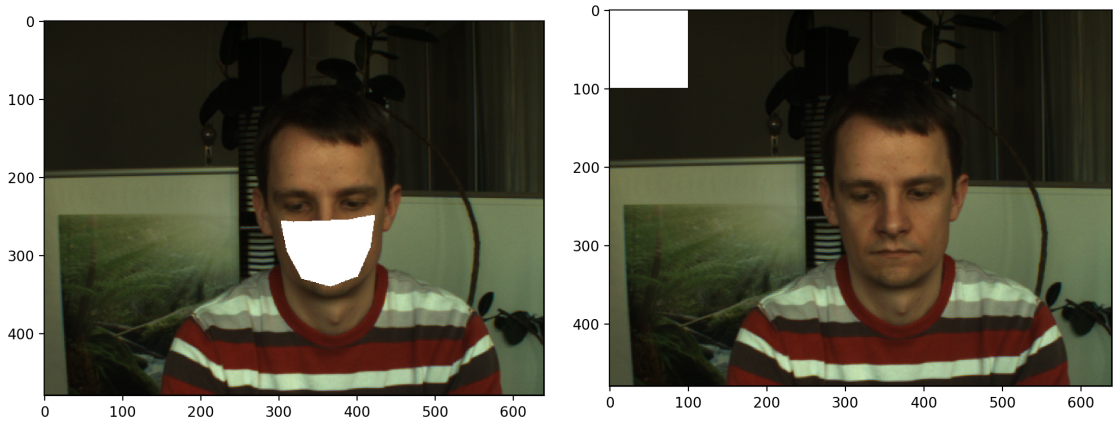


Figure 3.6: The selected face landmark points for building the face mask

and background region of a video frame.



(a) Face mask overlaid on the original frame **(b)** Background mask overlaid on the original frame

Figure 3.7: Signal extraction regions on an original video frame.

The algorithm then uses a normalized least mean square filtering method to filter the extracted background signal to complete the illumination rectification step. Fig. 3.8 provides the plots of this step.

After that, the algorithm removes the filtered background signal from the facial ROI signal. It divides the signal into segments and removes the segments having a high standard deviation to eliminate the motion impact. In the final step, the algorithm applies a detrending filter, moving average filter, and a Hamming window-based finite impulse response bandpass filter to build the rPPG signal. Fig. 3.9 presents each filtering step.

3.4. Power Spectral Density (PSD) and Heart Rate (HR) Estimation

After obtaining a pulse signal with one of the rPPG algorithms, it is possible to obtain the PSD of it with the use of Welch's method after converting the signal to the frequency domain. After finding the maximum frequency in the spectrum between the [0.7, 4] Hz values, the multiplication of this value with 60 provides an HR value between 42 beat-per-minute (bpm) to 240 bpm [40]. Fig. 3.10 presents a sample plot of a PSD and estimated heart rate value.

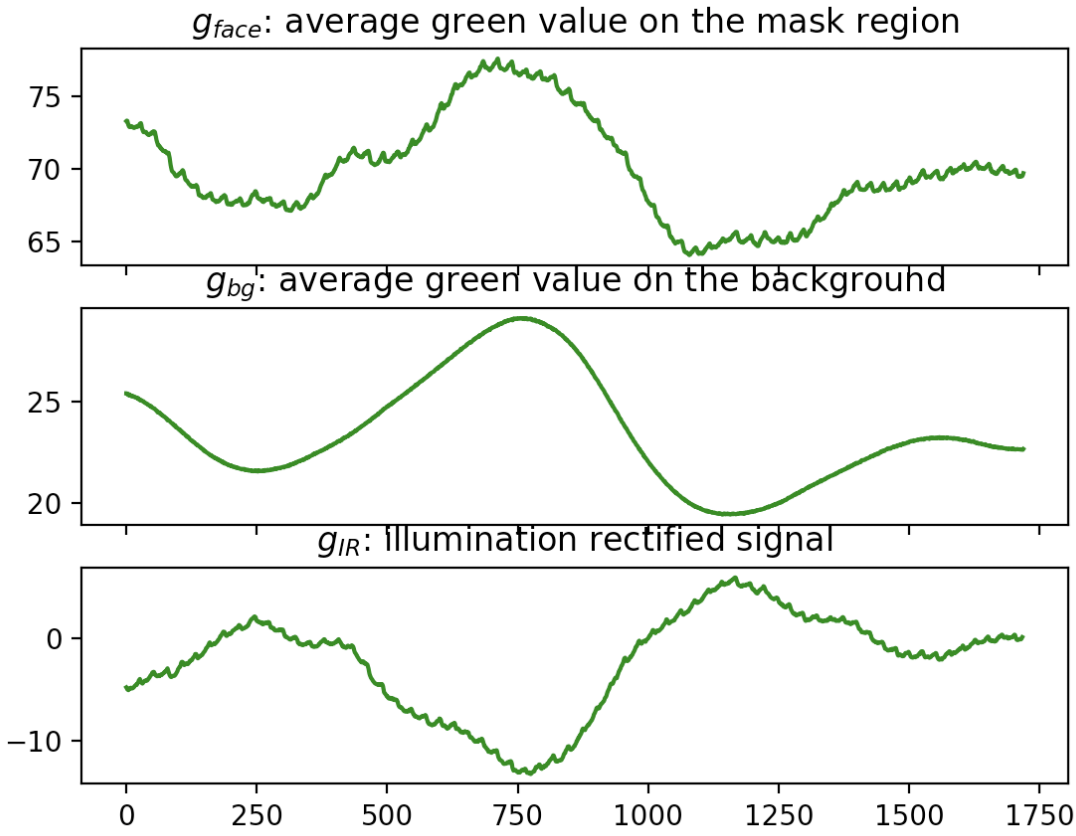


Figure 3.8: The illumination rectification step of Li's CVPR14 algorithm.

3.5. Experimental Comparison

For our experiments, we compared the above 3 method for heart rate estimation. We use the Bob toolbox [42] for the implementation of 2SR, CHROM, and Li's CVPR14 rPPG signal extraction algorithms and extraction of PSD and HR outputs. We test these three rPPG algorithms on PURE and UBFC datasets. For the performance evaluation, we use the root mean square error (RMSE) metric with considering computed HR values and ground truth (gt) HR values for each subject:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(HR(i) - HR_{gt}(i))^2} \tag{3.2}$$

Table 3.1 contains the corresponding output metric values. For our 2SR, CHROM, and Li's CVPR14 algorithm implementations and experiments, it is evident that some more optimization is plausible to enhance the accuracy of the estimated heart rate. Since

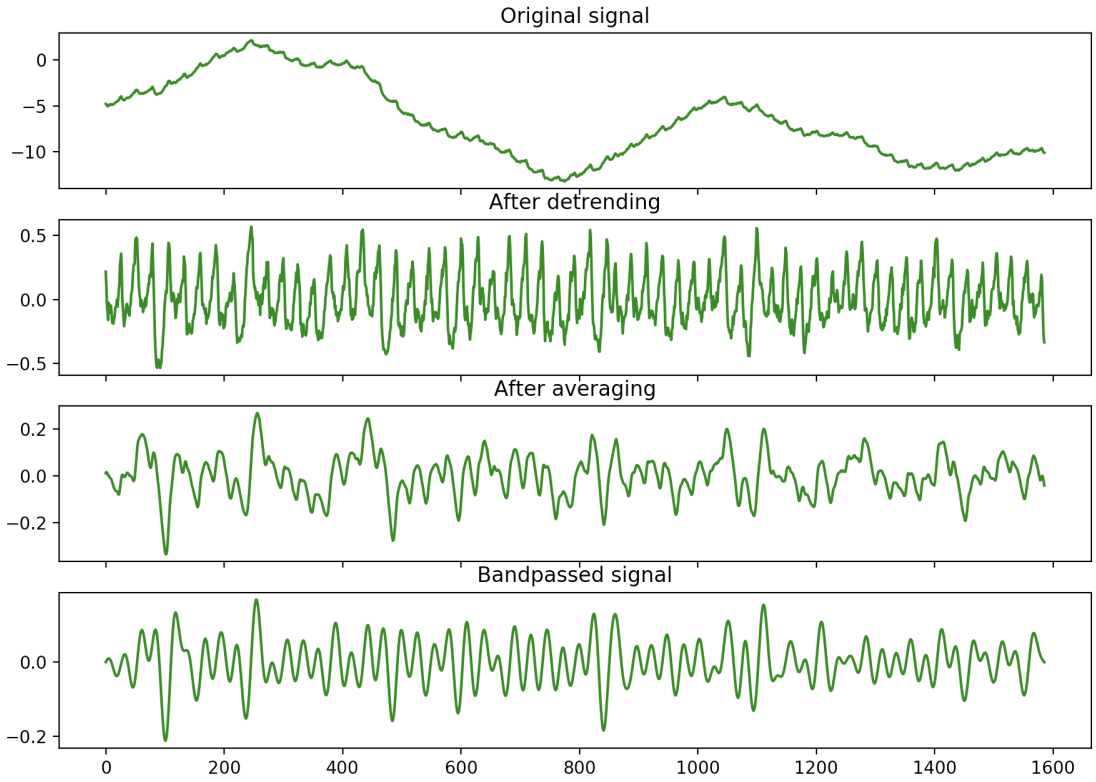


Figure 3.9: The detrending filter, moving average filter, and bandpass filter process before building the final rPPG signal.

our main aim in this study is not the correct estimation of heart rate, instead, we focused on the PAD experiment part to obtain clues for liveness detection.

Table 3.1: RMSE values [bpm] on PURE and UBFC-RPPG datasets for heart rate estimation.

	PURE	UBFC-RPPG
2SR [19]	3.40	25.52
CHROM [18]	20.90	8.44
Li's CVPR14 [40]	28.23	38.88

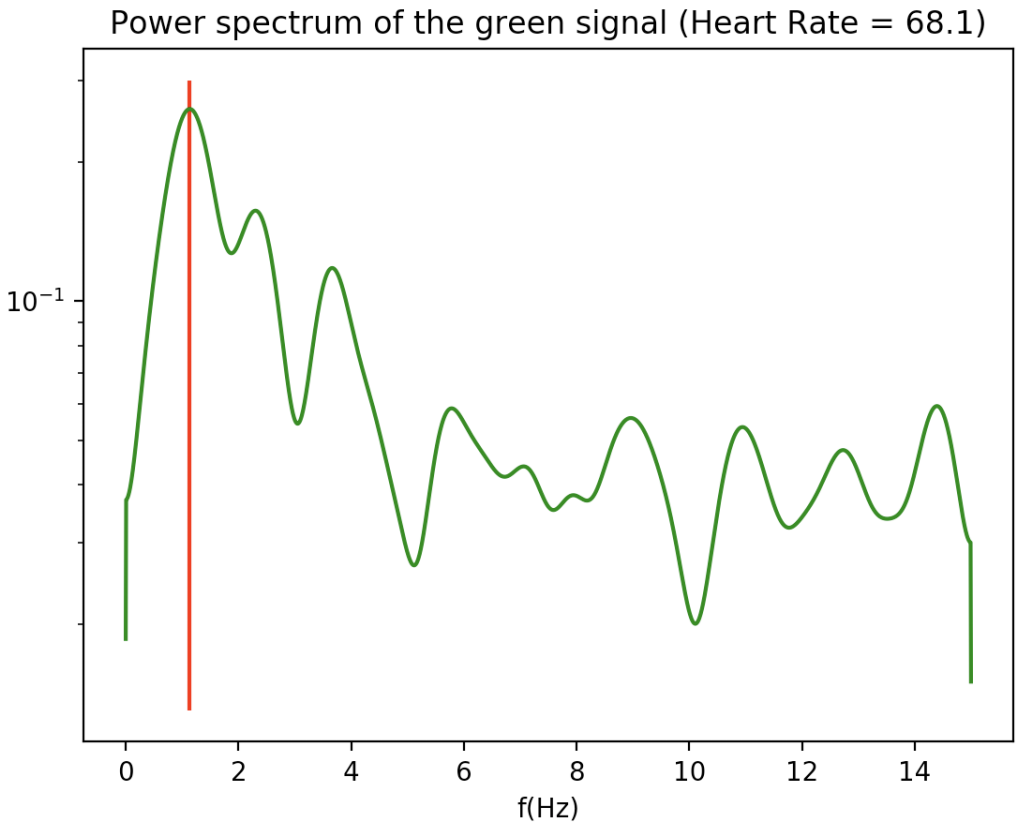


Figure 3.10: Sample power spectral density (PSD) diagram and estimated heart rate value.

4. REMOTE PHOTOPLETHYSMOGRAPHY BASED PRESENTATION ATTACK DETECTION METHODS

4.1. Presentation Attack Detection with 2D Photoplethysmography Features

4.1.1. Feature Extraction

As a pre-analysis study of our PPG based PAD experiments, we use 3DMAD, Replay-Attack, Replay-Mobile, and MSU-MFSD datasets. For each subject in each of these datasets, we extract the facial ROI employing the Viola-Jones algorithm. After estimating the rPPG signal using one of the three rPPG signal extraction algorithms described in the previous chapter above, the power spectrum density (PSD) of the rPPG signals $P(f)$ are estimated. Then, the peak of the PSD with the maximum power value is detected between [0.7Hz, 4Hz]. Finally, we represented the rPPG-based feature vector as follows:

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 find the following ratio [12]:

$$\Gamma = \frac{P(f_{max})}{\sum_{\forall f \in [0.7,4]} P(f)} \tag{4.1}$$

The two-dimensional rPPG feature vector is then formed as

$$R_1 = [P(f_{max}), \Gamma]. \tag{4.2}$$

4.1.2. Experimental Results

For the classification of 3DMAD dataset videos, we adopted a leave-one-subject-out-cross-validation (LOOCV) protocol similar to the application in [43]. Since there are 17 subjects in this dataset, we performed our experiments on the 3DMAD dataset using 17 folds. At each fold, we leave one subject for testing and use half of the remaining subjects for training and the other half for the development process. Fig. 4.1 presents the SVM classification processes' performance for each PPG algorithm and each fold and the average EER and HTER metrics values.

For Replay-Attack and Replay-Mobile datasets, we use the default provided train,

Table 4.1: HTER results of SVM classification with 2D PPG features extracted by 2SR, CHROM, and Li's CVPR14 algorithms.

	3DMAD	Replay-Attack	Replay-Mobile	MSU-MFSD
2SR	22.64%	45.62%	**29.26%**	**40.00%**
CHROM	**21.76%**	**33.75%**	45.42%	57.08%
Li's CVPR14	37.35%	42.47%	42.37	50.41%

development, and test partitions of the datasets. For the MSU-MFSD dataset, we use default train and test partitions. We use the support vector machine (SVM) algorithm to classify subjects as a genuine or fake utilizing the LIBSVM library. We use the RBF kernel during the training of SVM. With only the use of 2D PPG features extracted with different rPPG algorithms, we get HTER metrics presented in Table 4.1.

Fig. 4.2, 4.3, 4.4 presents the receiver operating characteristics (ROC) curve for the SVM classification results for Replay-Attack, Replay-Mobile, and MSU-MFSD datasets using 2D PPG features.

As it can be observed from the initial results, the CHROM algorithm gives better results for 3DMAD and Replay-Attack datasets compared to 2SR and Li's CVPR14 algorithms. On the other hand, the 2SR algorithm performs better on Replay-Mobile and MSU-MFSD datasets.

4.2. Presentation Attack Detection with Photoplethysmography Magnitude Features

4.2.1. Feature Extraction

In order to improve the PAD results, we try to use the magnitude of whole PSD as a feature vector to the SVM:

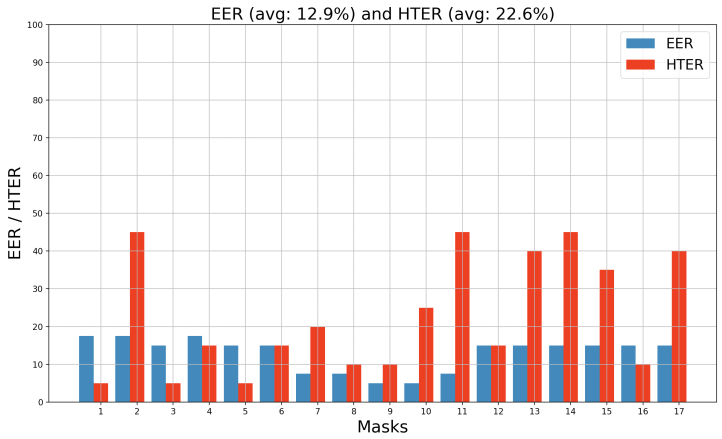
The magnitude power spectrum of the rPPG signal $P(f)$ as a feature vector after band-pass filtering ([0.7Hz, 4Hz]) is denoted as $R_2 = |P(f)|$, which gives us a vector of length approx. 500 (depending on the frame rate of the video).

Table 4.2: HTER results of SVM classification with the magnitude of PSD as feature vector.

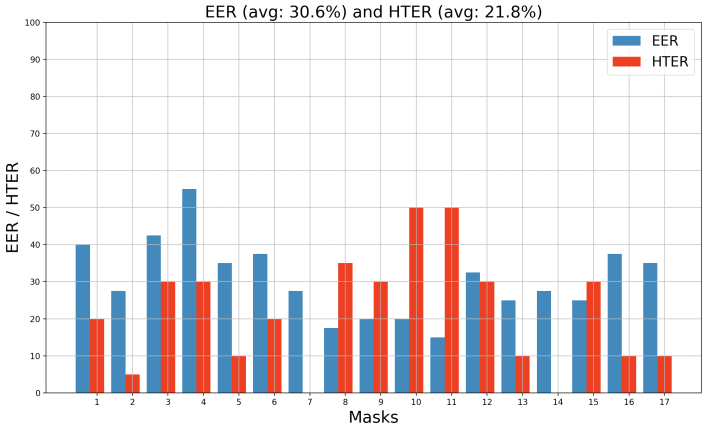
	3DMAD	Replay-Attack	Replay-Mobile	MSU-MFSD
2SR	11.76%	49.37%	47.27%	**34.58%**
CHROM	**8.82%**	**11.25%**	54.8%	57.49%
Li's CVPR14	41.02%	31.12%	**24.43%**	50.00%

4.2.2. Experimental Results

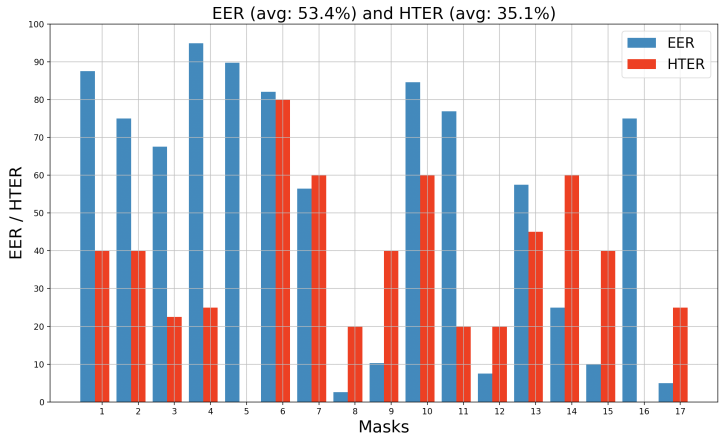
For the binary classification process with a magnitude feature vector (R_2), we utilize an SVM classifier with RBF kernel again. Using the magnitude feature vector, we get improvements and better HTER metrics with the CHROM algorithm on the performance of the classification process of 3DMAD datasets, as presented in Table 4.2. On the other hand, our classification process's performance only improves on the MSU-MFSD dataset when we use magnitude features extracted by the 2SR algorithm. For the other datasets, performance decreases for the 2SR algorithm compared to using only two pulse features. The magnitude features extracted by Li's CVPR14 algorithm also improves the classification performance on Replay-Attack and Replay-Mobile datasets. However, using magnitude features decreases performance on 3DMAD and MSU-MFSD datasets. Improvement of the results of the 3DMAD dataset with the CHROM magnitude feature vector is observable in Fig. 4.5. The ROC curves in Fig. 4.6, 4.7, 4.8 shows that the magnitude feature vector obtained by the Li's CVPR14 algorithm improves the classification performance for the Replay-Attack and Replay-Mobile datasets.



(a) 2SR

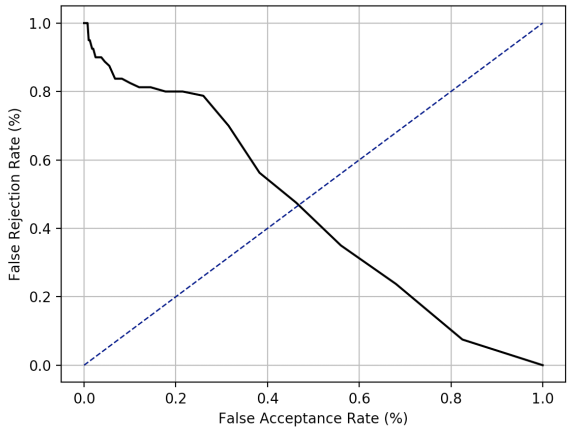


(b) CHROM

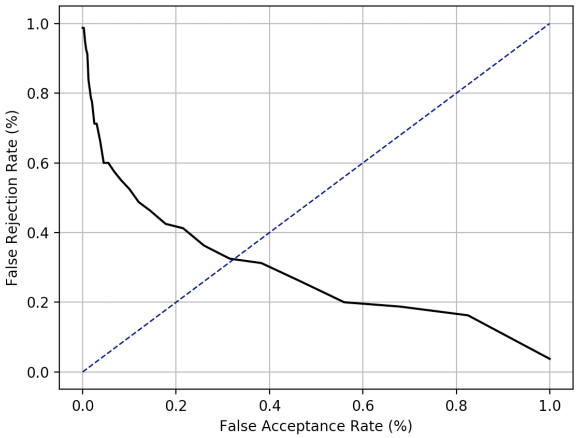


(c) Li's CVPR14

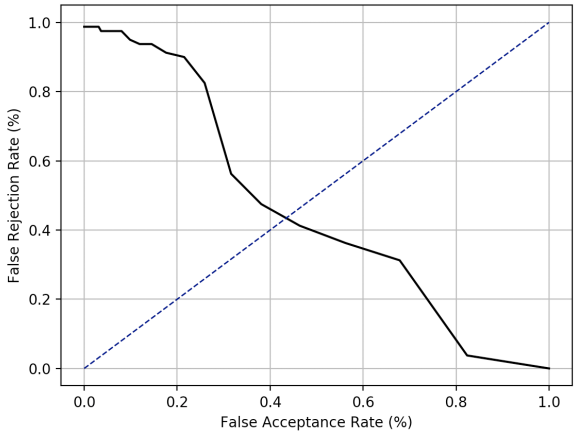
Figure 4.1: EER and HTER metrics for SVM classification results of the 3DMAD dataset in 17 fold using 2D PPG features extracted with the 2SR, CHROM, and Li's CVPR14 algorithms



(a) 2SR - Replay-Attack

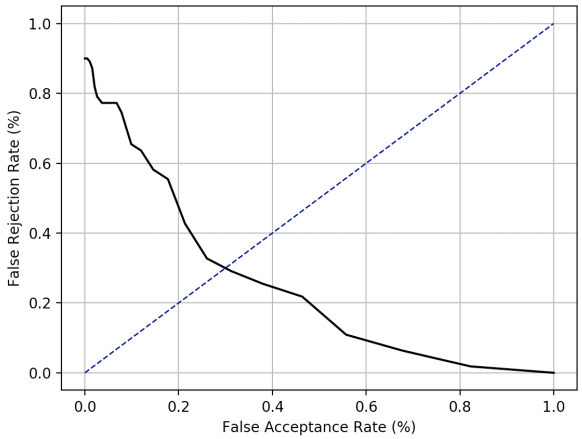


(b) CHROM - Replay-Attack

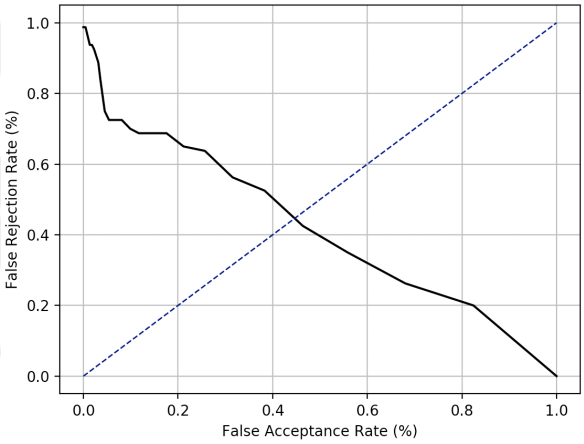


(c) Li's CVPR14 - Replay-Attack

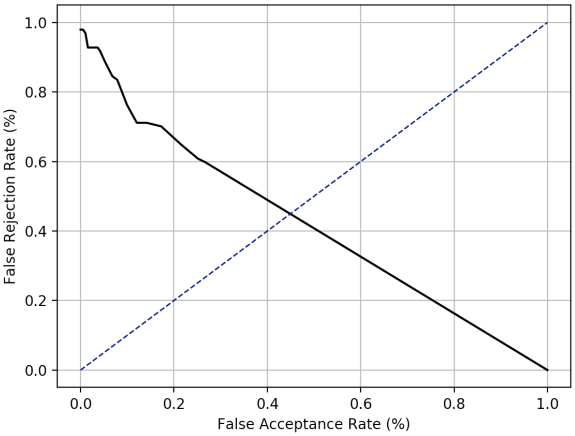
Figure 4.2: ROC curves for the SVM classification (with RBF kernel) results of Replay-Attack dataset using 2D PPG features



(a) 2SR - Replay-Mobile

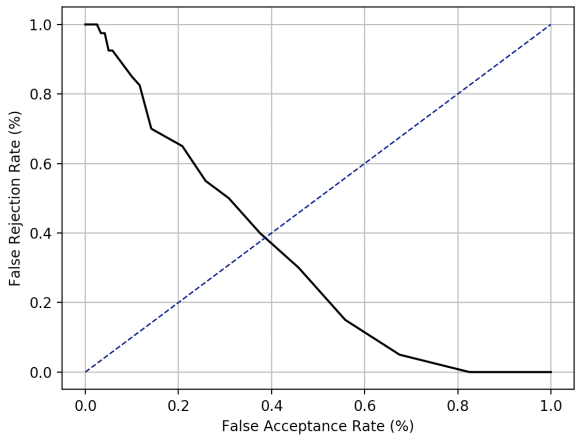


(b) CHROM - Replay-Mobile

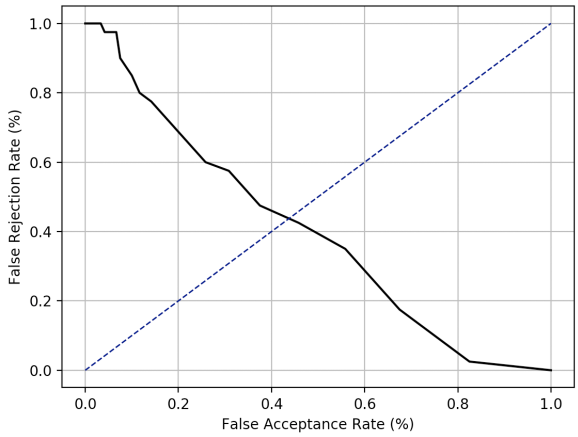


(c) Li's CVPR14 - Replay-Mobile

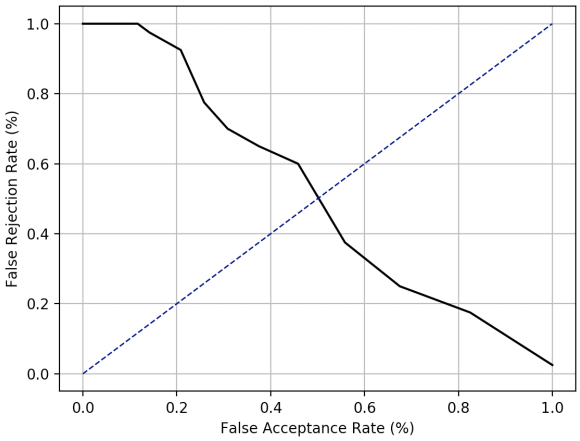
Figure 4.3: ROC curves for the SVM classification (with RBF kernel) results of Replay-Mobile dataset using 2D PPG features



(a) 2SR - MSU-MFSD

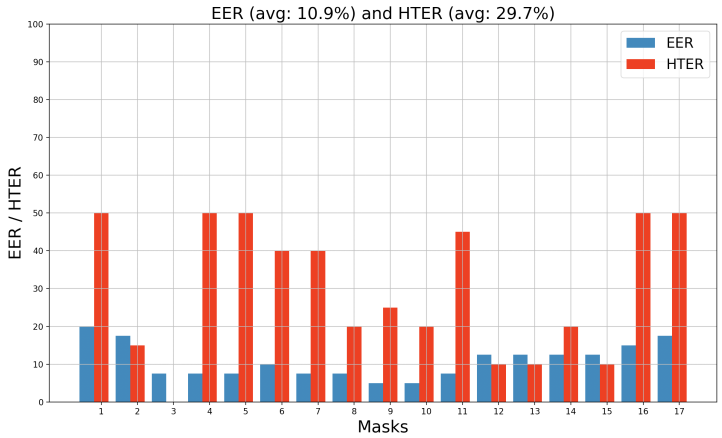


(b) CHROM - MSU-MFSD

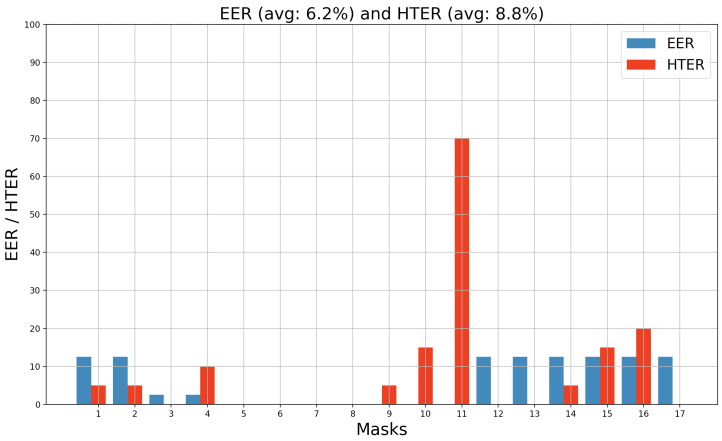


(c) Li's CVPR14 - MSU-MFSD

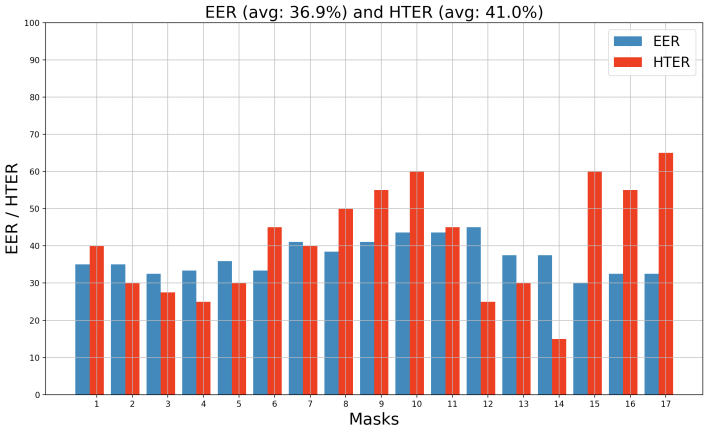
Figure 4.4: ROC curves for the SVM classification (with RBF kernel) results of MSU-MFSD dataset using 2D PPG features.



(a) 2SR

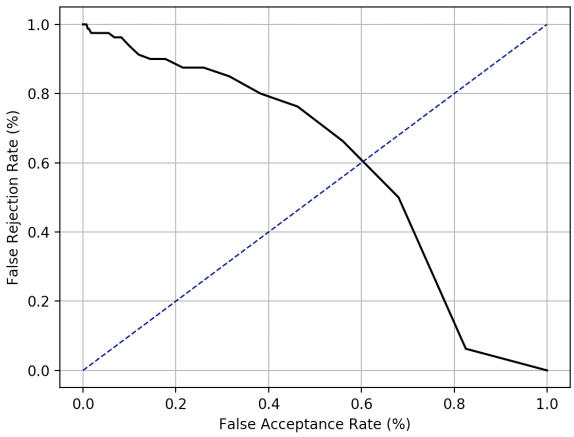


(b) CHROM

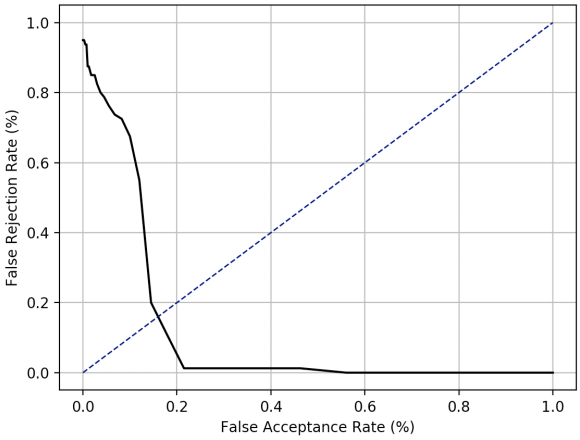


(c) Li's CVPR14

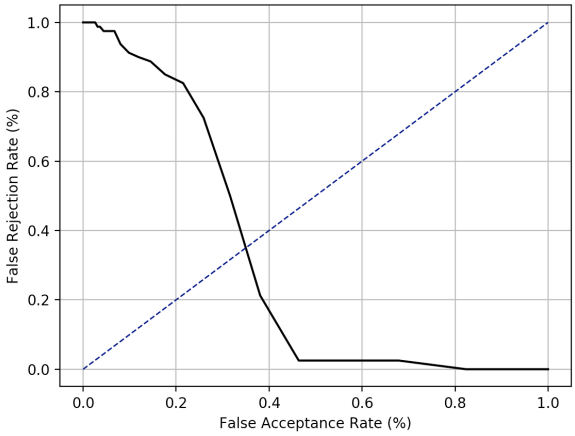
Figure 4.5: EER and HTER metrics for SVM classification results of the 3DMAD dataset in 17 fold using PPG magnitude features extracted with the 2SR, CHROM, and Li's CVPR14 algorithms



(a) 2SR - Replay-Attack

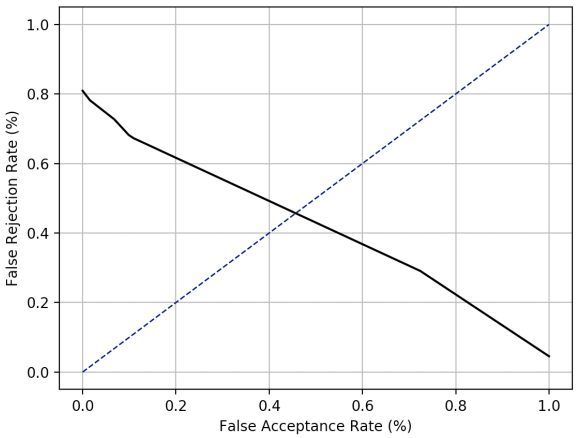


(b) CHROM - Replay-Attack

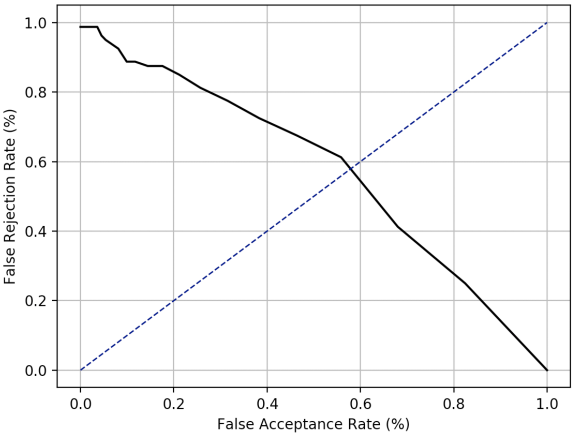


(c) Li's CVPR14 - Replay-Attack

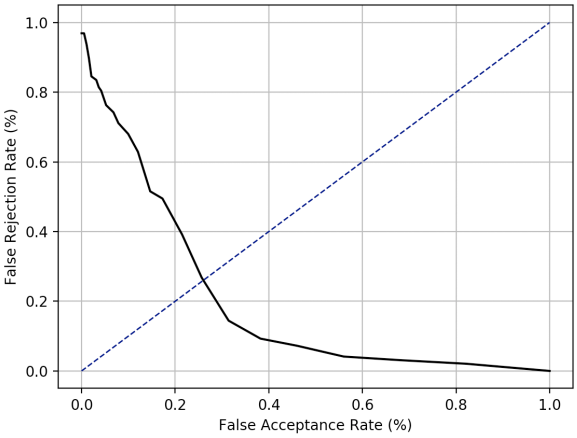
Figure 4.6: ROC curves for the SVM classification (with RBF kernel) results of Replay-Attack dataset using PPG magnitude features



(a) 2SR - Replay-Mobile

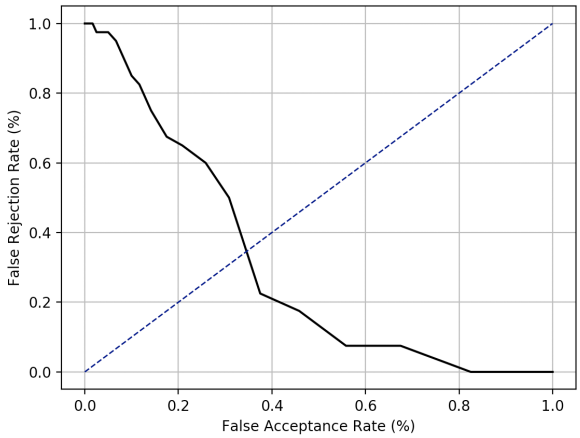


(b) CHROM - Replay-Mobile

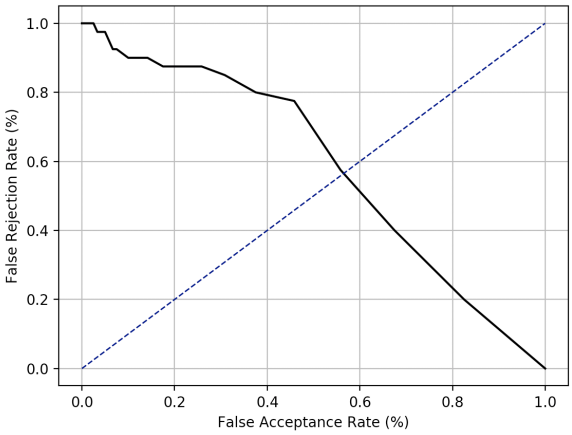


(c) Li's CVPR14 - Replay-Mobile

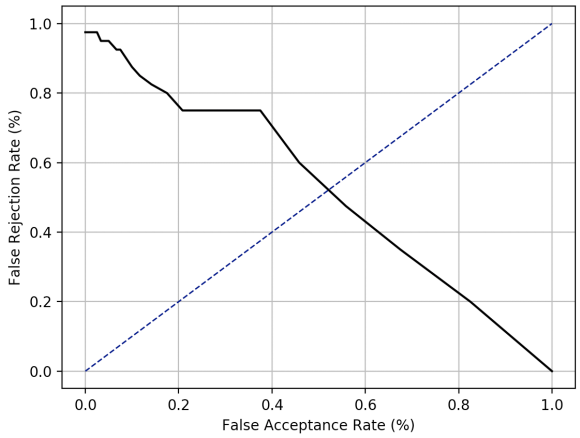
Figure 4.7: ROC curves for the SVM classification (with RBF kernel) results of Replay-Mobile dataset using PPG magnitude features



(a) 2SR - MSU-MFSD



(b) CHROM - MSU-MFSD



(c) Li's CVPR14 - MSU-MFSD

Figure 4.8: ROC curves for the SVM classification (with RBF kernel) results of MSU-MFSD dataset using PPG magnitude features

5. CASCADED FUSION FOR PRESENTATION ATTACK DETECTION

Feature diversity is crucial for the success of a PAD system. On the other hand, the utilization method of features is just as important as the variety of the feature set. The combination method of features dealing with the curse of dimensionality, hyper-parameter optimization, decision for binary classification, or multi-class classification are the most critical factors affecting a PAD system's performance. In this section, we introduce a novel PAD method and try to deal with these PAD performance-related concerns effectively. We also present the experimental studies that we obtained for the best PAD classification results.

5.1. Overview of the Cascaded Fusion Architecture

The overall view of the proposed cascaded fusion system PAD is shown in Fig. 5.1. 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. The feature extraction block consists of the extraction of three types of features: rPPG features, motion-based features (consisting of head-pose, eye-gaze, blinks), and texture-based features. These blocks will be explained in more detail below.

After the 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 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 vectors are used as feature vectors for the second phase of SVM classification. The details of each block will be described below.

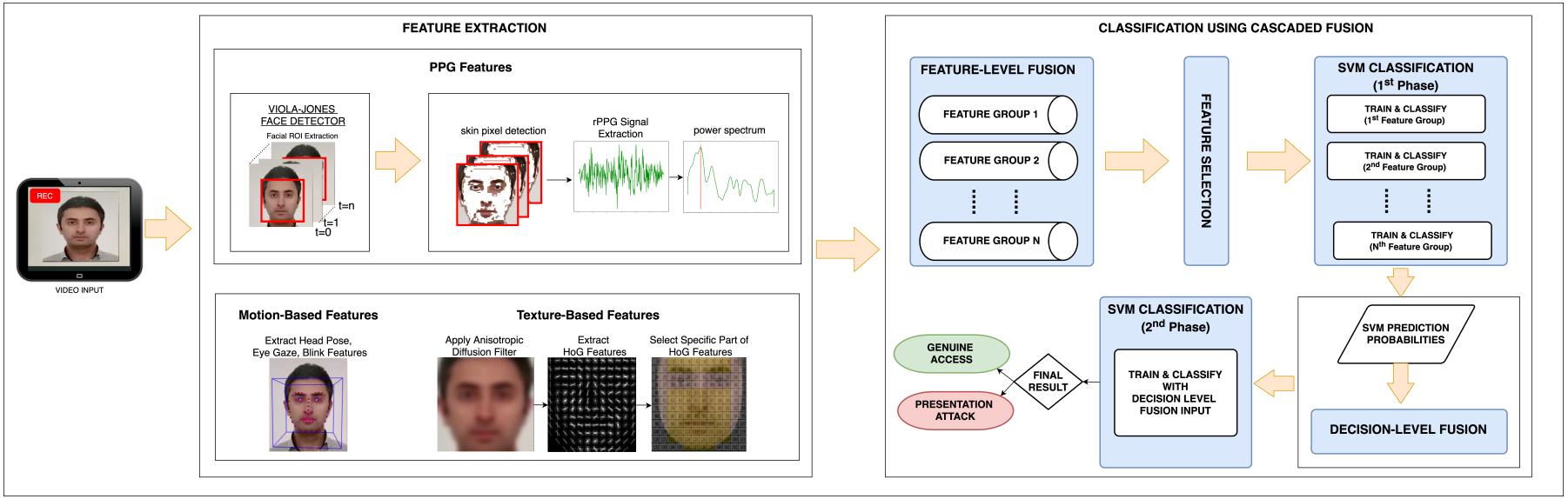


Figure 5.1: PAD Flow of Proposed Method.

5.2. Motion and Texture-Based Features for Presentation Attack Detection

PPG based features are useful for discrimination between genuine and attack presentations. In the previous section, we obtained a HTER value of 8.82% on the 3DMAD dataset. However, the HTER metric gets worse for other datasets. This result is directly dependent on the structure of datasets. The 3DMAD dataset only contains mask attack types. On the other hand, the other datasets contain different types of presentations, like photo and video replay attacks. Since the video replay attacks still contain pulse signals, it is challenging to distinguish video replay attacks from genuine presentations. It is inevitable to introduce additional features to increase the PAD classification success on all datasets. Motion-based features could provide useful clues for differentiating photo attacks. Texture based features could provide useful discriminators for flat and constant patterns such as the same fabric masks or moire patterns occurring on video replay attacks. In this section, we introduce new features for the improvement of our proposed PAD architecture.

5.2.1. Motion-Based Feature Extraction

Although rPPG features are beneficial for liveness detection, they are not sufficient to diversify all kinds of presentation attacks as the pulse signal still exists in a video replay attack. Focusing on each attack type and producing new features could be useful for getting better PAD classification results. Hence, we also used motion-based features of the head and face for PAD. Motion-based features like an eye blink, eye gaze angle, and head pose characteristics could be valuable for the detection of photo attacks.

We utilized the OpenFace 2.0 [44, 45, 46, 47, 48] toolbox for the extraction of motion features. We obtain the blink detection, gaze angle, and head pose features at each frame and calculate their mean to use as features. These motion features are expected to be useful for the detection of photo attacks, since they have no blinks, and eye gaze changes.

The blink feature will be denoted by B, which is one-dimensional. If there is no blink at a video frame, the value of blink feature will be 0, otherwise it will be larger than 0.

Let's denote the gaze angle feature as $G = [\angle_x, \angle_y]$, which represents the average eye gaze angle of both eyes in radians with respect to the camera frame of reference. For the estimation of eye gaze, OpenFace 2.0 toolbox detects eyelids, iris and pupil utilizing a constrained local neural field (CLNF) detector. Then pupil and eye location is used for the calculation of eye gaze vector for each eye [44].

The head-pose feature vector is six-dimensional $H = [T_x, T_y, T_z, R_x, R_y, R_z]$, where T_x, T_y, T_z represents the location of the head with respect to the camera, and R_x, R_y, R_z represents the rotation angles around x, y, z axes.

Figure 5.2 presents a sample illustration of motion-based features.

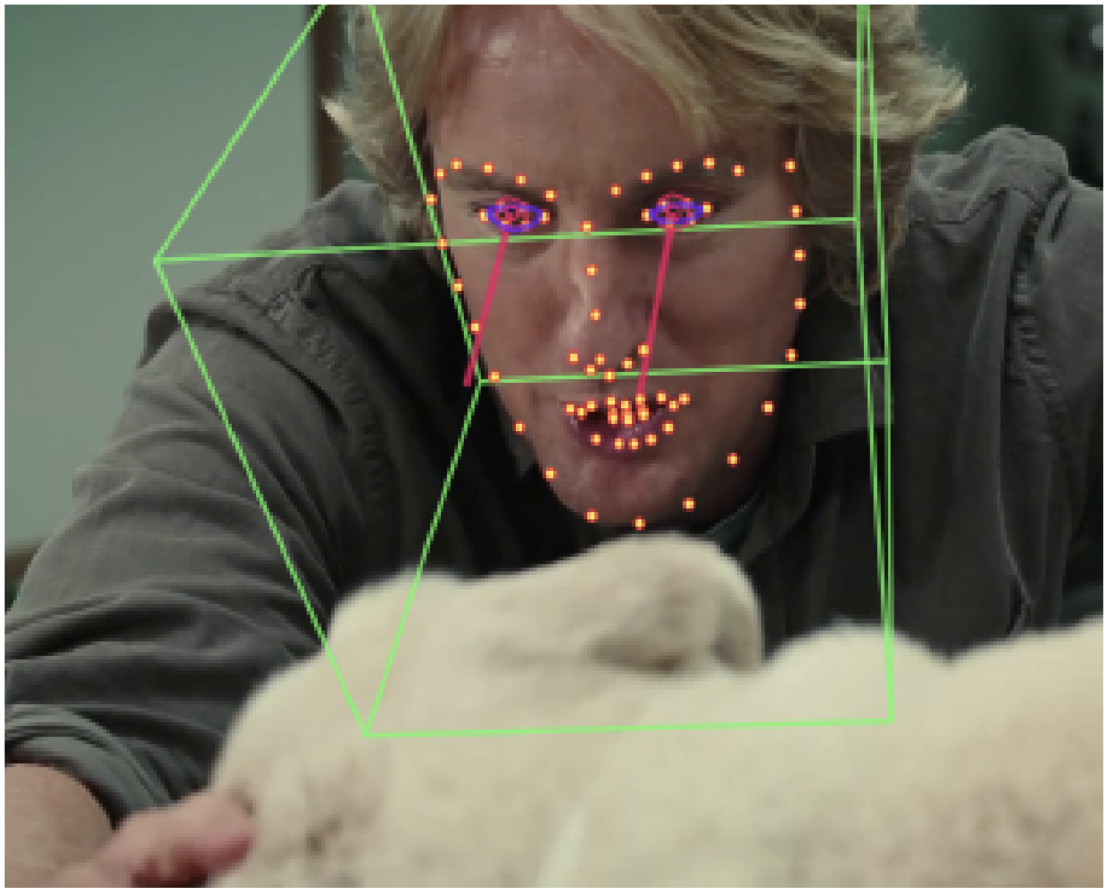


Figure 5.2: Face landmarks, eye gaze and headpose [44].

5.2.2. Texture-Based Feature Extraction

Texture-based features could contain useful clues for discovering flat patterns like the surface of the masks used in the 3DMAD dataset. Texture-based features could

be fed into image classification algorithms like SVM to obtain desired results. In our cascaded PAD system, we utilize the histogram of gradients (HoG) [49, 50] features for this purpose.

HoG is a feature descriptor mainly used for object detection purposes in computer vision research area. HoG works on grayscale images similar using sliding windows on the image pixels. The magnitude and the direction of the change in the intensities create the gradients for each cell.

For the calculation of the HOG feature descriptor of a cropped face region of an image input, facial ROI needs to be divided into patches. In our experiments the cropped face region is divided into 12 x 12 = 144 patches. For each image patch, it is required to calculate the horizontal and vertical gradients. It is possible to obtain the gradients when we filter the image patch with a Sobel operator according to predefined kernel. Magnitude and direction of the gradients can be calculated using the below equations:

$$g = \sqrt{g_x^2 + g_y^2} \tag{5.1}$$

$$\theta = arctan\frac{g_y}{g_x} \tag{5.2}$$

With the use of the magnitude and the direction of the gradients, the histogram of gradients is calculated. In our experiments, the HoG feature descriptor consists of 31 bins that correspond to the angles 0, 12, 24, 36...336, 348, 360. The magnitude values are added proportionally to the corresponding bins according to their direction.

Visualization of the HoG descriptor could be done via plotting the 31x1 histograms in each image patch. By plotting the HoG features of a facial region, we can observe the structure of a subject's face. Fig. 5.3 illustrates a sample plot of an HoG descriptor. The left side of the figure contains a sample plot of a face image for 144 patches. The right side of the figure contains the patch numbers and the projection of the corresponding 144 facial regions of an aligned face. We consider the yellow marked cells in our PAD method. The intuition behind using the HoG features in our study is to discover the different structures of the masks compared to a real face.

For the 3DMAD dataset, using HoG features could be useful to capture analog patterns of the masks. Before the extraction of HoG features on the datasets, we apply an additional anisotropic diffusion filter on facial ROI as a preprocessing step to reveal more differences between the videos of the attack and genuine presentations before getting the HoG feature vector of the facial region. Fake face images become more blurry after application of an anistropic diffusion filter [51]. Figure 5.4 presents the impact of anistropic diffusion filter on genuine and fake face images. After applying an anisotropic diffusion filter, we consider only the first two frames that we can detect a face in the subject video stream and take the average of them for HoG feature extraction. Let us denote the texture feature vector by T, which is 2511 dimensional.

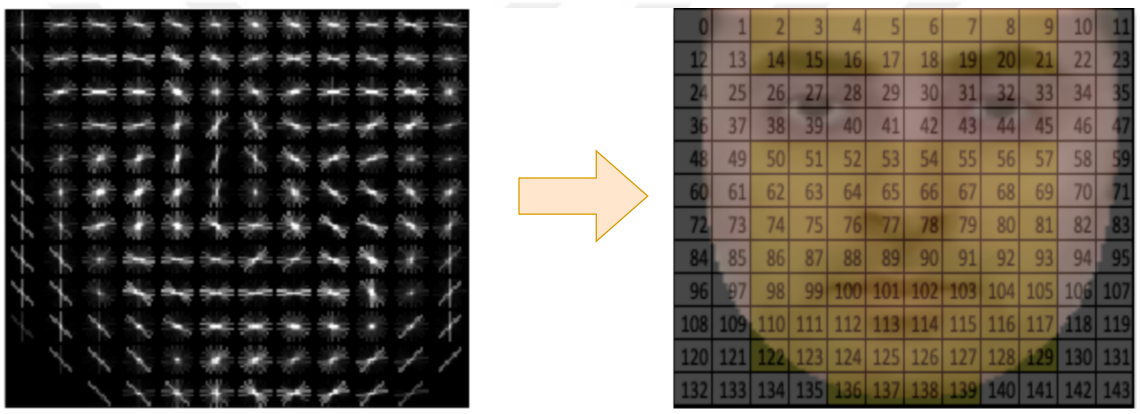


Figure 5.3: HoG features and corresponding facial region cells

Figure 5.4: Anisotropic diffusion filter impact on genuine and fake face images. (a), (b) and (c) are genuine face images, (d), (e) and (f) are fake face images 5.4.

5.3. Grid Search

In our PAD system, we utilize grid search functionality to optimize SVM classification parameters both for linear and RBF kernels. The linear SVM kernel only uses the cost (C) hyper-parameter for the optimization. The RBF SVM kernel uses both C and gamma (γ) hyper-parameters for the optimization.

C and γ parameters determine the performance of an SVM classifier. C parameter corresponds to the cost of misclassification by SVM. Large C values provide low bias and high variance, as we penalize the cost of misclassification too much. On the other hand, small C values give higher bias and low variance and can underfit the training set. We require γ hyper-parameter to handle non-linear classification process. If the points in a training set are not linearly separable in two dimensions, it is possible to map them to a higher dimension to make them linearly separable. Small γ values give low bias and high variance. On the contrary, large γ provides higher bias and low variance.

We use the python scikit-learn library [52] to employ a grid search process over the custom-defined ranges for C and γ parameters. For the C hyper-parameter, we define the range input as [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288]. For the γ hyper-parameter, we define the range input as [0.1, 0.01, 0.001, 0.0001, 0.000001, 0.0000001]. We implement a parallel grid search functionality in our PAD system that tries to optimize the hyper-parameters in both linear and RBF kernel and chooses the best performing one.

5.4. Binary and Multi-class Support Vector Machine Classification

The final decision of a PAD system is usually a binary output as a genuine or attack presentation. However, some feature groups used for the PAD system could provide better classification results with multi-class classification. To support both binary and multi-class classification in our experiments, we utilize multi-class SVM classification as a three-class classification problem as a photo-attack, video-attack, and genuine classes in addition to the binary classification procedure.

5.5. Feature Selection

To improve the classification accuracy and reduce overfitting, we apply an automatic and manual feature selection process over some feature groups. For HoG features, we first apply a manual feature selection process to only include extracted values from the forehead, cheeks, and chin area of the subjects' facial region, as shown by the yellow region in the right image in Fig. 5.3. After that, we apply an automatic feature selection by utilizing an extra-trees classifier implementation of the scikit-learn library.

Extra trees classifier is an algorithm to obtain feature importances. Each decision tree uses the original training sample. For each tree, k features are provided, and each tree selects the best feature to split the data on some mathematical criteria like the Gini index (5.3) or entropy (5.4). This random sampling method creates multiple de-correlated decision trees. Each feature's importance is calculated according to the selected criteria, and the features are listed in descending order. Moreover, the top n features are selected, and others are eliminated.

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{5.3}$$

$$Entropy = \sum_{i=1}^{c} -(p_i) log_2(p_i) \tag{5.4}$$

where p_i is the probability of the i^{th} class.

It is expected that the Gini index equals to 0 for a 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. This gives us the value of the Gini index. The sum of the calculated Gini indexes of each split is used to calculate feature importances.

On the other hand, entropy calculation weights the probability of a class with its base two logarithm. When using entropy as a metric, the feature importances is defined by the calculation of an information gain value. Information gain calculation is done by subtracting the child nodes' entropy values from the parent node's entropy value:

$$Gain(S, f) = Entropy(S) - \sum_{v \in Values(f)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5.5}$$

where $Gain(S, f)$ is the information gain for the dataset S for the feature f. S_v is the dataset after the split.

Smaller values of entropy are better because this makes the difference between the parent node's entropy and child nodes' entropy larger.

In our experiments, we empirically set the total number of HoG features to keep as 2048 and use 4096 estimators that use a maximum of 256 features for determining the importance sorting of the features. For PPG magnitude features, we use the 2048 estimators utilizing max 256 features to keep 512 features as a result of the feature selection process. We use the Gini index as selection criterion. We do not apply any manual feature selection process for the PPG magnitude feature set.

5.6. Feature-Level Fusion

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 $\{R_1, R_2, B, G, H, T\}$ in a parametric way to form N feature groups (FG). Then, we perform a feature selection process over the feature groups using extra trees classifier to compute each feature's importance. We eliminate the less important features 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, which can output the probability vector for two-class classification (genuine/attack) or three-class classification (genuine/photo attack/video-replay attack).

5.7. Decision-Level Fusion

We introduce a decision level fusion step to combine the results of N feature groups. We concatenate the N genuine class probabilities produced by the N SVM classifiers in the 1^{st} phase corresponding to the N feature groups. Let the N-dimensional feature vector obtained after this concatenation be denoted as F. Then, we train another SVM (2^{nd} phase) to classify the input video as an attack or genuine presentation (see Fig. 5.1).

5.8. Experimental Work

In our experiments, we focused on the impact of the feature-level and decision-level fusion phases of our PAD system. We present our experiments on four different configurations given in Table 5.1. In the first configuration **(conf1)**, we fuse all types of features (PPG, texture-based, and motion-based) in a single feature group. In the second configuration **(conf2)**, we only use rPPG features obtained by the CHROM algorithm. In the third setup **(conf3)**, we place each type of feature in a single feature group and

Table 5.1: The Tested Configurations for Feature-Level and Decision-Level Fusion Phases

Id	Feature-Level Fusion Details*	1st Phase SVM	2nd Phase SVM
conf1	FG1: $\{R_{2-chrom}, B, G, H, T\}$	2-class	-**
conf2	FG1:$\{R_{1-chrom}, R_{2-chrom}\}$	2-class	-**
conf3	FG1: $\{R_{1-chrom}\}$	2-class	2-class
	FG2: $\{R_{2-chrom}\}$	2-class	
	FG3: $\{R_{1-2sr}\}$	2-class	
	FG4: $\{R_{2-2sr}\}$	2-class	
	FG5: $\{B\}$	2-class	
	FG6: $\{G\}$	2-class	
	FG7: $\{H\}$	2-class	
	FG8: $\{T\}$	2-class	
conf4	FG1: $\{R_{1-chrom}, R_{2-chrom}\}$	2-class	2-class
	FG2: $\{T\}$	2-class	
	FG3: $\{B, G, H\}$	3-class	

*FG: Feature Group.

**Decision-level fusion does not exist when there is only one feature group.

try to see the impact of decision-level fusion. In the fourth configuration **(conf4)**, 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.

The experimental results for the proposed cascaded fusion are given in Table 5.2. We also compare our results with other methods in the literature, which utilize rPPG features. It can be observed that fusing all types of features in a single group (conf1) only gives the best result for the 3DMAD dataset, which has only mask attacks. However, for other datasets, which contain different types of attacks, concatenating all features does not give the best results. Using only rPPG features (conf2) provides the best HTER for the Replay-Attack dataset. Nevertheless, the same setup does not work on the Replay-Mobile

Table 5.2: rPPG Based PAD Results (HTER [%]) for Four Datasets

	3DMAD	**Replay Attack**	**Replay Mobile**	**MSU-MFSD**
Li CVPR + LTSS [16]	17.0	16.1	32.5	35.0
Nowara et al. [16], [20]	43.0	25.5	35.9	31.7
CHROM + LTSS [16]	29.0	20.9	38.1	50.6
2SR + LTSS [16]	13.0	5.9	37.7	43.3
Li et al. PPG [12]	7.94	-	-	36.67
Our method (conf1)	**0.58**	50.0	24.92	48.33
Our method (conf2)	41.17	**4.62**	54.64	52.08
Our method (conf3)	1.47	10.25	14.23	**27.5**
Our method (conf4)	0.88	17.0	**11.69**	35.0

dataset, which contains videos with higher resolution. After using the decision-level fusion setups (conf3 and conf4), we see the real contribution of decision-level fusion property of our PAD system on the Replay Mobile and MSU-MFSD datasets as compared to using only the feature-level fusion. Our results are better than other methods in the literature that utilize rPPG features.

6. RESULTS AND DISCUSSION

This section discusses the contribution of the cascaded fusion system and additional features to the PAD classification performance and the impacts of the dataset structure on experimental results.

6.1. Impacts of Cascaded Fusion on Presentation Attack Detection Performance

Table 5.2 demonstrates the performance of our PAD detection system for each dataset compared to PPG based other methods in the literature. With only the use of feature-level fusion during SVM classification on separated feature groups, it is impossible to achieve these HTER values on Replay Attack, Replay Mobile, and MSU-MFSD datasets. Decision-level fusion process improves the accuracy and HTER value on the classification process. ROC curves illustrated in Fig. 6.1, 6.2, and 6.3 justify that decision-level fusion improves PAD process performance compared to the ROC curves obtained with the only use of 2D PPG and PPG magnitude features in a feature-level fusion.

6.2. Impacts of Motion-Based and Texture-Based Features on Presentation Attack Detection Performance

The utilization of HoG features is beneficial on the 3DMAD dataset as it is observable in Fig. 6.4, there is a significant improvement on the HTER results of each experiment fold. The pattern of uniform masks used in this dataset reveals discriminative features that help us achieve a better HTER result during the SVM classification process compared to using PPG features alone.

For the Replay-Attack dataset, we see that a good feature selection and enhanced grid search range take the classification results to a much more satisfying level only with the use of PPG features. This result emphasizes that hyper-parameter optimization is crucial to obtain more reliable SVM scores and probability predictions.

For the Replay-Mobile dataset, we see that sometimes the use of multi-class SVM classification as a first step provides better results than binary classification. Fusing the

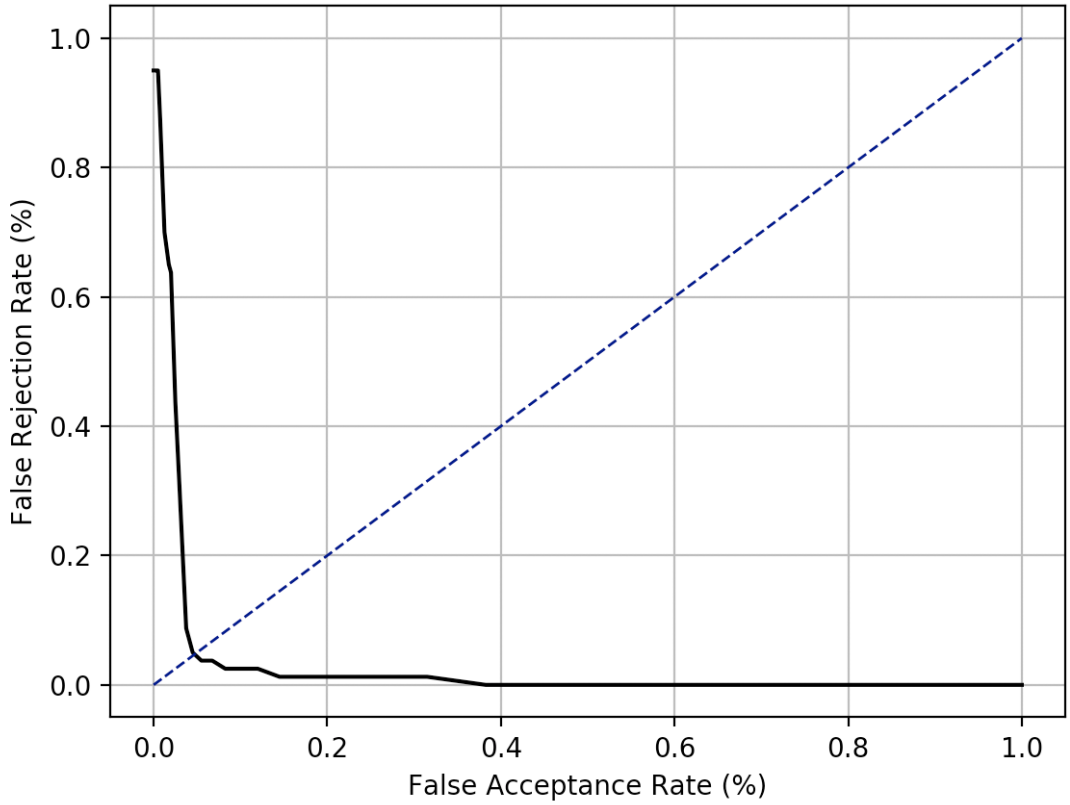


Figure 6.1: ROC curve for the SVM classification (with linear kernel) results of Replay-Attack dataset using proposed PAD system

probability results of distinctive SVM classifiers to provide them as an input to a second step SVM classification process improves the final decision on PAD. Decision-level fusion also improves the results for the MSU-MFSD dataset. Nevertheless, still, there is room for improvement in the PAD process of the Replay-Mobile and MSU-MFSD datasets.

The ROC curves illustrated in Fig. 6.5, 6.6, 6.7 prove the contribution of cascaded fusion and additional features to the classification performance.

6.3. Impacts of Dataset Structure on Presentation Attack Detection Performance

The experimental results show that the structure of the dataset directly impacts PAD performance. For instance, the Replay-Mobile dataset contains videos that have higher resolution in comparison to the Replay-Attack dataset. In this case, the PPG signal in the Replay-Mobile dataset could be preserved much more in this dataset's video attack videos.

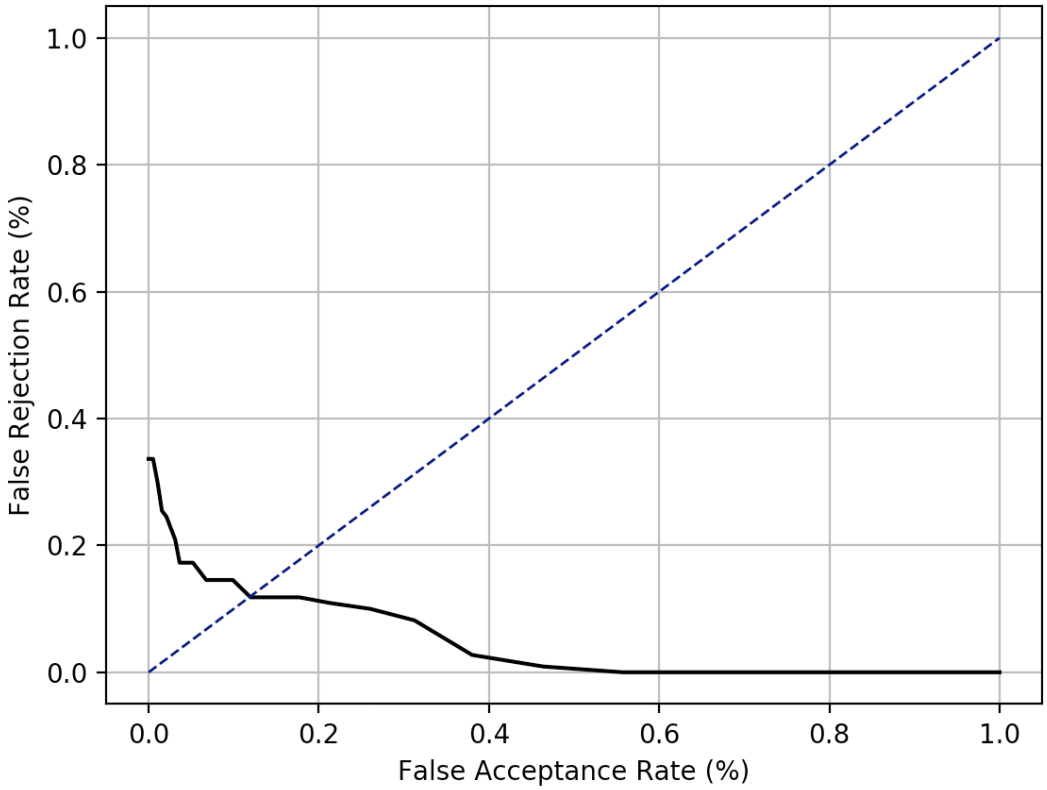


Figure 6.2: ROC curve for the SVM classification (with linear kernel) results of Replay-Mobile dataset using proposed PAD system

Even though the organizational structure of the Replay-Attack and the Replay-Mobile dataset is similar, the PAD results obtained in each dataset are very different, presumably the difference in data recording resolutions.

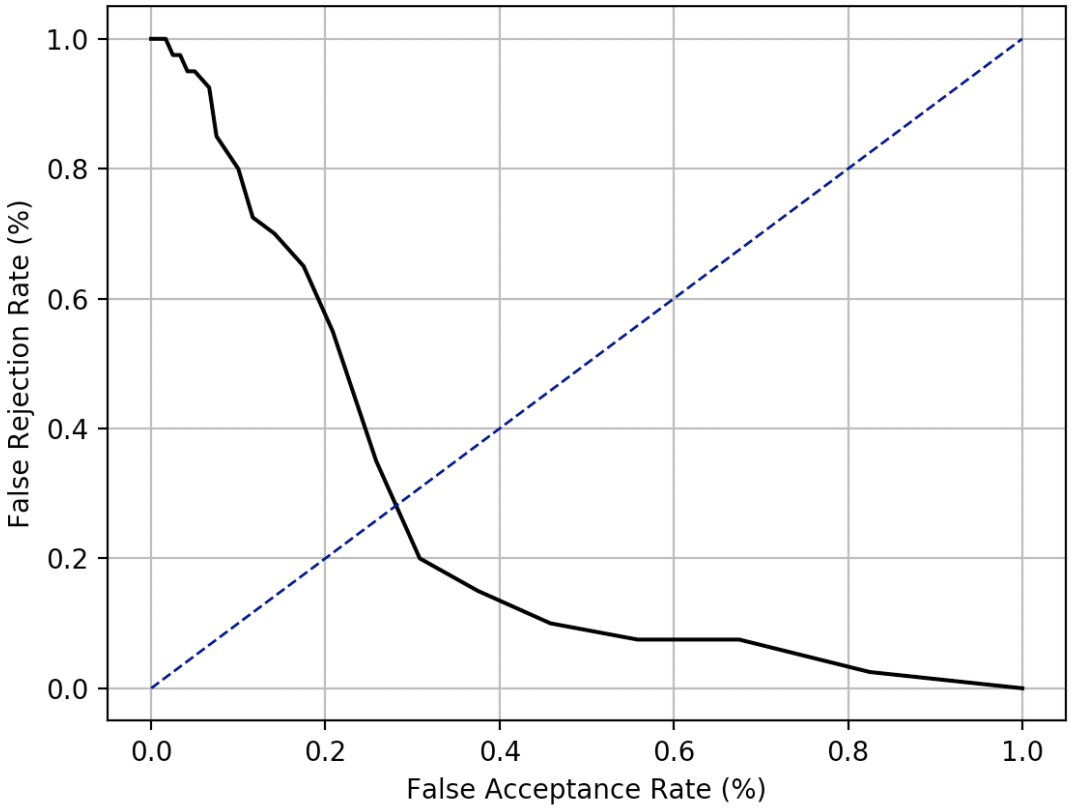


Figure 6.3: ROC curve for the SVM classification (with linear kernel) results of MSU-MFSD dataset using proposed PAD system

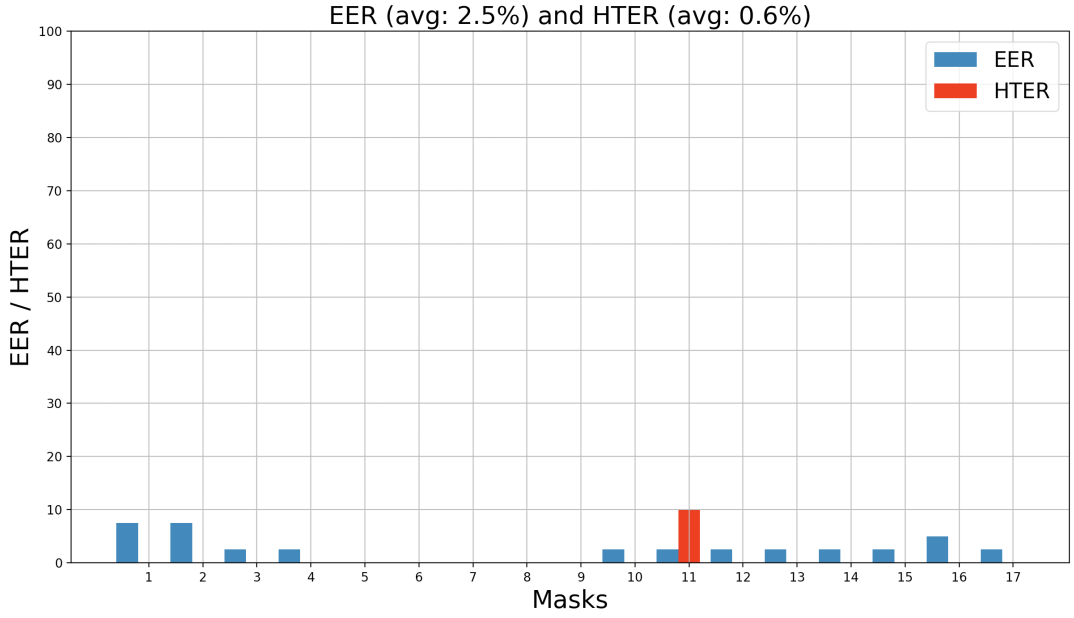


Figure 6.4: EER and HTER metrics for SVM classification results of the 3DMAD dataset using PPG CHROM magnitude and HoG features

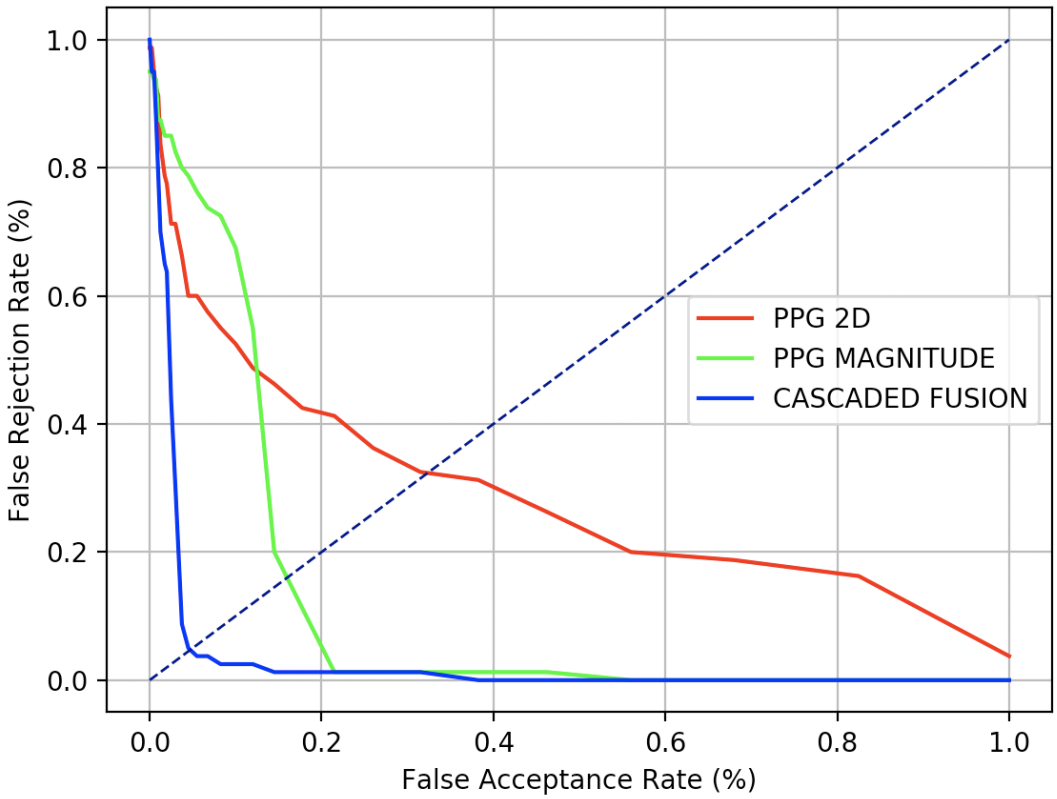


Figure 6.5: ROC curve comparison of Replay-Attack dataset for the PPG 2D, PPG Magnitude and cascaded fusion classification methods (conf2 in Table 5.1)

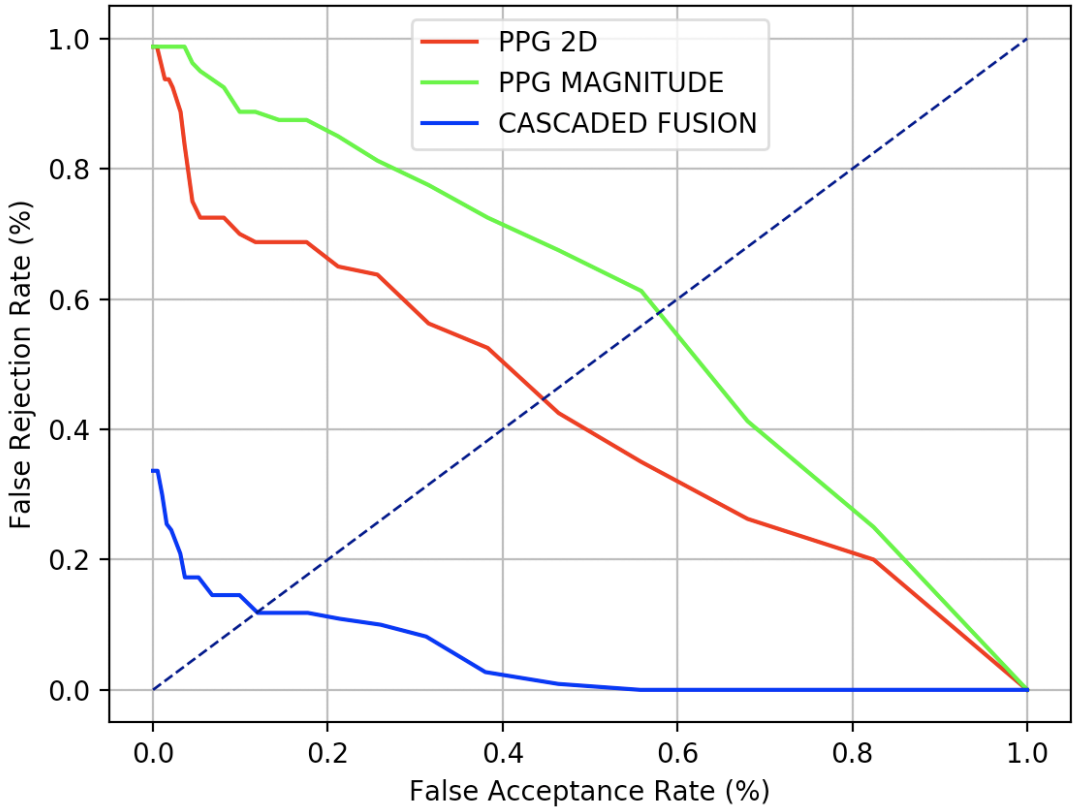


Figure 6.6: ROC curve comparison of Replay-Mobile dataset for the PPG 2D, PPG Magnitude and cascaded fusion classification methods (conf4 in Table 5.1)

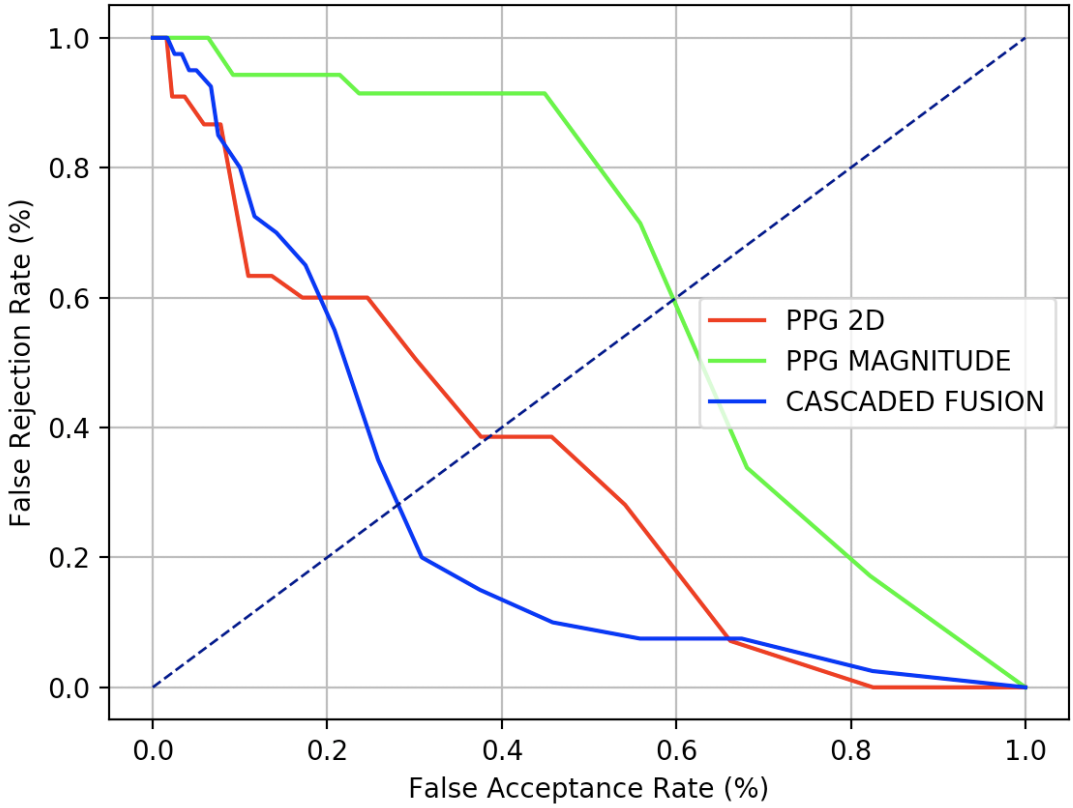


Figure 6.7: ROC curve comparison of MSU-MFSD dataset for the PPG 2D, PPG Magnitude and cascaded fusion classification methods (conf3 in Table 5.1)

7. CONCLUSIONS AND FUTURE WORK

In this thesis, we investigated the use of rPPG for PAD. As we observed from the results of our experiments, PPG features are not enough for every type of attack scenario. After enrichment of PPG features with additional discriminative motion-based and texture-based features concerning other attack types, the results that we obtain show that performance increases are possible, even with linear SVM classification. Decision level fusion is also an essential factor for improving PAD performance when dealing with different groups of features. The results of our experiments demonstrate that the contribution of the same features to the performance of a PAD system in decision level fusion increases as compared to using them only in feature level fusion. As a result of this study, there is still a valid expectation that it is possible to exhibit more discriminative and successful PAD architecture with some preprocessing before feature extraction and introduce more discriminative features.

On the other hand, as it has been observed from the results of the experiments, different feature groups provide the best results for each dataset. There is no collective feature group and fusion structure that provides the optimal result for all datasets. This situation is also known as a domain-shift problem and impacts the cross-dataset performance of PAD experiments, and there is no de facto solution in the literature for this issue. Fig. 7.1 provided in [8] illustrates this problem with a t-Distributed Stochastic Neighbor Embedding (t-SNE) [53] plot of four different datasets. (t-SNE) is a kind of dimensionality reduction technique for the visualization of high-dimensional datasets. As visualized in that figure, data among different datasets is clustered on different regions of the plot, which explain the domain-shift problem in a more clear way.

As future work, it is possible to enhance this study by introducing deep learning-based methods that could also promise better results on PAD. There could be a more convenient way to capture deep temporal and spatial patterns by utilizing a convolutional neural network (CNN). Subtle temporal patterns in an extracted PPG signal could be learned better in a sufficiently deep network, and it is possible to make improvements on PAD architecture by doing more experiments and with an efficient hyper-parameter optimization.

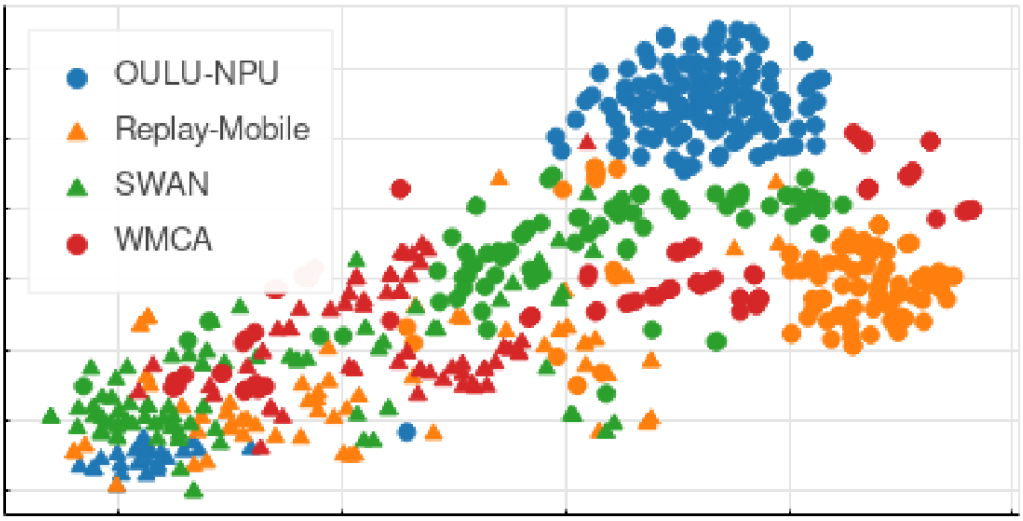


Figure 7.1: t-SNE [53] plot of the embeddings of a CNN-based PAD system (DeepPixBiS [54]) for four datasets. Samples with the same color belong to the same face-PAD dataset. Triangles are genuine samples and circles are presentation attack samples

Addressing the domain-shift problem is still an important research area to explore. Different from the classical approaches, adversarial approaches with utilizing GANs could be adopted for PAD problem. Doing a corruption on a first-hand recording of a genuine subject video with an adversarial method could output significant differences as compared to doing same kind of adversarial corruption on a spoofing video recording obtained from a spoofing medium. Concordantly, a deepfake detector solution could also provide satisfactory results on a PAD problem if it is fused and supported with liveness detection solutions. It could be more sensible to work on cross-dataset experiments after addressing and handling the domain-shift problem in a convenient way.

REFERENCES

[1] Lei Li et al. “Face Presentation Attack Detection in Learned Color-liked Space”. In: *Computing Research Repository(CoRR)* abs/1810.13170 (2018). arXiv: 1810.13170. URL: http://arxiv.org/abs/1810.13170.

[2] X. Chen et al. “Video-Based Heart Rate Measurement: Recent Advances and Future Prospects”. In: *IEEE Transactions on Instrumentation and Measurement* 68.10 (Oct. 2019), pp. 3600–3615. ISSN: 1557-9662. DOI: 10.1109/TIM.2018.2879706.

[3] Raghavendra Ramachandra and Christoph Busch. “Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey”. In: *ACM Computing Surveys* 50.1 (Mar. 2017). ISSN: 0360-0300. DOI: 10.1145/3038924. URL: https://doi.org/10.1145/3038924.

[4] R. Shao, X. Lan, and P. C. Yuen. “Joint Discriminative Learning of Deep Dynamic Textures for 3D Mask Face Anti-Spoofing”. In: *IEEE Transactions on Information Forensics and Security* 14.4 (Apr. 2019), pp. 923–938. ISSN: 1556-6021. DOI: 10.1109/TIFS.2018.2868230.

[5] Berthold K.P. Horn and Brian G. Schunck. *Determining Optical Flow*. Tech. rep. USA, 1980.

[6] H. Li et al. “Learning Generalized Deep Feature Representation for Face Anti-Spoofing”. In: *IEEE Transactions on Information Forensics and Security* 13.10 (Oct. 2018), pp. 2639–2652. ISSN: 1556-6021. DOI: 10.1109/TIFS.2018.2825949.

[7] Xiao Song et al. “Discriminative Representation Combinations for Accurate Face Spoofing Detection”. In: *Computing Research Repository(CoRR)* abs/1808.08802 (2018). arXiv: 1808.08802. URL: http://arxiv.org/abs/1808.08802.

[8] A. Mohammadi, S. Bhattacharjee, and S. Marcel. “Improving Cross-Dataset Performance of Face Presentation Attack Detection Systems Using Face Recognition Datasets”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 2947–2951.

[9] Z. Li et al. “Unseen Face Presentation Attack Detection with Hypersphere Loss”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 2852–2856.

[10] A. Liu et al. “Multi-Modal Face Anti-Spoofing Attack Detection Challenge at CVPR2019”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1601–1610.

[11] X. Yang et al. “Face Anti-Spoofing: Model Matters, so Does Data”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3502–3511.

[12] Xiaobai Li et al. “Generalized face anti-spoofing by detecting pulse from face videos”. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. Dec. 2016, pp. 4244–4249. DOI: 10.1109/ICPR.2016.7900300.

[13] P. Viola and M. Jones. “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*. Vol. 1. Dec. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.

[14] A. Asthana et al. “Robust Discriminative Response Map Fitting with Constrained Local Models”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 3444–3451. DOI: 10.1109/CVPR.2013.442.

[15] J. Hernandez-Ortega et al. “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)*. June 2018, pp. 657–6578. DOI: 10.1109/CVPRW.2018.00096.

[16] G. Heusch and S. Marcel. “Pulse-based Features for Face Presentation Attack Detection”. In: *2018 IEEE 9th International Conference on Biometrics Theory,*

Applications and Systems (BTAS). Oct. 2018, pp. 1–8. DOI: 10.1109/BTAS.2018.8698579.

[17] X. Li et al. "Remote Heart Rate Measurement from Face Videos under Realistic Situations". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 4264–4271. DOI: 10.1109/CVPR.2014.543.

[18] G. de Haan and V. Jeanne. "Robust Pulse Rate From Chrominance-Based rPPG". In: *IEEE Transactions on Biomedical Engineering* 60.10 (Oct. 2013), pp. 2878–2886. ISSN: 1558-2531. DOI: 10.1109/TBME.2013.2266196.

[19] W. Wang, S. Stuijk, and G. de Haan. "A Novel Algorithm for Remote Photoplethysmography: Spatial Subspace Rotation". In: *IEEE Transactions on Biomedical Engineering* 63.9 (Sept. 2016), pp. 1974–1984. ISSN: 1558-2531. DOI: 10.1109/TBME.2015.2508602.

[20] E. M. Nowara, A. Sabharwal, and A. Veeraraghavan. "PPGSecure: Biometric Presentation Attack Detection Using Photopletysmograms". In: *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*. May 2017, pp. 56–62. DOI: 10.1109/FG.2017.16.

[21] Umur Aybars Ciftci and Ilke Demir. "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals". In: *Computing Research Repository (CoRR)* abs/1901.02212 (2019). arXiv: 1901.02212. URL: http://arxiv.org/abs/1901.02212.

[22] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. "OpenFace: A general-purpose face recognition library with mobile applications". In: 2016.

[23] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 27:1–27:27.

[24] Keyur Patel et al. "Live Face Video vs. Spoof Face Video: Use of Moire Patterns to Detect Replay Video Attacks". In: May 2015. DOI: 10.1109/ICB.2015.7139082.

[25] Ivana Chingovska and André Anjos. “On the Use of Client Identity Information for Face Antispoofing”. In: *IEEE Transactions on Information Forensics and Security* 10 (2015), pp. 787–796.

[26] Huiling Hao, Mingtao Pei, and Meng Zhao. “Face Liveness Detection Based on Client Identity Using Siamese Network”. In: *Pattern Recognition and Computer Vision Lecture Notes in Computer Science* (2019), pp. 172–180. DOI: 10.1007/978-3-030-31654-9_15.

[27] Ivana Chingovska, André Anjos, and Sébastien Marcel. “On the effectiveness of local binary patterns in face anti-spoofing”. In: *BIOSIG Biometrics Special Interest Group 2012*. Ed. by Arslan Brömme and Christoph Busch. Bonn: Gesellschaft für Informatik e.V., 2012, pp. 183–194.

[28] Artur Costa-Pazo et al. “The Replay-Mobile Face Presentation-Attack Database”. In: Sept. 2016. DOI: 10.1109/BIOSIG.2016.7736936.

[29] D. Wen, H. Han, and A. K. Jain. “Face Spoof Detection With Image Distortion Analysis”. In: *IEEE Transactions on Information Forensics and Security* 10.4 (Apr. 2015), pp. 746–761. ISSN: 1556-6021. DOI: 10.1109/TIFS.2015.2400395.

[30] Nesli Erdogmus and Sébastien Marcel. “Spoofing in 2D Face Recognition with 3D Masks and Anti-spoofing with Kinect”. In: 2013.

[31] Z. Zhang et al. “A face antispoofing database with diverse attacks”. In: *International Association for Pattern Recognition International Conference on Biometrics (IAPR ICB)*. 2012, pp. 26–31.

[32] Z. Boulkenafet et al. “OULU-NPU: A mobile face presentation attack database with real-world variations”. In: 2017.

[33] Y. Liu, A. Jourabloo, and X. Liu. “Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 389–398. DOI: 10.1109/CVPR.2018.00048.

[34] Keyurkumar Patel, Hu Han, and A.K. Jain. " Secure Face Unlock: Spoof Detection on Smartphones". In: *IEEE Transactions on Information Forensics and Security (TIFS)* (2016).

[35] Shifeng Zhang et al. "CASIA-SURF: A Dataset and Benchmark for Large-scale Multi-modal Face Anti-spoofing". In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* (2020).

[36] Brian Dolhansky et al. "The Deepfake Detection Challenge (DFDC) Preview Dataset". In: *arXiv preprint arXiv:1910.08854* (2019).

[37] H. Demirezen and C. E. Erdem. "Remote Photoplethysmography Using Nonlinear Mode Decomposition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 1060–1064.

[38] R. Stricker, S. Müller, and H. Gross. "Non-contact video-based pulse rate measurement on a mobile service robot". In: *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. 2014, pp. 1056–1062.

[39] Serge Bobbia et al. "Unsupervised skin tissue segmentation for remote photoplethysmography". In: *Pattern Recognition Letters* (Oct. 2017). DOI: 10.1016/j.patrec.2017.10.017.

[40] X. Li et al. "Remote Heart Rate Measurement from Face Videos under Realistic Situations". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4264–4271.

[41] Vahid Kazemi and Josephine Sullivan. "One Millisecond Face Alignment with an Ensemble of Regression Trees". In: June 2014. DOI: 10.13140/2.1.1212.2243.

[42] André Anjos et al. "Bob: a free signal processing and machine learning toolbox for researchers". In: Oct. 2012. DOI: 10.1145/2393347.2396517.

[43] N. Erdogmus and S. Marcel. "Spoofing Face Recognition With 3D Masks". In: *IEEE Transactions on Information Forensics and Security* 9.7 (July 2014), pp. 1084–1097. ISSN: 1556-6021. DOI: 10.1109/TIFS.2014.2322255.

[44] T. Baltrusaitis et al. “OpenFace 2.0: Facial Behavior Analysis Toolkit”. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 2018, pp. 59–66.

[45] Amir Zadeh, Tadas Baltrusaitis, and Louis-Philippe Morency. “Deep Constrained Local Models for Facial Landmark Detection”. In: *Computing Research Repository (CoRR)* abs/1611.08657 (2016). arXiv: 1611.08657. URL: http://arxiv.org/abs/1611.08657.

[46] T. Baltrusaitis, P. Robinson, and L. Morency. “Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild”. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 354–361.

[47] E. Wood et al. “Rendering of Eyes for Eye-Shape Registration and Gaze Estimation”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3756–3764.

[48] T. Baltrušaitis, M. Mahmoud, and P. Robinson. “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)*. Vol. 06. 2015, pp. 1–6.

[49] P. F. Felzenszwalb et al. “Object Detection with Discriminatively Trained Part-Based Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32.9 (2010), pp. 1627–1645.

[50] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1.

[51] Changyong Yu et al. “Diffusion-based Kernel Matrix Model for Face Liveness Detection”. In: *Image and Vision Computing* 89 (July 2019). DOI: 10.1016/j.imavis.2019.06.009.

[52] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[53] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[54] Anjith George and Sébastien Marcel. “Deep Pixel-wise Binary Supervision for Face Presentation Attack Detection”. In: *International Conference on Biometrics*. 2019.

RESUME

MEHMET FATİH GÜNDOĞAR

Marmara University
Computer Engineering Department, Faculty of Engineering,
Goztepe Campus, Kadikoy, Istanbul, Turkey
Email: fatih_gundogar@yahoo.com

Education

- M.S., Computer Science Marmara University, Faculty of Engineering, Istanbul, Turkey, 2020.
  - *Thesis Topic:* Detection of Presentation Attacks for Face Recognition Systems
- B.S., Computer Science Ege University, Faculty of Engineering, Izmir, Turkey, 2006.
  - *Thesis Topic:* 3D Visualization on Geographical Information Systems

Work Experience

December 2019 - present

- Functional Architect, Architecht Bilisim Sistemleri

September 2011 – December 2019

- Solution Architect, Vodafone Turkey

October 2006 – September 2011

- Software Engineer, Atos

Research Interests

- Presentation Attack Detection, Machine Learning, Deep Learning

Foreign Languages

- Turkish (Native), English (Fluent)