Guillermo Estrada Domech

An Assessment of Presentation Attack Detection Methods for Face Recognition Systems

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
Programa de Pós-graduação em Engenharia Elétrica

Rio de Janeiro
August 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Guillermo Estrada Domech

An Assessment of Presentation Attack Detection Methods for Face Recognition Systems

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor : Prof. Raul Queiroz Feitosa
Co-advisor: Dr. Gilson Alexandre Ostwald Pedro da Costa

Rio de Janeiro
August 2018

Pontifícia Universidade Católica do Rio de Janeiro

Guillermo Estrada Domech

An Assessment of Presentation Attack Detection Methods for Face Recognition Systems

Dissertation presented to the Programa de Pós-Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof. Raul Queiroz Feitosa
Advisor
Departamento de Engenharia Elétrica PUC-Rio

Dr. Gilson Alexandre Ostwald Pedro da Costa
Co-advisor
UERJ

Dra. Aura Conci
UFF

Dr. Leonardo Alfredo Forero Mendoza
UERJ

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico PUC-Rio

Rio de Janeiro, August 16th, 2018

Guillermo Estrada Domech

The author received his bachelor's degree in Biomedical Engineering at the Havana University of Technologies José Antonio Echeverría (ISPJAE), Havana, Cuba.

Bibliographic data

Estrada Domech, Guillermo

An Assessment of Presentation Attack Detection Methods for Face Recognition Systems / Guillermo Estrada Domech; advisor: Raul Queiroz Feitosa; co-advisor: Gilson Alexandre Ostwald Pedro da Costa. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2018.

v., 84 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Detecção de Fraude;. 3. Contramedida;. 4. Sistema Anti-fraude;. 5. Sistema de Reconhecimento Facial. I. Feitosa, Raul Queiroz. II. Costa, Gilson Alexandre Ostwald Pedro. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Acknowledgments

I am truly grateful to my advisors, Prof. Raul Queiroz Feitosa and Prof. Gilson Alexandre Ostwald Pedro da Costa, for the encouragement, their worth advices and talks, their patience, and comprehension throughout the development of my dissertation.

I thank my parents and my sister for their support and unconditional love.

I want to thank all my colleagues from the Computer Vision Lab in Pontifical Catholic University of Rio de Janeiro - PUC-Rio for the companionship and valuable scientific discussion. Especially, I want to thank Pedro Soto for the availability of the code used for the Convolutional Autoencoder (CAE) implementation.

I thank CAPES for the financial support.

Abstract

Estrada Domech, Guillermo; Feitosa, Raul Queiroz (Advisor); Costa, Gilson Alexandre Ostwald Pedro (Co-Advisor). **An Assessment of Presentation Attack Detection Methods for Face Recognition Systems**. Rio de Janeiro, 2018. 84p. Dissertação de mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The vulnerabilities of Face Recognition Systems (FRS) to Presentation Attacks (PA) have been recently recognized by the biometric community, but there is still a lack of generalized software-based facial Presentation Attack Detection (PAD) techniques that perform robustly in realistic authentication scenarios. The main objective of this dissertation is to analyze, evaluate and compare some of the most relevant, state-of-the-art feature-based methods for facial PAD in a variety of conditions, considering three of the facial spoofing databases publicly available 3DMAD, REPLAY-MOBILE and OULU-NPU. In the current work, PAD methods based on LBP-RGB, BSIF-RGB and IQM hand-crafted texture descriptors were investigated. Additionally, a Convolutional Autoencoder (CAE), a learned feature descriptor, was also implemented and evaluated. Furthermore, one-class and two-class classification approaches were implemented and evaluated. The experiments conducted in this work were designed to measure the performance of different PAD schemes in two conditions, namely: (i) intra-database and (ii) inter-database (or cross-database). The results revealed the effectiveness of the features learned by CAE in two-class classification PAD schemes provide, in general, the best performance in intra-database evaluation protocols. The results also indicate that PAD schemes based on one-class classification approach are not inferior as compared to their two-class counterpart in the inter-database evaluations.

Keywords

Presentation Attack Detection; Countermeasure; Antispoofing System; Face Recognition System

Resumo

Estrada Domech, Guillermo; Feitosa, Raul Queiroz; Costa, Gilson Alexandre Ostwald Pedro. **Avaliação de métodos de Detecção de Fraude em Sistemas de Reconhecimiento Facial**. Rio de Janeiro, 2018. 84p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

As vulnerabilidades dos Sistemas de Reconhecimento Facial (FRS) aos Ataques de Apresentação (PA) foram recentemente reconhecidas pela comunidade biométrica, mas ainda existe a falta de técnicas faciais de Detecção de Ataque de Apresentação (PAD) baseadas em software que apresentam desempenho robusto em cenários de autenticação realistas. O objetivo principal desta dissertação é analisar, avaliar e comparar alguns dos métodos baseados em atributos do estado-da-arte para PAD facial em uma variedade de condições, considerando três dos bancos de dados de fraude facial publicamente disponíveis 3DMAD, REPLAY-MOBILE e OULU-NPU. No presente trabalho, os métodos de PAD baseados em descritores de texturas LBP-RGB, BSIF-RGB e IQM foram investigados. Ademais, um Autoencoder Convolucional (CAE), um descritor de atributos aprendidos, também foi implementado e avaliado. Também, abordagens de classificação de uma e duas classes foram implementadas e avaliadas. Os experimentos realizados neste trabalho foram concebidos para medir o desempenho de diferentes esquemas de PAD em duas condições: (i) intra-banco de dados e (ii) inter-banco de dados. Os resultados revelaram que a eficácia dos atributos aprendidos pelo CAE em esquemas de PAD baseados na abordagem de classificação de duas classes fornece, em geral, o melhor desempenho em protocolos de avaliação intra-banco de dados. Os resultados também indicam que os esquemas de PAD baseados na abordagem de classificação de uma classe não são inferiores em comparação às suas contrapartes de duas classes nas avaliações inter-banco de dados.

Palavras-chave

Detecção de Fraude; Contramedida; Sistema Anti-fraude; Sistema de Reconhecimento Facial

Table of contents

List of figures

List of tables

List of Abreviations

AE	Autoencoder
AD	Average Difference
APCER	Attack Presentation Classification Error Rate
BIQI	Blind Image Quality Index
BPCER	Bonafide Presentation Classification Error Rate
BSIF	Binarized Statistical Image Features
CNN	Convolutional Neural Network
CAE	Convolutional Autoencoder
DCT	Discrete Cosine Transform
DL	Deep Learning
DNN	Deep Neural Network
DoG	Difference of Gaussian
DRM	Dichromatic Reflection Model
EM	Expectation-Maximization
FRS	Face Recognition Systems
GME	Gradient Magnitude Error
GMM	Gaussian Mixture Model
GPE	Gradient Phase Error
HDF5	Hierarchical Data Format
HLFI	High-Low Frequency Index
ID	Identifier
IDA	Image Distortion Analysis
IQA	Image Quality Assessment
IQM	Image Quality Measurement
JQI	JPEG Quality Index
LMSE	Laplacian MSE
LBP	Local Binary Patterns
dLBP	directional-coded Local Binary Patterns
mLBP	modified Local Binary Patterns
tLBP	transitional Local Binary Patterns
LBP-TOP	Local Binary Patterns from Three Orthogonal Planes

LFC	Light Field Camera
LLR	Likelihood Ratio
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAS	Mean Angle Similarity
MAMS	Mean Angle Magnitude Similarity
MD	Maximum Difference
MSE	Mean Squared Error
ML	Maximum Likelihood
NAE	Normalized Absolute Error
NIQE	Naturalness Image Quality Estimator
NN	Neural Networks
NK	Normalized Cross-Correlation
PA	Presentation Attack
PAD	Presentation Attack Detection
PAI	Presentation Attack Instrument
PAIS	Presentation Attack Instrument Species
PDF	Probabilities Density Function
PIN	Personal Identification Number
PSNR	Peak Signal to Noise Ratio
RAMD	R-Averaged MD
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RRED	Reduced Ref. Entropic Difference
rPPG	Remote Photoplethysmography
SAE	Sparse Autoencoder
SC	Structural Content
SME	Spectral Magnitude Error
SNR	Signal to Noise Ratio
SPE	Spectral Phase Error
SPP	Spatial Pyramid Pooling
SSIM	Structural Similarity Index
SVM	Support Vector Machine
TCD	Total Corner Difference
TED	Total Edge Difference
VIF	Visual Information Fidelity

1
INTRODUCTION

1.1
Overview

In modern society, the relevance of biometrics recognition systems has been reinforced by the need for the reliable identification of individuals in real-time in many applications, which include forensics, computer security, physical and logical access control and e-commerce, among others [1]. Traditionally, knowledge-based (e.g., passwords or PINs) and token-based (e.g., ID cards or physical keys) mechanisms have been extensively used for identifying individuals and user credentials [2]. However, these mechanisms have showed to be insufficient because they can be easily lost, stolen, shared or manipulated thereby compromising intended security. Moreover, these methods cannot provide crucial functions such as non-repudiation, or multiple instances detection [3].

With the proliferation of web-based services (e.g., online banking, credit card transactions) and surrounded by networks with constant information flow, we have adopted a computerized life that relies on extensive use of smartphones, social media, and cloud computing. Hence, it has become vital to deploy more reliable identification systems that can provide higher degrees of security and stronger authentication schemes. With the advent of biometrics, it is now possible to establish an identity based on "who you are" rather than by "what you possess", such as a physical key, or "what you remember", such as a PIN. Biometric recognition offers a natural and reliable solution to certain aspects of identity management, since it recognizes individuals based on their biological and behavioral characteristics which do not normally change over time (e.g., face, fingerprint, palmprint, iris, palm/finger vein, and voice). In the biometric literature, these characteristics are referred to as *traits*, *indicators*, *identifiers* or *modalities* [1].

Being intrinsically linked to the user, biometric traits can be safely argued to have the unique advantage to truly verify that a person is in fact who he claims to be. In this regard, face biometric offers some advantages: it is natural, easy to use, less human-invasive, non-intrusive data and employs low-cost sensors [4]. Specially, in the context of border control, face recognition has the obvious advantage that the comparison can be conducted with visual evidence in a case of a false-negative decision by the system. In spite of the widespread use of face recognition systems as an alternative solution for conventional identification methods, recent works have revealed its vulnerability to spoofing attacks [5].

Identity theft is an issue that hinders the general adoption of biometrics as an actual form of identification in high-security applications [6]. In contrast to traditional security means, face biometric information is widely available and extremely easy to sample. We cannot claim them to be secret, once our facial images can be captured by surveillance cameras, in a non-intrusive manner at a long distances, or disclosed voluntarily to be shared on social media platforms. Users should not realize that their biometric samples can be dishonestly used. In this new scenario, attackers hack authentication procedures by capturing and replicating facial image samples. These factors have stimulated various researchers to address the challenges of Presentation Attack Detection (PAD), also referred to as antispoofing countermeasures, for facial biometric systems.

It has been suggested in the past the use of multimodal biometrics systems in order to increase authentication security [7]. However, it has been shown in [8] that a multimodal system based on traditional fusion schemes (i.e. Likelihood Ratio (LLR) or weighted sums), can be intrinsically less secure than unimodal ones by spoofing only one of the biometrics (e.g., face trait). Therefore, each biometric trait needs to be taken care of by its own specialized countermeasures.

Today, no matter what security measures are in place, there is no system completely spoof-proof [6]. The identification of counterfeits is a challenging task, especially, in face verification applications. Therefore, face spoofing (or presentation attack) concern should be well solved with high priority before face recognition systems could be widely applied in our daily life as replacement of traditional methods of person authentication in unsupervised environments.

1.2 Motivation

In recent years, a large variety of research in the field of the Presentation Attack Detection (PAD) has been reported in the literature [9–12]. Among the antispoofing techniques available, feature-level methods (usually denoted in the literature as *software-based methods*) for addressing facial presentation attacks at sensor level have received much attention of the biometric community. This is mostly because these approaches are known to be cost-effective, easy to integrate with existing face recognition systems and do not require user cooperation, neither specialized hardware [4].

Face recognition systems, in particular, are known to respond weakly to presentation attacks for a long time [13] and are easily spoofed using one of three categories of counterfeits [14]: (1) a photograph, (2) a video or (3) a 3D model of the enrolled person's face. A considerable number of face PAD approaches have been studied in previous works, and recently proposed methods have achieved good performances over different databases and challenges. In this regard, handcrafted-based techniques, such as Image Quality Measurement (IQM) [14], Binarized Statistical Image Features (BSIF), Local Binary Patterns (LBP) [15], among others, have been widely applied in antispoofing methods. Additionaly, over the past few years, Deep Neural Networks (DNNs) have demonstrated great successes in image representation [16, 17] and has achieved impressive results on face recognition in unconstrained environments [18]. This has motivated some of the latest works to employ a variety of architectures based on Convolutional Neural Network (CNN), Autoencoders (AE), among others; obtaining comparable accuracies and even outperforming previously reported, state-of-the-art methods [19–24].

Another aspect worth mentioning is the availability of large-scale public datasets that contains a variety of spoofing data. Currently, there are twelve face presentation attack databases that comprise most attacking scenarios, namely: NUAA Impostor Database [25], Yale-Recaptured Database [26], Print-Attack Database [27], Replay Video Attack Database [9], CASIA FAS Database [28], MSU-MFSD Database [29], GUC Light Field Face Artifact Database [30], 3D Mask Attack Database [31], MSU-MFD Database [32], REPLAY-MOBILE Database [33], MS-Face Database [34] and OULU-NPU Face Presentation Attack Database [35].

Finally, the diversity of spoofing attacks, including new, and previously unknown biometric artifacts based on novel technologies makes PAD an extremely challenging issue. Traditionally, antispoofing solutions have been developed by formulating the detection problem as a conventional two-class discrimination task (i.e., bonafide versus attack presentation), however, some researchers have reported that face PAD systems using such approaches have failed to generalize across both different datasets and unseen presentation attacks [12, 36].

Even though recently new formulations have been proposed to address the generalization issue, including one-class classification and domain adaptation approaches, a general solution is yet to be found.

1.3 Objectives

The general objective of this dissertation is to compare some of the most relevant state-of-the-art methods for facial Presentation Attack Detection (PAD) in different publicly available facial spoofing databases.
Furthermore, this research pursues the following specific objectives:

1. Evaluate facial PAD techniques based on the combination of handcrafted and learned features, with four classifiers.

2. Develop and evaluate Convolutional Autoencoder (CAE) learned features, and compare them to some of the most relevant state-of-the-art features, in the context of facial PAD.

3. Analyze the performance of the one-class and two-class classification approaches in different facial PAD schemes.

4. Evaluate the performance of the implemented facial PAD techniques considering intra-database and inter-database evaluation protocols.

1.4 Organization of the dissertation

The following parts of this work are structured as follows:

1. Chapter 2 presents the fundamental concepts of the Face Recognition Systems (FRSs) and an overview of their vulnerable points with respect to spoofing attacks. In this chapter some of the most relevant Presentation Attack Detection (PAD) methods for FRSs reported are introduced, with special focus on software-based approaches.

2. Chapter 3 reviews the theoretical background of the existing facial PAD methods assessed in this study.

3. Chapter 4 details the algorithms involved in each stage of workflow for the implemented face PAD methods.

4. In Chapter 5, after giving a brief description of the databases used, the *intra* and *inter-database* evaluation experiments are described, followed by a discussion about the obtained results.

5. Chapter 6 presents the final conclusions and discusses the future directions that could be followed for the extension of this research.

2
RELATED WORKS

This chapter introduces the fundamental concepts of Face Recognition Systems (FRS) and its main vulnerable points to different kind of attacks. In addition, some of the most relevant works related to Presentation Attack Detection (PAD) techniques are presented, with emphasis on software-based methods.

2.1
Face Recognition Systems Under Spoofing Attacks: Vulnerabilities

Since the dawn of the facial biometric technologies, the possibility of recognition subversion systems by determined adversaries has been widely acknowledged [37], since FRS focus on maximizing the discrimination capacity and not in determining whether the presented trait originates from a living legitimate client.

Attacks to biometrics systems can be classified as direct and indirect [38]. Figure 1 shows a block diagram of a typical face recognition system, indicating vulnerable points where possible attacks can occur [37].

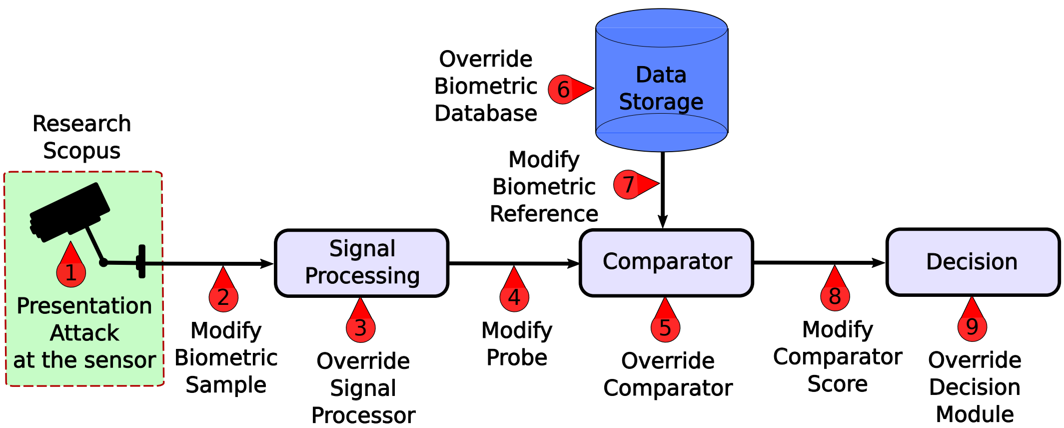

Figure 1: Possible vulnerable points in a FRS (inspired by figure in [37]).

Indirect attacks require that an intruder gains access to the internals of the biometric systems. Once inside, indirect attackers can: tamper feature extractors or comparators (vulnerabilities represented with red arrows 3 and 5 in Figure 1); manipulate trait references (vulnerability represented with red arrow 6 Figure 1); override the decision module to output the intended decision (vulnerability represented with red arrow 9 Figure 1); or exploit possible weak points in communication channels (vulnerabilities represented with red arrows 2, 4, 7 and 8 Figure 1). On the other hand, direct attacks (vulnerability represented with red arrow 1 Figure 1) are carried out at the sensor level and involves presenting a face biometric artifact of the enrolled user as an input to the sensor. An artifact is termed as an artificial object or representation presenting a copy of biometric characteristics or synthetic biometric patterns [37]. This kind of attack is known as a presentation attack: a presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system [37].

A Presentation Attack Instrument (PAI), according to [37], is defined as the biometric characteristic or object used in a presentation attack. The PAIs can be divided into two types: (i) Artificial, which involves artificial means for generating a PAI; and (ii) Human characteristics, which involves using human as PAI. Artificial PAIs can be classified as: (a) complete, which involves the generation of a complete artificial PAI (e.g., a 2D face print, a video of a face, a 3D face mask); and (b) partial, referring to the use of an artificial PAI that can show partial biometric characteristics (e.g., a face video with sunglasses or a partially visible face). Human characteristics PAIs can be classified as (a) lifeless (a cadaver part); (b) altered (the mutation of faces and cosmetic surgery); (c) non-conformant (e.g., the use of facial expression); (d) coerced (e.g., using the face of an unconscious human); and (e) conformant (zero-effort impostor attempts). In addition, PAI species (PAIS) can be termed as the class of presentation attack instruments created using a common production method and based on different biometric characteristics (e.g. printed photo with a laser jet and a printed photo with an inkjet printer as photo print PAI species).

Within the past few years, facial artificial PAIs have been one of the main topics of the biometric community to address vulnerabilities of facial biometric systems [3–5, 35, 39]. Facial PAIs can be easily generated from a photograph of a genuine user who is enrolled in the biometric system. These type of artifacts can be created using: (i) a printed photo with a laser jet [27, 28], (ii) a printed photo with an inkjet printer [30], (iii) an electronic display of photograph or a video of a face [33, 35], or (iv) a 3D facial mask [31].

2.2 Presentation Attack Detection Techniques

Vulnerabilities of FRSs to different types of the aforementioned PAIs have posed a demand to detect and mitigate such attacks in order to improve both the intended security and reliable facial biometric recognition. According to [37], a Presentation Attack Detection (PAD) method (also referred to in the literature as a countermeasure or an antispoofing technique) can be termed an automated determination of a presentation attack. From a general overview, PAD techniques can be classified into two types: (i) hardware- and (ii) software-based methods (as shows Figure 2). The state-of-the-art hardware-based approaches can be divided into three types: *sensor characteristics, blink detection and challenge response.*

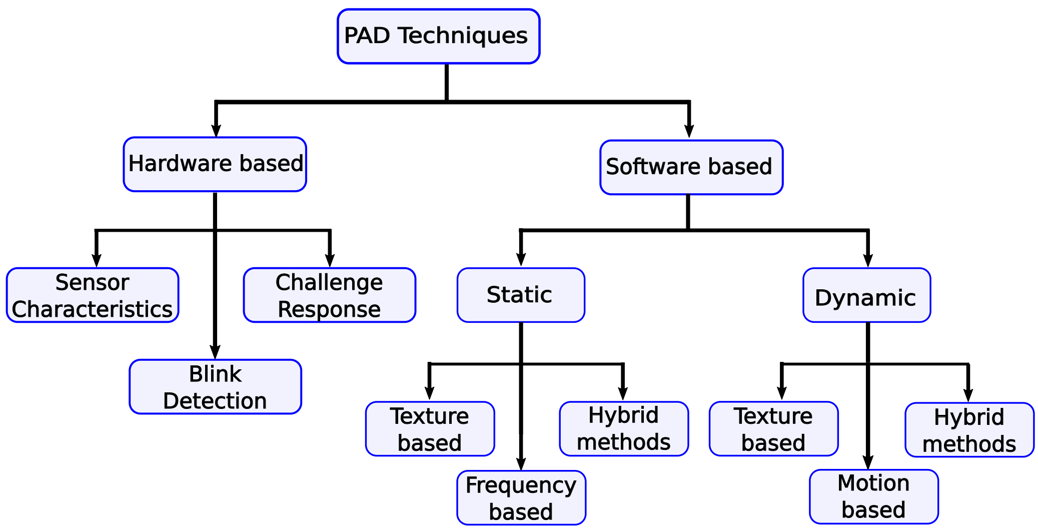

Figure 2: Classification of facial PAD methods (inspired by figures in [4, 38]).

2.3 Hardware-based Presentation Attack Detection Techniques

Hardware-based Sensor Characteristics PAD techniques exploit characteristics of the capture subsystem used to sample the face image (or video). Such characteristics depend on the type of sensor used to capture the face data , for instance: measuring the variation of the focus with a light field camera (LFC) [30]; measuring the reflectance from a near-infrared/thermal/multispectral [34] face sensor; or measuring the reflectance in a 3D scan. In spite such methods present good generalization, they have moderate computational costs and usually rely on expensive sensors.

Blink Detection PAD techniques have been widely employed in liveness detection as a typical countermeasure to spoofing [40, 41], which aims at continuously tracking the spontaneous action of eye blinks that are performed unconsciously by the user. Hence, they present a good effectiveness for display and printed photo attack detection. However, they are associated to high computation costs and are not effective for video replay attacks.

Challenge Response PAD techniques require user cooperation, as their aim is to detect voluntary (behavioral) or involuntary response (reflex reactions) to specific (random) action requirements (challenges or external stimuli), and then analyze the user activity in order to check whether the required action was actually performed (response). For instance, some methods consists on tracking the gaze of the user towards a lighting event (reflex) [41], or the head movement following a random path determined by the system (behavioural) [42]. Although, these techniques show reasonable generalization and good effectiveness for both printed photo and display attacks, they are not effective for replay video attacks, they present some user inconveniences, and they also require a high computation effort and dedicated hardware.

From a general perspective, one may prefer sensor characteristics-based approaches (either by using spectral analysis or a LFC) over methods based on blink detection and challenge response, as the latter techniques have higher computational costs or require a high level of user collaboration; moreover, their performance is limited to tackle the fairly simple photo and display attacks. These constitute key points that have motivated recent research on software-based techniques.

2.4 Software-based Presentation Attack Detection Techniques

Software-based schemes (also known as feature-based approaches) basically involve an algorithm that can discriminate between an attack presentation or a bonafide presentation. Recently reported works have demonstrated the outstanding performances achieved by using them in a variety of scenarios [12, 23, 36]. This aspect together with their well-known cost-effectiveness, easy integration with existing FRSs, and the aforementioned drawbacks of hardware-based techniques have motivated the development of a large number of software-based PAD approaches. In general, software-based techniques can be divided into two types (as shows Figure 2), namely: (i) static approaches, designed to work on a single image without the need for temporal information; and (ii) dynamic approaches, which exploit the temporal information from the video replay presented to a face recognition system.

2.4.1 Software-based Static Face PAD Techniques

In spite of being designed to work on single face images, software-based static PAD techniques can also be applied to video attacks by performing the analysis in a frame-by-frame way and using fusion score techniques (e.g., majority voting) in a later stage to generate a final decision from the combination of individual frame scores. Depending on the nature of the subjacent algorithm, static PAD techniques can be further categorized into three groups: (i) texture-based, (ii) frequency-based, and (iii) hybrid schemes.

Texture-based methods basically consist in the analysis of microtextural patterns of face regions in an image, as it is likely that bonafide faces and fake ones present different texture patterns because of image quality degradation associated to recapturing process, and also because of disparities in surface and reflectance properties. The broadly use of these algorithms have to do with their ability to efficiently discriminate PAI characteristics such as the presence of pigments (due to printing effects), specular reflection, and shades. A representative example is the Local Binary Patterns (LBP) feature extraction method, extensively used to address these issues [15]. Määtä et al. [43] addressed the print photo PAD problem by using three LBP variants: $LBP_{8,1}^{u2}$ (operator in 8 neighborhood pixels located at the circle of radius 1 using uniform patterns), $LBP_{8,2}^{u2}$ (operator in 8 neighborhood pixels located at the circle of radius 2 using uniform patterns), and $LBP_{16,2}^{u2}$ (operator in 16 neighborhood pixels located at the circle of radius 2 using uniform patterns). Feature vectors are created from the concatenation of the respective histograms, and used to classify samples as bonafide or attack presentation. That study was successfully expanded by Chingovska et al. [9], which investigate the effectiveness of the LBP and its extended versions proposed in [44], namely: transitional LBP (tLBP), direction-coded LBP (dLBP), and modified LBP (mLBP); in the detection of replay attacks. Furthermore, Erdogmus and Marcel have examined the effectiveness of LBP and its variants to 3D mask presentation attacks [31]. More recently, Convolutional Neural Networks (CNNs) have been adopted for face PAD schemes. Yang et al. [45] proposed the use of the AlexNet [16] for feature extraction, and Support Vector Machines (SVM) for classification. Lucena et al. [23] showed that pre-trained network based on ImageNet [46] can be successfully transferred to face PAD scenario.

Frequency-based methods exploit the analysis of facial appearance properties by assuming that the disparities between genuine faces and artificial material can be observed in single visual spectra images. The early study carried out by Li et al. [47] describes a method based on the analysis of the 2D Fourier spectrum for detecting face print photo attacks, by assuming that a photograph contains fewer high-frequency components compared to bonafide faces. However, although this method may work well for down-sampled photos, it is likely to fail for higher-quality images. Recently, the same technique was extended by Liu [48] to detect video replay attacks by computing Fourier spectra from the head hair instead of the face. Moreover, other frequency-based features have been used for face PAD, such as Discrete Cosine Transforms (DCTs) [49] and Difference of Gaussian (DoG) filters [28].

Hybrid methods basically involve the combination of various features, such as: Image Quality Measurement (IQM) [14, 29, 50], shape and texture [43], contextual information [51], micro-frequency information (2D Cepstrum) with the Binarized Statistical Image Features (BSIF) descriptor [52], the characterization of the defocus property of the captured face image [53], or the use of client identity information [54].

Summing up, static algorithms are well-known for their outstanding performance over several publicly available facial spoofing databases, and for their low computational cost. Moreover, they are faster as compared to dynamic-based approaches. The Major drawback of static approaches (especially those based on texture analysis) is that rather high resolution input images are required in order to extract the fine details needed for discriminating bonafide from attack presentation samples.

2.4.2 Software-based Dynamic Face PAD Approaches

Dynamic approaches tend to model the temporal information from video replay attacks by exploiting the relative motion frame-by-frame. The existing state-of-the-art methods in the state-of-the-art can be further divided into three types, namely: (i) texture-based, (ii) motion-based, and (iii) hybrid schemes.

Texture-based methods exploit the dynamic texture change across the captured video. A recent study carried out by Pereira et al. [55] extended the analysis of facial microtexture patterns to the spatiotemporal domain by applying LBP over Three Orthogonal Planes (LBP-TOP) [56] for describing specific dynamic events, e.g., facial motion, shaking, and sudden characteristic reflections of planar display media, which might differentiate bonafide faces from fake ones.

The second type of dynamic methods capture the unconscious motion cues particularly exhibited by the muscles in the face due to the movement of the head [42], mouth [57] or eyes [58]. The use of motion vectors based on optical flow to detect unconscious movement of the head was reported in [59]. Motion extraction based on optical flow was also employed in [60] to detect photo attacks by assuming they present a certain measure of unnatural motion, such as swinging and bending.

Finally, hybrid methods involve the analysis of both motion- and texture-based features, by exploiting scenic cues for determining whether a display device is present in the observed scene. Yan et al. [61] introduced the use of multiple scenic clues to address video replay attacks to FRSs. Furhermore, Anjos et al. [62] used context-based motion extraction to differentiate the face from the background. Shao et al. [63] proposed the use of deep convolutional dynamic texture, coupled with a channel-discriminability constraint to distinguish different subtle facial motion patterns between bonafide faces and 3D masks. Feng et al. [64] presented a pre-trained layer-wise Sparse Autoencoder (SAE) that fuses features such as shearlet-based image quality, face motion, and scene motion clues to discriminate between genuine faces and 3D masks. More recently, the ability of Long Short-Term Memory (LSTM) units to find long relations in input sequences has been combined with Convolutional Neural Networks (CNNs) to address the facial PAD issue, showing significant performance improvement when compared to the basic CNN architecture or hand-crafted features [65]. Liu et al. [66] proposed a network architecture that combines a CNN and Recurrent Neural Network (RNN) to estimate the depth of face images and Remote Photoplethysmography (rPPG) signals of face video to discriminate between real and fake faces.

Generally, dynamic-based PAD approaches achieve very competitive performance. However, they cannot be used in FRSs where only a single face image of the user is available (e.g., passport related applications). Moreover, even in scenarios where video data has been recorded (e.g., surveillance applications), it is not rare to find that only a very few non-consecutive frames are suitable for facial analysis, which also limits their final use and accuracy. These aspects have motivated current research to assess the performance of some of the most relevant software-based static PAD techniques reported in the literature, which will be detailed in the next chapter.

3
THEORETICAL FOUNDATIONS

This chapter introduces the main theoretical foundations of the algorithms implemented and evaluated in the current study. The general structure of the PAD system adopted in this research is presented, as well as the algorithms used in each one of its subsystems.

3.1
General Workflow of Face Presentation Attack Detection Schemes

Presentation Attack Detection (PAD) usually involves the steps shown in Figure 3, which are similar to those of a biometric recognition process.

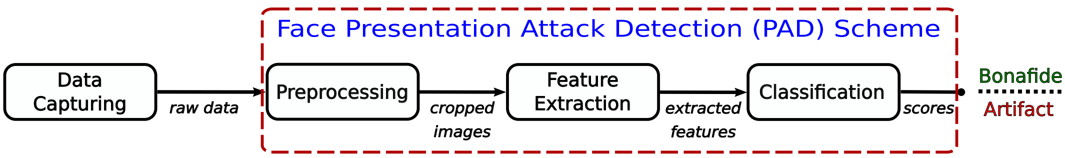

Figure 3: Typical workflow of Presentation Attack Detection.

Firstly, the face image is acquired by the data capture subsystem using a sensor (typically, a camera). The captured image is preprocessed in order to prepare for the following steps. Subsequently, feature extraction is carried out on the preprocessed image. Finally, a classifier is trained to discriminate between bonafide and artifact presentation.

In the next sections, each step of the PAD workflow will be described. Additionally, the fundamentals of the respective algorithms will be explained.

3.2
Preprocessing

Generally, biometric measurements are noisy and contain redundant information that is not necessary for the analysis (e.g., facial images containing non-face background information). The aim of the data preprocessing stage is to clean up the raw facial biometric data so that it is in the best possible state to make recognition or PAD easier. For instance, this stage includes face cropping from the background, photometrical enhancement (face normalization alignment), among others.

3.3
Feature Extraction

Although the preprocessing step produces cleaner biometric data, the resulting data is usually very large and still contains a lot of redundant information. The feature extraction stage involves extracting features that are necessary for recognizing an individual or to discriminate between bonafide or artifact presentation. It is important that reader notes that prior to feature extraction, some algorithms (e.g., deep learning based feature extraction methods) require a training stage (to help the extractor to learn which features to extract) that uses training data provided by a face spoofing database.

3.3.1
Local Binary Patterns (LBP)

Originally designed for texture description, Local Binary Patterns (LBP) operator assigns a label to every pixel of an image by thresholding the 3×3 neighborhood of each pixel with the center pixel value and considering the result as a binary number (as shows equation 3-2). Then, the histogram of the labels can be used as a texture descriptor [15]. This technique was extended by Ojala et al. [67] in order to be able to deal with textures at different scales, by defining the local neighborhood as a set of points evenly spaced on a circle centered at the pixel to be labeled allows any radius R and number of sampling points P.

$$LBP_{P,R}(x_c, y_c) = \sum_{n=1}^{P} \delta(r_n - r_c) \times 2^{n-1} \tag{3-1}$$

where

$$\delta(x) = \begin{cases} 1, & \text{if } x >= 0 \\ 0, & \text{otherwise} \end{cases} \tag{3-2}$$

r_c and $r_n, \forall n = \Big\{1, 2, ..., P\Big\}$ denote the intensity values of the central pixel (x_c, y_c) and its P neighborhood pixels located at the circle of radius $R(R > 0)$, respectively.

The occurrences of the different binary patterns are collected into a histogram to represent the image texture information. Thus, the authors proposed another extension to the original operator as so-called *uniform patterns*. LBP pattern is defined as uniform if its binary code contains at most two transitions from 0 to 1 or from 1 to 0. For example 01110000 (2 transitions) and 00000000 (0 transitions) are uniform patterns. In the literature a common notation for the LPB operator is: $LBP^{u_2}_{P,R}$ which indicates applying the operator in a (P, R) neighborhood, using only uniform patterns (u_2). Figure

4 shows an example of the LBP operator for a neighborhood 8 pixels located at the circle of radius 1.

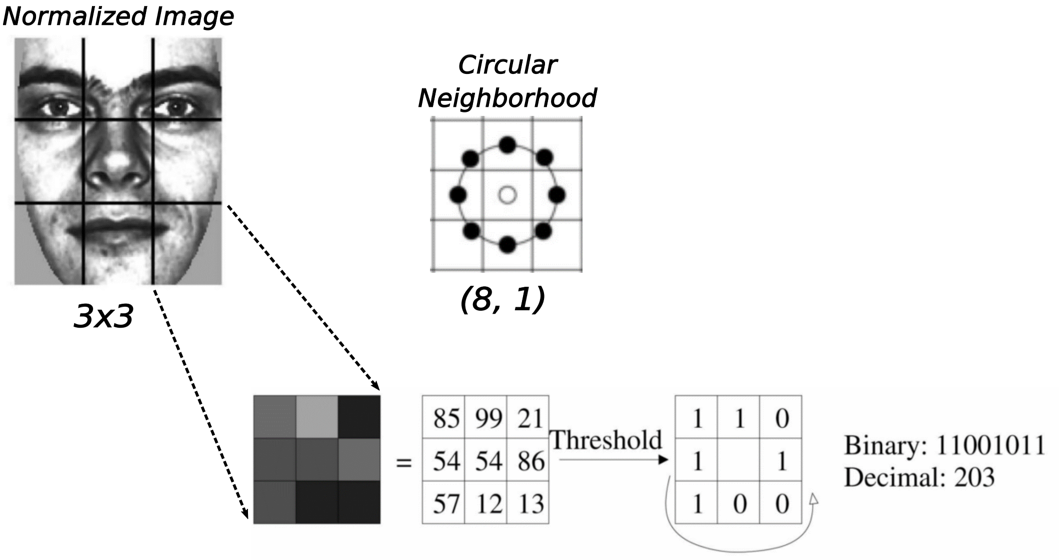

Figure 4: The basic $LBP^{u_2}_{8,1}$ operator in a neighborhood 8 pixels located at the circle of radius 1, modified from [68].

3.3.2 Binarized Statistical Image Features (BSIF)

Similar to the LBP, the idea of the Binarized Statistical Image Features (BSIF) is to represent each pixel as a binary code obtained by performing a convolution operation between the 2D images and a set of filters [69]. The number of the used filters determines the length of the binary code. Thus, given an image patch X of size $l \times l$ pixels and a linear filter W_i of the same size, the filter response s_i is obtained as follows

$$s_i = \sum_{u,v} W_i(u,v)X(u,v) \tag{3-3}$$

where u and v denote the row and column of the image patch and W_i, $i = \Big\{1, 2, ..., n\Big\}$ denote the linear filters. The combined filter response is turn binarized to obtain the binary string (Equation 3-4)

$$b_i = \begin{cases} 1, & \text{if } s_i > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3-4}$$

In order to obtain a statistically meaningful representation of the image data and efficient encoding using simple element-wise quantization, the fixed set linear filters are learned from a set of image patches by maximizing the statistical independence of the filter responses using independent component analysis (ICA). We refer to [69] for further details on BSIF.

Figure 5 shows how is performed the BSIF feature extraction using image patches and linear filter of 9×9 size. In the figure, the symbol "$*$" denotes the convolution operation of the image patch with each of the eight linear filters.

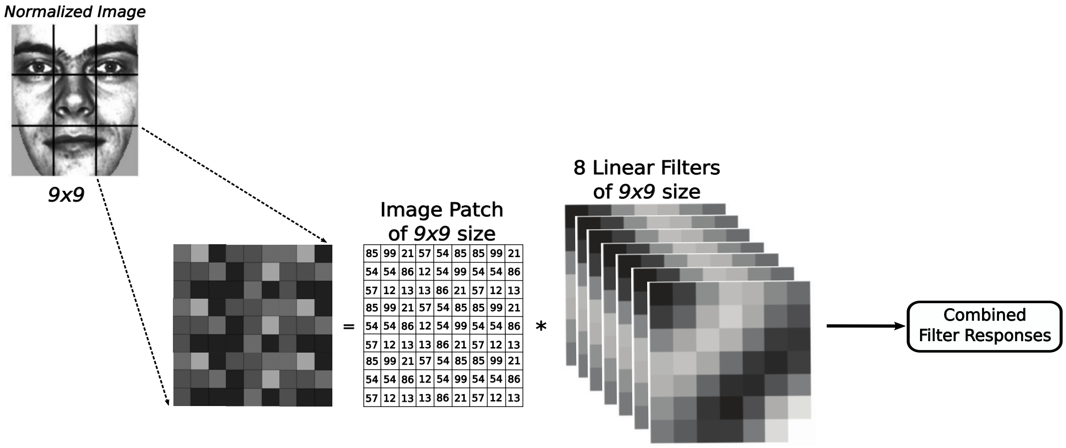

Figure 5: Flow diagram of the BSIF feature extraction using image patches and linear filter of 9×9 size.

3.3.3 Image Quality Measurement (IQM)

Image Quality Assessment (IQA), also referred as Image Quality Measurement (IQM), was first employed by Galbally et al. [14, 50] as a liveness detection method, based on the assumption that a recaptured image has different quality than a real sample, acquired in the normal operation scenario for which the sensor was designed. Expected quality differences between bonafide and fake samples may include: degree of sharpness, color and luminance levels, local artifacts, amount of information found in both types of images (entropy), structural distortions or natural appearance [50]. For instance, face images captured from a mobile device will probably be over- or under-exposed.

Motivated by this different-quality hypothesis, the authors proposed a system that uses a novel parameterization of 25 objective IQMs [50], which provides a quantitative score that describes the level of distortion of the input image. Two types of IQMs are present in the 25-feature set used as discriminative characteristics: Full-Reference and No-Reference.

Full-Reference IQMs (FR-IQMs) rely on the availability of an ideal undistorted reference image against which the quality of a test sample is compared. Since in the case of spoofing attack detection there is no access to such a sample, the authors simulate it by filtering the input image with a low-pass Gaussian kernel ($\sigma = 0.5$ and size 3×3). The first 21 features based on FR-IQMs used in [50] comprise error sensitivity measures, structural similarity measures and information theoretic measures.

No-Reference IQMs (NR-IQMs) (also referred as blind IQMs), unlike FR-IQMs, try to assess the visual quality of images in the absence of a reference by using pre-trained statistical models [70]. Depending on the images used to train this model and on the *a priori* knowledge required, the methods are coarsely divided into one of three groups: distortion-specific approaches, training-based approaches and natural scene statistic approaches. The first 21 FR-IQMs computed in [50] were concatenated with 4 NR-IQMs to created the final 25 image quality feature vector. A general flow diagram of the IQM feature extraction is presented in Figure 6.

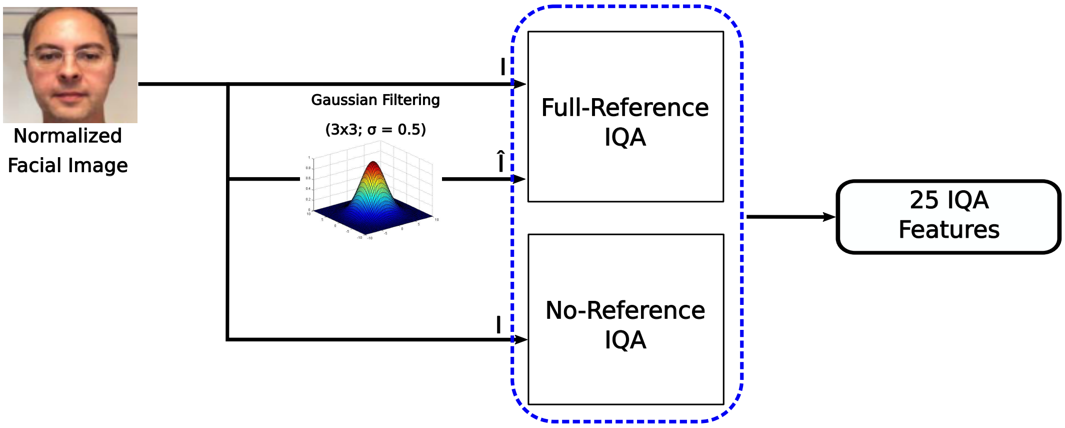

Figure 6: Flow diagram of the IQM feature extraction.

3.3.4
Image Distortion Analysis (IDA)

Image Distortion Analysis (IDA) was proposed by Wen et al. [29] for addressing the face spoofing detection issue. Based on the analysis of Dichromatic Reflection Model (DRM), the authors assume that the major distortions in a face spoof image include: (i) specular reflection from the printed paper surface or a display screen; (ii) image blurriness due to camera defocus; (iii) image chromaticity and contrast distortion due to imperfect color rendering of a printer or display screen; and (iv) color diversity distortion due to limited color resolution of the printer or the display.

Firstly, the specular reflection component is separated from the input face image or video frame by assuming that the illumination is (i) from a single source, (ii) of an uniform color, and (iii) not over-saturated. After computing the specular reflection component image, the specularity intensity distribution is represented with three-dimensional features: (i) specular pixel percentage, (ii) mean intensity of specular pixels, and (iii) variance of specular pixel intensities.

The blurriness features are computed based on two methods: (i) the difference between the original input image and its blurred version, and (ii) the average edge width in the input image. Both methods output a non-reference (without a clear image as a reference) blurriness score between 0 and 1.

Since the absolute color distribution is dependent on illumination and camera variations, the authors proposed invariant features to detect abnormal chromaticity in spoof faces. Chromatic moment features are extracted by computing the mean, deviation, and skewness of each channel from the normalized facial image after converting from the RGB space into the HSV (Hue, Saturation, and Value) space. Besides these three features, the percentages of pixels in the minimal and maximal histogram bins of each channel are used as two additional features. So the dimensionality of the chromatic moment feature vector is 15.

According to the authors, differences between bonafide and spoof faces can be established based on the color diversity. Color diversity features are extracted measuring the image color diversity by first performing a color quantization (with 32 steps in the red, green and blue channels, respectively) on the normalized face image[1]. Two measurements are then joined from the color distribution: (i) the histogram bin counts of the top 100 most frequently appearing colors, and (ii) the number of distinct colors appearing in the normalized face image. The dimensionality of the color diversity feature vector is 101.

Finally, the above four types of feature (specular reflection, blurriness, chromatic moment, and color diversity) are concatenated together, resulting in an IDA feature vector with 121 dimensions, extracted from the facial region containing only image distortion information. A general flow diagram of the IDA feature extraction is presented in Figure 7.

[1]Usually, the face image normalization is part of the preprocessing step performed by PAD techniques or Face Recognition Systems (FRSs), which will be detailed in Chapter 4.

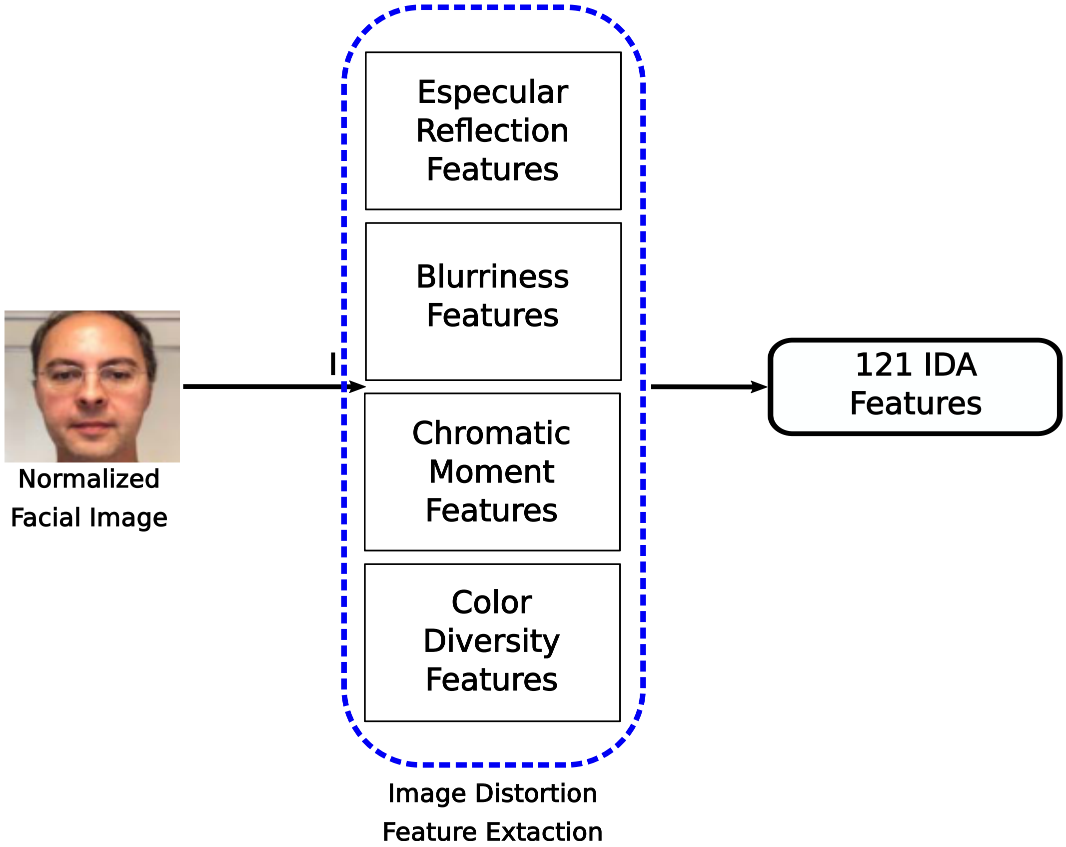

Figure 7: Flow diagram of the IDA feature extraction.

3.3.5
Autoencoders (AEs)

Basically, an auto-encoder is an unsupervised neural network that creates a compact data representation from which the original data can be accurately reconstructed. It usually has two parts: an encoder and a decoder [71], often implemented by a single hidden layer network (as shows Figure 8).

The encoder, denoted as f , maps the input data $x \in \mathbb{R}^d$, to a compact representation $z \in \mathbb{R}^k$ through the activations of the k neurons in the hidden layer, whereby $k < d$. The function f has the form:

$$h = f(x) = s(Wx + \beta) \tag{3-5}$$

where $W \in \mathbb{R}^{k \times d}$ is the matrix containing the learned coefficients of the non-linear transformation, $\beta \in \mathbb{R}^k$ denotes the bias vector and $s(\cdot)$ is the so-called "element-wise activation function", which is usually a non-linear function, such as the sigmoid or the hyperbolic tangent.

The decoder, denoted as g, aims at mapping the representation z back to the input x, formally:

$$\hat{x} = g(z) = s(\hat{W}z + \hat{\beta}) \tag{3-6}$$

where $\hat{W}$ is usually constrained to be equal to W^T and $\hat{\beta} \in \mathbb{R}^d$ the reconstruction bias. The parameters $W, \beta, \hat{W}$ and $\hat{\beta}$ are determined by minimizing the loss function:

$$[W, \beta, \hat{W}, \hat{\beta}] = \min_{W,\beta,\hat{W},\hat{\beta}} \sum_{i=1}^{N} \|x_i - g(f(x_i))\|_2^2 \tag{3-7}$$

where x_i corresponds to the i^{th} out of N training samples. equation 3-7 can be solved by gradient descent methods.

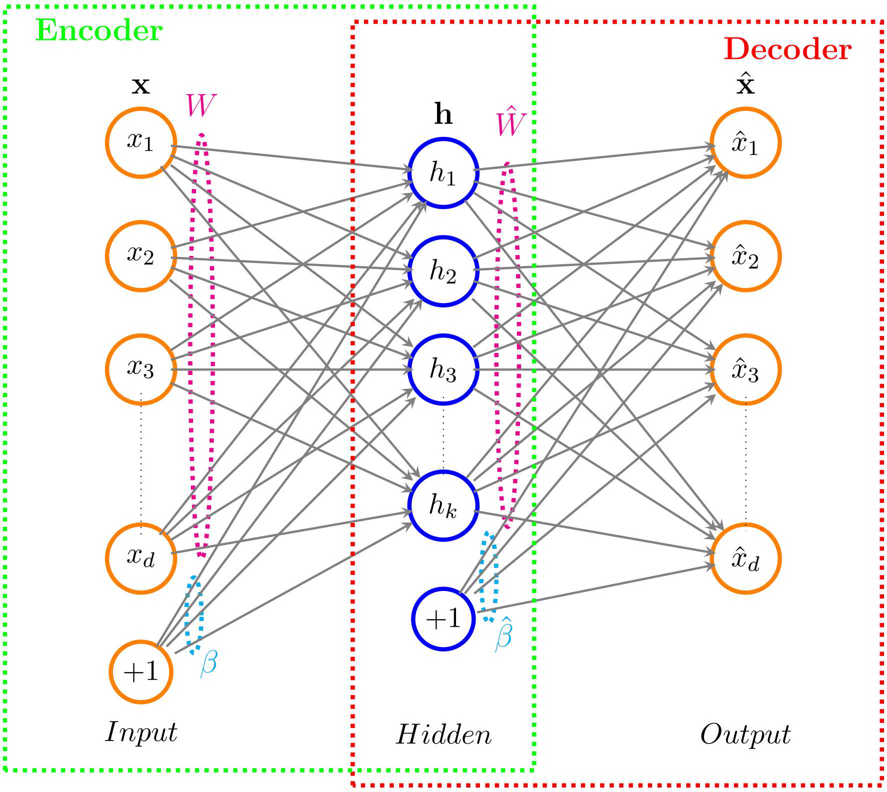

Figure 8: Autoencoder's architecture, example case for input data $\mathbf{x}$.

3.3.6
Convolutional Neural Networks (CNNs)

The Convolutional Neural Networks (CNNs, or ConvNets) is one of the most notable discriminative deep learning approaches where multiple layers are trained in a robust manner [72]. Briefly, the training of the network consists of two stages, namely: (i) forward stage and (ii) backward stage. The main goal of the first stage is to represent the input image with the current parameters (weights and bias) in each layer. Then the loss cost is computed with the ground truth labels by using the prediction output. In the second stage, the gradients of each parameter are calculated with chain rules from the loss cost. The network learning can be stopped after completing sufficient iterations of the forward and backward stages [73].

A typical CNN is composed by many layers with hierarchy including layers for feature representations (or feature maps) whose convolutional layers alternate with pooling layers, followed by some fully connected layers (as shows Figure 9).

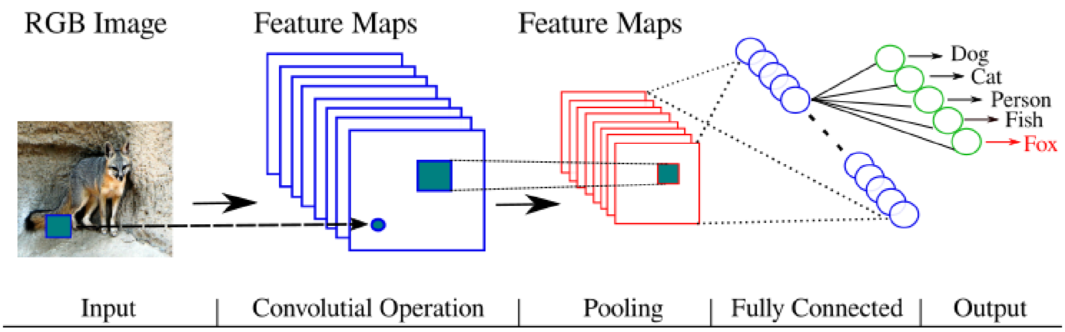

Figure 9: An overview of a typical CNN architecture.

Convolutional layer utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. The value of each unit in a feature map is the result of convolution operation between the learned filter and the local region of the image, called receptive field. This is evaluated by a nonlinear activation function:

$$y_j^{(l)} = s(\sum_{i=1} w_{ij} * x_i^{(l-1)} + b_j) \tag{3-8}$$

where $y^{(l)}$ is the j^{th} output for the l^{th} convolution layer C_l; $s(\cdot)$ is a nonlinear function like *sigmoid*, *tanh* and *Rectified Linear Unit (ReLU)*. The symbol $*$ represents a discrete convolution operator and b_j is a bias. Note that each filter w_{ij} can connect to all or a portion of feature maps in the previous layer.

Pooling layer follows the convolutional layers in order to reduce the dimensions of feature maps and network parameters. For example, given a 8×8 feature map, the output map is reduced to 4×4 dimensions, with a pooling strategy which has size 2×2 and stride 2. The pooling layer reduces the spatial resolution of the feature map, thus providing some level of distortion invariance, because their computations take neighboring pixels into account [74]. Although, average pooling and max pooling are the most commonly used strategies, alternatives such as stochastic pooling [75], Spatial Pyramid Pooling (SPP) [76] and def-pooling [77].

Fully-connected layer is usually located following the last pooling layer in the network, as seen in Figure 9. The main goal of the fully-connected layer is to convert the 2D feature maps into a 1D feature vector, for further feature representation. It enables to feed forward the neural network into a vector with a pre-defined length. It is possible to either feed forward the vector into a certain number categories for image classification [16] or to take it as a feature vector for subsequent processing.

Due to the large number of parameters introduced in the deep architectures a commom problem that can occur during training is overfitting. In addition to the stochastic pooling mentioned above, which can be used to address the overfitting issue, some regularization techniques have been proposed in order to improve the training performance.

For instance, the dropout technique [78] prevents complex co-adaptations on the training data and enhance the generalization ability by randomly omitting a percentage of the feature detector (or neurons) during each training phase. Furthermore, the data augmentation technique has been used when CNN is applied to visual object recognition in order to generate additional samples, without introducing extra labeling costs (e.g., image translations, reflections or even modifications of the intensities of the RGB channels in training images) [16].

On the other hand, the transfer learning technique [79] can be used to apply previously learned knowledge of a relevant visual recognition problem to a new, desired task domain. Depending on the size and similarity between the pre-training database and the new dataset, transfer learning can be applied in two different approaches: (i) by fine-tuning the pre-trained network weights using the new dataset via backpropagation, or (ii) by directly utilizing the learned weights in the desired problem to extract and later classify features [79].

3.3.7 Convolutional Autoencoder (CAE)

Fully connected AEs and its variants[2] (Sparse Autoencoder, Denoising Autoencoder, etc.) ignore the 2D image structure. This is not only a problem when dealing with realistically sized inputs but also introduces redundancy in the parameters, forcing each feature to be global (i.e., to span the entire visual field) [81]. CAEs address the filter definition task by letting the model learn the optimal filters that minimize the reconstruction error. Once these filters have been learned, they can be applied to any input in order to extract features. These features, then, can be used to do any task that requires a compact representation of the input, like classification.

The main characteristic of CAE is that this kind of model shares weights among all locations in the input, preserving spatial locality [73]. In comparison to CNNs, they are trained only to learn filters able to extract features that

[2]More details about Deep Learning-based achitectures can be found in [73, 80].

can be used to reconstruct the input. For a mono-channel input x the latent representation of the k^{th} feature map is given by

$$h^{(k)} = s(x * W^k + \beta^k) \tag{3-9}$$

where $s(\cdot)$ is an activation function (*sigmoid*, *tanh*, etc.), $*$ denote the 2D convolution operation and the bias is broadcasted to the whole map. The reconstruction is obtained by

$$y = s(\sum_{k \in H} h^k * \hat{W}^k + c) \tag{3-10}$$

where there is one bias c per input channel; H identifies the group of latent feature maps; $\hat{W}$ identifies the flip operation over both dimensions of the weights. The parameters are optimized, minimizing an appropriate cost function over the training set (similar procedure as aforementioned for AE section). A typical CAE architecture is presented in Figure 10.

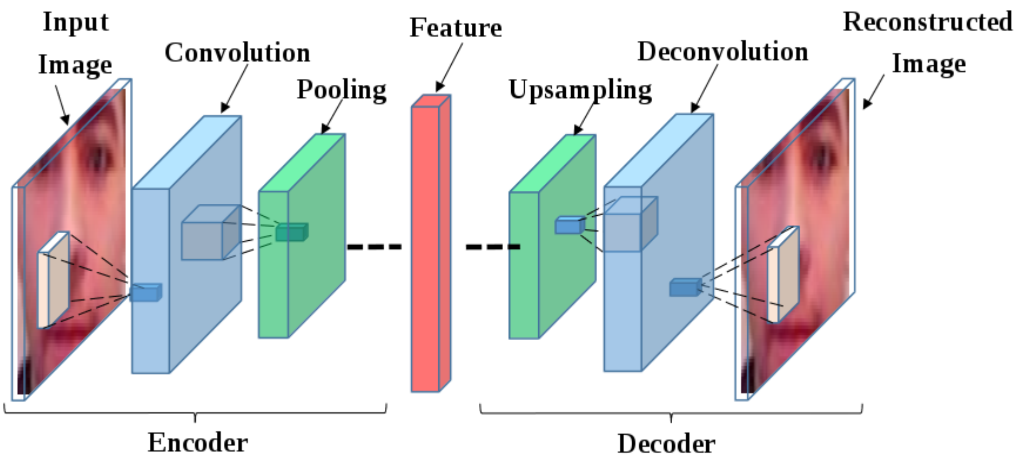

Figure 10: Typical Convolutional Autoencoder (CAE) architecture.

In CAE models, deconvolutional layers (also known upsampling layers) are often used during the reconstruction process (or decode stage). In the encoding stage of a CAE model, the data goes through several convolutional and pooling layers resulting in the feature maps with smaller sizes. This process is followed by a decoding stage that reconstructs the input data making use of deconvolution operations. In practice, the deconvolution operation implements a transposed convolution operator and can be seen as a convolutional layer with backward and forward passes inverted [82]. The transpose convolution relocates the activations of the previous layer in the upsampled grid and performs a convolution for end-to-end learning by backpropagation from the pixelwise loss. Figure 11 shows the transpose of convolving a 3×3 kernel over a 5×5 input padded with a 1×1 border of zeros (known as zero-padding operation) using 2×2 strides, which it is equivalent to convolving a 3×3 kernel over a 3×3 input (with one zero inserted between inputs) padded with a 1×1 border of zeros using unit strides.

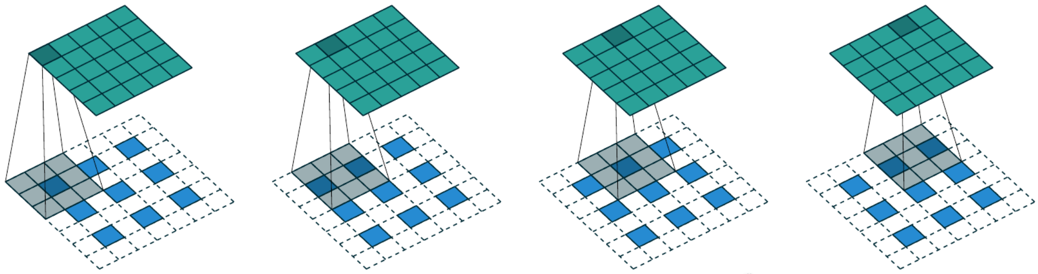

Figure 11: Principles of the transposed convolution (deconvolution or upsampling) operation (taken from [82]).

3.4 Classification

This stage involves the comparison between the features extracted by the PAD algorithm with a stored PAD criteria [37]. These criteria may be common for all subjects or specific to each subject. For instance, when involuntary reactions, physiological functions, voluntary reactions or subject behaviours are used to detect presentation attacks, the presentation-attack criteria may be common for all subjects if they are measured roughly, while the criteria may be specific to each subject if they are measured precisely. Generally, this stage requires a classifier training which produces a score for each probe sample. Then, the provided score will be used to discriminate between bonafide and artifact presentation.

3.4.1 Support Vector Machine (SVM)

Originally proposed by Vapnik [83, 84], Support Vector Machines (SVMs) are a popular set of supervised learning methods for classification, regression, and distribution estimation (also known as outliers detection). (so-called functional margin). As the feature space may have a high dimension (which results in very expensive to compute), it is common to apply a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$, that can be evaluated efficiently. The most used kernel functions include:

- linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^T \mathbf{x}_j$
- polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)^d, \gamma > 0$
- Radial Basis Function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2), \gamma > 0$

– sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)$

During the training process, γ, d and r represent the kernel parameters to be learned.

3.4.2
C-Support Vector Classification (C-SVC)

C-Support Vector Classification (C-SVC) is one of the formulations of SVM applied to two-class or multi-class classification task. In two-class classification, given a training vector $\mathbf{x}_i \in \mathbb{R}^n$, $\forall i = \{1, 2, ..., l\}$, where $x \in \mathbb{R}^n$, $y \in \{1, -1\}$ and l represents the number of support vectors, a SVM classifier is constructed from the sums of kernel functions of the form:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \leq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \tag{3-11}$$

where the training vectors $\mathbf{x}_i$ are mapped into a higher dimensional space by the function ϕ, in order to find a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 1$ is the penalty parameter of the error term.

An important fact is the use of kernel functions which, depending on its nature, allows SVMs to construct hyperplanes that correspond to a nonlinear decision function in input space. The nonlinear decision function can take the form:

$$S(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{3-12}$$

where α_i (Lagrange multiplier associated with the i^{th} support vector), y_i (the corresponding classification label class, i.e., in the particular case of face PAD issue, $y_i = +1$ if $\mathbf{x}$ belongs to a bonafide presentation and $y_i = -1$ if $\mathbf{x}$ belongs to the fake presentation) and b (learned constant), represent the kernel function parameters.

Schölkopf et al. [85] introduced the ν-Support Vector Classification (ν-SVC) as a reparameterization of the C-SVC formulation. The proposed formulation introduces a new parameter ν, which controls the number of support vectors and training errors. The parameter $\nu \in (0, 1]$ has an upper bound on the fraction of training errors and a lower bound on the fraction of support vectors.

3.4.3
Distribution Estimation (one-class SVM)

Several applications require being able to decide whether a new observation belongs to the same distribution as existing observations (referred to as an inlier), or should be considered as different (referred as an outlier). These outliers in the data can be caused by errors in the measurement of feature values, resulting in an exceptionally large or small feature value in comparison with other training objects. Instead of modeling the density of data, however, this approach aims to find a smooth boundary enclosing a region of high density. In order to address this issue, two approaches have been proposed by using SVMs.

The method proposed by Schölkopf et al. [86] tries to find a hyperplane that separates all but a fixed fraction ν of the training data from the origin, at the same time maximizing the distance (margin) of the hyperplane from the origin. Then, if further observations lay within the frontier-delimited subspace, they are considered as coming from the same population than the initial observations. Otherwise, if they lay outside the frontier, we can say that they are abnormal This approach is used for detecting anomalies in new observations by assuming that training data is not polluted by outliers. In this regard, one of the kernel usually chosen for this approach is the Radial Basis Function (RBF).

3.4.4
Gaussian Mixture Model (GMM)

A Gaussian mixture model (GMM) is a probabilistic model for density estimation, which assumes the feature vectors follow a Gaussian distribution [87, 88]. Generally, the GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm [89], which can guarantee monotonic convergence to the set of optimal parameters (in the Maximum-Likelihood sense). A Gaussian Mixture Model can be expressed as a weighted sum given K component densities (or mixtures) [88],

$$p(\mathbf{x}|\lambda) = \sum_{k=1}^{K} \varrho_k \mathcal{N}(\mathbf{x}|\mu_k, \xi_k) \tag{3-13}$$

where $\mathbf{x} \in \mathrm{R}^d$ is an input data vector (e.g., feature vector), ϱ_k are the mixture weights that satisfies $\sum_{k=1}^{K} \varrho_k = 1$ and $\mathcal{N}(\mathbf{x}|\mu_k, \xi_k)$ represents the component Gaussian densities, which can be formulated for a d-variate Gaussian function as follows,

$$\mathcal{N}(\mathbf{x}|\mu_k, \xi_k) = \frac{1}{(2\pi)^{d/2}|\xi_k|^{1/2}} e^{-\frac{1}{2}(\vec{\mathbf{x}_k}-\vec{\mu_k})^T \xi_k^{-1} (\vec{\mathbf{x}_k}-\vec{\mu_k})} \tag{3-14}$$

with mean vector $\mu_k \in \mathrm{R}^d$ and covariance matrix $\xi_k \in \mathrm{R}^{d\times d}$. From the K mixtures, a GMM can be parameterized by the mean vectors, covariance matrices and mixture weights as follows,

$$\lambda\Big\{\varrho_k, \mu_k, \xi_k\Big\}, \quad \forall_k = \Big\{1, 2, ..., K\Big\} \tag{3-15}$$

3.4.5
Anomaly Detection (one-class GMM)

Basically, the task of clustering consist in assigning a number of points, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$, into K groups or clusters. In the last years, GMMs became one of the most popular clustering algorithms [88]. This approach is employed under the assumption that the points, which belong to the same cluster, are distributed according to the same Gaussian distribution of unknown mean and covariance matrix. Each mixture component defines a different cluster. To accomplish this task, the EM algorithm is run over the available data points to provide, after convergence, the posterior probabilities $p(k|\mathbf{x}_n), k = 1, 2, ..., K, n = 1, 2, ..., N$, with k corresponding to a cluster and n to a data point. Each point can be assigned to a cluster k as follows,

$$\text{assign } \mathbf{x}_n \text{ to cluster } k = \arg\min_i p(i|\mathbf{x}_n)\forall_i = \Big\{1, 2, ..., K\Big\}.$$

The one-class GMM approach can be trained with a dataset containing only real samples (i.e., samples belonging to the target class, in our case bonafide presentations) following the procedure described above. If the computed probability is below a threshold, the sample is considered a fake presentation, therefore, probability means that is not probable that a given presentation is genuine.

3.4.6
Logistic Regression (LR)

In Bayesian classification, the assignment of a pattern to a class is performed based on the posterior probabilities, $P(C_i|\mathbf{x})$. These posteriors are estimated via the respective conditional Probabilities Density Functions (PDFs). However, an alternative way to directly model the posterior probabilities is using the Logistic Regression (LR) method. In the two-class LR case, the ratio of posteriors is formulated as,

$$\ln \frac{P(C_1|\mathbf{x})}{P(C_2|\mathbf{x})} = \mathbf{w}^T\mathbf{x}. \tag{3-16}$$

Taking into account that $P(C_1|\mathbf{x}) + P(C_2|\mathbf{x}) = 1$, the posterior probability of class C_1 can be written as a logistic sigmoid acting on a linear function of the feature vector $\mathbf{w}$, so that:

$$P(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) \tag{3-17}$$

with $P(C_2|\mathbf{x}) = 1 - P(C_2|\mathbf{x})$ and $\sigma(\cdot)$ representing the *logistic sigmoid* or *sigmoid link function* defined by

$$\sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 - e(-\mathbf{w}^T\mathbf{x})} \tag{3-18}$$

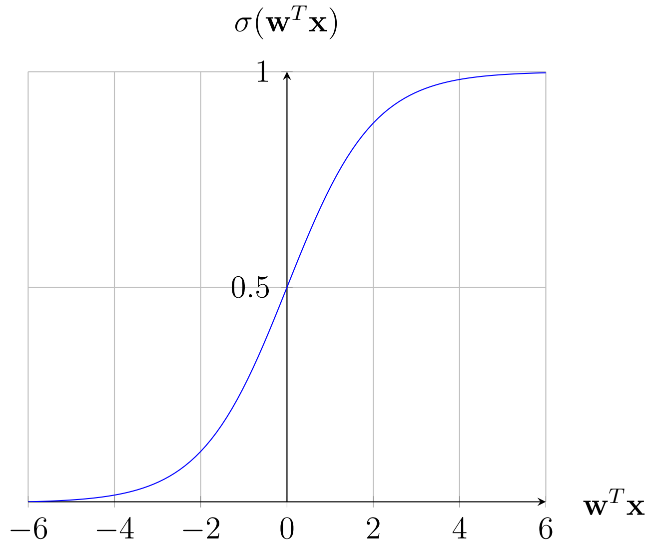

Figure 12: Plot of the logistic sigmoid function $\sigma(\cdot)$ defined in equation 3-18.

For a set training samples $(\mathbf{x}_n, y_n)$, with $n = 1, 2, ..., N, y_n = \Big\{0, 1\Big\}$, the parameter vector, $\mathbf{w}$, can be estimated via the Maximum Likelihood (ML) method. The likelihood function can be formulated as

$$P(y_1, ..., y_N; \mathbf{w}) = \prod_{n=1}^{N} (\sigma(\mathbf{w}^T\mathbf{x}_n))^{y_n}(1 - \sigma(\mathbf{w}^T\mathbf{x}))^{1-y_n} \tag{3-19}$$

Usually, an error function can be defined by taking the negative logarithm of the likelihood (log-likelihood), which give the *cross entropy* error function in the form:

$$L(\mathbf{w}) = -\sum_{n=1}^{N}[y_n \ln \sigma_n(\mathbf{w}^T\mathbf{x}_n) + (1 - y_n)\ln(1 - \sigma_n(\mathbf{w}^T\mathbf{x}_n))] \tag{3-20}$$

It is worth noting that ML can exhibit severe overfitting for data sets that are linearly separable. This arises because the maximum likelihood solution occurs when the hyperplane corresponding to $\sigma = 0.5$, equivalent to $\mathbf{w}^T\mathbf{x} = 0$ (see Figure 12), separates the two classes and the magnitude of $\mathbf{w}$ goes to infinity. To deal with this issue, it is common to include a penalty term $(||\mathbf{w}||^2)$ in the respective cost function, redefined in the form:

$$L(\mathbf{w}) = -\sum_{n=1}^{N}[y_n \ln \sigma_n(\mathbf{w}^T\mathbf{x}_n) + (1-y_n)\ln(1-\sigma_n(\mathbf{w}^T\mathbf{x}_n))] + \frac{\lambda}{2}||\mathbf{w}||^2 \quad (3\text{-}21)$$

where λ is a new hyperparameter added to control the regularization strength.

4
METHODS

This chapter describes the steps involved in the methodology followed to accomplish the three goals of this work. At firstly, implementation details of the workflow adopted will be discussed, by focusing on the settings of the PAD methods that will be evaluated in Chapter 5.

4.1
Evaluation Methodology

The PAD techniques evaluated in the current study are composed of three main steps, namely: (i) preprocessing, (ii) feature extraction, and (iii) classification. Figure 13 shows the workflow adopted for all the PAD techniques considered in this work. The PAD methods were implemented with the Facial Presentation Attack Detection Library[1]. This software package is provided by Bob[2] [90], a free signal-processing and machine learning toolbox originally developed by the Biometrics group at IDIAP Research Institute, Switzerland. The toolbox is written in Python and C++ and is designed to be efficient and reduce development time. It is composed of a reasonably large number of methods for image, audio and video processing, machine learning, pattern recognition, and a lot more task-specific packages.

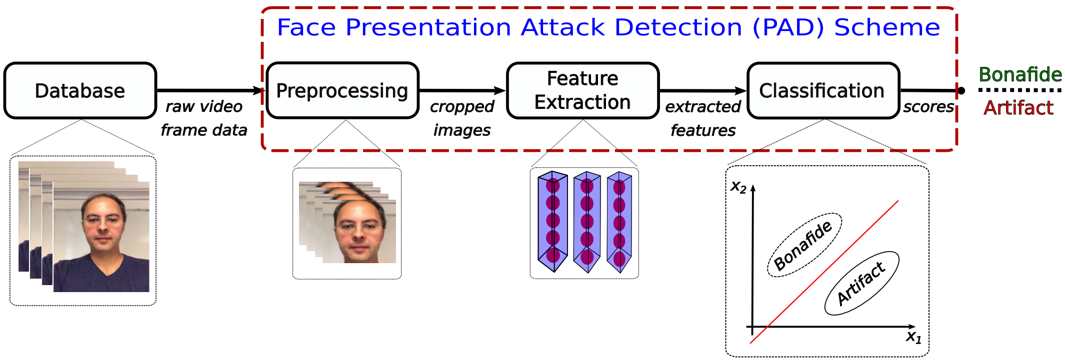

Figure 13: Workflow adopted for all the PAD methods evaluated in this work.

The communication between any two steps in the PAD workflow is file-based by using a binary Hierarchical Data Format (HDF5) interface version 5.

[1]Implements tools for spoofing or presentation attack detection in face biometrics.
[2]Bob is a free signal-processing and machine learning toolbox repository.

The exception is the classification step, which uses score files in text format. As the implementation of some of the descriptors (features) used in the PAD methods are not available in the Bob Framework, implementations provided by authors were used.

Implementation details of each stage of the workflow will be given in the next sections, as well as the settings of the PAD methods that will be assessed in the experiments reported in the Chapter 5.

4.1.1 Preprocessing

Basically, the same preprocessing step was carried out for all evaluated PAD techniques in order to assess them under equal conditions. Since the biometric samples provided by most of facial PA databases considered in this study are videos, the preprocessing stage involved the extraction of faces on a frame-by-frame basis from the annotations defining the facial region. The frames with a face smaller than 50×50 pixels were discarded. Similar to prior works [12, 14, 31, 35], the cropped images are the then normalized to the identical size of 64×64 pixels.

Figure 14 shows some facial images from the three sessions in which the 3DMAD database was collected after the preprocessing stage. The first two rows represent bonafide accesses (from top to down sessions 1 and 2 in which data was collected, respectively), and the third row represents mask attacks (session 3).

Figure 14: Facial images from 3DMAD Database after the preprocessing stage.

4.1.2 Feature Extraction

In the current work, Local Binary Patterns (LBP), Binarized Statistical Image Features (BSIF) and Image Quality Measurement (IQM) hand-crafted texture descriptors were used as input of the classification step. Additionally, a Convolutional Autoencoder (CAE), a learned feature descriptor, was also implemented, and used as input to the classification step. All feature descriptors were computed from facial images in RGB color space delivered by the preprocessing step.

The LBP feature considered in this work was obtained by a $LBP^{u2}_{8,1}$ operator. LBP features for all pixels in the image were computed, and from those values a single histogram was produced (per-image calculated features).

Contrary to the approach of dividing the image into blocks and calculating LBP histograms for each of the blocks separately to form a final feature vector by their concatenation (per-block computed features), which is a common procedure for facial recognition, the LBP variant used in this work (per-image calculated features) has been successfully applied to address the face PAD issue [9, 12, 35, 91, 92], achieving better performance than the per-block approach.

Considering the total number of bins in the histogram per channel of the RGB image, the number of dimensions of the feature vector is $177 =$ (59 bins in the histogram) $\times$ (3 channels). In this work, a modified version of the code[3] publicly provided by Boulkenafet et al. [91, 92] was used for extracting LBP features.

The image quality based technique assessed in this work uses a feature vector obtained from the concatenation of the IQMs introduced in [50] and the IDA-based features proposed in [29]. Note that Galbally et al. [50] proposed 25 quality measures in their paper. However, Bob framework only implements 18 of the features (listed in Table 1) described in the paper. Additionally, the image distortion-based features (specularity, image-blur, color-diversity) proposed by Wen et al. [29] were computed to obtain a 121-D feature vector. The IQM-based PAD implemented in this study uses a combination of those two sets of features, which results in a 139-D feature vector.

In the BSIF based PAD scheme, the feature vectors were obtained using eight filters of size 7×7. In a way similar to [91, 92], in the experiments reported in section 5.3, the set of filters provided by the authors of [69], which were learned from a set of natural image patches, were used. The final number of dimensions of the feature vector is 768. In this work, a modified version of

[3]LBP implementation is publicly available at Boulkenafet's repository.

the code[4] provided by Boulkenafet et al. [91, 92] was used for extracting BSIF features.

Table 1: Image quality measures adopted in [50]. (FR denotes Full-Reference-based approaches while NR stands for no-reference approaches).

Attribute	**Approach**	**Name**
1	FR	Mean Squared Error (MSE)
2	FR	Peak Signal to Noise Ratio (PSNR)
3	FR	Average Difference (AD)
4	FR	Structural Content (SC)
5	FR	Normalized Cross-Correlation (NK)
6	FR	Max. Difference (MD)
7	FR	Laplacian MSE (LMSE)
8	FR	Normalized Absolute Error (NAE)
9	FR	Signal to Noise Ratio (SNR)
10	FR	R-Averaged MD (RAMD)
11	FR	Mean Angle Similarity (MAS)
12	FR	Mean Angle Magnitude Similarity (MAMS)
13	FR	Spectral Magnitude Error (SME)
14	FR	Gradient Magnitude Error (GME)
15	FR	Gradient Phase Error (GPE)
16	FR	Structural Similarity Index (SSIM)
17	FR	Visual Information Fidelity (VIF)
18	NR	High-Low Frequency Index (HLFI)

The CAE[5] architecture adopted in this work was implemented with the Keras Library [93] and Tensorflow as backend [94]. The architecture is summarized in Table 2.

The CAE was composed of 9 convolution operations (4 in the encode and 5 in the decode), 3 max-pooling (in the encode) and 3 up-sampling (in the decode). For each convolution operation a ReLU activation function was used, and zero padding was applied to obtain the same dimensions in the output feature maps. A dropout regularization strategy of 50 % was used to address the overfitting issue, and the Adadelta optimizer [95] was used in the training procedure. The loss function was based on the Mean Squared Error (MSE). Finally, the trained model was used as a feature extractor, and a 4096-dimensional feature vector was obtained by reshaping the $8 \times 8 \times 64$ feature map of the third max pooling operation in the encoding stage of the model.

[4]BSIF implementation is publicly available at Boulkenafet's repository.
[5]CAE code available under request at PUC-Rio Computer Vision Laboratory (LVC) website.

Table 2: The structure of CAE implemented in this study. The input and output sizes are described in $(rows \times cols \times \#filters)$. The kernel is specified as $rows \times cols \times \#filters, stride$.

Layer	size-in	size-out	kernel	parameters
conv2d_1	(64, 64, 3)	(64, 64, 8)	$5 \times 5 \times 8, 1$	608
conv2d_2	(64, 64, 8)	(64, 64, 16)	$5 \times 5 \times 16, 1$	3216
pooling2d_1	(64, 64, 8)	(32, 32, 16)	$2 \times 2 \times 16, 2$	0
conv2d_3	(32, 32, 16)	(32, 32, 32)	$5 \times 5 \times 32, 1$	12832
pooling2d_2	(32, 32, 32)	(16, 16, 32)	$2 \times 2 \times 16, 2$	0
conv2d_4	(16, 16, 32)	(16, 16, 64)	$1 \times 1 \times 64, 1$	2112
pooling2d_3	(16, 16, 64)	(8, 8, 64)	$2 \times 2 \times 16, 2$	0
up_sampling2d_1	(8, 8, 64)	(16, 16, 64)	$2 \times 2 \times 64, 2$	0
conv2d_5	(16, 16, 64)	(16, 16, 64)	$5 \times 5 \times 64, 1$	102464
up_sampling2d_2	(16, 16, 64)	(32, 32, 64)	$2 \times 2 \times 64, 2$	0
conv2d_6	(32, 32, 64)	(32, 32, 32)	$3 \times 3 \times 32, 1$	18464
up_sampling2d_3	(32, 32, 32)	(64, 64, 32)	$2 \times 2 \times 32, 2$	0
conv2d_7	(64, 64, 64)	(64, 64, 16)	$3 \times 3 \times 16, 1$	4624
conv2d_8	(32, 32, 64)	(64, 64, 8)	$1 \times 1 \times 8, 1$	136
conv2d_9	(64, 64, 8)	(64, 64, 3)	$1 \times 1 \times 3, 1$	27
Total				**144483**

4.1.3 Classification

Most studies in the field of face PAD consider the task as a two-class classification problem. In this case, the two-class classifier was trained to predict the class of the input samples as bonafide or artifact. Some recent studies are, however, based on one-class classification,such as the works by Arashloo et al. [36] and Nikisins et al. [12]. In these works, the one-class classifiers are trained solely on bonafide samples either to model the probability density of data or to find a smooth boundary enclosing a region of high density.

In the current work, both one-class and two-class approaches are investigated. Support Vector Machine (SVM) and Logistic Regression (LR) are used in the two-class classification approach, while one-class GMM and one-class SVM models are employed in the one-class approach. A list of the classification schemes evaluated in the experiments is shown in Table 3, in which the top rows correspond to the two-class approach, whereas the bottom rows refer to the one-class approach. The implementation of the four classifiers were done with the Bob Framework [90].

In the case of the one-class classifiers, one-class GMM corresponds to a generative model [36], while one-class SVM is regarded as a discriminative model [36]. The output score of the one-class GMM is a log-likelihood, whereas the output of the one-class SVM is a confidence score, similar to that obtained with the LIBSVM [96]. In fact, Bob's SVM implementation is based on LIBSVM and offers different options such as kernel type, multiclass classification and cross-validation. In our experiments, we used the RBF kernel function because it can handle the case when the relation between class labels and attributes is nonlinear; it has fewer hyperparameters than the polynomial kernel.

Table 3: Presentation Attack Detection (PAD) schemes assessed in the current work.

Attribute	**Name**
LR+IQM	The Logistic Regression classifier trained using the IQM features.
SVM2+IQM	The two-class SVM classifier trained using the IQM features.
LR+LBP	The Logistic Regression classifier trained using the LBP features.
SVM2+LBP	The two-class SVM classifier trained using the LBP features.
LR+BSIF	The Logistic Regression classifier trained using the BSIF features.
SVM2+BSIF	The two-class SVM classifier trained using the BSIF features.
LR+CAE	The Logistic Regression classifier trained using the CAE features.
SVM2+CAE	The two-class SVM classifier trained using the CAE features.
SVM1+IQM	The one-class SVM classifier trained using the IQM features.
GMM1+IQM	The one-class GMM classifier trained using the IQM features.
SVM1+LBP	The one-class SVM classifier trained using the LBP features.
GMM1+LBP	The one-class GMM classifier trained using the LBP features.
SVM1+BSIF	The one-class SVM classifier trained using the BSIF features.
GMM1+BSIF	The one-class GMM classifier trained using the BSIF features.
SVM1+CAE	The one-class SVM classifier trained using the CAE features.
GMM1+CAE	The one-class GMM classifier trained using the CAE features.

The one-class GMM is set with 50 gaussians, as preliminary experiments demonstrated that working with more components brought no significant gain in performance, which is consistent with what was reported in [12]. The one-class classifiers were trained using only bonafide samples of the training set.

The two-class classifiers, LR and two-class SVM, are set as reported in [12]. Bob's LR implementation permits to set the regularization constant (C), which was set to $C = 1$ in the experiments. The RBF kernel was selected for the two-class SVM. For both classifiers, the output score is a probability of a sample being a bonafide class. These two classifiers were trained using both bonafide and artifact samples of the training set.

It is worth to mention that for all SVM-based PAD schemes, the penalty parameter C and kernel parameters values were determined through cross-validation. Then, the best parameters were used in the training with the whole training set.

5
EXPERIMENTAL ANALYSIS

This chapter describes the experiments conducted to evaluate the face Presentation Attack Detection (PAD) methods investigated in this study. Section 5.1 describes the publicly available databases used in the experiments, which represent the heterogeneity of the type of attacks considered in each of the evaluation protocols. Section 5.2 describes the metrics used for performance assessment. Finally, Section 5.3 describes the experimental settings and the results of the experiments.

5.1
Face Spoofing Databases

This section gives a brief overview of the databases used in the experiments performed to evaluate the face PAD methods proposed.

5.1.1
3D Mask-Attack DB (3DMAD)

The 3D MASK-ATTACK DB (3DMAD) [31], constitutes the first public database that considers mask attacks, it provides 2D data, in addition to depth information.

The database is publicly available at the IDIAP Research Institute website[1] and it is composed of genuine and attack access attempts of 17 different users recorded by the Microsoft Kinect sensor. This sensor provides both regular 2D RGB data (8-bit) and depth data (11-bit), with a resolution of 640×480 pixels, at 30 frames per second. Overall, the dataset is composed of: 255 color videos with 300 frames (170 real sequences and 85 mask attacks), and the same number of 2.5D sequences[2] with the corresponding depth information. In Figure 15, the first two sessions (first two columns) are bonafide samples, while the third session (third column) represents a 3D mask attack.

[1]Available link to download the 3D MASK-ATTACK DB at IDIAP website.

[2]Erdogmus and Marcel in [31] refer to depth data as 2.5D sequences (or depth maps), which are grayscale images which contain information relating to the distance of the surfaces of 3D objects from a viewpoint.

Figure 15: Example color (top row) and depth (bottom row) images from three different sessions for a particular subject in 3DMAD [31].

The database was captured in three different sessions: two real-access sessions held two weeks apart, and a third session that represents a mask attack. In each video, the eye-positions are manually labelled for every 1^{st}, 61^{st}, 121^{st}, 181^{st}, 241^{st} and 300^{th} frames and they are linearly interpolated for the remaining frames. Masks (as seen in Figure 16) were manufactured using the service provided by "ThatsMyFace.com", which only requires a frontal and two profile pictures of each person to generate a 3D mask. The diversity provided by the database allows broad flexibility to conduct research on the face presentation attack issue by considering 2D and 3D face PAD approaches and their fusion. Table 4 summarize the main statistcs of the 3DMAD database.

Figure 16: Seventeen facial Presentation Attack Instruments (PAIs) from 3DMAD [31].

Table 4: Summary of the main statistics of 3DMAD database.

Number of clients				17
Number of videos				255
Bonafide videos				170
PA videos				85
Video resolution				640 × 480 pixels
Types of attacks				3D masks
Distribution of videos per class				
Sets	**Number of clients**	**Bonafide**	**Attacks**	**Total**
Training	7	70	35	105
Development	5	50	25	75
Evaluation	5	50	25	75

5.1.2 REPLAY-MOBILE Database

The REPLAY-MOBILE face Presentation Attack Database was introduced by Costa-Pazo et al. [33] and is publicly available at the IDIAP Research Institute website[3].

This dataset contains 10 seconds long HD (720×1280) resolution videos corresponding to 40 identities. The samples were recorded using two mobile devices, namely: (i) an iPad Mini 2 tablet and (ii) an LG-G4 smartphone. The bonafide videos accesses were collected under five different lighting conditions (*controlled*, *adverse*, *direct*, *lateral* and *diffuse*). Figure 17 shows samples from bonafide accesses captured on a smartphone (top row), samples captured on a tablet (bottom row) and video frames in controlled, adverse, direct, lateral, and diffuse scenarios (represented in the columns from left to right, respectively).

In addition, to produce the attacks, high-resolution photos and videos from each subject were taken under conditions similar to those in their authentication sessions (*lighton* and *lightoff*). To generate the PAIs a Nikon Coolpix P520 camera was used to capture high-resolution images (18 Mpixel) for photo-based attacks, whereas video-based attacks were recorded by using the back camera of the LG-G4 smartphone, which records 1080p FullHD video clips.

[3]Available link to download the REPLAY-MOBILE face Presentation Attack Database at IDIAP website.

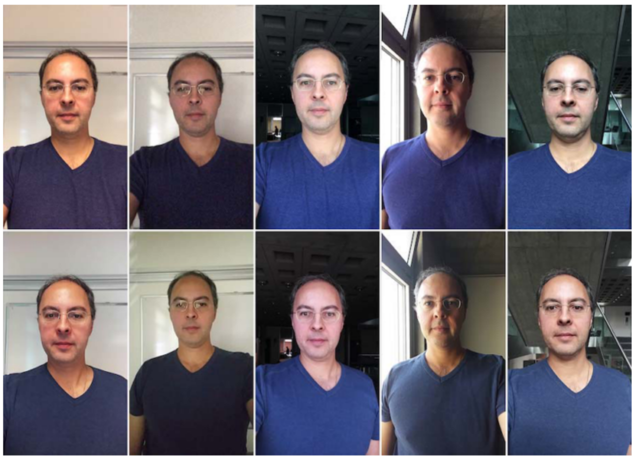

Figure 17: Examples of bonafide accesses in different scenarios provided by REPLAY-MOBILE [33].

PAs represented in this database have been constructed using two PAIs: matte-paper for print attacks and matte screen monitor for digital-replay attacks. For each PAI, two kinds of attacks have been recorded: one where the user holds the recording device in hand, and the second where the recording device is stably supported on a stand. Thus, four kinds of attacks are represented in the database. Figure 18 shows samples of attack accesses captured on a smartphone (top row), samples captured on a tablet (bottom row) and examples of *mattescreen-lighton*, *mattescreen-lightoff*, *print-lighton*, and *print-lightoff* (represented in the columns from left to right, respectively). Table 5 summarize the main statistcs of the REPLAY-MOBILE database.

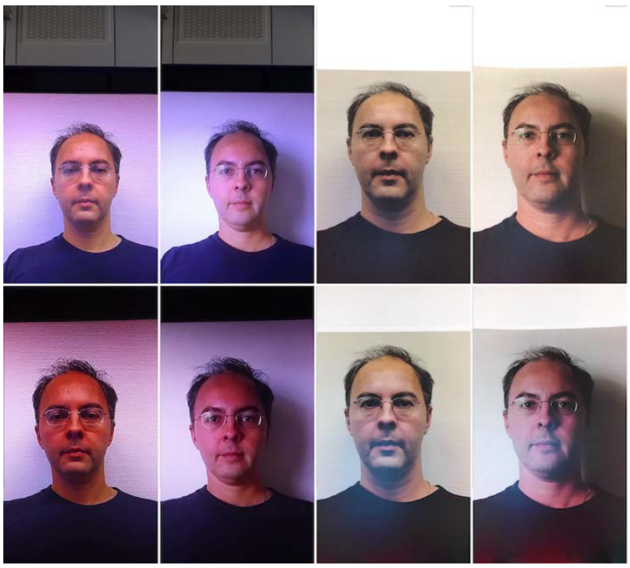

Figure 18: Samples of the different presentations attack instruments (PAIs) available in REPLAY-MOBILE [33].

Table 5: Summary of the main statistics of REPLAY-MOBILE database.

Number of clients				40
Number of videos				1030
Bonafide videos				390
PA videos				640
Video resolution				720 × 1280 pixels
Print attacks				A4 prints
Replay attacks				PC matte-screen
Distribution of videos per class				
Sets	**Number of clients**	**Bonafide**	**Attacks**	**Total**
Training	12	120	192	312
Development	16	160	256	416
Evaluation	12	110	192	302

5.1.3 OULU-NPU Face Presentation Attack Database

Published in 2017, OULU-NPU[4] face presentation attack database [35] consists of 4,950 bonafide accesses and artifact face videos corresponding to the 55 subjects.

The samples were recorded using the front cameras of six mobile devices (Samsung Galaxy S6 edge, HTC Desire EYE, MEIZU X5, ASUS Zenfone Selfie, Sony XPERIA C5 Ultra Dual, and OPPO N3). Some frame examples of a subject are shown in Figure 19. This figure, from left to right, shows examples of samples acquired using Samsung, HTC, MEIZU, ASUS, Sony, and OPPO, respectively.

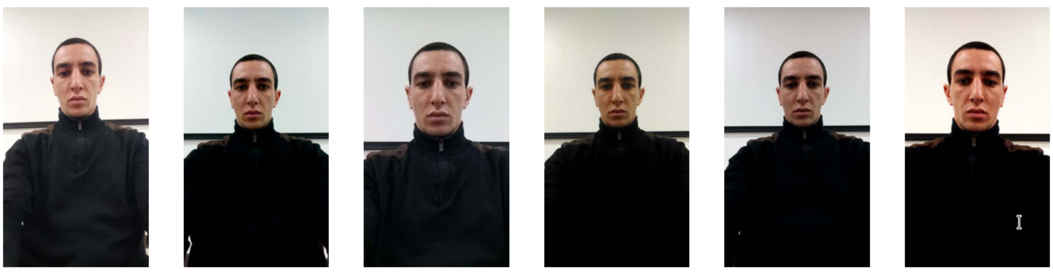

Figure 19: Sample images showing the image quality of the different camera devices for a user in OULU-NPU [35].

The bonafide and artifact videos clips were collected in three sessions with different illumination conditions (Session 1, Session 2 and Session 3).

In order to simulate realistic mobile authentication scenarios, the video length was limited to five seconds and the subjects were asked to hold the

[4] Available link to download OULU-NPU database.

mobile device like they were being authenticated but without deviating too much from their natural posture in normal device usage. The artifact species considered in the OULU-NPU database were print and video-replay. These types of PAIs were created using two printers (high resolution photos printed on A3 glossy paper using a Canon imagePRESS C6011 (Printer 1) and a Canon PIXMA iX6550 (Printer 2)) and two display devices (high-resolution videos replayed on a 19” Dell UltraSharp 1905FP display with 1280×1024 resolution (Display 1) and an early 2015 Macbook 13” laptop with Retina display of 2560×1600 resolution (Display 2)). Table 6 summarize the main statistcs of the OULU-NPU database.

Table 6: Summary of the main statistics of OULU-NPU database.

<table>
<tr><td colspan="2">Number of clients</td><td colspan="3">55</td></tr>
<tr><td colspan="2">Number of videos</td><td colspan="3">4950</td></tr>
<tr><td colspan="2">Bonafide videos</td><td colspan="3">990</td></tr>
<tr><td colspan="2">PA videos</td><td colspan="3">3960</td></tr>
<tr><td colspan="2">Video resolution</td><td colspan="3">1920 × 1080 pixels</td></tr>
<tr><td colspan="2">Print attacks</td><td colspan="3">A3 prints (using two printers)</td></tr>
<tr><td colspan="2">Replay attacks</td><td colspan="3">PC and notebook display</td></tr>
<tr><td colspan="5">Distribution of videos per class</td></tr>
<tr><td>Sets</td><td>Number of clients</td><td>Bonafide</td><td>Attacks</td><td>Total</td></tr>
<tr><td>Training</td><td>20</td><td>360</td><td>1440</td><td>1800</td></tr>
<tr><td>Development</td><td>15</td><td>270</td><td>1080</td><td>1350</td></tr>
<tr><td>Evaluation</td><td>20</td><td>360</td><td>1440</td><td>1800</td></tr>
</table>

5.2 Metrics

According to [97], Attack Presentation Classification Error Rate (APCER) is defined as the proportion of attack presentations using the same PAI species incorrectly classified as bonafide presentations at the PAD subsystem in a specific scenario. Additionally, Bonafide Presentation Classification Error Rate (BPCER) is defined as the proportion of bonafide presentations incorrectly classified as presentation attacks at the PAD subsystem in a specific scenario.

For a given Presentation Attack Instrument Species (PAIS), the APCER is calculated as follows:

$$\text{APCER}_{PAIS} = 1 - \left(\frac{1}{N_{PAIS}}\right) \sum_{i=1}^{N_{PAIS}} Res_i \tag{5-1}$$

where N_{PAIS} is the number of attack presentations for the given Presentation Attack Instrument PAI species [97]. In this regard, Res_i, the classifier response, takes the value 1 if the i^th presentation is classified as an attack presentation and a value of 0 if classified as a bonafide presentation. On the other hand, the BPCER is computed as:

$$\text{BPCER}_{PAIS} = \left(\frac{1}{N_{BF}}\right) \sum_{i=1}^{N_{BF}} Res_i \tag{5-2}$$

where N_{BF} is the number of bonafide presentations, Res_i takes the value 1 if the i^th presentation is classified as an attack presentation and value 0 if classified as a bonafide presentation.

Since both the APCER and the BPCER depend on a decision threshold τ, the development set operates as a separate validation set for fine tuning the system parameters and estimating the threshold value to be used on the test set. Here, τ is defined on the development data as the intersection point of the APCER and BPCER. This intersection point is termed as the Equal Error Rate (EER).

To summarize the overall system performance in a single value, the Half Total Error Rate (HTER) is used. This metric is computed as the average of the APCER and the BPCER at the decision threshold. Finally, Receiver Operating Characteristic (ROC) curves outline the APCER versus the BPCER on the evaluation set.

5.3 Experiments

The evaluation protocols used in this work is similar to what is reported in the literature [24, 45, 91]. They are designed to measure the performance of the PAD schemes in two conditions, namely: (i) *intra-database* and (ii) *inter-database* (or *cross-database*). The first evaluation protocol consists in training and testing the PAD schemes on data from a single database, in which bonafide and artifact accesses are acquired using the same set of sensors settings and attacks are attempted with the same set of PAIs. Furthermore, the evaluation is performed by computing the true and false positive detection rates on the test data available in the dataset. The second evaluation protocol is devised to measure the generalization capacity of the PAD scheme, which is, in this case, trained using training samples the other spoofing database.

The next section shows the performance of the PAD schemes according to the intra-database evaluation protocol. In the following section the generalization capacity of the facial PAD is assessed using the cross-database testing protocol.

5.3.1
Intra-Database Evaluation Protocol

The purpose of these experiments is to evaluate the performance of the facial PAD schemes, by testing in the same database where the method is trained.

The evaluation protocol used on the IDIAP databases (REPLAY-MOBILE and 3DMAD) is similar to that of prior works [31, 33]. The PAD schemes are trained using all videos available in the training set of each database. The only difference with respect to those works, is the way of sampling each video in the training set: by selecting a sample from the video frames with a step size of 3 (i.e. every 3 frames). The same protocol was used for the OULU-NPU database.

The results corresponding to the intra-database evaluation protocol are presented below.

Results and Discussion

Table 7 shows the performance of the face PAD schemes for the intra-database evaluation protocol, for each database. The performance is reported as a measure of EER (computed on the development set) and HTER (computed on the evaluation set) values. The best HTERs are highlighted in bold.

The results show that the utilization of features learned with the CAE model in the respective two-class PAD schemes provide, in general, the best detection rates. LR+CAE and SVM2+CAE outperform other schemes on the REPLAY-MOBILE and OULU-NPU databases, with HTER values of 5.17% and 12.11%, respectively. These results demonstrate the effectiveness of learned feature with the CAE in the intra-database testing protocol, and reveal that OULU-NPU database is the most challenging one.

Table 7: The performance of the face PAD schemes for the intra-database evaluation protocol on each database.

Systems		**REPLAY-MOBILE**		**3DMAD**		**OULU-NPU**	
		Dev.	**Test**	**Dev.**	**Test**	**Dev.**	**Test**
		EER	**HTER**	**EER**	**HTER**	**EER**	**HTER**
LR	**+ IQM**	3.69	7.15	0.00	2.29	12.73	15.65
SVM2	**+ IQM**	2.65	5.84	2.90	1.00	9.38	12.24
SVM1	**+ IQM**	27.89	35.51	32.12	51.79	28.70	32.84
GMM1	**+ IQM**	25.17	27.48	38.09	32.72	27.99	31.92
LR	**+ LBP-RGB**	6.03	6.83	17.36	10.96	14.82	20.11
SVM2	**+ LBP-RGB**	2.69	7.25	19.92	12.00	14.24	15.79
SVM1	**+ LBP-RGB**	27.85	32.13	44.45	37.29	35.69	33.43
GMM1	**+ LBP-RGB**	19.96	22.85	39.40	45.46	33.81	32.75
LR	**+ BSIF-RGB**	9.33	5.82	0.12	**0.93**	11.17	18.69
SVM2	**+ BSIF-RGB**	10.26	6.89	0.32	4.89	12.00	18.36
SVM1	**+ BSIF-RGB**	25.30	24.96	28.28	25.87	39.35	38.25
GMM1	**+ BSIF-RGB**	27.60	27.46	30.00	32.64	40.79	38.22
LR	**+ CAE**	2.91	**5.17**	13.48	7.86	8.64	14.82
SVM2	**+ CAE**	5.65	7.55	9.49	3.16	8.75	**12.11**
SVM1	**+ CAE**	36.49	39.69	49.08	42.29	47.22	48.93
GMM1	**+ CAE**	24.65	22.64	36.48	37.36	36.12	42.39

However, the best two-class PAD scheme based on the features learned with the CAE (SVM1+CAE) was outperformed by the schemes based on BSIF-RGB and IQM hand-crafted features in the 3DMAD database, which achieved HTER values of 0.93% and 1.00%, respectively. In the case of 3DMAD, as expected, the CAE decreased its performance owing to the influence of the small number of training samples provided by this database.

Furthermore, the results obtained by LR+BSIF and SVM2+IQM schemes on 3DMAD are comparable with those reported in the state-of-the-art, and the HTER of 0.93 % achieved by LR+BSIF system outperformed the baseline results reported in [33].

The performances of the PAD schemes based on one-class classification approach measured in the current protocol were inferior in comparison to their two-class counterparts, which take advantage of the number of training samples, as the training of the schemes based on the one-class classification approach use much less training data (bonafide accesses only).

It is worth highlighting that the best one-class PAD schemes are based on the GMM classifier, GMM1+CAE and GMM1+IQM, which achieved HTER values of 22.64% and 31.92% on REPLAY-MOBILE and OULU-NPU databases, respectively. However, the SVM1+BSIF outperformed the other one-class PAD schemes on the 3DMAD database, achieving a HTER value of 25.87%.

To evaluate the performance of the PAD schemes more comprehensively, the ROC curves for each database are presented. Figure 20 shows ROC curves for PAD systems on REPLAY-MOBILE database, whereas Figure 21 and 22 correspond to the results obtained on 3DMAD and OULU-NPU databases, respectively.

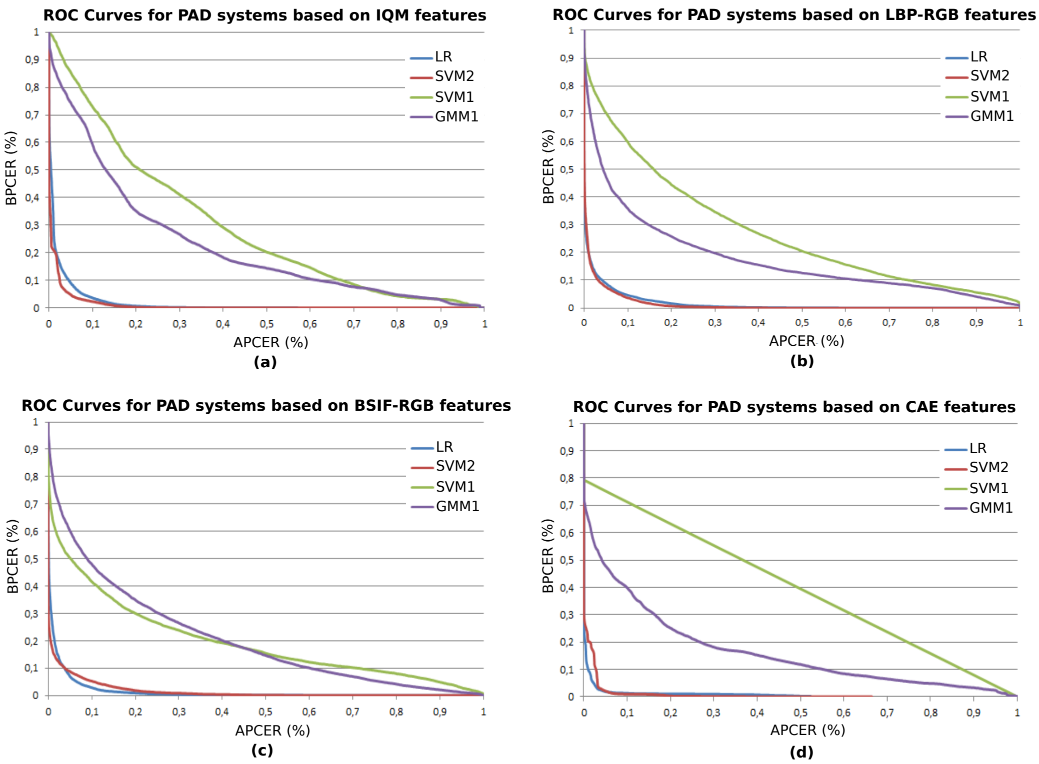

Figure 20: ROC curves for PAD systems on REPLAY-MOBILE database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

In the particular case of IQM features, it can be seen that the PAD schemes based on one-class GMM classifier outperformed those based on one-class SVM for all databases. The trend manifested by these PAD schemes on the three databases is similar to what is reported in the literature [12], when they are evaluated in other databases. In this regard, the ROC curves confirm that one-class PAD schemes based on the features learned by CAE as the worse performance obtained on the 3 databases.

From a general perspective, the results obtained here confirm that when the PAD schemes based on one-class classification approach are restricted to application environments in a same domain, their discrimination capacity degrades considerably. Moreover, the use of IQM and LBP features is more beneficial for the intra-database evaluation protocol, when those features are combined with the one-class SVM classifier, whereas IQM and LBP features seem to be a better option for one-class GMM.

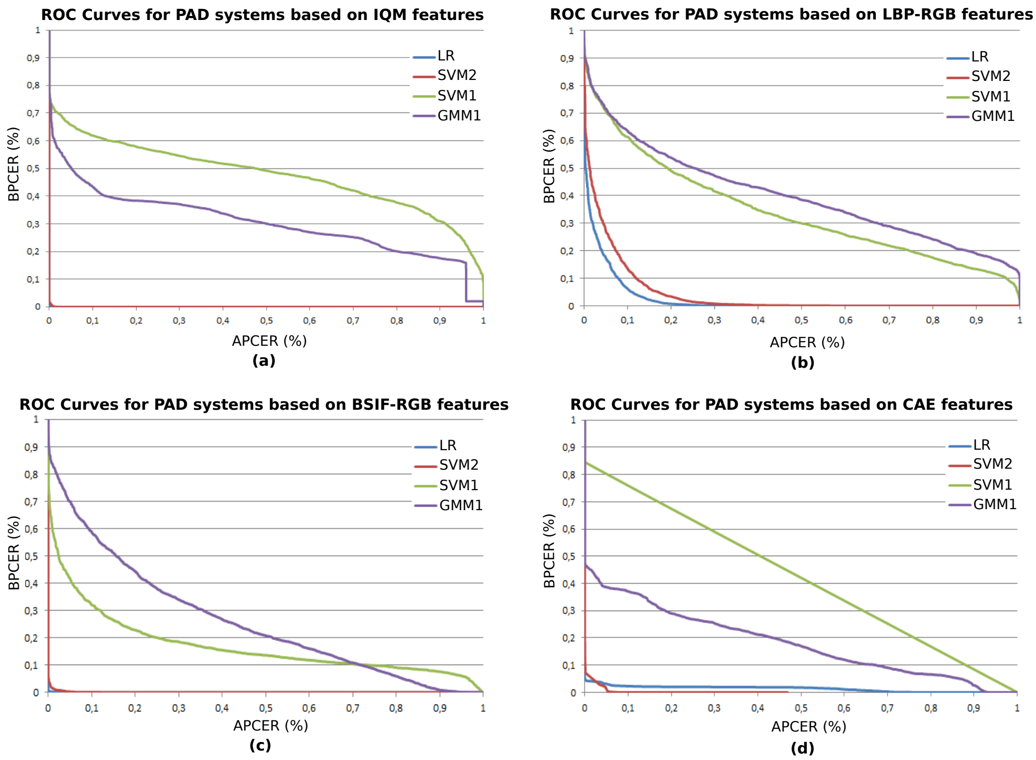

Figure 21: ROC curves for PAD systems on 3DMAD database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

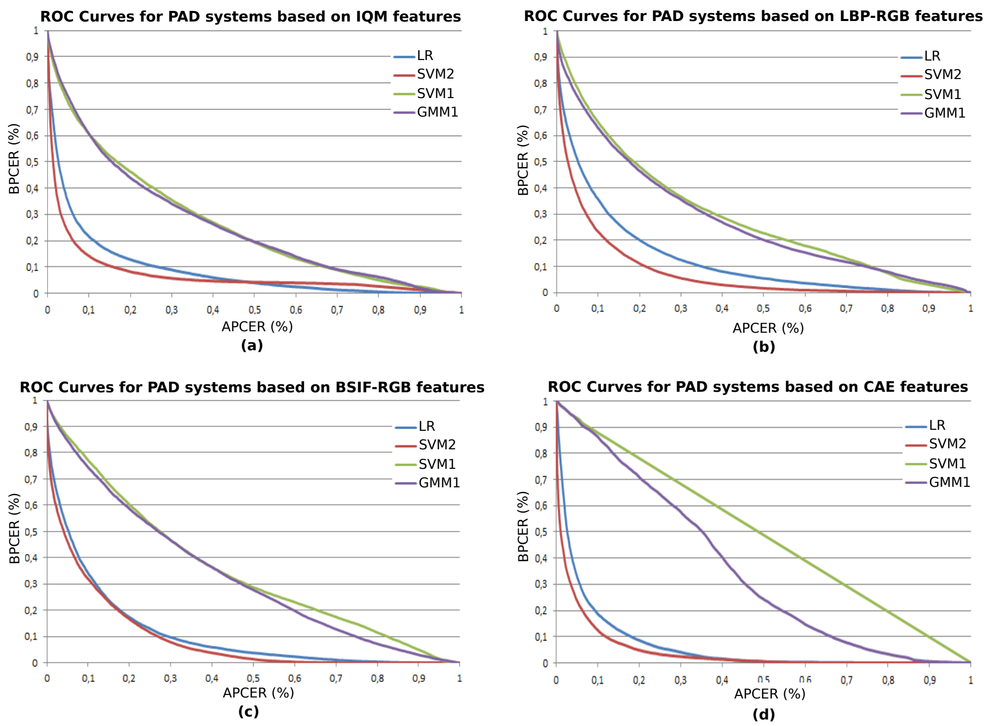

Figure 22: ROC curves for PAD systems on OULU-NPU database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

5.3.2
Inter-Database Evaluation Protocol

The purpose of theses experiments is to study the generalization capacity of the facial PAD schemes, across different databases. To accomplish this aim, we follow two strategies for the inter-database evaluation protocol, by assuming that:

1. Training and development data from one database is available. The models are trained and developed using samples from that database and then evaluated using samples from the two other databases.

2. Training and development data from two databases are available. The models are trained and developed using samples from both databases and then evaluated on the third database.

It is worth emphasizing that regardless of the strategy followed, the two-class schemes are trained using bonafide and artifact data, and the one-class schemes are trained solely with bonafide accesses. The results corresponding to the current evaluation protocol are presented in the following.

Results and Discussion

Table 8 shows the performance of the face PAD schemes for the inter-database evaluation protocol on the REPLAY-MOBILE database. The performance is reported as a measure of EER (computed on the development set) and HTER (computed on the evaluation set) values. The best HTERs are highlighted in bold.

Table 8: The performance of the face PAD schemes for the inter-database evaluation protocol on REPLAY-MOBILE database.

Testing		**REPLAY-MOBILE**					
Training		**3DMAD**		**OULU-NPU**		**3DMAD + OULU-NPU**	
Methods		**Dev.**	**Test**	**Dev.**	**Test**	**Dev.**	**Test**
		EER	**HTER**	**EER**	**HTER**	**EER**	**HTER**
LR	**+ IQM**	0.00	50.34	12.73	41.93	12.32	**38.51**
SVM2		2.09	**45.53**	9.38	**33.93**	10.09	40.59
SVM1		32.12	50.00	28.70	46.51	29.15	45.83
GMM1		38.09	50.00	27.99	49.26	41.48	47.40
LR	**+ LBP-RGB**	17.36	55.68	14.82	62.38	16.35	64.34
SVM2		19.92	50.91	14.24	41.61	12.48	43.05
SVM1		44.45	50.00	35.69	50.00	31.83	50.00
GMM1		39.40	50.00	33.81	49.54	30.42	49.72
LR	**+ BSIF-RGB**	0.12	54.40	11.17	54.55	11.67	57.50
SVM2		0.32	49.78	12.00	47.82	50.00	50.00
SVM1		28.28	50.00	39.35	50.00	33.83	50.00
GMM1		30.00	50.00	40.79	48.79	34.68	47.14
LR	**+ CAE**	13.48	56.05	8.64	55.69	9.60	44.00
SVM2		9.49	50.00	8.75	50.00	12.32	45.18
SVM1		49.08	50.00	47.22	50.00	42.41	50.00
GMM1		36.48	50.00	36.12	50.00	42.81	47.88

As it can be seen in Table 8, the results reveal the SVM2+IQM and LR+IQM as the best two-class PAD schemes on the REPLAY-MOBILE database when trained on 3DMAD database and its combination with OULU-NPU, which achieved 42.77% and 38.51%, respectively. However, in general, the results confirm the effectiveness of the one-class PAD schemes on the REPLAY-MOBILE database when trained with samples from OULU-NPU database. The best performance was obtained by GMM1+BSIF-RGB, with an HTER value of 30.30%, followed by GMM1+LBP-RGB and SVM1+IQM with 30.41% and 32.28%, respectively. It is worth noting that the strategy of training with samples from two databases did not ensure a significant performance enhancement for the two-class PAD schemes, and also, it deteriorated detection rates of their one-class counterparts.

Table 9 shows the performance of the face PAD schemes for the inter-database evaluation protocol on the 3DMAD database. The performance is reported as a measure of EER (computed on the development set) and HTER (computed on the evaluation set) values. The best HTERs are highlighted in bold.

Table 9: The performance of the face PAD schemes for the inter-database evaluation protocol on 3DMAD database.

Testing		3DMAD					
Training		REPLAY-MOBILE		OULU-NPU		REPLAY-MOBILE + OULU-NPU	
Methods		Dev.	Test	Dev.	Test	Dev.	Test
		EER	HTER	EER	HTER	EER	HTER
LR	+ IQM	3.69	41.42	12.73	**14.14**	15.29	43.31
SVM2	+ IQM	2.65	**15.79**	9.38	**14.14**	17.14	45.81
SVM1	+ IQM	27.89	50.00	28.70	50.00	33.65	50.00
GMM1	+ IQM	25.17	50.00	27.99	50.00	37.75	50.00
LR	+ LBP-RGB	6.03	57.06	14.82	40.27	19.08	68.59
SVM2	+ LBP-RGB	2.69	50.00	14.23	57.46	23.41	54.07
SVM1	+ LBP-RGB	27.85	50.00	35.69	50.00	37.34	50.00
GMM1	+ LBP-RGB	19.96	50.00	33.81	50.00	39.64	47.50
LR	+ BSIF-RGB	9.33	50.00	11.17	49.50	16.83	50.03
SVM2	+ BSIF-RGB	10.26	49.68	12.00	52.01	50.00	50.00
SVM1	+ BSIF-RGB	25.30	52.96	39.35	44.76	46.86	49.42
GMM1	+ BSIF-RGB	27.60	45.63	40.78	46.10	55.17	**21.76**
LR	+ CAE	2.91	50.00	8.63	49.12	9.25	50.53
SVM2	+ CAE	5.65	50.00	8.75	50.00	7.77	57.72
SVM1	+ CAE	36.49	50.00	47.22	50.00	44.34	39.58
GMM1	+ CAE	24.65	50.00	36.12	50.00	49.73	74.74

The results show that the SVM2+IQM and LR+IQM obtained the best detection rates on the 3DMAD, which considerably outperformed the remaining PAD schemes when trained on the OULU-NPU and REPLAY-MOBILE databases, respectively. In the particular case of IQM features, it can be seen that the PAD schemes based on one-class GMM classifier outperforms those based on one-class SVM for all databases. The HTER value of 3.44% obtained by SVM2+IQM represents a one sixth of the best HTER value, 21.76%, achieved by GMM1+BSIF-RGB, which take advantage of training with samples from the combination of REPLAY-MOBILE and OULU-NPU databases.

Table 10 shows the performance of the face PAD schemes for the inter-database evaluation protocol on the 3DMAD database. The performance is reported as a measure of EER (computed on the development set) and HTER (computed on the evaluation set) values. The best HTER are highlighted in bold.

As it can be seen in Table 10, the best performing two-class schemes for the OULU-NPU database when trained on REPLAY-MOBILE is LR+IQM, with an HTER of 37.60%, whereas SVM1+LBP-RGB and GMM1+BSIF-RGB were the best one-class schemes, both with an HTER of 40.53%. Examining Table 10 reveals that, in general, the GMM1+BSIF scheme performs better than the two-class schemes.

Table 10: The performance of the face PAD schemes for the cross-database evaluation protocol on OULU-NPU database.

Testing		OULU-NPU					
Training		REPLAY-MOBILE		3DMAD		REPLAY-MOBILE + 3DMAD	
Methods		Dev.	Test	Dev.	Test	Dev.	Test
		EER	HTER	EER	HTER	EER	HTER
LR	+ IQM	3.69	44.14	0.00	49.71	5.42	49.19
SVM2	+ IQM	2.65	45.84	2.09	**46.04**	2.54	44.68
SVM1	+ IQM	27.89	**42.39**	32.12	50.00	30.95	48.10
GMM1	+ IQM	25.17	49.28	38.09	50.00	31.44	49.10
LR	+ LBP-RGB	6.03	57.18	17.36	50.09	9.52	53.97
SVM2	+ LBP-RGB	2.69	50.63	19.92	46.69	11.61	54.74
SVM1	+ LBP-RGB	27.85	49.17	44.45	50.00	22.84	49.92
GMM1	+ LBP-RGB	19.96	46.68	39.40	50.00	22.74	47.02
LR	+ BSIF-RGB	9.33	50.84	0.12	51.20	10.34	52.61
SVM2	+ BSIF-RGB	10.26	51.07	0.32	50.40	6.66	50.84
SVM1	+ BSIF-RGB	25.30	49.92	28.28	50.00	24.98	48.47
GMM1	+ BSIF-RGB	27.60	44.73	30.00	50.00	38.54	**43.23**
LR	+ CAE	2.91	50.00	13.48	49.03	5.32	51.24
SVM2	+ CAE	5.65	50.00	9.49	50.00	4.16	55.52
SVM1	+ CAE	36.49	50.00	49.08	50.00	30.12	54.29
GMM1	+ CAE	24.65	50.00	36.48	50.00	27.13	49.03

The schemes trained using samples from two databases, in general, outperformed to those trained using samples from one single database in terms of EER, which is illustrated in the Table 10 corresponding to the cross-database testing protocol on OULU-NPU database. In terms of HTER the opposite occurred.

To evaluate the performance of the best PAD schemes more comprehensively, the ROC curves for each training database are presented. Figure 23 shows ROC curves of the best PAD scheme on REPLAY-MOBILE database, whereas Figure 24 and 25 correspond to the results obtained on 3DMAD and OULU-NPU databases, respectively. The remaining ROC curves achieved in the current testing protocol are presented in the Appendix A.

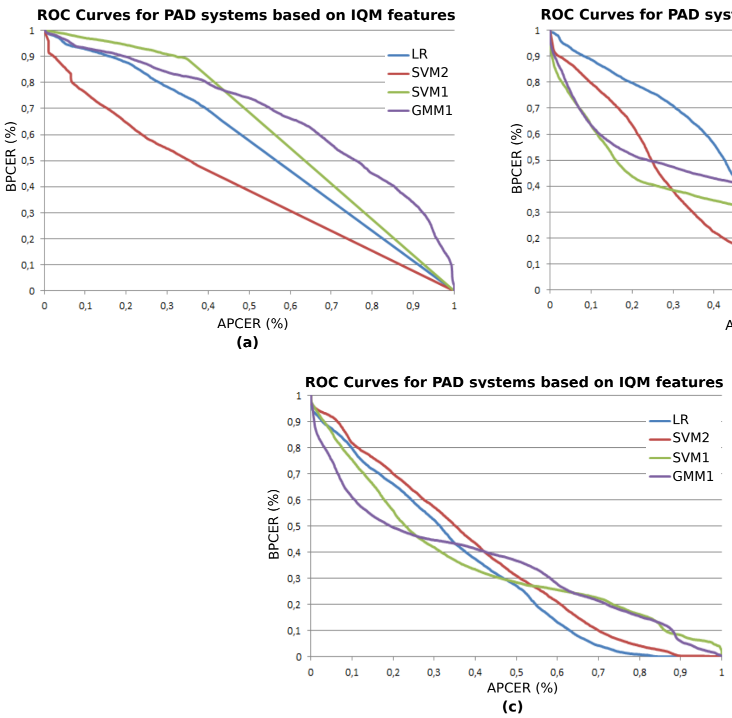

Figure 23: ROC curves for the best PAD systems on REPLAY-MOBILE database are shown, considering each training database: (a) 3DMAD, (b) OULU-NPU, and (c) the combination of 3DMAD and OULU-NPU databases.

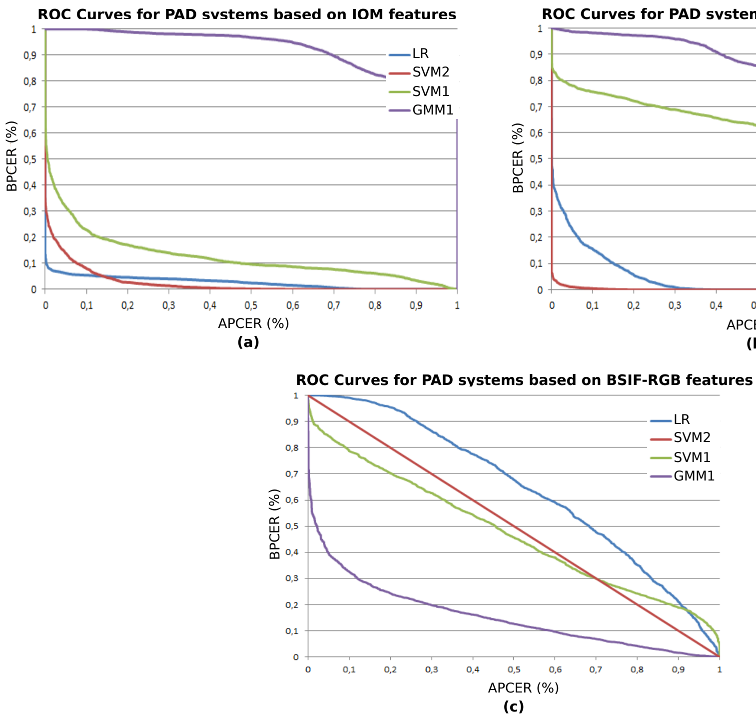

Figure 24: ROC curves for the best PAD systems on 3DMAD database are shown, considering each training database: (a) REPLAY-MOBILE, (b) OULU-NPU, and (c) the combination of REPLAY-MOBILE and OULU-NPU databases.

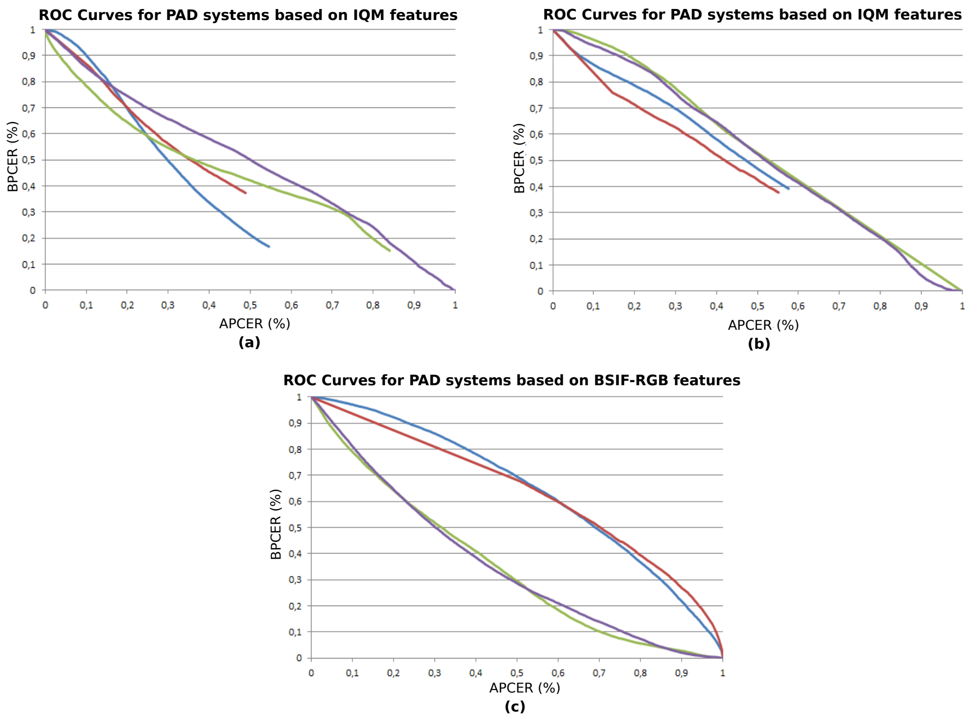

Figure 25: ROC curves for the best PAD systems on OULU-NPU database are shown, considering each training database: (a) 3DMAD, (b) REPLAY-MOBILE, and (c) the combination of REPLAY-MOBILE and 3DMAD databases.

According to the results obtained, inter-database anti-spoofing is far from satisfactory. Owing to different devices, illuminations, races, etc., there are some inevitable biases among two datasets. In this case, the inter-database evaluation protocol can hardly obtain analogous performance as compared to the intra-database counterpart. However, the ability of the schemes based on one-class classifier approaches to discriminate between bonafide and artifact attacks could be better than that of two-class approaches, as in essence one-class classifiers aim to encapsulate the bonafide access data and any deviations from the norm, including different samples in the new domain should be detectable.

It can be concluded that neither the two-class systems nor the one-class approaches perform well enough and more research should be conducted to enhance current systems.

6
CONCLUSIONS AND FUTURE WORKS

In this work, we have evaluated and compared some of the most relevant feature-based state-of-the-art methods for facial PAD using three facial spoofing databases publicly available, which represent the heterogeneity of presentation facial attacks. For this purpose, we tested sixteen different PAD schemes, which represent the combination of four feature descriptors and two classification approaches: one-class and two-class.

The results obtained in the experiments show that the performances of the PAD schemes based on one-class classification, measured in application environments of the same attack domains are inferior, as expected, in comparison to their two-class counterparts. Additionally, the experiments revealed that PAD schemes that use the features learned by CAE in combination with the two-class classification approach provide, in general, the best performances in REPLAY-MOBILE and OULU-NPU databases, whereas PAD schemes based on IQM and BSIF-RGB features descriptors can perform better in scenarios with a limited number of training samples.

By comparing the results of the inter-database experiments, through the ROC Curves, EER and HTER metrics for each of the evaluated PAD schemes, we can concluded that the performance of both formulations (one-class and two-class) was not adequate, and more research is required to enhance the detection rates in such scenarios.

As a desirable extension of this study, it would be interesting to test other feature descriptors, or modified versions of those evaluated here. For example, the LBP-TOP and BSIF-TOP features which have been associated to good accuracies when combined with one-class classification [36]. The combination of the outputs delivered by the different PAD schemes evaluated in a final decision provided by an ensemble it is worth exploring in future research.

Additionally, it would be interesting to examine the utilization of other dimensions of the feature vectors learned by the CAE, as well as, the implementation of data augmentation during the training procedure of the model to boost the performance in application environments with a limited number of samples.

Bibliography

1 JAIN, A. K.; NANDAKUMAR, K. ; ROSS, A.. **50 years of biometric research: Accomplishments, challenges, and opportunities**. Pattern Recognition Letters, 79:80–105, 2016.

2 JAIN, A. K.; FLYNN, P. ; ROSS, A. A.. **Handbook of Biometrics**. Springer Science + Business Media, New York, 1 edition, 2008.

3 RATHA, N. K.; CONNELL, J. H. ; BOLLE, R. M.. **Enhancing security and privacy in biometrics-based authentication systems**. IBM Systems Journal, 40(3):614–634, 2001.

4 RAMACHANDRA, R.; BUSCH, C.. **Presentation Attack Detection Methods for Face Recognition Systems: A Comprehensive Survey**. ACM Computing Surveys, 50(1):1–37, 2017.

5 MOHAMMADI, A.; BHATTACHARJEE, S. ; MARCEL, S.. **Deeply vulnerable: a study of the robustness of face recognition to presentation attacks**. IET Biometrics, 7(1):15–26, 2018.

6 SCHUCKERS, S. A. C.. **Spoofing and Anti-Spoofing Measures**. Information Security Technical Report, 7(4):56–62, 2002.

7 ROSS, A. A.; NANDAKUMAR, K. ; JAIN, A. K.. **Handbook of Multibiometrics**. Springer Science + Business Media, New York, 1 edition, 2006.

8 RODRIGUES, R. N.; LING, L. L. ; GOVINDARAJU, V.. **Robustness of multimodal biometric fusion methods against spoof attacks**. Journal of Visual Languages and Computing, 20(3):169–179, 2009.

9 CHINGOVSKA, I.; ANJOS, A. ; MARCEL, S.. **On the Effectiveness of Local Binary Patterns in Face Anti-spoofing**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE OF THE BIOMETRICS SPECIAL INTEREST GROUP (BIOSIG), Darmstadt, Germany, 2012. IEEE.

10 CHINGOVSKA, I.; YANG, J.; LEI, Z.; YI, D.; LI, S. Z.; KAHM, O.; GLASER, C.; DARNER, N.; KUIJPER, A.; NOUAK, A.; KOMULAINEN, J.; PEREIRA, T.; GUPTA, S.; KHANDEL WA, S.; BANSAL, S.; RAI, A.; KRISHNA, T.;

GOYAL, D.; WARIS, M. A.; ZHANG, H.; AHMAD, I.; KIRANYAZ, S.; GABBOUJ, M.; TRONCI, R.; PILI, M.; SIRENA, N.; ROLI, F.; GALBALLY, J.; FICRRCZ, J.; PINTO, A.; PEDRINI, H.; SCHWARTZ, W. S.; ROCHA, A.; ANJOS, A. ; MARCEL, S.. **The 2nd competition on counter measures to 2D face spoofing attacks**. Proceedings - 2013 International Conference on Biometrics, ICB 2013, 2013.

11 BAGGA, M.; SINGH, B.. **Spoofing detection in face recognition: A review**. In: 3RD INTERNATIONAL CONFERENCE ON COMPUTING FOR SUSTAINABLE GLOBAL DEVELOPMENT, p. 2037–2042, New Delhi, India, 2016. IEEE.

12 NIKISINS, O.; MOHAMMADI, A.. **On Effectiveness of Anomaly Detection Approaches against Unseen Presentation Attacks in Face Anti-Spoofing**. In: THE 11TH IAPR INTERNATIONAL CONFERENCE ON BIOMETRICS, número March, 2018.

13 NIXON, M. S.; LI, S. Z. ; BIOMETRICS, T.. **Handbook of Biometric Anti-Spoofing**. Springer, London, 2014.

14 GALBALLY, J.; MARCEL, S.. **Face anti-spoofing based on general image quality assessment**. In: PROCEEDINGS - INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, p. 1173–1178, 2014.

15 OJALA, T.; PIETIKÄINEN, M. ; HARWOOD, D.. **A comparative study of texture measures with classification based on featured distributions**. Pattern Recognition, 29(1):51–59, 1996.

16 KRIZHEVSKY, A.; SUTSKEVER, I. ; HINTON, G. E.. **Imagenet classification with deep convolutional neural networks**. In: PROCEEDINGS OF THE 25TH INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS - VOLUME 1, NIPS'12, p. 1097–1105, USA, 2012. Curran Associates Inc.

17 LE, Q. V.; RANZATO, M.; MONGA, R.; DEVIN, M.; CHEN, K.; CORRADO, G. S.; DEAN, J. ; NG, A. Y.. **Building high-level features using large scale unsupervised learning**. In: PROCEEDINGS OF THE 29TH INTERNATIONAL COFERENCE ON INTERNATIONAL CONFERENCE ON MACHINE LEARNING, ICML'12, p. 507–514, USA, 2012. Omnipress.

18 HU, G.; YANG, Y.; YI, D.; KITTLER, J.; CHRISTMAS, W.; LI, S. Z. ; HOSPEDALES, T.. **When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for**

Face Recognition. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOP (ICCVW), Santiago, Chile, 2015. IEEE.

19 LI, Y.; PO, L. M.; XU, X.; FENG, L. ; YUAN, F.. **Face liveness detection and recognition using shearlet based feature descriptors**. In: 2016 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), p. 874–877, 2016.

20 ABBAS, Y.; REHMAN, U.; PO, L. M. ; LIU, M.. **Deep Learning for Face Anti-Spoofing: An End-to-End Approach**. In: SIGNAL PROCESSING: ALGORITHMS, ARCHITECTURES, ARRANGEMENTS, AND APPLICATIONS (SPA), p. 195–200. IEEE, 2017.

21 ASIM, M.; ZHU, M. ; JAVED, M. Y.. **CNN based spatio-temporal feature extraction for face anti-spoofing**. 2017 2nd International Conference on Image, Vision and Computing, ICIVC 2017, p. 234–238, 2017.

22 GAN, J.; LI, S.; ZHAI, Y. ; LIU, C.. **3D Convolutional Neural Network Based on Face Anti-spoofing**. In: Wuhan, C., editor, 2ND INTERNATIONAL CONFERENCE ON MULTIMEDIA AND IMAGE PROCESSING (ICMIP), número 1, p. 1–5. IEEE, 2017.

23 LUCENA, O.; JUNIOR, A.; MOIA, V.; SOUZA, R.; VALLE, E. ; LOTUFO, R.. **Transfer Learning Using Convolutional Neural Networks for Face Anti-spoofing**. In: Karray, F.; Campilho, A. ; Cheriet, F., editors, IMAGE ANALYSIS AND RECOGNITION, p. 27–34, Cham, 2017. Springer International Publishing.

24 LI, H.; LI, W.; CAO, H.; WANG, S.; HUANG, F. ; KOT, A. C.. **Unsupervised Domain Adaptation for Face Anti-Spoofing**. IEEE Transactions on Information Forensics and Security, 6013(c):1–1, 2018.

25 TAN, X.; LI, Y.; LIU, J. ; JIANG, L.. **Face liveness detection from a single image with sparse low rank bilinear discriminative model**. In: PROCEEDINGS OF THE 11TH EUROPEAN CONFERENCE ON COMPUTER VISION: PART VI, ECCV'10, p. 504–517, Berlin, Heidelberg, 2010. Springer-Verlag.

26 PEIXOTO, B.; MICHELASSI, C. ; ROCHA, A.. **Face liveness detection under bad illumination conditions**. In: 2011 18TH IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, p. 3557–3560, 2011.

27 ANJOS, A.; MARCEL, S.. **Counter-measures to Photo Attacks in Face Recognition: A Public Database and a Baseline**. In:

PROCEEDINGS OF THE 2011 INTERNATIONAL JOINT CONFERENCE ON BIOMETRICS, IJCB '11, p. 1–7, Washington, DC, USA, 2011. IEEE Computer Society.

28 ZHANG, Z.; YAN, J.; LIU, S.; LEI, Z.; YI, D. ; LI, S. Z.. **A face antispoofing database with diverse attacks**. In: 2012 5TH IAPR INTERNATIONAL CONFERENCE ON BIOMETRICS (ICB), p. 26–31, 2012.

29 WEN, D.; HAN, H. ; JAIN, A. K.. **Face Spoof Detection With Image Distortion Analysis**. IEEE Transactions on Information Forensics and Security, 10(4):746–761, 2015.

30 RAGHAVENDRA, R.; RAJA, K. B. ; BUSCH, C.. **Presentation Attack Detection for Face Recognition Using Light Field Camera**. IEEE Transactions on Image Processing, 24(3):1060–1075, 2015.

31 ERDOGMUS, N.; MARCEL, S.. **Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect**. In: 2013 IEEE SIXTH INTERNATIONAL CONFERENCE ON BIOMETRICS: THEORY, APPLICATIONS AND SYSTEMS (BTAS), p. 1–6, Arlington, VA, USA, 2013. IEEE.

32 PATEL, K.; HAN, H. ; JAIN, A. K.. **Secure Smartphone Unlock: Robust Face Spoof Detection on Mobile**. Technical Report 10, Department of Computer Science, Michigan State University, Michigan, USA, 2015.

33 COSTA-PAZO, A.; BHATTACHARJEE, S.; VAZQUEZ-FERNANDEZ, E. ; MARCEL, S.. **The Replay-Mobile Face Presentation-Attack Database**. In: 2016 INTERNATIONAL CONFERENCE OF THE BIOMETRICS SPECIAL INTEREST GROUP (BIOSIG), p. 1–7, 2016.

34 CHINGOVSKA, I.; ERDOGMUS, N.; ANJOS, A. ; MARCEL, S.. **Face Recognition Systems Under Spoofing Attacks**, p. 165–194. Springer International Publishing, Cham, 2016.

35 BOULKENAFET, Z.; KOMULAINEN, J.; LI, L.; FENG, X. ; HADID, A.. **OULU-NPU: A Mobile Face Presentation Attack Database with Real-World Variations**. In: 2017 12TH IEEE INTERNATIONAL CONFERENCE ON AUTOMATIC FACE GESTURE RECOGNITION (FG 2017), p. 612–618, 2017.

36 ARASHLOO, S. R.; KITTLER, J. ; MEMBER, L.. **An Anomaly Detection Approach to Face Spoofing Detection : A New Formulation and Evaluation Protocol**. IEEE Access, 5:13868 – 13882, 2017.

37 INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Information technology - Biometric presentation attack detection -Part 1: Framework**. ISO ISO/IEC 30107-1:2016, International Organization for Standardization, Geneva, Switzerland, 2016.

38 GALBALLY, J.; MARCEL, S. ; FIERREZ, J.. **Biometric antispoofing methods: A survey in face recognition**. IEEE Access, 2:1530–1552, 2014.

39 CHAKRABORTY, S.; DAS, D.. **An Overview of Face Liveness Detection**. International Journal on Information Theory, 3(2):11–25, 2014.

40 PAN, G.; WU, Z. ; SU, L.. **Liveness Detection for Face Recognition**. Recent Advances in Face Recognition, (December), 2008.

41 KOLLREIDER, K.; FRONTHALER, H. ; BIGUN, J.. **Verifying liveness by multiple experts in face biometrics**. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, 2008.

42 DE MARSICO, M.; NAPPI, M.; RICCIO, D. ; DUGELAY, J. L.. **Moving face spoofing detection via 3D projective invariants**. Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012, p. 73–78, 2012.

43 MAATTA, J.; HADID, A. ; PIETIKAINEN, M.. **Face spoofing detection from single images using texture and local shape analysis**. IET Biometrics, 1(1):3, 2012.

44 TREFNÝ, J.; MATAS, J.. **Extended set of local binary patterns for rapid object detection**. Computer Vision Winter Workshop, p. 1–7, 2010.

45 YANG, J.; LEI, Z. ; LI, S. Z.. **Learn Convolutional Neural Network for Face Anti-Spoofing**. CoRR, abs/1408.5, 2014.

46 DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K. ; FEI-FEI, L.. **ImageNet: A Large-Scale Hierarchical Image Database**. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. IEEE, 2009.

47 LI, J.; WANG, Y.; TAN, T. ; JAIN, A. K.. **Live face detection based on the analysis of fourier spectra**. In: BIOMETRIC TECHNOLOGY FOR HUMAN IDENTIFICATION, volumen 5404, p. 296–303, 2004.

48 LIU, W.. **Face liveness detection using analysis of Fourier spectra based on hair**. In: INTERNATIONAL CONFERENCE ON WAVELET ANALYSIS AND PATTERN RECOGNITION, volumen 2014-Janua, p. 75–80, 2014.

49 TEJA, M. H.. **Real-time live face detection using face template matching and DCT energy analysis**. In: INTERNATIONAL CONFERENCE OF SOFT COMPUTING AND PATTERN RECOGNITION, SOCPAR 2011, p. 342–346, 2011.

50 GALBALLY, J.; MARCEL, S. ; FIERREZ, J.. **Image quality assessment for fake biometric detection: Application to Iris, fingerprint, and face recognition**. IEEE Transactions on Image Processing, 23(2):710–724, 2014.

51 KOMULAINEN, J.; HADID, A. ; PIETIKÄINEN, M.. **Context based face anti-spoofing**. In: 2013 IEEE SIXTH INTERNATIONAL CONFERENCE ON BIOMETRICS: THEORY, APPLICATIONS AND SYSTEMS (BTAS), Arlington, VA, USA, 2013. IEEE.

52 RAGHAVENDRA, R.; BUSCH, C.. **Presentation attack detection algorithm for face and iris biometrics**. In: EUROPEAN SIGNAL PROCESSING CONFERENCE, p. 1387–1391, Lisbon, Portugal, 2014. IEEE.

53 YANG, L.. **Face liveness detection by focusing on frontal faces and image backgrounds**. In: INTERNATIONAL CONFERENCE ON WAVELET ANALYSIS AND PATTERN RECOGNITION, volumen 2014-Janua, p. 93–97, Lanzhou, China, 2014. IEEE.

54 CHINGOVSKA, I.; DOS ANJOS, A. R.. **On the Use of Client Identity Information for Face Antispoofing**. IEEE Transactions on Information Forensics and Security, 10(4):787–796, 2015.

55 PEREIRA F., T.; KOMULAINEN, J.; ANJOS, A.; MARTINO M., J.; HADID, A.; PIETIKAINEN, M. ; MARCEL, S.. **Face liveness detection using dynamic texture**. EURASIP Journal on Image and Video Processing, 2, 2014.

56 ZHAO, G.; PIETIKAINEN, M.. **Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(6):915–928, 2007.

57 KOLLREIDER, K.; FRONTHALER, H.; FARAJ, M. I. ; BIGUN, J.. **Real-Time Face Detection and Motion Analysis With Application in " Liveness " Assessment**. IEEE Transactions on Information Forensics and Security, 2(3):548–558, 2007.

58 PAN, G.; PAN, G.; SUN, L.; SUN, L.; WU, Z.; WU, Z.; LAO, S. ; LAO, S.. **Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera**. In: IEEE 11TH INTERNATIONAL CONFERENCE ON COMPUTER VISION, Rio de Janeiro, Brazil, 2007. IEEE.

59 BAO, W.; LI, H.; LI, N.; JIANG, W. ; FIELD, A. O. F.. **A Liveness Detection Method for Face Recognition Based on Optical Flow Field**. In: INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND SIGNAL PROCESSING, p. 0–3, Taizhou, China, 2009. IEEE.

60 JING, B. Z.; CHAN, P. P.; NG, W. W. ; YEUNG, D. S.. **Anti-spoofing system for RFID access control combining with face recogntion**. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, ICMLC 2010, volumen 2, p. 698–703, Qingdao, China, 2010. IEEE.

61 YAN, J.; ZHANG, Z.; LEI, Z.; YI, D. ; LI, S. Z.. **Face liveness detection by exploring multiple scenic clues**. In: 12TH INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION, ICARCV 2012, p. 188–193, Guangzhou, China, 2012. IEEE.

62 ANJOS, A.; CHAKKA, M. M. ; MARCEL, S.. **Motion-based countermeasures to photo attacks in face recognition**. IET Biometrics, 3(3):147–158, 2014.

63 SHAO, R.; LAN, X. ; YUEN, P. C.. **Deep Convolutional Dynamic Texture Learning with Adaptive Channel-discriminability for 3D Mask Face Anti-spoofing**. In: IEEE INTERNATIONAL JOINT CONFERENCE ON BIOMETRICS, p. 748–755, Denver, CO, USA, USA, 2017. IEEE.

64 FENG, L.; PO, L.-M.; LI, Y.; XU, X.; YUAN, F.; CHEUNG, T. C.-H. ; CHEUNG, K.-W.. **Integration of Image Quality and Motion Cues for Face Anti-spoofing**. J. Vis. Comun. Image Represent., 38(C):451–460, 2016.

65 XU, Z.; LI, S. ; DENG, W.. **Learning temporal features using LSTM-CNN architecture for face anti-spoofing**. In: Kuala Lumpur, M., editor,

3RD IAPR ASIAN CONFERENCE ON PATTERN RECOGNITION, p. 141–145. IEEE, 2016.

66 LIU, Y.; JOURABLOO, A. ; LIU, X.. **Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision**. In: THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2018.

67 OJALA, T.; PIETIKÄINEN, M. ; MÄENPÄÄ, T.. **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):971–987, 2002.

68 AHONEN, T.; MEMBER, S.; HADID, A.; MEMBER, S. ; PIETIKA, M.. **Face Description with Local Binary Patterns: Application to Face Recognition**. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 28(12):2037–2041, 2006.

69 KANNALA, J.; RAHTU, E.. **BSIF: Binarized statistical image features**. 21st International Conference on Pattern Recognition (ICPR), (Icpr):1363–1366, 2012.

70 SAAD, M. A.; BOVIK, A. C. ; CHARRIER, C.. **Blind image quality assessment: a natural scene statistics approach in the DCT domain.** IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 21(8):3339–52, 2012.

71 KAN, M.; SHAN, S.; CHANG, H. ; CHEN, X.. **Stacked progressive auto-encoders (SPAE) for face recognition across poses**. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, p. 1883–1890, 2014.

72 LECUN, Y.; BOTTOU, L.; BENGIO, Y. ; HAFFNER, P.. **Gradient-based learning applied to document recognition**. Proceedings of the IEEE, 86(11):2278–2323, 1998.

73 GUO, Y.; LIU, Y.; OERLEMANS, A.; LAO, S.; WU, S. ; LEW, M. S.. **Deep learning for visual understanding: A review**. Neurocomputing, 187:27–48, 2016.

74 XUE-WEN CHEN; XIAOTONG LIN. **Big Data Deep Learning: Challenges and Perspectives**. IEEE Access, 2:514–525, 2014.

75 ZEILER, M. D.; FERGUS, R.. **Stochastic Pooling for Regularization of Deep Convolutional Neural Networks**. CoRR, abs/1301.3, 2013.

76 HE, K.; ZHANG, X.; REN, S. ; SUN, J.. **Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9):1904–1916, 2015.

77 OUYANG, W.; LUO, P.; ZENG, X.; QIU, S.; TIAN, Y.; LI, H.; YANG, S.; WANG, Z.; XIONG, Y.; QIAN, C.; ZHU, Z.; WANG, R.; LOY, C. C.; WANG, X. ; TANG, X.. **DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection**. CoRR, abs/1409.3, 2014.

78 SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I. ; SALAKHUTDINOV, R.. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**. Journal of Machine Learning Research, 15:1929–1958, 2014.

79 PAN, S. J.; YANG, Q.. **A Survey on Transfer Learning**. IEEE Trans. on Knowl. and Data Eng., 22(10):1345–1359, 2010.

80 DENG, L.. **A tutorial survey of architectures, algorithms, and applications for deep learning**. APSIPA Transactions on Signal and Information Processing, 3(2014):1–29, 2014.

81 MASCI, J.; MEIER, U.; CIRESAN, D. ; SCHMIDHUBER, J.. **Stacked convolutional auto-encoders for hierarchical feature extraction**. In: PROCEEDINGS OF THE 21TH ICANN, volumen 6791 LNCS, p. 52–59, 2011.

82 DUMOULIN, V.; VISIN, F.. **A guide to convolution arithmetic for deep learning**. arXiv preprint arXiv:1603.07285, 2016.

83 BOSER, B. E.; GUYON, I. M. ; VAPNIK, V. N.. **A Training Algorithm for Optimal Margin Classifiers**. In: PROCEEDINGS OF THE FIFTH ANNUAL WORKSHOP ON COMPUTATIONAL LEARNING THEORY, COLT '92, p. 144–152, New York, NY, USA, 1992. ACM.

84 CORTES, C.; VAPNIK, V.. **Support-Vector Networks**. Mach. Learn., 20(3):273–297, 1995.

85 SCHÖLKOPF, B.; SMOLA, A. J.; WILLIAMSON, R. C. ; BARTLETT, P. L.. **New Support Vector Algorithms**. Neural Comput., 12(5):1207–1245, 2000.

86 SCHÖLKOPF, B.; PLATT, J. C.; SHAWE-TAYLOR, J. C.; SMOLA, A. J. ; WILLIAMSON, R. C.. **Estimating the Support of a High-Dimensional Distribution**. Neural Comput., 13(7):1443–1471, 2001.

87 TOGNERI, R.; PULLELLA, D.. **An overview of speaker identification: Accuracy and robustness issues**. IEEE Circuits and Systems Magazine, 11(2):23–61, 2011.

88 THEODORIDIS, S.. **Machine Learning: A Bayesian and Optimization Perspective**. Academic Press, Inc., Orlando, FL, USA, 1st edition, 2015.

89 BILMES, J.. **A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models**. Technical report, INTERNA TIONAL COMPUTER SCIENCE INSTITUTE, Berkeley , California, 1998.

90 ANJOS, A.; EL-SHAFEY, L.; WALLACE, R.; GÜNTHER, M.; MCCOOL, C. ; MARCEL, S.. **Bob: A Free Signal Processing and Machine Learning Toolbox for Researchers**. In: PROCEEDINGS OF THE 20TH ACM INTERNATIONAL CONFERENCE ON MULTIMEDIA, MM '12, p. 1449–1452, New York, NY, USA, 2012. ACM.

91 BOULKENAFET, Z.; KOMULAINEN, J. ; HADID, A.. **Face Spoofing Detection Using Colour Texture Analysis**. IEEE Transactions on Information Forensics and Security, 11(8):1818 – 1830, 2016.

92 BOULKENAFET, Z.; KOMULAINEN, J. ; HADID, A.. **Face anti-spoofing based on color texture analysis**. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2015, p. 2636–2640. IEEE, 2015.

93 CHOLLET, F.; OTHERS. **Keras**. `https://keras.io`, 2015.

94 ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y. ; ZHENG, X.. **TensorFlow: Large-scale machine learning on heterogeneous systems**, 2015. Software available from tensorflow.org.

95 ZEILER, M. D.. **ADADELTA: an adaptive learning rate method**. CoRR, abs/1212.5701, 2012.

96 CHANG, C.-C.; LIN, C.-J.. **LIBSVM: A library for support vector machines**. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

97 INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Information technology — Biometric presentation attack detection — Part 3: Testing and reporting**, 2016.

A
ROC Curves of the Inter-Database Evaluation Protocol

Below, the Receiver Operating Characteristic (ROC) curves obtained in the evaluation protocol are presented. ROC curves for PAD schemes the on the testing database when trained on different databases. PAD techniques are grouped based on the feature used.

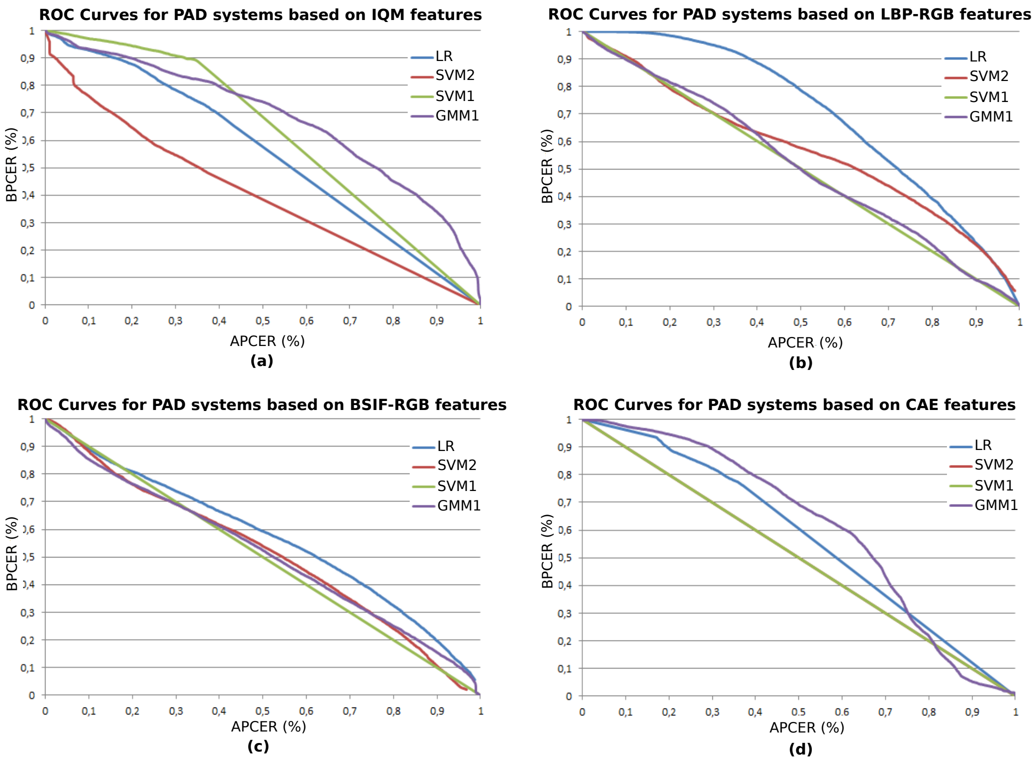

Figure A.1: ROC curves for PAD systems on REPLAY-MOBILE database when trained on 3DMAD database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

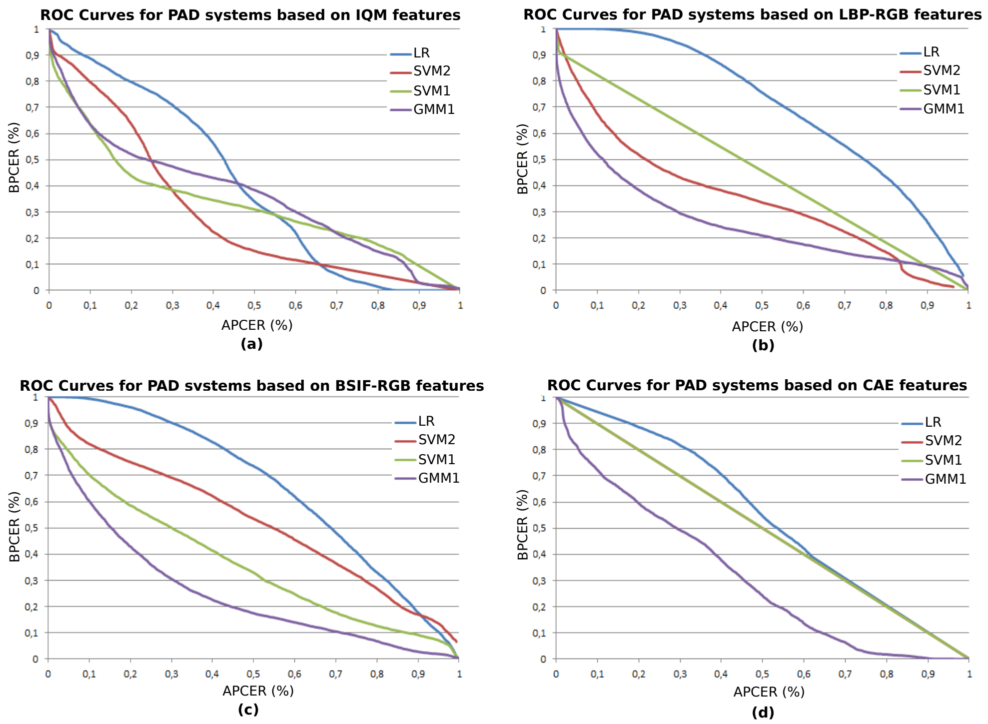

Figure A.2: ROC curves for PAD systems on REPLAY-MOBILE database when trained on OULU-NPU database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

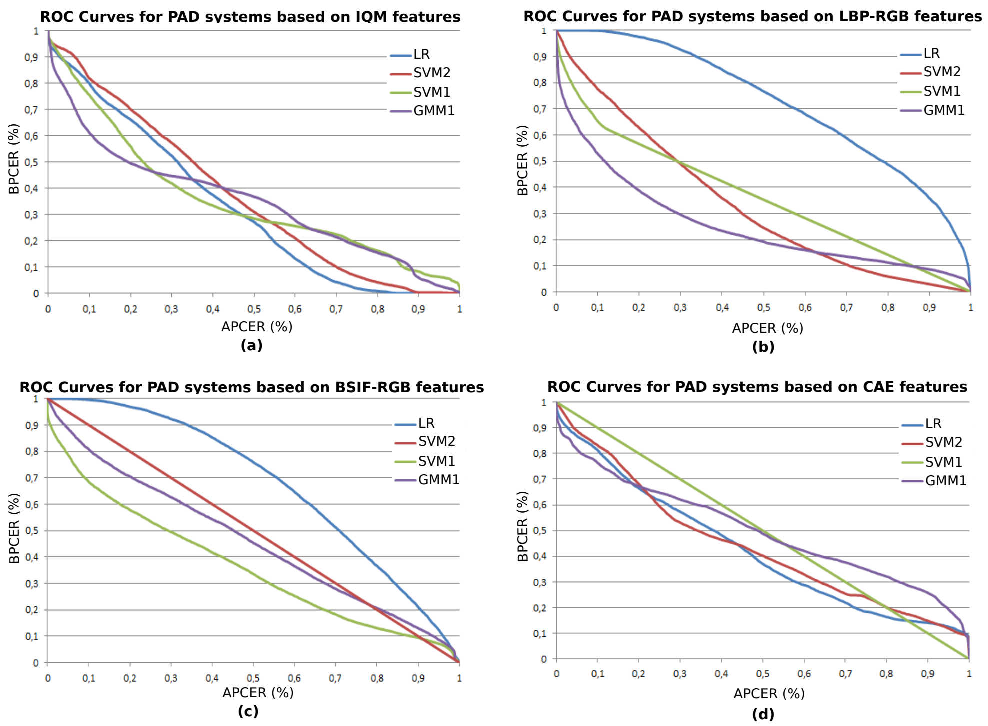

Figure A.3: ROC curves for PAD systems on REPLAY-MOBILE database when trained on the combination of 3DMAD and OULU-NPU databases. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

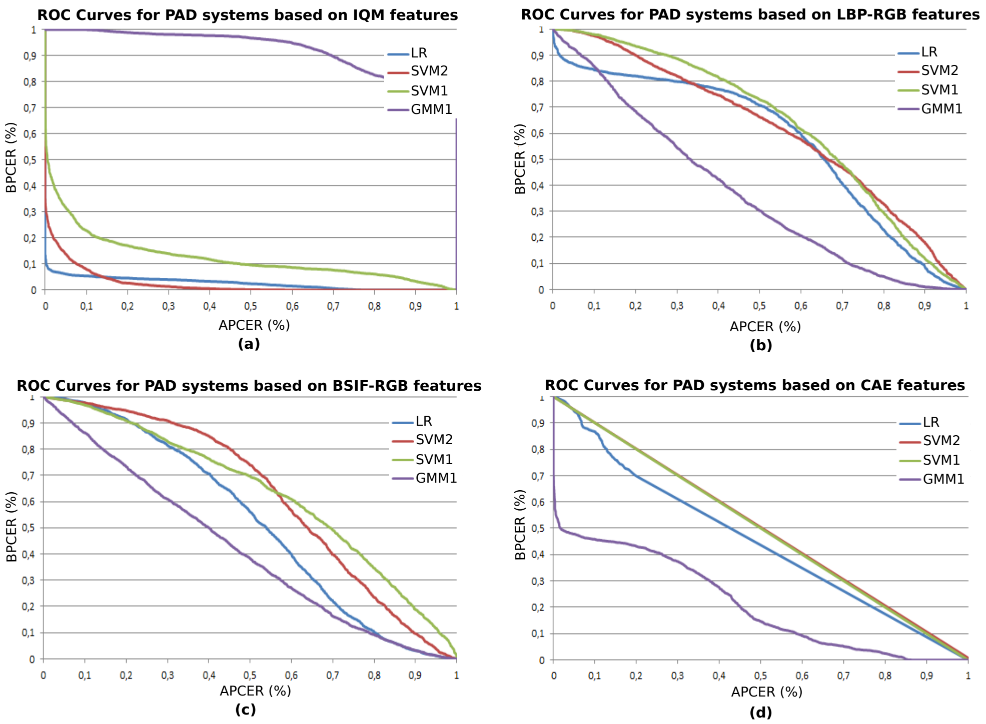

Figure A.4: ROC curves for PAD systems on 3DMAD database when trained on REPLAY-MOBILE database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

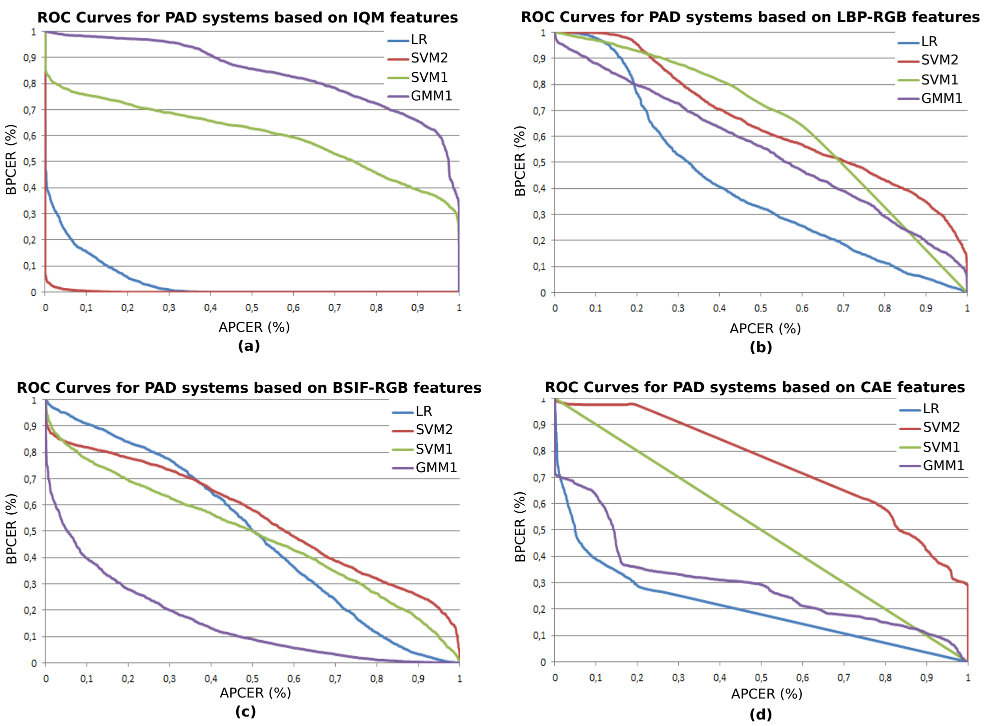

Figure A.5: ROC curves for PAD systems on 3DMAD database when trained on OULU-NPU database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

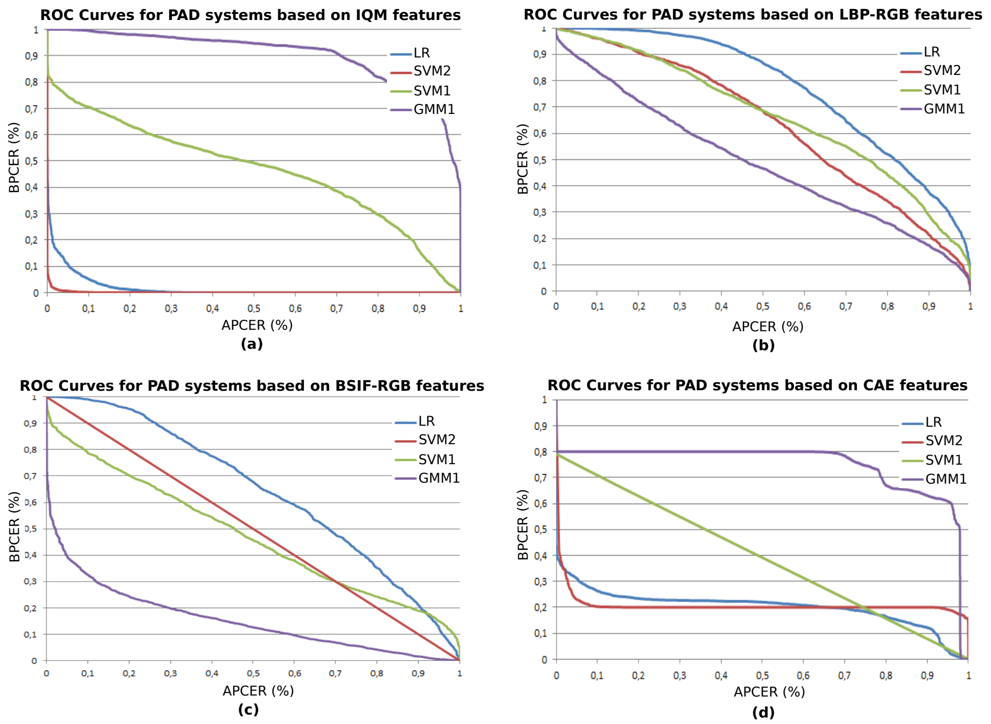

Figure A.6: ROC curves for PAD systems on 3DMAD database when trained on the combination of REPLAY-MOBILE and OULU-NPU databases. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

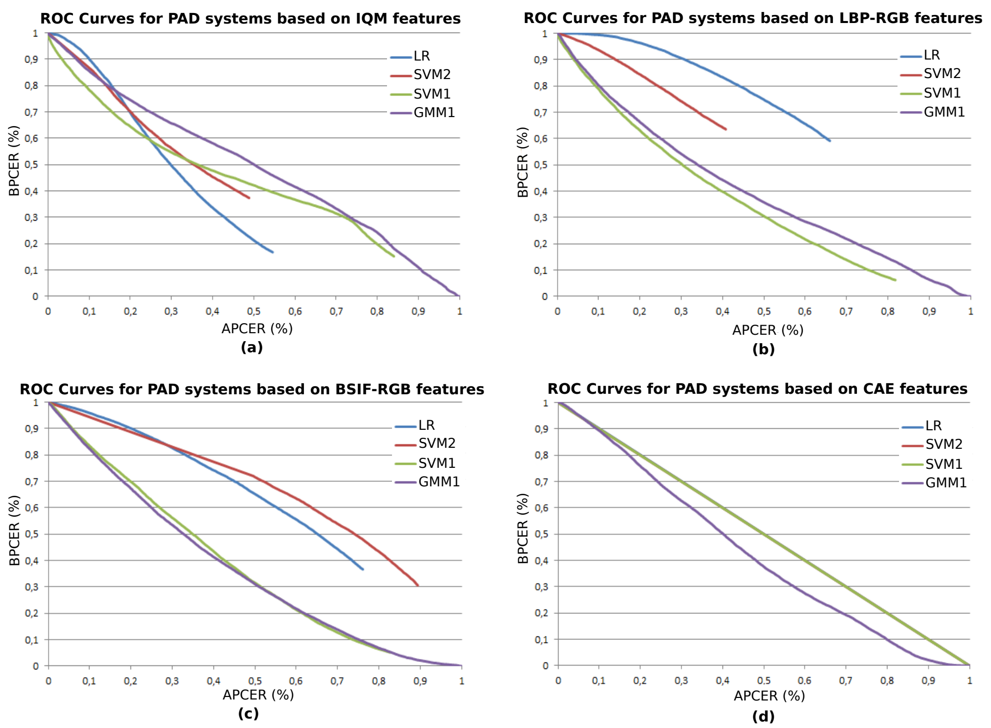

Figure A.7: ROC curves for PAD systems on OULU-NPU database when trained on REPLAY-MOBILE database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

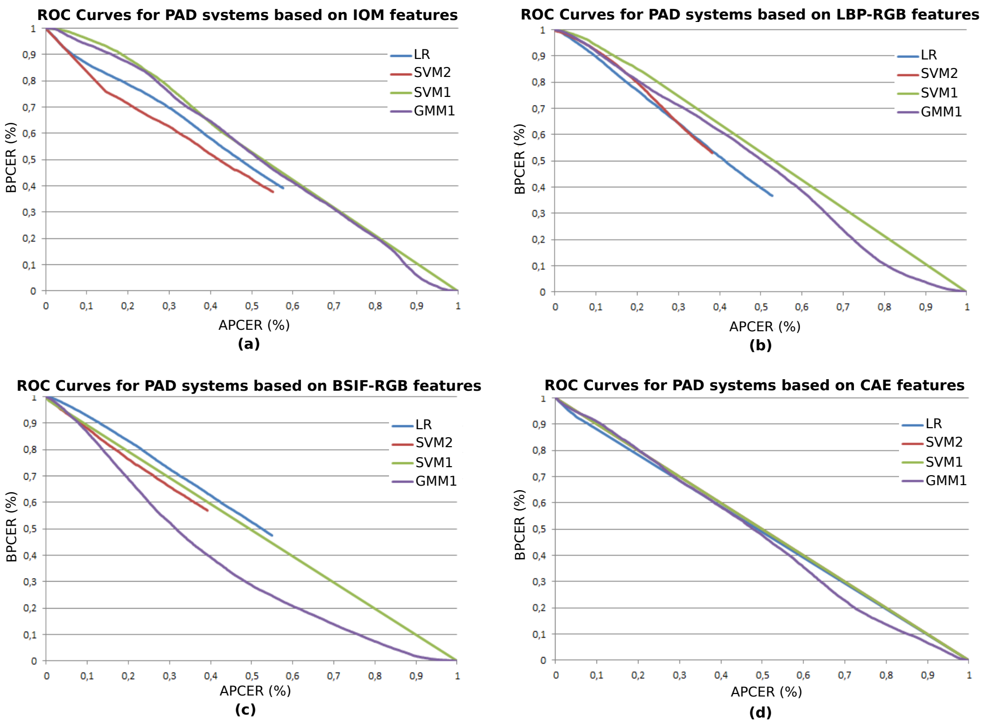

Figure A.8: ROC curves for PAD systems on OULU-NPU database when trained on 3DMAD database. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.

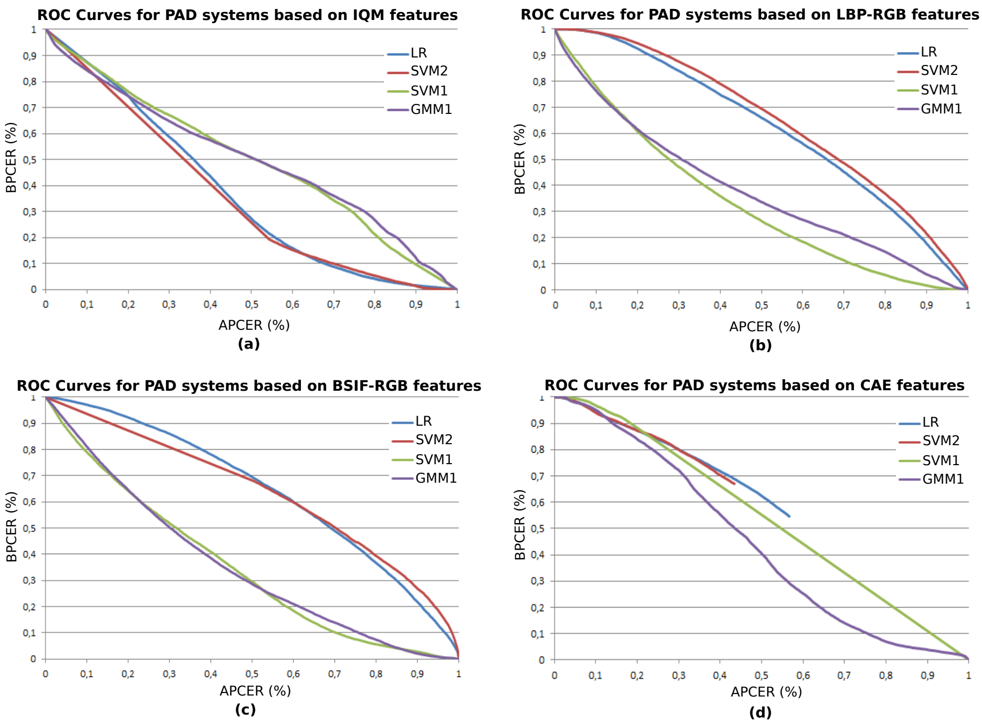

Figure A.9: ROC curves for PAD systems on OULU-NPU database when trained on the combination of REPLAY-MOBILE and 3DMAD databases. PAD systems based on: (a) IQM, (b) LBP-RGB, (c) BSIF-RGB, and (d) CAE features.