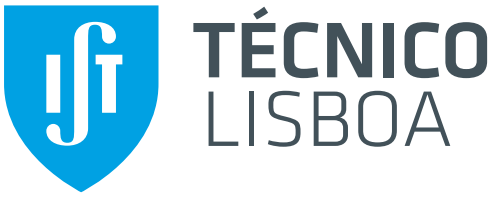


Detection of Attacks to Face Recognition Systems

Daniel Bento de Sousa

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Paulo Luís Serras Lobato Correia

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Paulo Luís Serras Lobato Correia
Member of the Committee: Prof. Ana Luísa Nobre Fred

November, 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

I would like to thank my parents, in the first place, because of everything they have ever done for me, believing in me since the very start and for their incredible support. Providing with everything I ever wanted and needed, for loving and taking care of me. I just truly hope I always make you proud. Secondly I would like to thank my friends, for helping me get through this tough time, giving me motivation and helping me during times of need, I will treasure all these moments forever. To my girlfriend, for her unceasing presence in my life during these times, hope it never changes. There were a lot of times I was a pain in the neck but you helped me cut through thick and thin, together, as always. I owe you very much.

I would also like to thank my supervisor, prof. Paulo, for the given opportunity, for his trust and guidance during this thesis. Special thanks to his advice, infinite patience and unceasing support. Finally, a thank you note to Instituto de Telecomunicações (IT), which I was working to, for providing me with the necessary tools to accomplish this thesis, as well as a scholarship, under the name ”Research Grant - BIL/Nº25 - 19/04/2018 - UID/EEA/50008/2013 - SeLF-ICN”, showing the great value of my work and dedication. I hope I have kept up with the expectations.

Resumo

Sistemas de reconhecimento facial são cada vez mais relevantes na sociedade actual, sendo maioritariamente utilizados como medida de segurança. Desde itens do quotidiano, como os telemóveis e computadores, até sistemas cuja segurança é crucial, como o controlo de acesso aos aeroportos, são exemplos da utilidade do reconhecimento facial. Devido à sua popularidade, estes sistemas biométricos são vulneráveis a uma vasta gama de ataques, que se têm tornado cada vez mais complexos. Assim, o desenvolvimento de contra-medidas eficazes é necessário.

O objectivo desta tese é o desenvolvimento de uma ferramenta de deteção de intrusões ao nível do sensor, conhecidas como Ataques de Apresentação (AA). Para este propósito, as contribuições do estado da arte são revistas de forma a compreender as suas principais limitações e possilidades de melhoria. Quando comparado com métodos anteriores, o deep learning atinge um alto desempenho, o que consequentemente tem vindo a aumentar a sua popularidade. Assim, uma abordagem baseada em transferência de conhecimento usando o modelo de uma rede neuronal previamente treinada é apresentada. Esta rede é então adaptada e optimizada de acordo com o problema. Ao longo do processo, uma nova abordagem implementando uma camada que se baseia na análise de vídeo é proposta. Esta distingue-se da usual análise de frame a frame, sendo usada para classificação de ações, e é conhecida como camada de Long Short Term Memory (LSTM). Para além disso, uma comparação com outros algoritmos do estado da arte é feita, o método proposto atinge uma metade da taxa total de erro de 1,09%, na base de dados Replay-Attack.

Finalmente, uma conclusão sobre a deteção de ataques a sistemas de reconhecimento facial é traçada, constatando a razão pela qual este tópico é ainda um problema em aberto, apesar do alto desempenho atingido por vários algoritmos do estado da arte em bases de dados exigentes.

Palavras-chave: Reconhecimento Facial, Sistemas Biométricos, Ataques de Apresentação, Transferência de Conhecimento, Rede Neuronal, LSTM

Abstract

Face recognition systems are increasingly important in today's society, being mainly employed as a security measure. Everyday items, such as mobile phones and laptops, or more crucial security systems, such as the airport access control, are examples of face recognition usages. Due to its popularity, these biometric systems are vulnerable to a wide range of attacks, which are becoming more and more complex. Therefore the development of effective counter-measures is necessary.

The objective of this thesis is to develop a tool which detects intrusions at the sensor level, known as Presentation Attacks (PA). For this, state of the art contributions are reviewed in order to understand their main limitations and possibilities of improvement. When compared to older methods, deep learning achieves a high performance, which consequently has increased its popularity. Thus, an approach based on transfer learning using a pre-trained Convolutional Neural Network (CNN) model is presented. This network is then adapted to the problem and several steps are taken to optimise it accordingly. Along the process, a novel approach is proposed by implementing a layer that performs video analysis for action classification instead of the regular frame-by-frame analysis, known as Long Short Term Memory (LSTM). Furthermore, a comparison against other state of the art algorithms is made, where the proposed method achieves a half total error rate (HTER) of 1.09% in the Replay-Attack database.

Finally, a conclusion is made about the detection of attacks to facial recognition systems and why is it still an open problem, even though state of the art methods show a high performance in such demanding databases.

Keywords: Face Recognition, Biometrics, Presentation Attacks, Transfer Learning, CNN, LSTM

Contents

List of Tables

List of Figures

Acronyms

CNN Convolutional Neural Network

DL Deep Learning

FAR False Acceptance Rate

FC Fully Connected

FRR False Rejection Rate

HOG Histogram of Oriented Gradients

HTER Half Total Error Rate

IBG International Biometric Group

LBP Local Binary Patterns

LFC Light Field Camera

LSTM Long-Short Term Memory

ML Machine Learning

NIR near infra-red

PA Presentation Attack

PAD Presentation Attack Detection

PAI Presentation Attack Instrument

RNN Recurrent Neural Networks

SVM Support Vector Machine

Chapter 1

Introduction

Biometric recognition is increasingly important in today's society. The term biometrics refers to metrics related to human characteristics. In this case, it represents a set of biological and behavioural unique identifiers which can be used as a form of identification and access control, such as faces, voices, fingerprints, signatures, gait, among others. Biometric features and processes have been used for quite some time. Humans can recognize each other by face or voice and even in the animal world some species use each other's individual scent in a similar way.

Nowadays the expectations are higher, researchers from many different fields such as image processing, computer vision or pattern recognition, have applied several different techniques [1] to improve the performance of biometric systems [2], this has permitted the use of biometrics in many diverse activities such as forensics, border and access control, surveillance and on-line commerce.

1.1 Motivation

Face recognition together with fingerprint scanning are among the most used technologies for implementing security measures based on biometrics. However, face recognition systems are the most frequent targets of attacks [1], which proves that there exists a need to build a fail safe system that allows to securely "keep the prize" away from intruders.

Right after the fingerprint, the face is the second most largely deployed biometric at world level in terms of market quota (according to the International Biometric Group (IBG) [3]). It is used worldwide not only because of the biometric passport (e-passport), which allows people to enter countries by simply comparing their face with the passport's picture, but also because of the increasingly presence of biometric applications for personal computers and mobile phones to an extent of opening/accessing your personal bank account using face identification [4]. Furthermore, even in developing countries are now using biometric technologies to create national identification programs. The India Unique Identification Authority is creating and providing an unique ID for every Indian resident based on facial recognition and fingerprint scanning in order to replace the old fashioned ID card. For all of the previous reasons, the face is one of the biometrics where most spoofing-related research has been conducted.

Unfortunately, these biometric systems are all vulnerable to different kinds of attacks as it is possible to see in Figure 1.1, being the focus of this work the Presentation Attack (PA) at the sensor, where a person tries to masquerade as another by creating a fake biometric trait of the user and presenting it to the sensor, thus posing as the real user. The other type of attacks usually fall on the hacking predicament, where the biometric sample can be altered mid processing, or different points of the process could be overridden as the comparator or even the database, for example. This work focuses particularly in trying to determine if the presented trait originated from a real legitimate client or not. This can be seen as liveness detection, since this term is quite often used as a close synonym for spoof detection in some fields, but, usually, liveness detection refers to a more limited problem of sensing vitality signs of the user, like the heartbeat or eye blinking [5]. In this thesis this term is treated as a subcategory for Presentation Attack Detection (PAD) methods. It is important to note that this type of attacks depend on the resolution of the device, or on the type of support used to present the fake copy (handheld or fixed support), also known as artefact. The external variability, such as illumination or background conditions, can also heavily influence the outcome.

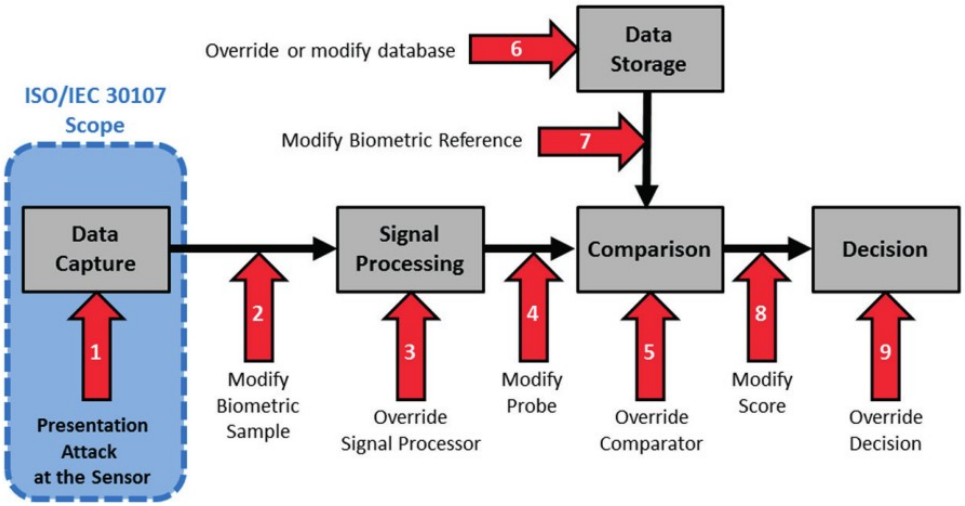


Figure 1.1: Different points of attacks in a biometric system. [5]

Furthermore, in each scan there exists an extremely vast layer of data, which needs to, ideally, be processed in a real-time scenario. Several PAD techniques depend on extracting features that are able to detect a presentation attack, usually simple features are extracted, processed and never connected. In some cases, correlation of these techniques could lead to an improvement on the overall outcome [6], this is specially hard to do since it is not possible to determine which techniques can be improved or not. Nowadays, this task is eased by the uncovering potential of Deep Learning (DL). Such algorithms are able to extract several distinct features that are deemed important, usually by the algorithm itself, although the programmer can have some influence, and successfully find correlations between these in order to produce a successful outcome. Sometimes, to optimize the network used, some features that are deemed useless to find the correct outcome can be discarded. These Machine Learning (ML) tools are considered powerful since it speeds up the entire process, can be trained and learn by their

mistakes, and can even find unexplained/unpredictable connections that help fulfil the task at hand [7].

1.2 Objectives

The objective of this work is to analyse what are the most employed type of attacks in this type of biometric systems and how can a system be protected against these, building a tool that can successfully detect said attacks. In order to do this, current state of the art methods are detailedly reviewed and compared, finding its most common shortcomings which are usually taken advantage by imposters. This knowledge is then used to develop a software based approach to improve robustness of face authentication systems to presentation attacks. The presented approach is a deep learning algorithm which analyses a small sequence of frames instead of the usual single frame analysis done by most state of the art algorithms, proving that a video analysis is fruitful to the problem.

1.3 Thesis Outline

In the remainder of this thesis, a presentation attack detection method based on deep learning is proposed and explained. The document is organized in the following manner:

- Chapter 2 contains a detailed description of state of the art in the PAD problem, generalising the different types of attacks to face recognition systems and the preferred techniques to detect these attacks;
- Chapter 3 presents the proposed algorithm, detailing its architecture and explaining how the used network was gradually constructed;
- Chapter 4 discusses the used experimental setup in order to fully construct the algorithm. What type of tools and frameworks are needed and utilised, discussing as well the different databases employed in training and testing the network;
- Chapter 5 displays the obtained results when using the proposed algorithm in chapter 3, validating its implementation and comparing with other state of the art methods described in chapter 2;
- Finally, in chapter 6 conclusions are drawn from the carried out work, explaining the obtained accomplishments and making references to future contributions that can be done.

Chapter 2

Presentation Attack Detection - State of the Art

This chapter is divided into three major sections. Firstly, the basic type of presentation attacks and the pre-requisites of a presentation attack method are reviewed, followed by the several types of different attacks, where the most are examined, explaining the points of exploitation they seek in the system. Section 2.1 is subdivided into three subsections where in each, a different type of attack is analysed. Afterwards, section 2.2 reviews the most used presentation attack detection methods, how are they segmented in several categories depending on their aim, discussing their advantages, disadvantages, being subdivided depending on the type of analysis: User Interaction, Image Sensor, Contextual Information and Feature Extraction. Lastly, in section 2.3, it is discussed how the recent use of machine learning improved PAD methods and why is it used so much nowadays, referencing some methods that were deemed important as it brought new techniques that try to solve the problem at hand.

As previously mentioned and as it can be seen in Figure 1.1, there are several possible attacks in a biometric system. These attacks can be divided into two classes [8]: indirect and direct attacks. The first one is performed inside the biometric system, usually by hackers, who successfully enter the system and manage to tamper with it, changing the database or bypassing the feature extractor/matcher. Firewalls, encryption and anti-virus are usually the securities applied to the hardware side of the system in order to prevent these kinds of attacks.

On the other hand, direct attacks, which are also named as presentation attacks, consist in the ability to fool the biometric system into recognizing an illegitimate user as a genuine one, by presenting a synthetic forged version of the original biometric trait to the sensor, which is labelled as 1 in Figure 1.1.

In addition, presentation attacks can be subdivided in two [5]: an **active impostor** presentation attack in which the attacker intends to be recognized as a different individual. This can, in turn, have one of the two objectives, the subject intends to be recognized as a specific user in the system, also known as impersonation, or it simply wants to pass as any other individual in the database; a **concealer** presentation attack in which the user simply does not want to be recognized as someone from the database of the system, also known as obfuscation. For example in a no flight list, where the imposter

might be forbidden to fly and can be detected during security check. Thus, the presentation attack can be conducted on the biometric system with the intent to gain access to the services provided to a real user or to hide an identity from being revealed.

A Presentation Attack Detection method has the ultimate goal to automatically distinguish between real biometric traits presented to the sensor and artificially forged artefacts which contain a replicated biometric trait of the genuine user. All of these methods are expected to fulfil the following pre-requisites [9]: **non-invasive**, it should not break the user interaction boundaries; **user friendly**, ought to work instinctively with as few user interactions as possible; **fast**, as it should converge to an acceptable solution with a high speed; **low cost**, for implementation sake and world wide accessibility, and **good performance**. In the next section the most commonly used Presentation Attacks to biometric systems are introduced.

2.1 Types of Face Presentation Attacks

Before discussing the different algorithms for PAD, it is important to also introduce and discuss the different categories and the most common PA techniques. It is essential to fully understand what are the most common types of presentation attacks, how do they work and what vulnerabilities of the system they seek. In the following subsections, these attacks are divided in three types: photo, video and mask attacks. The explanation and difference between these types of intrusions and some examples are also given. However, it is important to remark that even though these are the most frequent PA's, there are several other and/or variations of each discussed attack. This only shows how serious and difficult this problem is since new ways to overcome protection very much rely on the imagination and effort of the attackers, each system can be exploited in different ways or have an unknown weakness until now.

The biometric characteristic or object used in a presentation attack is referred to as Presentation Attack Instrument (PAI) [5]. As it can be seen in Figure 2.1, these instruments can be divided into two types: (1) Artificial and (2) Human. The first one refers to an artificial means when generating the PAI and can be subdivided into two more categories, (i) Complete and (ii) Partial; these differ in the way that the instrument is represented; one shows the complete artificial PAI, for example, a replay-attack, while the other one simply mimics partial biometric characteristics of the original user, such as make-up. In (2) it consists of using a human trait as a PAI and can be (i) Lifeless, like a severed body part; (ii) Altered, including cosmetic surgery to alter a biometric trait; (iii) Non-Conformant, where different facial expressions or deliberately partial biometric traits are given to the sensor in order to find flaws, as a covered face; (iv) Coerced, where the genuine subject is presented to the sensor while in distress, being forced to; (v) Conformant, which includes zero-effort imposter attempts such as the case of twin siblings [10]. The attacks discussed below are all in the artificial category, this does not mean that other types cannot be used, it simply states that, due to practicality and technology nowadays, these are the most common.

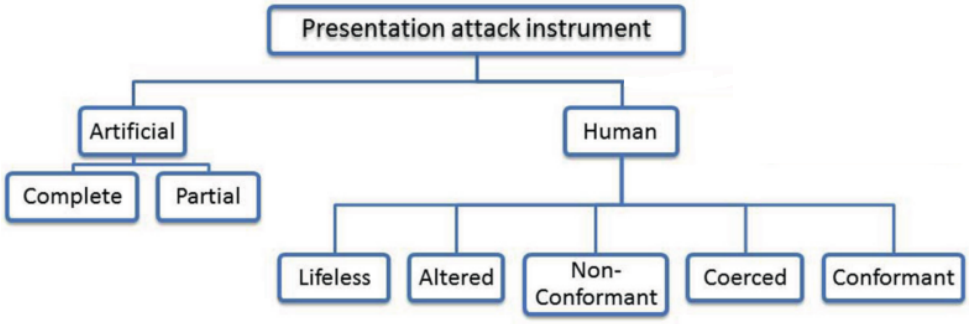


Figure 2.1: Different PA devices according to the International Standard ISO/EIC [5].

2.1.1 Photo Attacks

Photo attacks are carried out by presenting to the sensor a photograph of the original user. Such images can be printed out on a paper (print attacks), or displayed on the screen of a digital device (digital-photo attacks) as illustrated in Figure 2.2. There exists also the use of photographic masks, which are high-resolution printed photographs where the eyes and mouth are cut out; this happens so the impostor can wear it and portray natural movements, such as the iris movement or eye blinking and mouth twisting, to mimic natural human behaviours.

Figure 2.2: Spoofing attack with a hand-held photo, also referred to as a digital-photo attack. [11]

Even when a countermeasure can detect a plain photo-attack, the attacker can fool this by simulating facial and head movements by translating, rotating and warping a face print, which poses new challenges to motion based methods as illustrated in Figure 2.3.

Figure 2.3: Example sequence of a warped photo attack from the CASIA Face Anti-Spoofing Database. [12]

2.1.2 Video Attacks

Video Attacks may be also referred to as replay attacks, and this type of intrusions consists in the attacker replaying a video of the genuine user using a suitable device. These attacks are extremely difficult to detect, since not only the 2D face texture is captured but also the dynamics and movement of the genuine user. Even though these types of attacks are a more serious problem when compared to photo-attacks, high-quality videos of a targeted person are much more difficult to acquire compared to a frontal face photograph (easily acquired nowadays through social media). However, commercial animation software [13] can also be used for such a goal, creating an animation from a 2D image, exhibiting realistic liveness characteristics and motion.

2.1.3 Mask Attacks

In this intrusion type, a 3D mask of the genuine user is created and used to spoof the security system. There are several types of 3D masks but essentially they all capture completely the 3D model of the original person in detail so the normal use of depth cues, which can be used as a countermeasure to the previous two types of spoofing attacks, become ineffective in this kind of attacks. Mask spoofing is viewed as a high quality technology presentation attack since several resources to create a 3D mask model of the genuine user are needed, which is not cheap and the higher the quality of such mask the higher the chances of spoofing the system and more expensive the apparel. An example of such attack can be seen in Figure 2.4.

There are several types of masks, such as latex or silicone, or even simpler and cheaper synthetic appliances. The prices of these may range between 30€ to 650€, depending on the material and quality of the artefact [14]. These constraints make this type of spoofing less frequent than the previous two, but more problematic as they are harder to detect.

Figure 2.4: 3D silicone face mask ordered from `www.thatsmyface.com`. [11]

As expected, these different variations of attacks all depend on the characteristics, such as the resolution of the spoofing device or on the type of support used to present the artefact (fixed support or hand-held). The external variability also influences the conditions of said outcome since it may depend on the illumination and background circumstances.

2.2 Presentation Attack Detection Methods

As discussed in the section above, facial recognition systems are vulnerable to various presentation attacks, through several presentation attack instruments that can be nowadays generated with a high cost-effectiveness, making identity verification and recognition using facial information among the most active and challenging areas in computer vision. Even to this day, there are still ongoing major research challenges as the ageing of subjects, the complete change of facial style (e.g glasses, facial hair) or the complex outdoor lightning conditions among others. Some algorithms can achieve acceptable levels, with a low error rate (topic discussed in more detail later on, in section 4.1), and are good enough for consumer level applications. In this section some of the most used PAD methods are described in detail focusing on the different types of analysis made by each algorithm, subdividing these in different categories.

A PAD method can be defined as an *automated determination of a presentation attack* [5]. As illustrated in Figure 2.5, PAD methods can apply four different dimensions: User Interaction Support, Imaging Sensor, Contextual Information and Feature Extraction [15]. The combination of such dimensions is often used in order to achieve a more complete PAD algorithm.

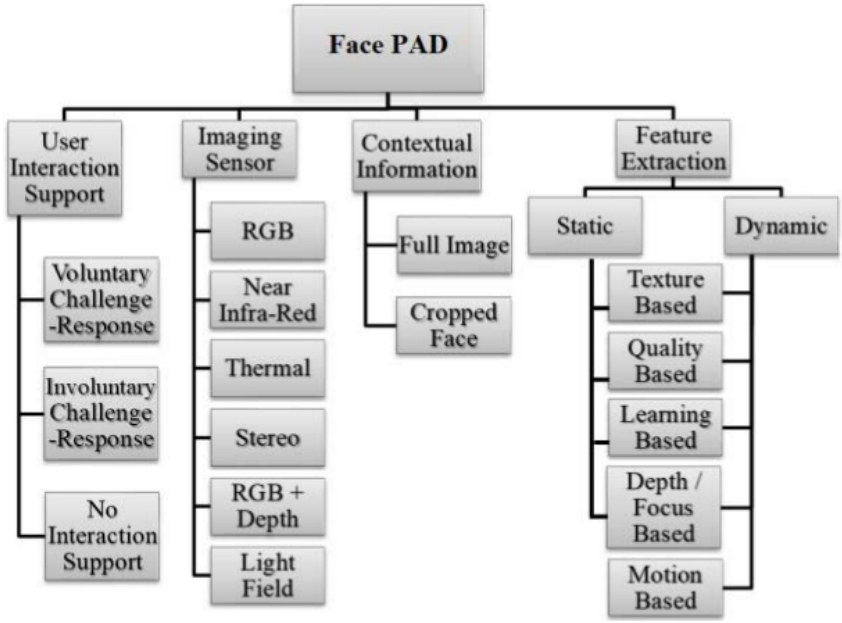


Figure 2.5: Classification of face presentation attack detection (PAD) algorithms. [15]

2.2.1 User Interaction Support

User interaction support may be applied when the user is willing to undergo a more thorough identity check. These approaches may or may not require interaction with the user itself, which then will often be used together with an analysis software to achieve a conclusion.

In such scenarios, PAD systems usually employ the so-called *challenge response* methods, which consist on presenting to the user several challenges that can be answered voluntarily or not. Voluntary

challenges would be portrayed as a pre-determined set of movements that are used to obtain several angles and aspects of the individual. On the other hand, involuntary challenges consist in detecting natural responses of the human body which can even be influenced by external stimuli or not. One example of a natural involuntary response may be *blink detection*, illustrated in Figure 2.6, which consists in continuously tracking eye movement and blinking that is unconsciously made by the user; this analysis can be carried out by hardware as well as by a software [16] and the mouth movement can also be taken into consideration.

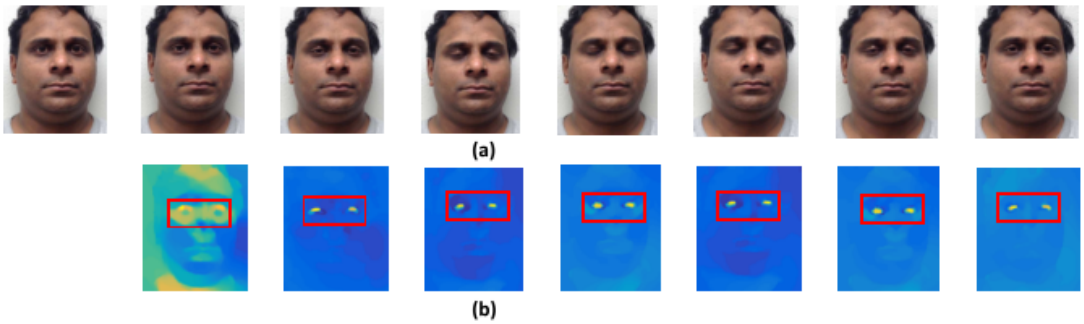


Figure 2.6: Eye blink detection as a PAD mechanism in (a) video frames and (b) the corresponding optical flow. [17]

As for involuntary challenges with stimuli, we have, for example, the natural pupillary response to changes in illumination, or the tracking of the gaze to predetermined stimuli [18]. Unfortunately, challenge response systems often fail against replay or mask attacks, with the eye region cut-out, need dedicated hardware, which increases the computation cost, and some can require frequent, troublesome user interaction, being often labelled as too intrusive. An in-depth eye scan and recognition could be made but then the problem would cease to be facial recognition and would transfer to eye verification. When trying to overcome user intrusion, imaging sensors might be a good option since they only require a single image analysis.

2.2.2 Imaging Sensor

The type of face analysis that can be applied usually depends on the type of sensor used to capture the image from the person. Typically, 2D RGB cameras are the most commonly used sensor nowadays, nevertheless, the recent availability of richer imaging sensors is opening new possibilities for improved PAD solutions. A low-cost depth sensor, such as the Microsoft Kinect, can be used to verify the existence of depth in the image which can be used to completely negate some print-attacks [19]. Another way of acquiring the sense of a depth model of the face is using a Light Field Camera (LFC) which is made up of several small cameras that capture both the direction and the intensity of incident light rays, rendering multiple images which, combined, can obtain the model of a face [20], see Figure 2.7. Another interesting sensor technique is the use of thermal imaging and near infra-red (NIR) images [21] which can be used to detect artefacts; the latter is used to capture images on a different spectrum, allowing to measure the reflectance levels of the supposed captured face and being able to distinguish between a real face and

an artefact since the human face has extremely low reflectance levels in the NIR spectrum.

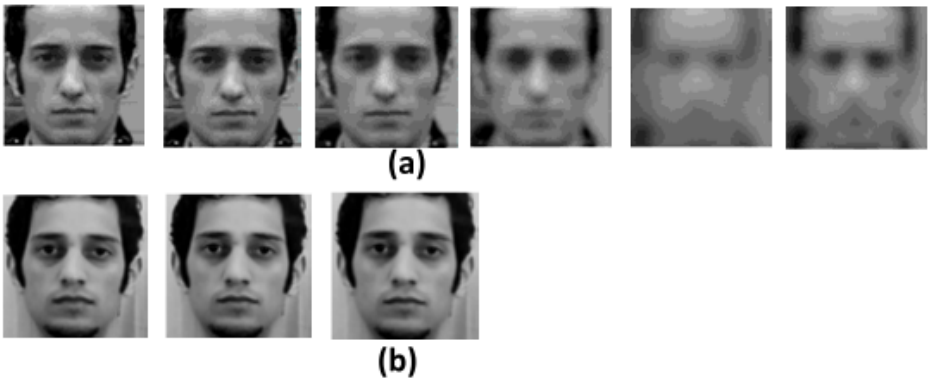


Figure 2.7: Multiple depth images rendered by a LFC being (a) on a real capture and (b) a print-attack. [20]

However, these depth sensors are completely powerless against any type of 3D mask attacks. Thermal imaging also has its flaws since, during capture of an image, it passes through materials, thus being useless against any type of wearable masks. In addition, NIR imaging is to be only used indoors in a controlled environment, since the sunlight causes severe perturbation to the sensor.

In conclusion, imaging sensor methods are not frequently used since they can be expensive and impractical, due to the installation of extra, specific hardware. Ideally, PAD methods would only need the already used camera system included in the facial capture. With this goal, a variety of other methods have been proposed, which have a need to be robust and reliable in order to detect any type of attack in different conditions of acquisition.

2.2.3 Contextual Information

This dimension is related to the possibility of using contextual information, for instance including background and scenic cues, to detect any presentation attacks. This is possible in scenarios where the image acquisition is done with a wide field of view camera and the PAD algorithm does not have to concentrate only on the cropped face region of the subject.

For example, a background analysis is possible. If the sensor is in a fixed location there is a fixed contextual information that can be used to detect any type of print attacks or handheld photographs. There may also exist motion in the background, which allows to completely determine any type of print or even replay attacks if the location in the background is different than usual [22].

2.2.4 Feature Extraction

Lastly, feature extraction schemes do not need user cooperation, have relatively low cost and can be subdivided into two classes: static or dynamic methods. Dynamic approaches usually take advantage of the temporal information from a video captured by the sensor by exploiting the motion across video frames, for example. Therefore, this type of methods require more time as well as more computational

effort compared to static based methods which only take into consideration a single captured frame by the sensor.

When taking texture based analysis into consideration, we can analyse micro-textural patterns in the captured sample and, this way, are able to successfully detect artefacts used in the attacks since it can find flaws, due to printing defects or due to the low resolution of the used artefacts. A widely used approach is based on Local Binary Patterns (LBP) [23]. It is a type of visual descriptor used for classification; it divides the image in groups of pixels and compares each pixel to its neighbours (the other 8 pixels), if a pixel is greater than its neighbours it is assigned the number 0, else 1, forming the image histogram. This way, LBP techniques are able to capture the pigments produced due to printing or the change of reflection caused by the quality of the artefacts, which can be seen in Figure 2.8. As illustrated, there is a lack of pigments in the printed attacks compared to the *bona fide* sample, in the case of an electronic display attack, the sample is better than the previous two although it is visible some screen reflection. In addition, off-the-shelf methods use LBP analysis and several variations of this base algorithm, proving to be one of the most effective PAD methods [24]. Even though it is widely used, this method does fail and lacks robustness due to the change in the acceptance threshold, which logically depends on the database and type of attack, and is only viable if the artefacts used have noticeable imperfections or the screens present a reflection, which might not be the case.

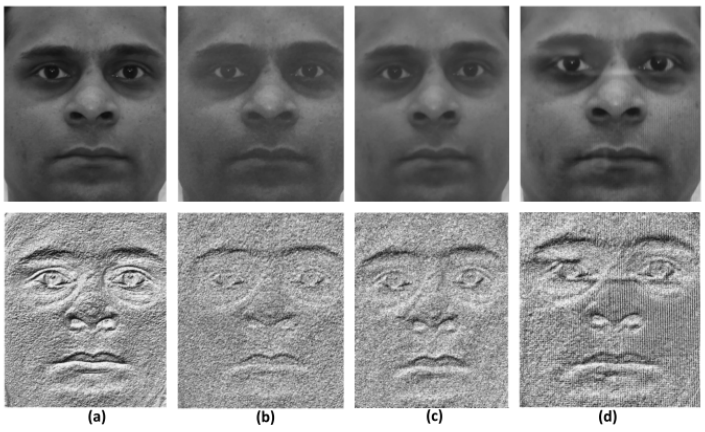


Figure 2.8: LBP as a PAD method. (a) is a *bona fide* image; (b) a laser printed artefact; (c) an ink-jet printed artefact and (d) a display attack in an iPad screen. [17]

Very similar to LBP methods and as often used, Binarized Statistical Image Features (BSIF) computes a binary code string for each pixel in an image by convolving the acquired sample with a filter and then binarizing the filter response. This binary code is considered as a descriptor of the image intensity pattern. In addition, an histogram of the pixel's code values can be obtained, which characterizes the textural properties within image subregions [25]. Another texture frequency descriptor, that is also quite similar to the LBP but specially deals with low quality and blurred images, is the Local Phase Quantization (LPQ) [26] which uses the local phase information extracted by the Short Term Fourier Transform (STFT) calculated in the neighbourhood of each pixel. Then, the basic LPQ features at each pixel are represented by a vector with the STFT of its neighbours, being each element of this vector quantized

(to restrict a continuous set of values to a discrete one) and its resulting binary quantized coefficients are represented by an integer number and collected into a histogram. The decision threshold in these two previously described techniques is made exactly as LBP, genuine faces present much more textural properties in the subregions, in case of BSIF, and higher STFT coefficients in case of LPQ.

These previous texture based techniques can be improved in several ways, one very utilized improvement is the change in colour space analysis [27]. Although RGB is the colour space used by most image acquisitions devices, it may not be the most suitable to process since RGB image analysis is quite limited. There is a high correlation between the three components (red, green and blue) and there is no separation between the luminance and the chrominance of a picture. This way, a grey-scale texture analysis was firstly used to detect any textural information of the image and it was found that it works well when there is enough spatial resolution to capture the face details. However, it fails when low resolution pictures are presented since it cannot distinguish any differences between genuine or fake faces [28]. Other methods use different colour spaces that preserve the luminance and the chrominance in its entirety of an image, such as the HSV (Hue and Saturation which have the chrominance information and Value for the luminance) or YC_bC_r (where Y is the luminance, Cb the chrominance blue and Cr the chrominance red), among others [29]. An example of these colour schemes can be seen in Figure 2.9

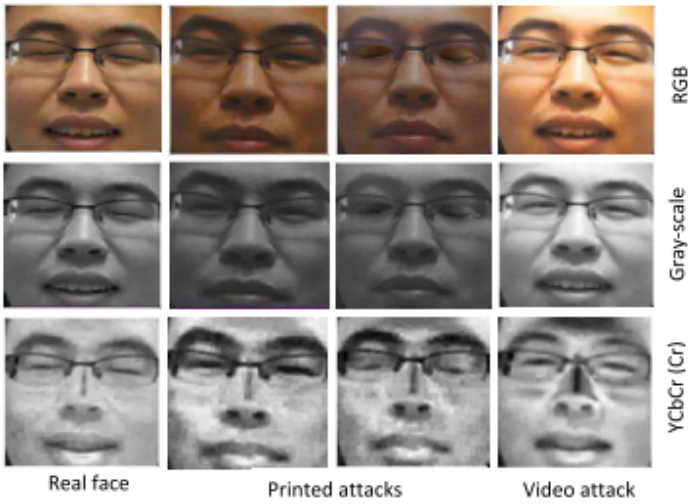


Figure 2.9: Example of a real face and the corresponding print and video attacks in RGB, grey-scale and YC_bC_r. [27]

Another type of methods worth mentioning is image quality based analysis, which differs fake faces from real ones based on a general physical model that describes the recapturing process, uses physical features and can even mix contextual background information. This concept assumes that when there is a recapture of a sample, in case of any PA, there is a loss in quality on a textural level, this method might even be improved if there is a benchmark of a captured image from the sensor to use for comparison.

In depth based analysis most methods use frequency algorithms to detect any 3D shape or part of the face. Early work was based on a simple Fourier spectrum analysis which assumes that the existence of a 3D shape leads to a difference in the low frequency region when compared to a simple face print

that happens due to illumination [30]. With this technique it is possible to identify several parts of the face, as the mouth and nose, due to the 3D frequency shape as it can be seen in Figure 2.10.

Another method is the eulerian motion magnification [31] that amplifies the frequencies within the human range and allows to magnify the slight colour and motion modifications in the human face due to the natural flowing of the blood as is even able to detect any distress the user might be experiencing because of the increasing cardiac beat.

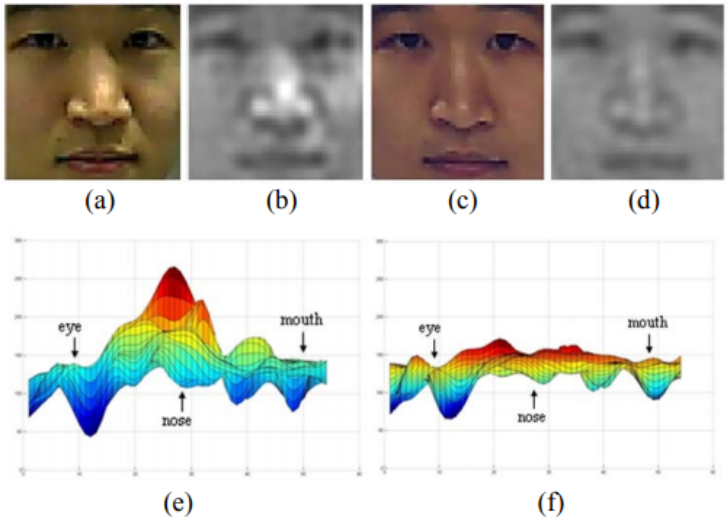


Figure 2.10: (a) is a live face image, (b) reconstructed by low frequency components. (c) is a fake face image, (d) reconstructed by low frequency components. (e) is the 3D intensity image of (b) and (f) is the 3D intensity image of (d). [30]

When using a digital screen to present the attack, it is possible to do a high-frequency analysis which becomes affected by the high brightness of the screen, blurring some pixels. Overall, this makes the fake images show less detail in the borders due to the high illumination when compared to a real face image [32]. This is possible by using a Difference of Gaussian (DoG) filter which basically consists in subtracting a blurred version of the image with another less blurred version of the same image, obtained applying different gaussian filters. This preserves the spatial information that lies between the range of frequencies of the two blurred images, detecting the borders which can be seen in Figure 2.11, working even with bad illumination conditions.

Figure 2.11: The original face with its DoG equivalent (first pair) and the recaptured image with its DoG correspondence (second pair). A loss in detail can be seen. [32]

Regarding focus analysis, variable focusing is a good example where the key approach is to utilize the variation of pixel values between two images sequentially taken with different focus levels. Such analysis is possible since in real faces focused regions are clear and others are blurred due to the depth perception. On the other hand, with printed images there is little difference between photographs with different focus levels. This is possible using the Depth of Field (DoF) [33], which determines the range of focus variation when comparing the nearest and farthest pixels. In this method, the Sum Modified Laplacian (SML) is used to measure the focus value and the difference between the summed patterns of the SML vector in a real face is usually consistent while in a fake face is not.

Lastly and relative only to dynamic systems, optical flow or motion based analysis [34] provides all the motion information when the difference of motion between two consecutive frames is short. It analyses and compares the pixels between two frames by calculating the gradient and matching the same pixels between these two frames. This way it can identify facial features and check for natural head movements to prevent any print attacks. In Figure 2.12 it is possible to observe crucial optical flow points. Even though it can be combined with other techniques, such as background analysis, it is too computationally complex to use in real-time applications, since its computation is for each pixel and higher resolution pictures present a better result but have a higher number in pixels, and may need special hardware in order to achieve trustworthy results. Overall dynamic methods that require motion have a high computational cost and a lack in generality, being only effective against any photo attacks, assuring the liveness measure of the user.

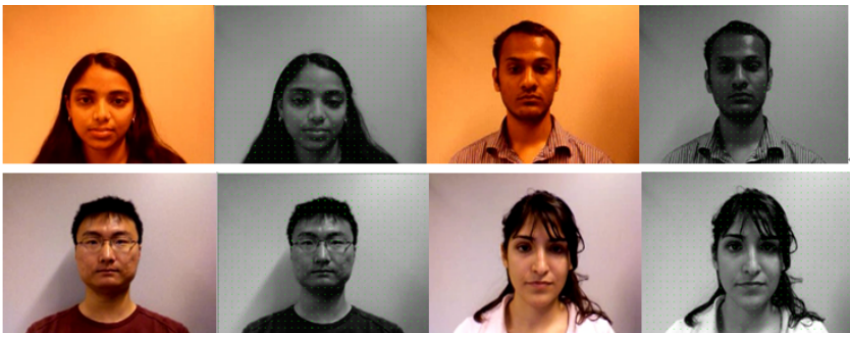


Figure 2.12: Original images and respective grey scale with marked optical flow points. [35]

It is also not uncommon to find face PAD algorithms combining two or more feature extraction methods. The combination of some of the previous types of dimensions is also a possibility, these joint systems are known as multi-biometric systems and may combine different hardware and software techniques. Naively, these fusion techniques were assumed to be the solution to every problem since it is only logical that the combination of several methods that analyse different biometric traits would increase the complexity of the system and, therefore, would be much more difficult to counterfeit. This was proven wrong due to the fact that the robustness of the system would not increase and by simply tricking one of the uni-modal systems it could be enough to break in [36]. It is also important to make sure that by fusing several techniques, the analysis of the captured data does not corrupt it or alter it such a way that changes the outcome of the rest of the techniques, used methods have to be completely independent

and complement one another. The two most used types of known fusions are at feature-level, where the features extracted by different, but related, complimentary techniques can be put together in order to achieve a better performance, and at score-level, where simply the output scores of the different algorithms are fused, trying to complement the disadvantages of a used technique with the advantages of another, using the average of the scores, for example. For instance, combining face and voice biometrics that cross correlate between lip movement and speech analysis [37]. In addition, almost every mobile phone or laptop companies nowadays are planning to equip their products with the standard microphone, camera and with some new emerging complimentary equipment, such as multi-spectral imaging [38], which can open new possibilities in the future of fusing these different type dimensions methods.

In conclusion, feature extraction methods are the most used since they can be practical to implement, requiring only the camera system to detect any attack, and can be made without no user intrusion whatsoever. Furthermore, each method focuses on solving a specific problem, lacking the robustness needed. Each algorithm has its strengths and weaknesses but fusing techniques blindly just to complement their downsides without first checking if they work well together only leads to worse results [36]. Nevertheless, the growth in technology and resources may lead to a higher computational power and a decrease in the price of hardware equipment, proving that a high computational cost method now or the combination of several techniques that complement each other might be the solution in a near future.

2.3 Machine Learning in Presentation Attack Detection

Nowadays Machine Learning techniques are being deployed in almost every field of study. This happens due to the natural high dimensionality of the problems, usually these techniques work in a high-dimensional space and can identify good solutions or connections out of many candidates, where some might not even be visible to the human being. In addition, what has been learned by ML can be transferred and scaled across multiple applications and millions of users, which is a much more practical approach to integrate expertise into data-intensive problems. Traditional software development uses deterministic instructions that are locked in place. This is an exceedingly unrealistic approach to building relevant products, considering that the environment software operates in is dynamic where the system's conditions keep on changing. What is much more realistic is an application that can learn from its environment and adapt to the shifting conditions and requirements that must be met, being the most realistic extension in problem solving and automation capable of dealing with high amounts of data.

For these algorithms to be able to find solutions and connections between data they need to be trained. There are several types of training, however the most common one, and the only one discussed in this thesis, is supervised learning. In supervised learning there is a dataset, named training set, which consists of labelled data, when a given input has an already known output, that is used to train the algorithm. The objective is to minimize the error, which is defined by the error function later explained, between the estimated output and the real output so when new inputs are given an acceptable result is achieved. Furthermore, if the training set is large enough it can be divided into two folds, training and

validation sets, where the latter is only used to adjust the parameters of the algorithm and to check for overfitting of the network, where too much of the same training might always lead to the same result.

Machine learning algorithms are frequently used when building a robust anti-spoofing system when a valid dataset is given and, as previously said, they are often used in two ways: standalone, where all of the analysis is being relied considerably on the machine learning algorithm, or as a complement to other algorithms, such as the classification based on several features, which is illustrated in Figure 2.13. In this type of methods the features are hand-crafted, as in being previously extracted using another technique, LBP for example, and then the trained classifier finds connections between all, or some, features that are deemed important by the classifier, performing a decision in whether it is a presentation attack or not. This is also known as feature classification.

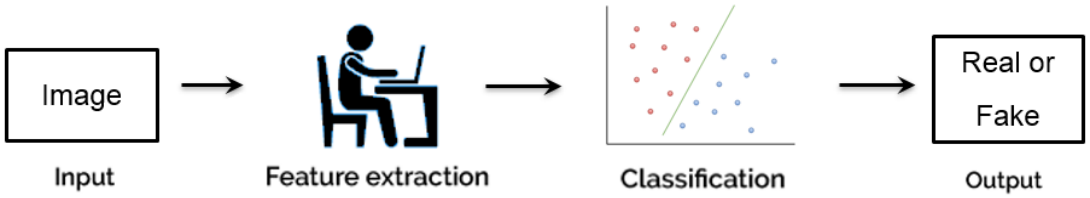


Figure 2.13: Feature extraction and classification done individually.

Usually, for feature classification, the most used classifier is a Support Vector Machine (SVM) [39]. A SVM separates data belonging to different classes, creating a representation of it as points in a hyperspace, separating each category by an hyperplane. The intriguing aspect about SVMs is that it can separate non-linear separable data, using the kernel trick. The kernel trick consists in letting the algorithm operate in a high-dimensional space without needing to compute the coordinates of the data in said space, but rather by computing the inner products between the data in feature space, which is much less computational demanding. However, a SVM can only separate two classes, meaning that when facing a situation with several classes, which is not the case, it has to separate each pair individually, increasing its computational effort. Another quite similar method to SVMs that can separate more than two classes at the same time, is the Linear Discriminant Analysis (LDA) [40] technique, which is also utilized as a dimensionality reduction and classification technique. LDA is not often used since it assumes that the data is normally distributed and linearly separable, which is not usually the case.

Machine learning methods can also be used in the wholesome of the process, being able to extract features and classify them accordingly. This type of methods are known as deep learning, as was previously stated, and can extract end-to-end features directly from raw data. This kind of deep representation is discriminative and generalizes well if the training data is sufficiently large. The tool that is mostly used for deep learning is a Convolutional Neural Network (CNN) [41]. These networks are a specialization of Neural Networks, which try to resemble the architecture of the human brain, these networks are constituted by several neurons interconnected, forming layers, as it can be seen in Figure 2.14. The input layer is given data and the network will try to learn and adapt following, similar to SVMs, a loss function, since each connection has a weight that suffers tuning during the training process and each neuron has an activation function that simply maps the resulting value accordingly, generating an

output response. There are also different types of layers in a network, usually in convolutional layers is where the computations occur, representing high-level features in the data, and towards the end of the network there are Fully Connected (FC) layers which provide the learning of non-linear combinations of said features, classification, generating a meaningful output. A conclusion could be made by analysing the output feature vector or the CNN may have the capability of fully classifying the input (imposter or genuine in case of PAs). Usually, Neural Networks in general have the disadvantage of depending heavily in the training set, more data often meaning better generalization and results.

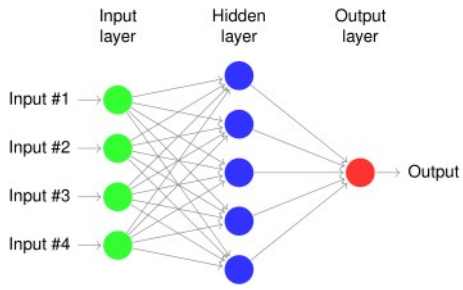


Figure 2.14: Basic Structure of a neural network.

Regarding PAD methods, early works used CNNs as a simple feature extractor, conducting a background or temporal analysis [42] to extract important features and then using it as input to a SVM to determine if the face was either real or fake. More recent methods become more imaginative regarding machine learning techniques. In *Alotabi et al.* [43], before using a deep convolutional neural network, it is first applied a nonlinear diffusion filter which is able to obtain the depth and preserve the boundary locations that help distinguish a fake image from a real image. Thus, the edges obtained from a flat image will fade out, whereas those from a real face will remain clearer as it can be seen in Figure 2.15. Afterwards, the image is given as input to a 5 layered Neural Network where it determines if it belongs to a genuine or imposter face.

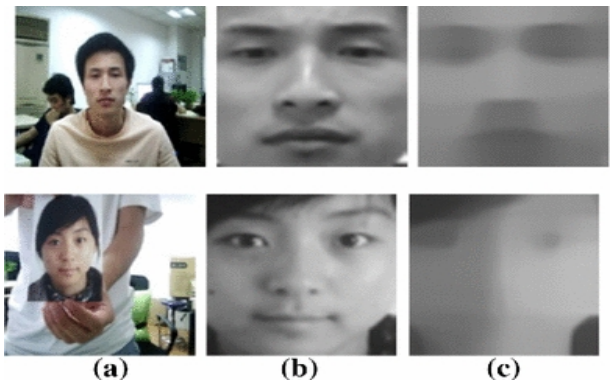


Figure 2.15: (a) the top image is a real face and the bottom a fake one; (b) Normalized face; (c) diffused image. [43]

Having a higher emphasis on the data pre-processing, *Yang et al.* [42] conclude that hand-crafted texture features techniques are unable to capture discriminative cues between genuine and fake faces. This way, they rely on the CNN to do an image quality analysis by giving as input a spatial augmented photograph rather than only the selected face image, displaying also the background so the network can find discriminative features in the scenario as blurred edges or abnormal specular reflections caused by the image recapture. Furthermore, they try to implement a temporal analysis as well, by extracting what they deem a temporal feature from consecutive frames and by giving these consecutive frames together to the neural network which, unfortunately, showed no improvement (the CNN architecture can be seen in Figure 2.16). This latter technique was improved by *Xu et al.* [7] since the temporal structure used in previous techniques was not fully correct because the CNN architecture itself cannot extract temporal features and these also exist in pixels that are not in the same plane. A newly layer was introduced, a Long-Short Term Memory (LSTM) layer, which has the ability to learn long range dependencies from the input sequences. This way, the problem starts being treated as a video classification, this layer is put on top of the CNN extracting temporal relations from different video frames.

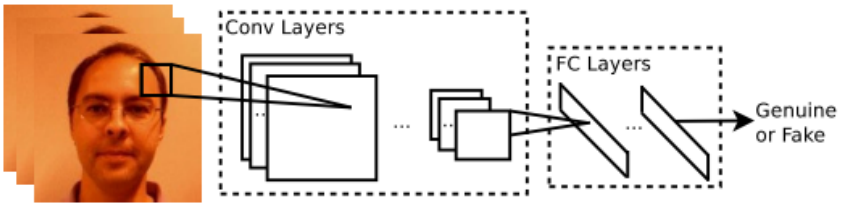


Figure 2.16: CNN architecture of *Yang et. al* [42].

Atoum et. al [44] uses a fusion approach. Their architecture is composed by two neural networks, each one doing their independent classification and afterwards fusing the scores, determining the output. Firstly there is a Patch-Based CNN where a deep neural network is trained to learn rich appearance features using different colour spaces, which originates better results when compared to RGB, as it was previously discussed (see subsection 2.2.4). This network analyses random patches from an image with the objective of increasing the number of training samples and to maintain the original image resolution so quality is not lost. Each patch has a score determining how reliable it is. The other CNN does a Depth-based analysis of each image, a depth label from a 2D image is generated using a method that can estimate the 3D face shape model and the camera projection parameters based on the face localization, orientation and geometry. Afterwards these features are given to a SVM which labels the feature vector, giving it a score as well. The overall architecture of the fusion of these cues and an image example can be seen in Figure 2.17.

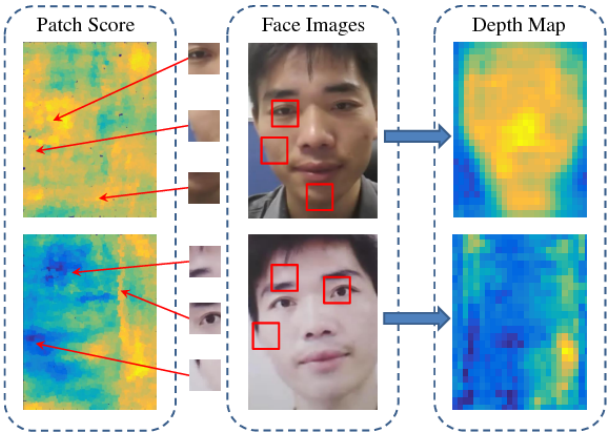


Figure 2.17: Fusion of a patch-based and depth CNN. Left column shows the output scores of the local patches for a live image (top) and for a fake image (bottom), where the blue/yellow represent a high/low probability of a presentation attack. The right column shows the output of the depth estimation, where yellow/blue represent closer/further points. [44]

As previously discussed, a considerable problem of deep neural networks is that it requires a large amount of data in order for the system to be robust. This proves true for presentation attack detection since there are far too many known variants and conditions of attacks to the system, as mentioned in section 2.1, and not that many training images. To enable the use of deeper networks for PAD applications, a newly method was introduced by *Y. Rehman et al.* [45]. Usually, a network is trained by iterating the whole dataset several times; each time all of the dataset is iterated through the neural network is denominated as an epoch. With this in consideration, instead of only shuffling the dataset once, before the whole training process as usual, this method shuffles the dataset before each epoch, using only part of it for the actual training, increasing the robustness of the algorithm since the training set varies in every epoch [45], preventing overfitting as well. Unfortunately, the training time and memory usage severely increase.

Chapter 3

Proposed Approach

As it was detailed in previous sections, presentation attacks have many variations depending on the type of attack, what system the imposter is trying to break in, and on lighting or capture conditions of the used artefact. Therefore, creating a presentation attack detection scheme robust enough to all possible scenarios proved to be extremely difficult. In the preceding section, the most important state of the art algorithms were discussed, presenting their advantages and disadvantages when facing each type of PA. Furthermore, deep learning algorithms proved to be very powerful in determining discriminative features from data, having, however, several disadvantages as the need for a big training set and taking a very long time to be properly trained.

In this chapter, an architecture using deep learning is proposed and illustrated in Figure 3.1. To begin with, a training set with facial images in order to train the network is needed, so several databases were used, later presented in section 4.2. However, these sets are bestowed as simple videos, therefore the capture of frames and detection of the correspondence facial region in each frame was needed, existing several algorithms that allow this. In section 3.1.1 an older, simple method which is the most used amongst state of the art techniques, is discussed. The Viola-Jones algorithm [46] is a feature-based machine learning algorithm which is generally used for face or eye detection, being, however, able to detect any object it was previously trained for. On the other hand, a newer approach would be using the famous concept of deep learning as it is the method utilized to detect presentation attacks as well. These are newer methods that present an increase in performance, presented in section 3.1.2.

Afterwards, in section 3.3, the used neural network's architecture is presented. This network is adapted to the PAD problem, by applying what is known as Transfer Learning [41], which is introduced in section 3.2 and allows to use already existing networks trained for similar tasks to newly proposed problems. This solves some big disadvantages of deep learning and proves fruitful when the training set of the task at hand is small, in terms of its dimension and robustness, which is often the case with PAD. Later on, in section 3.4, the modifications done to the neural network so it can be adjusted to the PAD problem are detailed. Firstly, using the proposed architecture, presented in 3.1, an analysis of single images is made and later on, that architecture is altered, allowing a full video analysis. In order to do this a new layer is used, explained in section 3.5, which was added at the end of the network, known

as Long-Short Term Memory (LSTM) [47], allowing a spatio-temporal analysis of a video, learning long range dependencies from the input sequences.

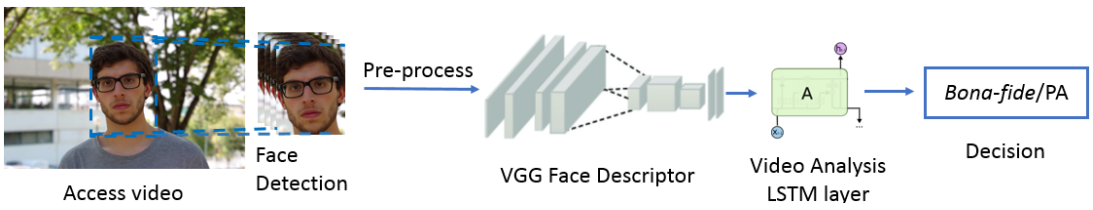


Figure 3.1: Architecture of the proposed solution.

3.1 Face Detection

The used network receives as input pre-processed facial images with a fixed size of 224 x 224 pixels. Therefore, facial regions in an image need first to be identified so they can be used as input to the network. There are various tools for detecting faces, for example, a deep learning network could be used as long as there exists high quantity of training data. With a deep learning approach a high performance in facial detection in every situation is expected to be reached. However, properly training a neural network takes time and a considerable memory usage, which proves to be inefficient if other simpler method that perform just as well might be available. When taking this in consideration, a simpler classifier can be an option. Even though this classifier needs to be trained as well, it has a simpler architecture, does not need much memory when compared to a neural network and leaves training faster. Although its performance is expected to be worse than the first approach, it should be only noticeable in unconstrained capture conditions, such as when faced with sided capturing angles rather than frontal or partially face occlusion in crowds. When regarding PAD techniques, usually the capturing software requires the user to be close and frontally facing the sensor so these errors do not occur. Nonetheless, a comparison between a conventional face detector and one based on deep learning is presented, discussing their different approaches to the face detection problem.

After detecting the face region in each image, the PAD neural network can now be trained. Since all of the databases used have a frame rate of 25 frames per second and present several videos, this will lead to thousands of images with very low variability between consecutive frames. To decrease the amount of redundant data, as well as to respect the memory usage of the system, the algorithm captures only a few frames every second, depending on the used database's size being usually between 3 to 5 frames, since it is enough to perform a complete analysis of the video. Afterwards these images are resized according to the input of the neural network, with a size of 224 x 224 pixels and, according to the type of analysis later explained, the colour space might also be converted. Finally and for each dataset, all images are subtracted by their average facial image previously calculated.

3.1.1 Face Detection - Haar Cascade Classifier

The Viola-Jones algorithm [46], introduced by P. Viola and M. Jones in the early 2000's, is a machine learning algorithm used to detect the eyes and faces in an image. It was a ground breaking technique since it was fast and computationally cheap, and it was one the first face detection algorithms to run on simple cameras and mobile phones. In order to do so the algorithm is trained with positive images (images of faces) and negative images (images without faces). To extract features from each image, a haar-feature extractor is used. As illustrated in Figure 3.2, each feature is a single value obtained by subtracting the sum of pixels that lay under the white rectangle from the sum of pixels that lay under the black rectangle.

During training, the best threshold which classifies the faces to positive or negative is found. To calculate all features within an image, all possible sizes and locations of each haar-block need to be used, which increases the computational cost and time drastically. In order to avoid this, the concept of Cascade of Classifiers is introduced [46]. Instead of applying all of the haar-features on a window, the features are first grouped into different stages of classifiers, being applied one-by-one. An example of different haar-features and its employment are illustrated in Figure 3.3. All of the sections of the image that are not relevant and do not include the face will contain very few number of features and can be discarded in the early steps of classification, not belonging to a face region.

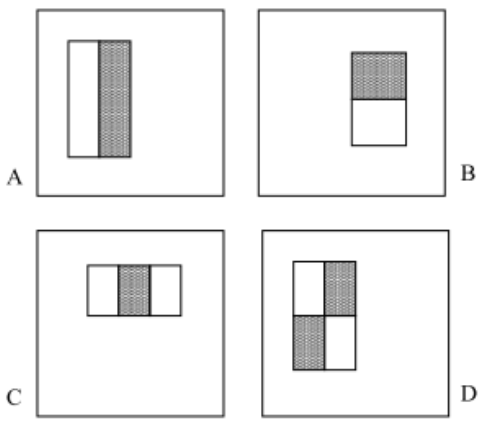


Figure 3.2: Examples of different haar-feature extractors, or haar-blocks, shown relative to a window. (A) and (B) show a two-rectangle features, (C) a three-rectangle feature and (D) a four-rectangle feature. [46]

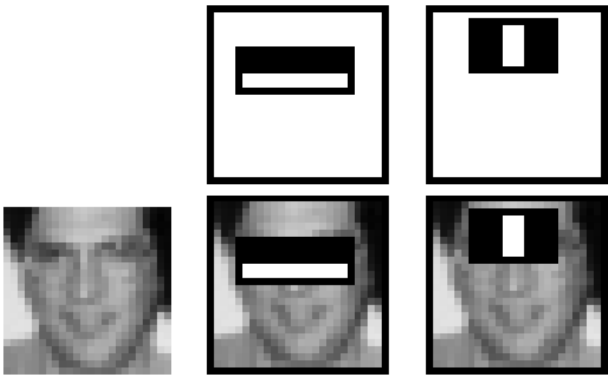


Figure 3.3: The top row are the haar-features which are then overlayed on a training face that is in the bottom row. The first feature measures the difference in intensity between the region of the eyes and the second compares the eye regions across the bridge of the nose. [46]

3.1.2 Face Detection - Deep Learning

Using neural networks to detect the facial regions has become increasingly more popular due to the high computational power demonstrated by these, which allows to achieve a high performance in adverse conditions. Usually the chosen approach when regarding the training of the neural network is the detection of important facial landmarks, which are present in every face, therefore identifying the facial region of the subject. However, training a neural network to detect faces is usually more computational expensive when compared to the Viola-Jones classifier, and a vast training set has to be prepared. Face Recognition CNN [48] is a state of the art face recognition algorithm built with deep learning, achieving an accuracy of 99.38% on the Labelled Faces in the Wild (LFW) [49] database. This database of face photographs was designed for studying the problem of unconstrained face recognition. All of its images were gathered from the web and present different lighting conditions or angles. These conditions are known as unconstrained since there are no constraints or control of the subjects' expressions/camera angle when regarding the conditions on the image capture.

In order to detect facial regions in images, this network uses a feature extraction technique known as Histogram of Oriented Gradients (HOG) [50]. An image is first converted to black and white and the gradient of each pixel is calculated. To do this, each pixel is analysed and compared to its neighbours, analysing in which direction the image becomes darker, as illustrated in Figure 3.4 by an arrow. By repeating this process for every set of pixels in the image, the arrows display the gradient of an image, showing the flow from light to dark pixels. This method is preferred rather than analysing each set of pixels since it is the variation of illumination that allows the identification of the face and not the pixel's values.

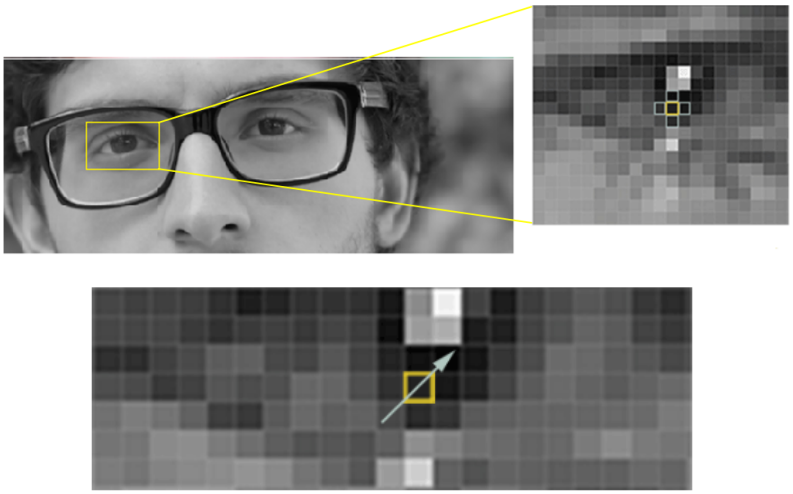


Figure 3.4: When taking in consideration the group of selected pixels, the image is getting darker towards the upper right.

In order to optimise the gradient's computation, and since only the basic flow of lightness/darkness is needed at a higher level, pixels are grouped in small squares with 16 x 16 size. The gradient is then computed, allowing the capture of basic structures from a face, as it can be seen in Figure 3.5.

Figure 3.5: Original image and its HOG transformation, capturing the major features of a face.

Using the HOG technique, a training set can be calculated so that the neural network is able to find any facial region by choosing the most similar pattern to a HOG facial region, denoted by a yellow square in Figure 3.5. Even though this technique works well with frontal face detection, it does not work when the sensor captures faces from different angles. To account for this, each face has to be warped so that it is always in the same sample place on an image. In order to do this, an algorithm known as face landmark estimation [51] is used. This algorithm allows to detect 68 specific points, or landmarks, that exist on every face, then the neural network is trained to detect these landmarks. After knowing where the eyes and mouth are, the image can be simply rotated and scaled so that it is centred as best as possible, as it is displayed in Figure 3.6.

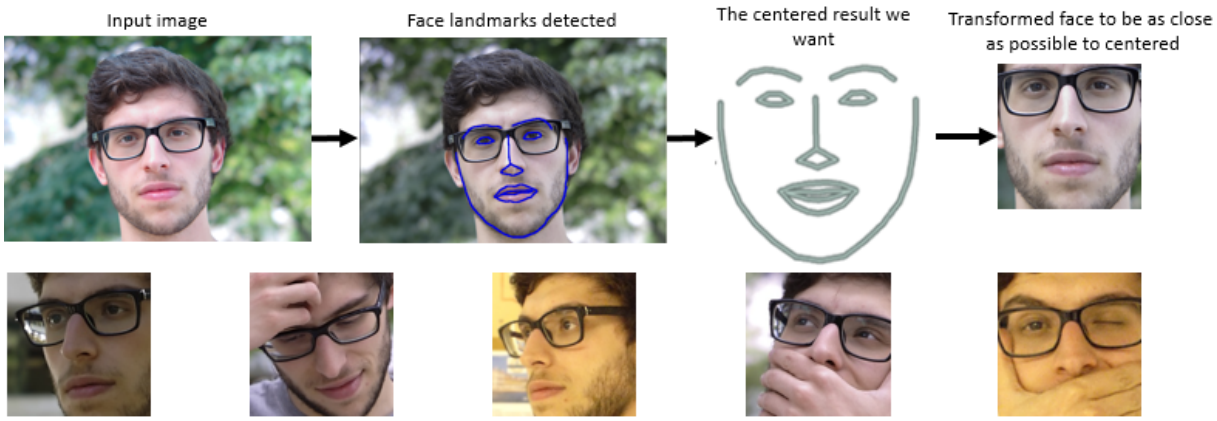


Figure 3.6: Input image with its face landmarks detected together with sample images with the detected faces.

This way, no matter how the face is turned, the algorithm is able to center the eyes and mouth so that they are all roughly in the same position in the image. By having the face centered, face detection using the previously explained HOG technique is possible.

3.2 Transfer Learning

Transfer learning is known as *the application of knowledge gained from completing one task to help solve a different, but related, problem* [41]. Machine learning algorithms are typically designed to address isolated tasks. During transfer learning, knowledge is leveraged from a source task to improve a new task; however if the latter does not occur, the transfer method may end up decreasing its overall performance, being nominated as a negative transfer. A major challenge when applying transfer learning is to choose which algorithm to adapt to the new problem and how to adapt it without changing any key, fundamental aspects of the original network which might negatively influence the result. This technique proves fruitful when both problems are similar and there is not enough robust training data available for the new problem, which is the case with most PAD methods.

It is important to note that in transfer learning the weights of the previous network, or knowledge, have to be reused and retuned for the new problem, this is also known as the pre-trained model approach [52]. This is possible since many research institutions release models trained on large and challenging datasets that may have a similar objective as the task at hand. This model may be seen as a starting point and can suffer several alterations from the original scheme in order to adapt the network to this new problem. Usually, after changing what is necessary, the model's weights may need to be refined or tuned in order to completely converge according to the task of interest.

Other different, widely used approach is to reuse only the neural network's architecture, resetting the weights and start training from the ground up. This is not known as Transfer Learning and it is often discouraged since architectures are built in accordance to the proposed objective. By using a randomly chosen architecture only because it had a high performance in general, it will probably not achieve as a

high performance in a completely different problem as a neural network built specifically for it.

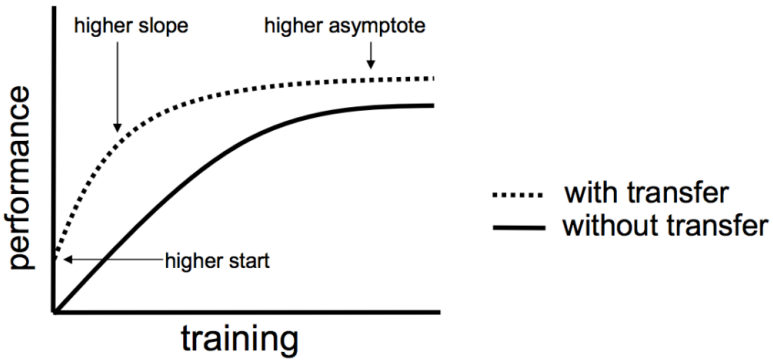


Figure 3.7: Three ways in which transfer may improve learning. [52]

Ideally, and when applying transfer learning, it is possible to obtain three distinctive benefits from this technique [52]. As illustrated in Figure 3.7 it might occur: a **higher start**, where the initial performance of the algorithm usually has a better starting point than the one that is completely trained from the ground up, this is mainly due to the similar objectives between the networks; a **higher slope**, the rate of improvement of skill during training of the model is steeper than it otherwise would be since we are fine tuning the weights; and a **higher asymptote**, where the converged skill of the model is usually higher than it otherwise would normally be as some previously trained weights might lead to better and faster converging results. Another aspect to have in consideration is that, since the problems are similar, there is no need to train the totality of the network. This happens because the first layers have very basic operations that suit both of the problems, so by tuning these layers the accuracy of the network might actually decrease. The results regarding this hypothesis can be seen in section 5.2.

There is a huge amount of available neural networks that can be used as a starting point. There are competitions performed yearly so the upcoming trends in machine learning can be introduced to the research community and these state of the art neural networks can be shared. Nevertheless, to achieve the best performance of the algorithm, a network with a similar objective must be found. Some of the most used networks, such as ImageNet or GoogleNet, could be adapted to this problem. However these networks deal with image classification problems, such as animals or action classification, and despite their outstanding results, their original training is not as close to the topic of interest as intended. Taking this into account, the chosen architecture considered as a starting point to the PAD problem is the VGG Face descriptor [53], a convolutional neural network used for deep face recognition, introduced by the Visual Geometry Group of the University of Oxford, which can identify more than two thousand celebrities. Even though this network identifies the identity of the user and not a presentation attack, it still deals with faces and its variations, also providing support to anybody that wants to use their pre-trained model, being, this way, the chosen architecture.

3.3 VGG Face Descriptor

The deep facial recognition descriptor from the Visual Geometry Group (VGG) is a convolutional neural network considered ”very deep”, in the sense that it comprises a long sequence of convolutional layers with several operations [53]. Its ultimate goal is the identification of two thousand, six hundred and twenty two (2622) people considered celebrities, being its input a face image of size 224 x 224 pixels with the average face (computed from the training set) subtracted. This network was derived from the VGGNet [53], a deep learning algorithm which was a runner-up in ImageNet Large Scale Visual Recognition Competition (ILSVRC) [54] in 2014, and showed that the depth of a network is a critical component for good performance, surpassing 140M weights as trainable parameters.

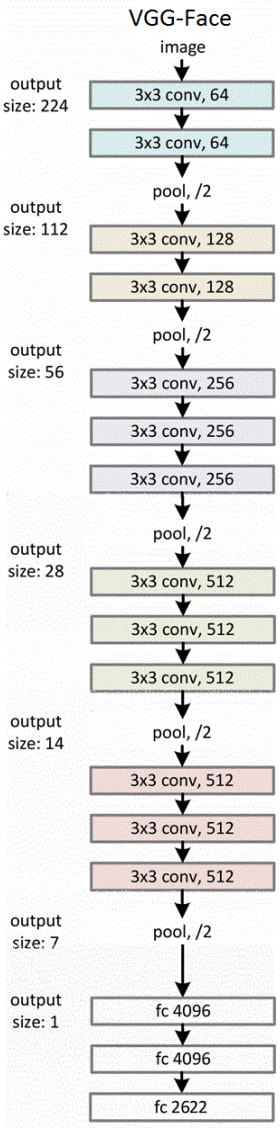


Figure 3.8: Architecture of the original VGG Face descriptor.

Regarding the architecture, and as illustrated in Figure 3.8, the network comprises 13 convolutional blocks, which and as previously said, are the layers where the computation of high-level features of the data take place, each containing a linear operator followed by one or more non-linearities such as ReLU and max pooling. The ReLU (Rectified Linear Unit) is used as an activation function, which decides whether a neuron should be activated or not or, in other words, decides if the information presented by the neuron is useful or not. This unit is applied after each convolutional block being ReLU quite used and accepted as one of the best activation functions for deep learning algorithms [55], which is mainly due to its high flexibility in order to account for non-linear operations and presenting a good representation of features. After each 2 or 3 convolutional blocks there is a max pooling function which is a sample-based discretization process. Its objective is to downsample the input features from the previous layer, reducing their dimensionality. This way, it is possible to aggregate the information over a small local patch with reduced dimensions when compared to the previous one, as illustrated in Figure 3.9, leading to faster computations and lower memory usage since the applied convolutional filters are smaller. However, by applying too many max pooling layers the features may be severely reduced, losing discriminative aspects important to the problem and can lead to a decrease in the overall performance of the algorithm. In order to avoid this, the depth of the layer is usually increased, which is the case.

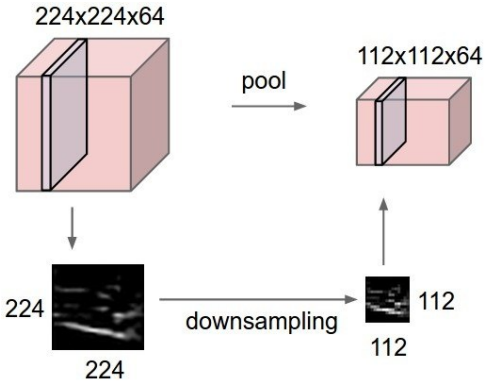
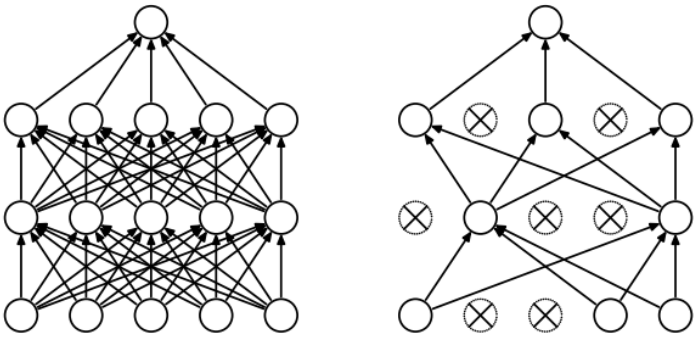


Figure 3.9: Example of a simple pooling operation. [41]

It is also important to note that throughout the network there are some dropout layers [56] which are used to avoid overfitting and complex connections between neurons, as it is illustrated in Figure 3.10. This is possible since, during training, a percentage (previously chosen by the programmer and, in this case, 50%) of random neurons are deactivated, or dropped-out, usually improving generalisation as it forces the others neurons to adapt to different changes. These have to fill in for the dropped-out neurons, avoiding a specific specialization. When conducting tests, the dropout layers are disabled, being only useful during training.



(a) Standard Neural Net (b) After applying dropout.

Figure 3.10: Dropout model layer. (a) is a standard neural net; (b) resulting neural net when applying a dropout layer. [56]

The last three blocks are instead called Fully Connected (FC) layers and, as previously said, usually provide the learning and combination of all of the previous features, matching the size of the input data. This way, each filter ”senses” the data from the entire image [41]. This happens because all neurons of the FC layers are activated and chosen as an important factor to make a decision.

The last FC layer has a softmax as an activation function which can be seen as a classification layer since it calculates the probability of the final feature vector belonging to each class, 2622 in the case of the original VGG. It is equivalent to a categorical probability distribution where the sum of all probabilities is 1. The function is represented in equation (3.1), where z represents the features with a vector size K, being σ the probability output for each class j. Furthermore, the softmax function assumes that the feature vector only belongs to exactly one class which proves to be true in this case, however, multi label problems do exist.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad for \quad j = 1, ..., K. \tag{3.1}$$

3.4 Adapting the VGG-Face Network to PAD

In order to adapt the VGG network to the problem at hand, several factors had to be taken into consideration. The goal is no longer about person identification, but rather presentation attack detection. Therefore the result of the last layer was changed to a binary output, with the two considered classes being either a *bona-fide* sample or a presentation attack. During training, and as previously introduced, the algorithm optimizes an error function, also known as loss or cost function, which allows the computation of the network's weights that lead to the best possible result, the minimum value of the loss. Thus it is important to choose a function that suits the problem well. Having this in consideration the chosen loss function was the cross-entropy, or the logarithmic loss, since it measures the performance of a classification model whose output is binary or a probability between 0 and 1. This function increases as the predicted probability diverges from the actual label as displayed in Figure 3.11. Binary loss functions usually achieve similar results, however, cross-entropy is often the preferred since it combines well with

the softmax classifier.

When other loss functions are used, the training can increase due to weight updating. In a binary problem, the output is either 1 or 0 and the weights need to be adapted accordingly. However, by using other loss functions rather than cross-entropy, the weight adjustment factor was found to decrease constantly while training, which may lead to an increase in training duration or the optimal solution might never be achieved [57].

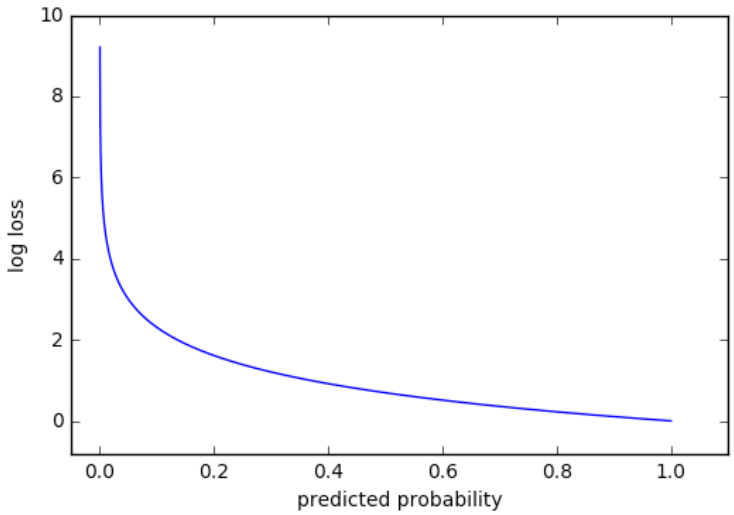


Figure 3.11: Cross-entropy function.

To calculate the loss function at each step, during training, it is needed to take into consideration the respective weights of the network and an optimizer. This optimization algorithm is simply a mathematical function that helps to minimize the error at every step, updating the weights in the direction of the optimal solution and applies a major role in the training process of the neural network [41]. Nowadays, the most used optimizer in neural networks is the Adaptive Moment Estimation, or Adam [58]. This optimizer is based on a gradient descent which calculates, using an adaptive rate, the ”direction” to where the function either increases or decreases. In order to do this, it updates the weights that lead to the desired minimum by calculating the gradient based on the current weights. The particularity of Adam, when compared to other optimizers, is that it computes individual adaptive rates for different weights of the network based on its previous values at the expense of memory, while in other optimizers there exists an unique adaptive rate to update all of the network.

In order to compare this optimizer with others, the authors of Adam used the IMDB movie review dataset problem [59]. This test consists in predicting the given score to a movie based only on its written review. By using the same algorithm but only changing its optimizers, the training cost over the dataset iterations was obtained and is summarized in Figure 3.12, where a lower cost means better efficiency.

This adaptive rate is also known as learning rate, a hyper parameter which controls how much adjusting do the weights suffer in an epoch. A learning rate too low might never lead to the convergence of the problem and too high might skip the optimal solution and never converge. This parameter is also tuned according to training and, for this problem, has the value of 10^{-5}. This rate is achieved

through experimentation, and values between $[10^{-3}, 10^{-6}]$ were tested. This very low value is due to the problem of Transfer Learning, as previously discussed, the weights need to be fine-tuned and not drastically changed, therefore the weight update has to be done smoothly. With the idea of optimizing the converging efficiency, during training, early stopping was applied. This concludes training earlier if the performance of the algorithm does not increase within a range of epochs, usually between 10 to 20, concluding that the algorithm already converged to the optimal solution and that training does not need to continue, allowing a faster training phase.

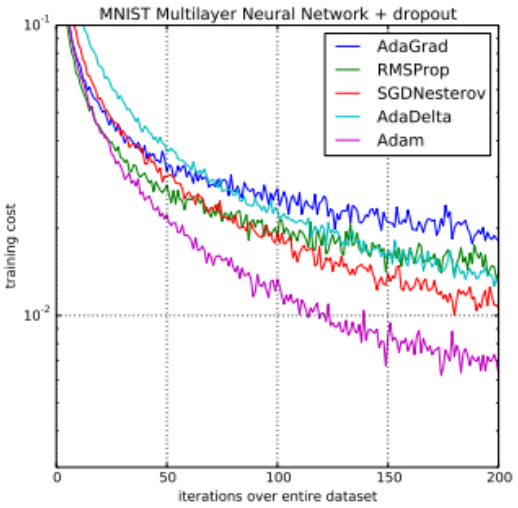


Figure 3.12: Comparison of the efficiency of the optimizer Adam with other known optimizers when facing the IMDB database problem. [58, 59]

It is also important to mention that this is just one adaptation of many of the VGG network. There are several variations of this network even when regarding other problems such as face detection or recognition. Concerning PAD, techniques discussed previously in section 2.3 mostly use this architecture and propose new alterations. In [60] VGG is altered only in the fully connected layers so the deep part features, supposedly richer textures, can be extracted from the network. Other example is the FASNet network [61], which proposes a presentation attack detection network based on the VGG architecture, adapting the top layers. These two previous approaches use RGB images and train all of the network. Other algorithms try to increase the efficiency of training deep learning algorithms using the VGG network for PAD [45]. Even though the architecture is the same, there are many different propositions when adapting any network to other problems and every one of them leads to different results, it is the changes done to the network and its training that influence the outcome, being the common architecture just a starting point.

3.5 Video Analysis

As discussed in previous sections, the analysis of a single image might not be enough to fully detect any type of fraud (see subsection 2.2.4). Having this in consideration, some techniques analyse a small

sequence of frames instead. Traditional neural networks accept a fixed-sized vector as input, a single frame in this case, and produce an outcome for each. This does not happen with Recurrent Neural Networks (RNN) since they allow to operate over a sequence of frames, or a small video, as input vectors.

3.5.1 Recurrent Neural Networks (RNNs)

In a RNN, as illustrated in Figure 3.13, A receives as input x_t and outputs a value h_t, also known as state vector, and there are loops which allow the output information to persist several iterations [41]. A RNN can be thought of multiple copies of the same network, each passing a message to its successor.

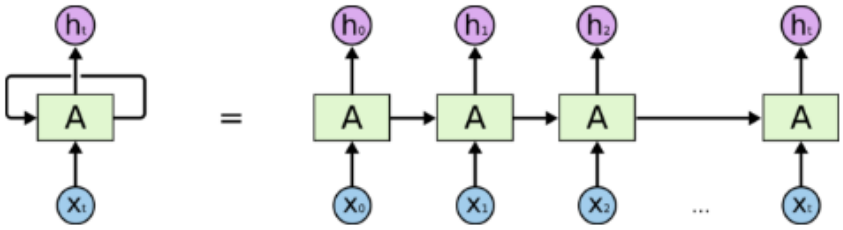


Figure 3.13: An unrolled recurrent neural network. [62]

Having this in consideration, these networks are intimately related to sequences and lists. Moreover, RNNs combine the input vector with the state vector to produce a new input, that depends on previous iterations, calculated in every loop. RNNs can have several architectures, as it can be seen in Figure 3.14. The **one-to-one** architecture is the standard neural network without any feedback loops or memory; **one-to-many** outputs a sequence, usually used in image captioning where an image can output a sequence of words, as image description; in **many-to-one** there is a sequence input, often being utilized in an action or sentence classification; **many-to-many** in asynchronous mode is usually employed in translation, where it first reads the sentence and, according to the context, translates it to another language; finally, **many-to-many** in synchronous mode is often applied in video classification where each frame is labelled, having an output.

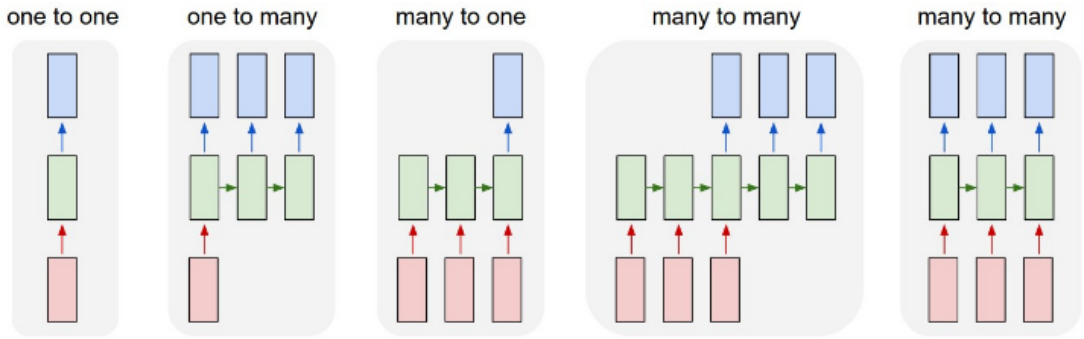


Figure 3.14: RNNs architectures. Red rectangles are input vectors, output vectors are blue and green vectors hold the RNN state, h_t. [63]

However, simple RNNs present a big disadvantage since they become unable to learn any connection between the presented information when the gap amidst the relevant data and the output increases, being known as the vanishing gradient problem [47]. When the information presented is too vast, or the input array is too long, the common RNN cannot find relevant connections between the data due to all of the noise introduced. One layer that solves this issue is the Gated Recurrent Unit (GRU). However, it does not allow to control or define a timestep, later explained, exposing the full hidden content without any control, which is not advantageous in this situation. The presented videos are too large so a limit in the analysis or grouping into smaller videos is a requirement. A layer that presents this feature and solves the vanishing gradient problem as well is presented in the following subsection.

3.5.2 Long Short Term Memory (LSTM) Layer

Long-Short Term Memory (LSTM) networks are a special kind of RNN, capable of learning long-term dependencies [47]. They are the chosen video analysis layer because the typical RNN disadvantage does not occur as it is able to remember information for long periods of iterations, although presenting a more complex module, as illustrated in Figure 3.15. The key aspect of LSTMs is the cell state [47], the horizontal line running through the top of the diagram, which is regulated by structures called gates. A gate is simply tasked to only let relevant information, learned by means of training, through.

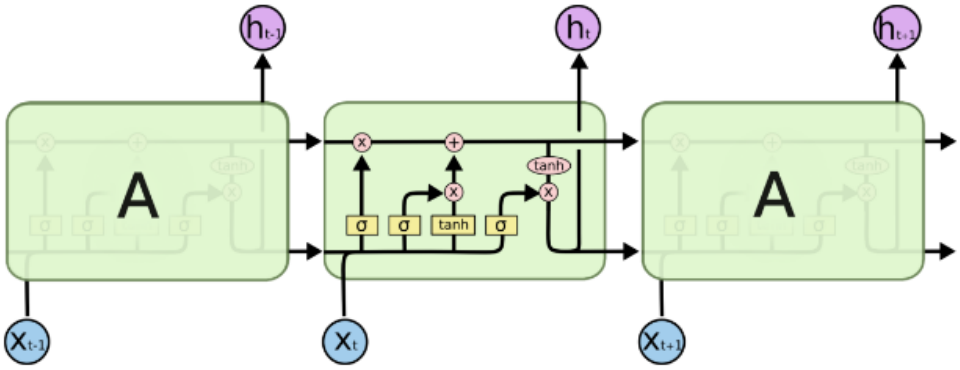


Figure 3.15: LSTM module. Each line carries a vector, from the output of one node to the input of others. The pink circle represents pointwise operations while the yellow boxes represent normal layers from the neural network. Lines merging represent concatenation while forking serve as a content copy going different locations. [62]

The first step of the LSTM is to decide what information is relevant to keep from the previous iteration. In order to do this, a forget gate layer is used, f_t, that is basically a sigmoid function (σ), which simply analyses h_{t-1}, the previous state, and the new input, x_t and outputs a number between 0 and 1 depending on the relevance of the information, for each number in the cell state C_{t-1}. This is all put together by W which represents the window of the operation. This can be simplified by the following equation:

$$f_t = \sigma(W_f.[h_{t-1}, x_t]) \tag{3.2}$$

Afterwards, new information that is considered important to the problem needs to be stored. For this,

another sigmoid function (σ) is used to determine which values will be updated, also nominated as input gate layer, i_t. Then, a new vector of candidate values, $\tilde{C}_t$, is created using the hyperbolic tangent function, tanh, as presented in the following equation:

$$\begin{aligned} i_t &= \sigma(W_i.[h_{t-1}, x_t]) \\ \tilde{C}_t &= tanh(W_c.[h_{t-1}, x_t]) \end{aligned} \tag{3.3}$$

Now to update the old cell state, C_{t-1}, into the new state, C_t, the first is multiplied by f_t, forgetting the no longer important information and adding the new candidate values, $i_t * \tilde{C}_t$, which are scaled according to their interest to the problem , leading to:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.4}$$

Finally, to calculate the output of the LSTM, a sigmoid function is used again to decide what parts of the cell state belong to the output, and its result is multiplied by a tanh that has the objective of comprising the values between -1 and 1:

$$\begin{aligned} \sigma_t &= \sigma(W_o.[h_{t-1}, x_t]) \\ h_t &= \sigma_t * tanh(C_t) \end{aligned} \tag{3.5}$$

Concluding, LSTM generalizes well, allowing a complex spatial and temporal analysis [47] even if the related inputs are separated by a relevant gap, bearing in mind that too large gaps may introduce too much noise in real problems.

3.5.3 LSTM Layer Implementation

In order to implement this layer in the already existing architecture, the adapted VGG previously discussed in section 3.4, the last FC layer was removed from the network, feeding the features directly to the newly added LSTM layer. These features are given to the LSTM layer which analyses their evolution over the time period. This time period, also known as timestep, is a new hyper parameter of the proposed architecture and it is a measure of how many steps does the layer keep in memory in order to make a decision. In an initial thought, the ideal value for this would be the time, in frames, of the total video, however such is not possible since the LSTM network cannot find significant relations between images that are temporally too much apart, meaning that the image content may have suffered significant changes.

By increasing the timestep, noise is added and more memory is needed while, on the other hand by decreasing its value, relevant connections between frames are lost. In section 5.4 the optimal timestep for this problem is discussed. After that timestep has passed, the memory of the LSTM layer is reset, starting its temporal analysis all over again with a new set of frames. For this to be possible, during the database pre-processing step, after the face detection and resizing, the frames are grouped into

sequences and are no longer seen as independent. The input layer of the network has to be altered in order to receive groups of frames that represent a sequence.

The neural network's full architecture can be seen in Figure 3.16, where a 224 x 224 face image, which was previously detected by the Viola-Jones algorithm and pre-processed, is given to the adapted VGG network. Then, there is the LSTM layer which is followed by a softmax classification layer that produces an output. The displayed architecture is during N timesteps and has a many-to-one display, however, in section 5.4 a comparison between the two possible architectures for this problem, many-to-many and many-to-one, is made.

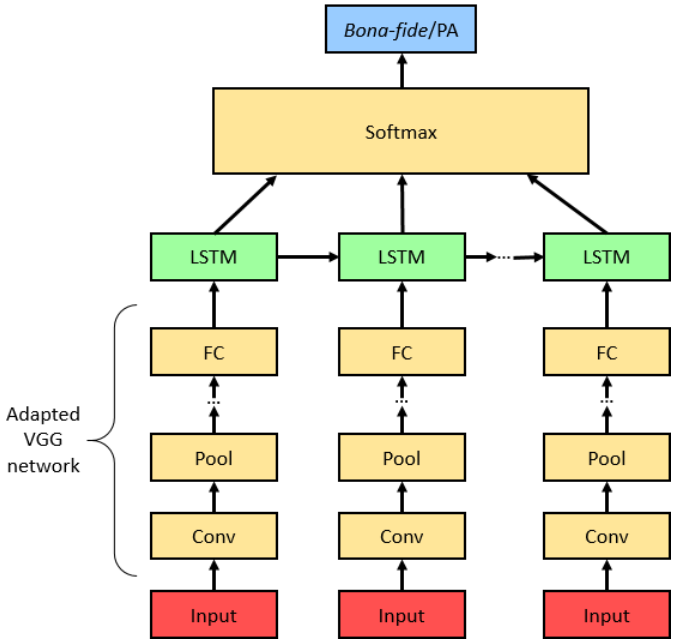


Figure 3.16: Architecture of the neural network used for Presentation Attack Detection in N timesteps with a many-to-one architecture. In red are displayed the inputs, in blue the output, in green the RNN layer and the rest is the adapted VGG network.

Chapter 4

Experimental Setup and Evaluation Criteria

In order to perform all the required tests, to evaluate and compare the performance between the proposed solution and state of the art techniques, different tools were used. Section 4.1 presents the performance evaluation methods utilized, clarifying what are the used error metrics, and detailing how are they calculated. Furthermore, and so that PAD techniques performance can be assessed, several databases are used for testing, following the corresponding evaluation protocols. Therefore, a detailed analysis of each employed database is presented in section 4.2, reporting on the conditions of capture, what types of attacks they present and the employed capturing sensors.

To develop the proposed PAD system and to process all the needed data, the toolboxes OpenCV and Keras were used. Open Source Computer Vision Library, or OpenCV [64], is the leading open source library for computer vision and image processing, being mainly used to support all of the operations related to imaging, such as the capture and detection of faces and its pre-process. In addition, Keras [65] is a deep learning library, which runs on top of TensorFlow, and it was developed with a focus on enabling fast experimentation. This framework has a high modularity and a high-level API (Application Programming Interface) in order to provide support to the user while creating its deep learning algorithm, providing high extensibility, making Keras suitable for advanced research. This latter package was used to support the development and tuning of the used neural network, allowing highly complex operations to its completeness.

4.1 Performance Evaluation Metrics

Performance evaluation is an important task in order to assess if the algorithm's contribution is what is expected or not. With this in mind, every state of the art technique is evaluated so that it can be compared. It is also crucial that the means of this assessment and its calculated metrics are the same to every algorithm and allow a robust and demanding performance analysis. Having this in consideration the used error metric is the Half Total Error Rate (HTER), later explained. Accuracy is quite used as well

by other techniques since it is simple to calculate even though it simply states a *hit or miss* and it does not provide important key conditions regarding misclassifications.

In order to fully understand the HTER, four major cases need to be considered [66] and summarized in the confusion matrix presented in Table 4.1:

(i) **True Negative** (TN) - if the sample provided to the sensor is real (*bona-fide*) and the user is in the database, being granted access;

(ii) **False Positive** (FP) - if the sample provided to the sensor is a *bona-fide* and the user is signed in the database, being, however, not granted access due to an algorithm misclassification;

(iii) **True Positive** (TP) - if it is presented to the sensor a PA which is safely detected by the algorithm;

(iv) **False Negative** (FN) - if it is presented to the sensor an attack which is not detected by the system, allowing the imposter to break-in.

According to this, it is possible to conclude that TP and TN show a well functioning of the algorithm, whereas FN and FP are errors. However, FN is seen as a more alarming error since the access is granted to the imposter. While in a FP, an attack is detected without being one, so the user needs only to retake the image assessment process.

Table 4.1: Confusion matrix for Face Presentation Attacks. Being **AD - Attack Detected**, or signalled by the algorithm, and **AND - Attack Not Detected**, or not signalled by the algorithm.

		Actual Value	
		Bona-Fide	**PA**
Prediction Outcome	**AD**	False Positive	True Positive
	AND	True Negative	False Negative

Even though false negatives are more important to detect when compared to false positives, the HTER [66] gives both error types the same weight. It is the used error metric by the scientific community in PAD, being calculated as the average of the False Acceptance Rate (FAR) and the False Rejection Rate (FRR), as follows:

$$HTER = \frac{FAR + FRR}{2} \tag{4.1}$$

FAR, also known as Attack Presentation Classification Error Rate (APCER) [67], represents the weight of the False Negatives, or the PAs that were not detected by the algorithm, when compared to all of the

PAs, and, as previously said, is the error rate that is most alarming, being calculated as follows:

$$FAR = \frac{FN}{FN + TP} = \frac{FN}{PA} \tag{4.2}$$

On the other hand, the FRR, or *Bona-Fide* Presentation Classification Error Rate (BPCER) [67], represents the weight of the False Positives that were deemed as intrusions without actually being one, when compared to all of the samples that are *bona-fide* and is calculated as:

$$FRR = \frac{FP}{FP + TN} = \frac{FP}{Bona - Fide} \tag{4.3}$$

This way, it is possible to obtain knowledge from key conditions of the algorithm's performance which are not available when simply calculating the accuracy and can safely conclude on the overall performance of the technique.

4.2 Databases

When assessing the effectiveness of PAD techniques, publicly available face spoofing databases are often used for benchmarking. Several of these have been collected and made available, using different capturing devices, conditions and operating scenarios. In this section a brief description of each used database is given, and in Table 4.2 an overall summary of all used databases can be seen. Even though the following presented databases are not the most recently available, they were selected due to their difficulty and popularity amongst other state of the art algorithms, so a wide comparison can be made. The contents of each database consists in recordings of genuine user accesses and several types of presentation attack attempts.

Table 4.2: Summary of the database characteristics used for presentation attack detection.

Database	# Subjects	Acquisition Devices	# Scenarios	Presentation Attacks	Genuine/Attack Samples
Replay-Attack [24]	50	1 laptop	2	high definition print, replay	200/1000
CASIA-FASD [12]	50	3 webcameras	1	warped and cut print, replay	150/450

4.2.1 Replay-Attack

The Replay-Attack database [24] consists in video sequences recorded by the built-in camera of a MacBook Air 13-inch laptop, with a resolution of 320 x 240 pixels, where all videos, including either genuine users or spoofing attacks, had the duration of at least 9 seconds. A total of 50 users participated in this database acquisition, which was recorded under two illumination conditions: controlled, with an illumination support, providing an homogeneous background; and adverse, where natural illumination

was used in a more complex background. The presentation attacks presented were: high-resolution print attacks, where photos and videos were captured using a Canon PowerShot SX150 IS to record 12.1 Mpixel photographs with a 720p quality; mobile display attacks, where photos and videos were shown on an iPhone 3GS, with a resolution of 480 x 320; and high quality photos and videos displayed on an iPad screen, with a resolution of 1024 x 768. These artefacts were either held by the attacker or supported on a fixed stand. All of these previous examples are illustrated in Figure 4.1.

For evaluation and robust testing, the subjects were divided and could only participate in one of three disjoint subsets, either in training or development, each one with 60 genuine and 300 attack videos, or in testing, with 80 genuine and 400 attack videos. In the particular case of presentation attack detection and by following the dataset evaluation protocol, the set for training should be used to train the algorithm, whereas the development set should be used has a validation and also for hyper-parameter tuning. Finally, the testing dataset should be used to assess the algorithm's overall performance.

It is also important to note that this was the primarily used database to fine tune the neural network and test its alterations, later detailed in chapter 5, since it is the most used database by state of the art techniques, as well as the most robust since it presents diverse types of attacks with different conditions among different people.

Figure 4.1: Examples of genuine accesses and spoofing attacks from the Replay-Attack database. Column from left to right show examples of real accesses, printed photograph, display attack and tablet attack. Moreover, the top row is from a controlled scenario whereas the bottom is from an adverse scenario. [24]

4.2.2 CASIA Face Anti-Spoofing

The CASIA Face Anti-Spoofing Database (CASIA-FASD) [12] contains video recordings of genuine users and PAs generated from high quality videos of the users. This database contains data captured from 50 subjects whereas the PAs were constructed from the high quality recordings of the genuine users. Three types of attacks were considered: warped photo attack, where the facial motion is simulated by bending a photograph, as displayed in Figure 2.3; cut photo attack, or photographic masks, where the eye regions are cut off and the imposter wears the mask in order to mimic eye blinking or movement; and replay attacks. When regarding the capture sensors, images were acquired with three different imaging qualities: low with a long-time used USB camera that degrades image quality; normal,

using a newly bought USB camera which keeps the original quality; and high, using a Sony NEX-5 camera for recording. These have resolutions of 640 x 480, 480 x 640 and 1280 x 720, respectively. When it comes to presentation attacks, videos were shown using an iPad with a resolution of 1280 x 720 and in prints the same Sony high quality camera was used.

The complete image set for a subject is illustrated in Figure 4.2. This resulted in 12 videos per subject, 3 genuine and 9 fake, resulting in 600 clips in the database. All of these subjects were divided into disjoint datasets, separating the 50 subjects in training and testing groups, with 20 and 30 subjects, respectively, for comparison purposes.

Figure 4.2: Complete image set for a subject. The top left images represent the low quality videos, the bottom left normal quality videos and the right set are the high quality videos. For each set of photos from left to right: genuine, warped photo attack, cut photo attack and replay attack. [12]

Chapter 5

Experimental Results

In this chapter a description is given on several testing procedures. This chapter consists of six sections where the first five correspond of intermediate results, tests in order to fine tune as well as optimize the algorithm's performance, and the latter section is where a comparison between the proposed solution and state of the art techniques is made. Throughout all sections, several graphs from the algorithm's training are shown, these represent its accuracy and loss during training, on both the training set and validation set. As previously mentioned, accuracy represents the correctness of the algorithm. Moreover loss, in this case cross-entropy, describes how well is the model performing for each respective set. However, when assessing the network using the testing set, the HTER together with FAR and FRR are the preferred metrics.

Firstly, in section 5.1 a comparison in performance between the two previously discussed face detection algorithms is made. Afterwards, in section 5.1 an assessment between the two proposed face detection classifiers is made. Moreover, in the following section, 5.2 the advantages of Transfer Learning are assessed by training only parts of the neural network and freezing the rest. Furthermore, in section 5.3, an analysis when representing the input images in different colour spaces is made to conclude whether changing the original network's colour space has an impact in the results. Section 5.4 presents the obtained results when introducing the LSTM layer in the architecture. For this to be possible new hyper-parameters such as the timestep as well as the network's architecture have to be taken into consideration and fully analysed. Finally, in section 5.5, several architecture fusion methods are taken into consideration. These fusion techniques ideally compensate errors that only happen in the standalone architectures, by matching features from both architectures it is possible to achieve a better performance.

After fully tuning the network a comparison between the already used techniques needs to be made, which is discussed in section 5.6. In this section, the overall performance of the proposed solution is assessed, using the databases already mentioned in previous sections. Moreover, and to determine the robustness of the algorithm, demonstrating why PAD is still an open problem, a more demanding inter-database test is performed.

5.1 Face Detection

When comparing an older, simpler algorithm as the Viola-Jones to a more complex face detection neural network it is expected that the latter has an increase in performance. Furthermore it is expected that the first algorithm fails when adverse conditions are presented, such as bad lighting, and when the subject is not facing the capturing sensor. To confirm this, a still in-development unconstrained database was used, which consists of 50 subjects, where for each 36 images were captured. These were taken under normal lighting conditions, indoors and outdoors with different occlusions, facial and capture angle variations. A few sample images from this database can be seen in Figure 5.1.

Figure 5.1: Small sample of the captured unconstrained database. Many variations are displayed such as different levels of zoom, lighting and occlusions by hand or by looking the other way.

The test consisted in running both algorithms through the set of photos and see which one achieved the best performance. As expected the deep learning algorithm successfully detected all faces present in every photo while the Viola-Jones algorithm managed only to detect the faces that were front facing the sensor without any type of occlusion. Some examples where the faces were not detected by the Viola-Jones algorithm can be seen in Figure 5.2. In Figure 5.1, the face of the subject in the second and third photographs were also not identified. As shown by both examples, the cases where Viola-Jones is unsuccessful usually happen when a photograph is captured with adverse lighting conditions and/or the subject's face is occluded. If the subject is not frontally facing the camera, as is the case of half-profile or full-profile pictures, it also remains undetected by the algorithm. On the other hand, the used facial identification neural network proves to be successful in all of the cases.

Figure 5.2: Small sample of the failed cases when using the Viola-Jones algorithm for face identification.

When employing face detection in the PAD problem we can conclude that even though the used neural network achieves a higher performance, PAD sensors require the subject to be facing frontally when the capture occurs, without any type of occlusion. Therefore, Viola-Jones disadvantages might prove useful to presentation attack detection as the algorithm does not recognize a face when the previously explained adverse conditions are met whilst in the high performing neural network all condition variations prove no challenge to the algorithm. Concluding, both algorithms prove useful depending on the objective. Furthermore, when the used PAD databases were tested, the Viola-Jones algorithm successfully detected all the available faces.

5.2 Neural Network Fine Tuning

As it was referred in section 3.2, there is a similarity between face recognition and face presentation attack detection, which stills needs to identify discriminative features of the face. So, when adapting the VGG neural network to the PAD problem, we know that initial convolutional layers capture low-level image features, such as edges, while deeper convolutional layers capture increasingly more complex details [41]. This way, an hypothesis can be made in which the proposed neural network adaptation should achieve better performance by only fine tuning the last parts of the neural network instead of the whole model. In order to confirm this, a test was made in which the entirety of the network was frozen, meaning that the weights cannot be altered during training, keeping its original values, and, subsequently, deeper layers were unfrozen and trained, testing the performance of the network to the PAD problem. Such results can be seen in Table 5.1.

Table 5.1: Performance of the network when only fine tuning some of its layers.

Trained	# Parameters	Accuracy (%)	FAR (%)	FRR (%)	HTER (%)
Fully Connected (FC)	119,554,050	76.9	40.6	86.4	64.5
FC + 1 conv. block*	126,633,474	81.7	29.3	58.9	44.0
FC + 2 conv. block*	132,533,250	83.0	28.4	50.5	39.5
FC + 3 conv. block*	134,008,578	**89.4**	14.4	32.0	**23.2**
Full Network	134,268,738	84.2	21.0	53.0	37.0

*Conv. block represents the group of 2/3 convolutional blocks before the max pooling operation as it was explained in Figure 3.8.

As it is demonstrated by the table, highest performance can be achieved by skipping the training for the first two convolutional blocks since they present introductory operations and, by tuning the weights of these convolutional blocks, robustness and performance is lost. In fact, if we train the whole network, performance decreases, showing the advantages of Transfer Learning and confirming the presented hypotheses.

When performing training the expected result is that the training loss decreases while accuracy increases. However, this might not happen in the validation set. Regarding the relation between the validation accuracy and loss, the latter decreases as the training process takes place, except for some fluctuations introduced by the optimizer or dropout layers (which introduces random noise as previously

discussed) whereas the first increases. If the validation loss decreases and the validation accuracy increases the training process is going well since the lower the loss the higher the robustness of the network. On the other hand, if the validation loss starts increasing and validation accuracy starts decreasing we are in the presence of overfitting. Finally, if both the validation loss and accuracy increase, then it means that the regularization techniques (e.g dropout) are working against overfitting. This is only true if afterwards the loss starts decreasing whilst the accuracy increases, else the model is diverging.

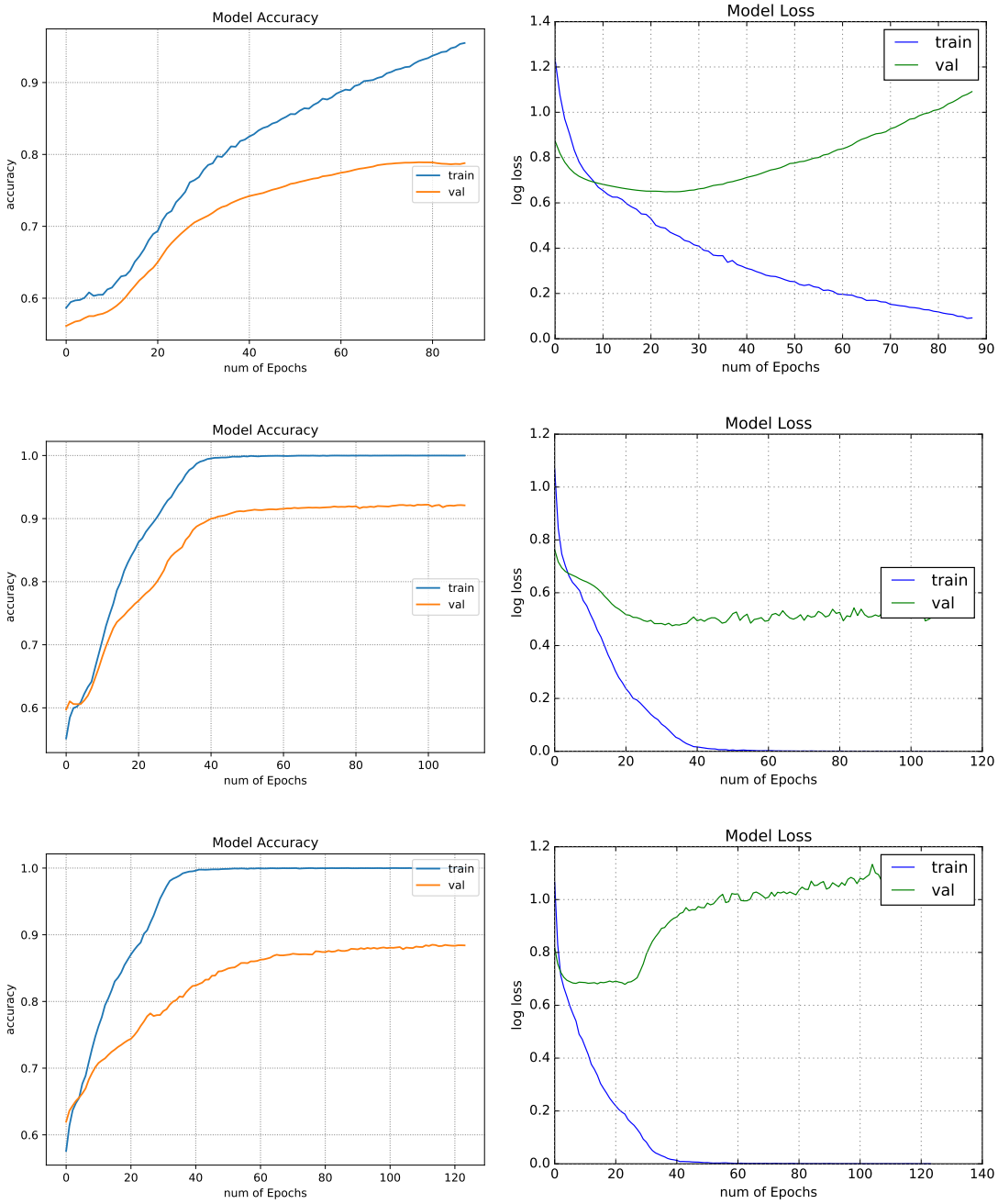


Figure 5.3: Network's performance of accuracy and loss during training. The first row corresponds to the training of only the FC layers, the second is of FC + 3 conv. block and third is the whole network. Curves labelled as "train" are originated from the training set and "val" are obtained when validating the network with the validation set.

The graphs relative to the training of the neural network are displayed in Figure 5.3. As it can be seen, training converges fairly quickly in all cases with the highest performance being when freezing the first two convulotional blocks of the architecture (as it can be seen in Figure 3.8). Even though the accuracy shows a high performance of the algorithm, above 90% in the validation set, the loss function presents a high error rate in the same set, which demonstrates a low robustness of the algorithm to other sets other than training. Furthermore, the calculated error rate in Table 5.1 is too high for the network to be considered a good solution to the problem. In order to further increase the algorithm's performance other colour schemes than RGB were considered and presented in the following subsection.

5.3 Colour Analysis

As discussed in section 2.2.4, performing an analysis in the RGB colour space can be quite limiting since there is a high correlation between the three channels, leading to a low quality feature extraction. Having this in consideration, other colour spaces that allow a separation between luminance and chrominance of a picture were considered. By fully separating these channels, the images are richer in features, presenting an increase in textural information. This way, a HSV and YC_bC_r analysis was performed. To do this and as previously mentioned, during the dataset pre-processing all images had to be converted to the corresponding colour space before giving it as input to the neural network. Furthermore, and as demonstrated in the preceding section, the algorithm achieved better performance when freezing the first two convolutional blocks of layers of the VGG descriptor. However, in this case, by changing the colour space the original training of the VGG network might not be fruitful to the problem since its original weights were trained using only RGB images. By changing the colour space of the training data, all of the weights might need to be adjusted to this new approach. In the following Table, 5.2, a fine tuning with different colour spaces analysis is displayed.

Table 5.2: Performance table of different colour spaces. Best values of each space highlighted in bold.

	RGB		HSV		YC_bC_r	
Trained	Accuracy (%)	HTER (%)	Accuracy (%)	HTER (%)	Accuracy (%)	HTER (%)
Fully Connected (FC)	76.9	63.5	93.2	14.6	87.4	27.7
FC + 1 conv. block*	81.7	44.0	96.2	12.4	89.1	22.8
FC + 2 conv. block*	83.0	39.5	95.0	10.5	93.6	13.3
FC + 3 conv. block*	**89.4**	**23.2**	96.7	7.0	94.4	11.7
Full Network	84.2	37.0	**96.9**	**6.5**	**95.7**	**8.5**

*Conv. block represents the group of 2/3 convolutional blocks before the max pooling operation as it was explained in Figure 3.8.

As displayed in the table, when using YC_bC_r and HSV a far better performance is achieved, being the highest value when performing a HSV analysis with only 6.5% of error (while in the RGB analysis the system presented an HTER of 23.2%). Furthermore, both the best values were obtained when training the whole network, which confirms the proposed hypothesis. Since the conditions of the current

training set, the colour space in this case, differ from the original training of the VGG network, all weights are no longer adapted to these new colour spaces and need tuning. The training graphs of the accuracy and loss of the model for both colour spaces can be seen in Figure 5.4. As it is possible to visualize, the validation accuracies are much higher when compared with the previous approach, displayed in the top row, and the losses are much lower. The solution also converges fairly quicker, taking leverage from the original network's weights proving that employing transfer learning still presents its benefits.

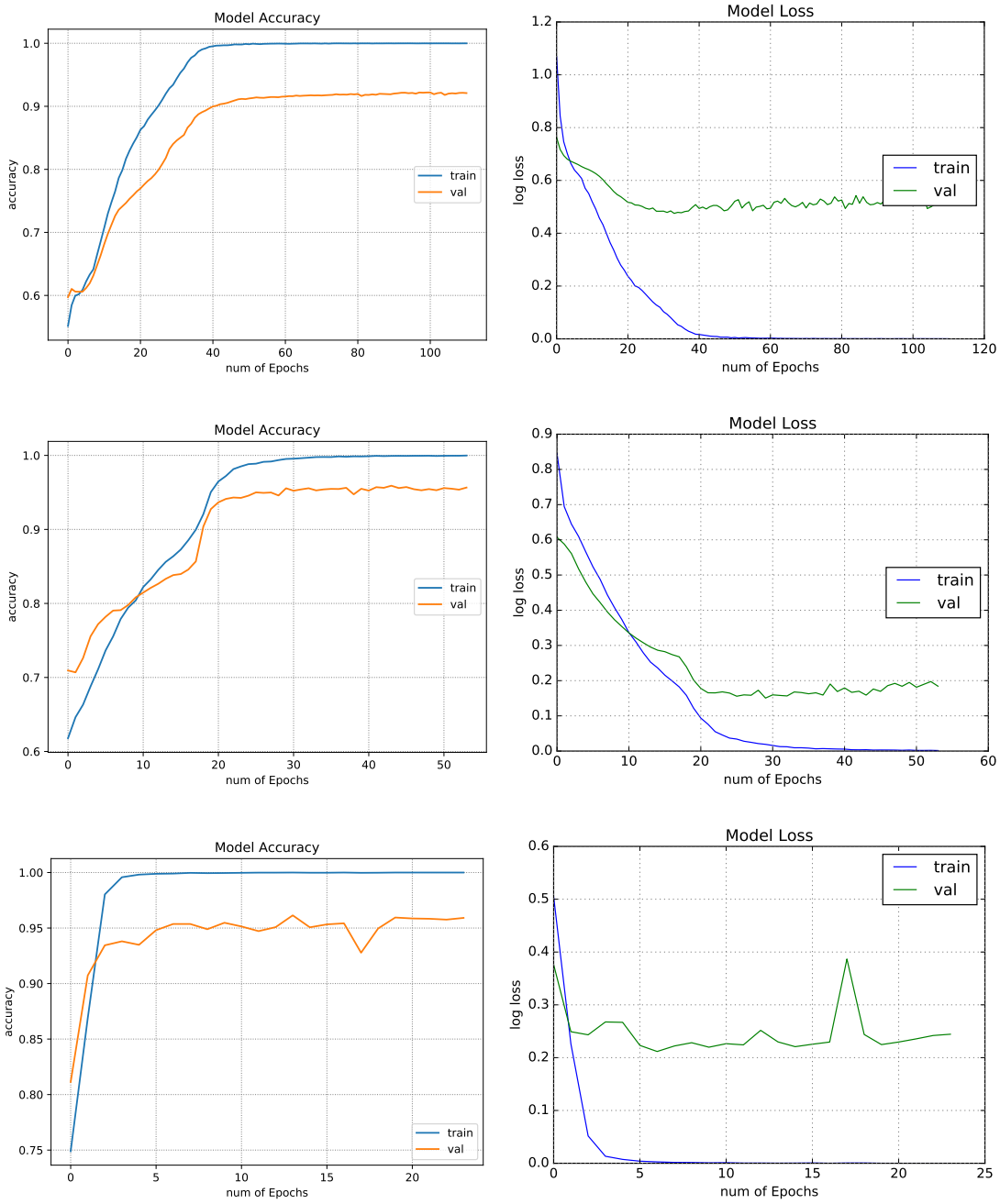


Figure 5.4: Network's performance of accuracy and loss during training. Top two graphs are from the RGB colour space, middle is HSV and bottom is YC_bC_r. Curves labelled as "train" are originated from the training set and "val" are obtained when validating the network through the validation set.

In addition, the spike that occurs in the YC_bC_r graph, where the highest loss and lowest accuracy occurs, shows the employed regularization techniques in action to avoid overfitting. The next step is to add the LSTM layer in the model so a time evolution analysis along a sequence of frames can be made, this way, the feature extraction will be much richer, extracting the already known features and adding the new, time and spatial related features.

5.4 Video Analysis

When regarding the analysis of consecutive frames, the usual type of used networks are the Recurrent Neural Networks (RNNs), as previously said, they introduce recursive operations and, through this, allow valuable information to persist throughout several iterations. However RNNs present a disadvantage, nominated as the vanishing gradient, where these networks become unable to learn any relevant connection between the presented information when the input array is large and the gap between the relevant data and the output increases. One type of RNN that solves this problem is the Long Short Term Memory (LSTM) layer.

When applying the LSTM, the RGB colour analysis was not considered and tests were only conducted using the HSV and YC_bC_r colour spaces since they presented the most promising results. During training the best models previously presented in Table 5.2 were used, meaning that the models with best performance for each colour space were selected. Consequently, they were then frozen and only the newly added LSTM layer is now being trained. This allows to test how adding the new RNN influences the system.

To begin with, a tuning of the LSTM layer hyper parameters has to be done. As it was detailed in section 3.5, a new assessment regarding different timesteps needs to be taken into consideration. As it was discussed, the timestep value allows to determine for how many frames, or for how long, the algorithm keeps analysing the video images before making a decision, forgetting previous iterations. If the timestep value is too low then it would result in a premature decision, which may lead to errors, and if it is too big it introduces noise, underperforming as well. Secondly, it is also important to experiment which architecture, discussed in section 3.5, is best for this specific analysis. From the four possible architectures, presented in Figure 3.14, only two were chosen for this specific problem as they are the most appropriate for the problem. The first is a many-to-one architecture where a sequence input has only one output and is often used for classification, as is the case. On the other hand, in a many-to-many synchronous architecture every input has an output classification, which might also be a possible solution for the problem. In order to apply these changes to the neural network's architecture, in the case of a many-to-many architecture, every output of the LSTM layer had to be given to the softmax, meaning that every frame produced a decision which had to be combined in the end by a softmax activation function, rather than having one final output produced only by the final LSTM layer, which is what happens when using the many-to-one architecture.

The results when tuning both of these hyper parameters were only obtained for the HSV colour space, since it had the best performance, and are summarized in Table 5.3. From the table it is possible to conclude that a many-to-one architecture is the most adequate for the problem as it achieves the lowest HTER values. This is mainly because this approach is better for action classification, which is the presented case, to classify whether the sensor is in the presence of a *bona-fide* sample or a presentation attack. Better performance is achieved by outputting an answer after analysing the whole group of frames rather than deciding in each singular frame what action is taking place, since we do have more information after the process. Regarding the timestep analysis, the best value found is seven, this shows that by analysing groups of seven frames, which corresponds between a three to five seconds window in the presented videos, the best possible outcome is achieved.

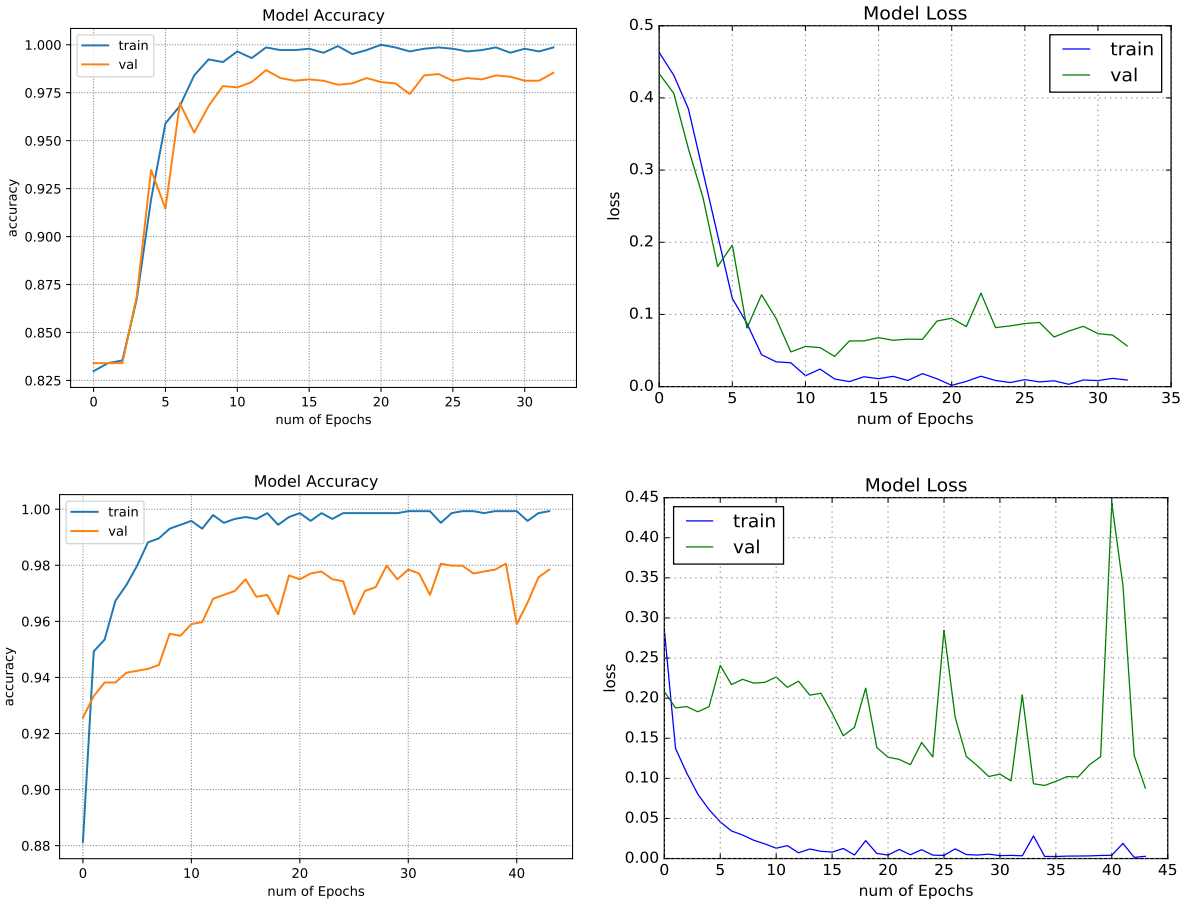


Figure 5.5: Network's performance when adding a LSTM layer with 7 as the chosen timestep with a many-to-one architecture. Top row is HSV analysis and bottom is YC_bC_r.

Table 5.3: Network's performance with various timesteps and different architecture. All values are in % and best highlighted in bold.

	Many-to-one				Many-to-many			
Timestep	Accuracy	FAR	FRR	HTER	Accuracy	FAR	FRR	HTER
5	98.5	1.6	7.8	4.7	97.8	4.5	6.1	5.3
6	98.8	2.3	4.1	3.2	98.1	4.3	3.9	4.1
7	**99.2**	0.6	1.6	**1.1**	98.4	4.0	2.8	3.4
8	97.9	4.3	2.4	3.4	97.7	6.5	3.1	4.8
9	97.1	9.0	3.0	6.0	96.0	11.0	4.4	7.7

After obtaining the hyper parameter values for the timestep as well as the network's architecture, tests were also conducted using the YC_bC_r colour space and the results with and without the LSTM layer are summarized in Table 5.4. Furthermore, in Figure 5.5, it is possible to visualise the training graphs of the models. In this case the accuracy reaches its best value when compared with the previously reported approaches, and the loss is extremely low (0,1 represents certainties near the 90% as was shown in Figure 3.11). By analysing both the table and the graph, it is possible to conclude that the LSTM layer does in fact provide with newer and richer features, allowing a more wholesome and versatile analysis.

Table 5.4: Comparison in both colour spaces with and without the LSTM layer. All values are in %.

	HSV				YCbCr			
	Accuracy	FAR	FRR	HTER	Accuracy	FAR	FRR	HTER
Without LSTM	96.9	2.3	10.7	6.5	95.7	11.8	5.3	8.5
With LSTM	99.2	0.6	1.6	1.1	99.0	0.5	2.5	1.4

5.5 Fusion Testing

As it was previously introduced in 2.2.4, the combination of two or more feature extraction methods is often possible, being referred to as multi-biometrics. These fusion systems increase the complexity of the system and allow, in the best case scenario, to correct mistakes of each standalone techniques. This is possible because if there is a condition where a method fails, it might no longer fail when combined with another that performs well in said condition. However, and as previously said, the blind combination of methods just to increase the complexity of the system often proves unfruitful and it even might reduce the technique's performance. This is mainly due to the manipulation of the capture data, used by one method, which corrupts or alters the same data, leaving it unusable for the other method, ruining the fusion. So for fusion techniques to be possible, the used methods need to be independent and able to complement one another.

In a multi-biometric system, fusion can be performed depending on the type of information available, usually being divided in four modules [68]: sensor, feature extraction, matcher and decision module.

Consequently, these four modules can lead to four different levels of fusion, as displayed in Figure 5.6, and explained as follows:

- Sensor level: This fusion happens prior to matching, which usually means that once the matcher of a biometric system is invoked the amount of information available decreases. This fusion level refers to the matching of either the same trait acquired from different sensors or from different traits acquired from the same sensor;
- Feature level: This level of fusion also happens prior to matching and it refers to a combination of different feature sets extracted from multiple biometric sources or techniques. When the feature sets are homogeneous, such as multiple measurements of the same person's hand geometry for example, a single feature vector might be calculated by combining the weights of each. When the feature sets are heterogeneous, sets that came from different biometric methods or are simply not relatable, concatenation is possible;
- Score level: This fusion happens after matching and it refers to the combination of match scores generated by different methods. The resulting score can further be classified into combination or classification. In combination, the individual matching scores are combined to generate a single scalar score, which is used to make a final decision, *e.g*: probability fusion such as average, sum, product, etc. On the other hand, in classification a feature vector is constructed using the matching scores output by the two methods and then classified as accepted or rejected;
- Decision level: In this level of fusion the final decision outputs by each standalone system is consolidated by different types of techniques. An example could be minutiae based matching, used in fingerprint biometrics where several algorithms are able to detect key point, minutiae, on the ridges in fingers. Other can be texture based matching that filters an image and combines with others.

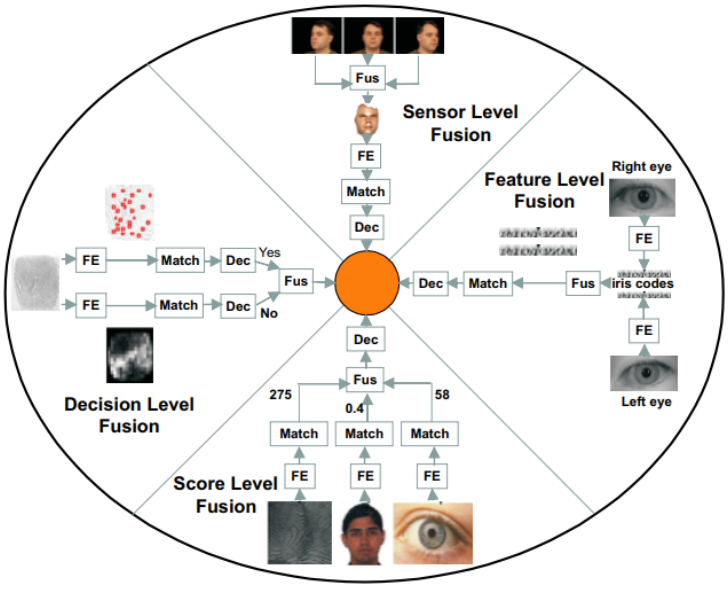


Figure 5.6: Different levels of fusion techniques in a biometric system. [68]

In order to conclude if any of the fusion levels introduced above can be employed using the standalone methods, first an analysis of its errors must be made. By fully analysing the best model's misclassifications, in each colour space, when using the test set of the Replay-Attack database summarized in Table 5.5, it is possible to notice that even though the number of errors are small, the majority of samples where they occur do not match. So, if a fusion of these colour space analysis were to be possible, the error rate might decrease.

The classification array of each colour space is given by the probabilities of a sample being *bona-fide* or a presentation attack. Ideally, in the presence of a bona-fide sample the output vector would be (1, 0), being the first array position the certainty, between 0 and 1, given by the network that it is in the presence of a genuine user and the latter the certainty, between 0 and 1 as well, of it being an attack.

In this case with the employed methods, two fusion techniques were experimented. The first fusion taking place at the score level, since both methods output probability arrays, a fusion of these arrays might be possible using score combination techniques. The second fusion can happen at the feature level, where the non homogeneous features, originated from different colour analysis, can be combined by concatenation. Sensor level and decision level fusion were not taken into consideration since these levels do not exist or cannot be specified in this architecture. It is also important to mention that the following fusion methods were performed using the output of the best models in each colour space, previously presented in Table 5.4.

Table 5.5: Number of misclassifications of each colour space when using the test set in the Replay-Attack database. There were a total of 360 bona-fide samples and 1080 presentation attack samples.

	HSV		YCbCr	
	Bona-fide	Presentation Attack	Bona-fide	Presentation Attack
Misclassifications	5/360[1]	10/1080[2]	8/360[3]	11/1080[4]

ID of samples that were misclassified [1]: 66, 68-71; [2]: 485, 1172-1175, 1676, 1768-1771; [3]: 232-239; [4]: 1136, 1144, 11280-1283, 1464, 1768-1771

5.5.1 Score Level Fusion

A score level fusion approach consists in running the two models independently from one another and fusing the probabilities of each class that is produced by the softmax classifier. Several score level methods were used:

- **Maximum**, being the higher probability of all the four cases, (*bona-fide* and PA for the two colour spaces) considered as the correct one;
- **Summation**, being the two probability vectors summed;
- **Product**, being the two probability vectors multiplied;

- **Weighted value between cross-misses**, this weight corresponds to the reliability of each colour space and its calculation is based on the errors from each class (*bona-fide* and PA) in the validation set, where W_1 and W_2 is the *bona-fide* and PA weight, respectively, and CS a colour space. Being the weights for one of the colour spaces given by the following expressions:

$$\begin{aligned} CS_1: \quad & W_1 = 1 - \frac{CS_1(FN)}{CS_1(FN) + CS_2(FN)}, \quad W_2 = 1 - \frac{CS_1(FP)}{CS_1(FP) + CS_2(FP)}; \\ CS_2: \quad & W_1 = 1 - \frac{CS_2(FN)}{CS_1(FN) + CS_2(FN)}, \quad W_2 = 1 - \frac{CS_2(FP)}{CS_1(FP) + CS_2(FP)}; \end{aligned} \tag{5.1}$$

- **Weighted values between correct cases**, which represents the confidence for each class, *bona-fide* or PA. It is calculated as the number of misclassifications divided by the sum of all existing cases:

$$\begin{aligned} CS_1: \quad & W_1 = 1 - \frac{CS_1(FN)}{CS_1(TN)}, \quad W_2 = 1 - \frac{CS_1(FP)}{CS_1(TP)}; \\ CS_2: \quad & W_1 = 1 - \frac{CS_2(FN)}{CS_2(TN)}, \quad W_2 = 1 - \frac{CS_2(FP)}{CS_2(TP)}; \end{aligned} \tag{5.2}$$

The results of these fusion metrics are displayed in Table 5.6. Even though there is an improvement when compared to the YC_bC_r colour space, the standalone HSV analysis has a higher performance than any of the fusion results which do not appear to have almost any difference in performance. After a thorough analysis it is possible to conclude that this happens due to the certainty of the algorithm which is always above 90%, even when it is the case of it being incorrect. With such high percentages any score fusion technique will not work well since it only helps in solving cases where there exists some uncertainty by the network.

5.5.2 Feature Level Fusion

Another method for joining the two colour spaces' results is at the feature level. This fusion type consists in joining the two network's architectures by concatenating the resulting features before giving them to the softmax classifier, as illustrated in Figure 5.7. This method has the objective of training the classifier with features from both colour spaces, this way, a more vast analysis can be made, since there are a more variety of features. In order to do this, and due to insufficient memory, the models had to be run in parallel and the output features, from the LSTM layer, were saved in both situations. Afterwards, a new model was created with only the softmax classifier which was trained with the concatenated feature vector.

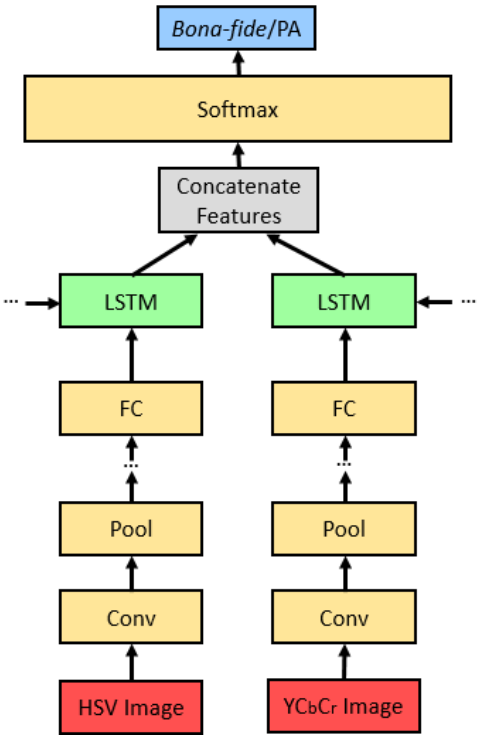


Figure 5.7: Architecture of a feature level fusion by concatenating the resulting features of the LSTM layer. In red are displayed the inputs, in blue the output, in green the RNN layer and the rest is the adapted VGG network.

The results are displayed in Table 5.6 and as it can be seen feature level fusion does not increase the algorithm's performance. This is mainly due to the high range of the feature vector, while these are useful and discriminative in their standalone architecture, when concatenating these features they become redundant, introducing too much noise in a way that any decision can be made. This way, it can be concluded that fusion techniques do not prove fruitful in this case since the standalone HSV architecture has the best performance when compared to the considered fusion techniques.

Table 5.6: Results of both score and feature fusion methods when compared to each standalone colour analysis.

	Accuracy (%)	HTER (%)
Standalone HSV	99.2	1.1
Standalone YC_bC_r	99.0	1.4
Maximum	98.1	2.7
Summation	99.1	1.2
Product	99.1	1.2
Weighted Cross-Miss	99.1	1.2
Weighted TN/TP	99.1	1.2
Feature level	96.6	3.3

5.6 Comparison with State of the Art Methods

After fully developing and tuning the proposed system, a performance assessment and comparison against state of the art methods is presented. The algorithms considered for comparison are the ones reporting the best performance values until this date, proposing mostly textural and/or motion analysis solutions, or other deep learning algorithms.

Two assessment evaluations were made, firstly each database was used for training and testing independent from one another, being named as intra-database test. On the other hand, to test the generalization ability of the proposed solution and others, in terms of robustness, an inter-database test was performed. In this latter assessment, one database is used to train the algorithm, but testing is performed with sets from the other database. This type of test allows assessing the full robustness of the systems in a more demanding cross-database scenario since by changing the databases every aspect of acquisition is completely altered: conditions, image quality, people, capturing sensor and so on.

The results of the first test are summarized in Table 5.7. It shows that amongst all of the tested databases, the algorithm performs well with quite good results when compared to other techniques. By analysing with more detail the conditions where misclassifications happen, it is possible to conclude that the majority of errors happen when high quality presentation attacks are given to the sensor, specifically when photographic masks are presented to the sensor, in the CASIA dataset. In a high quality image, usually with a high resolution, around 1024 x 720 pixels, the face is detected and then it is resized by an image with 224 x 224 pixels. This greatly downsamples the target image. With this, rich features that can identify the attacks are lost together with quality and, by analysing the remainder features it leads to an attack being incorrectly identified as genuine. An example of this is again the case of photographic masks, which are prints with a high quality, where both the eyes and mouth are cut out in order to mimic normal face movements. By downsampling these images the algorithm cannot detect the usual print attack features, such as the paper reflection and its low image texture when compared to a genuine face, but can detect movement which influences the system into accepting the sample as genuine.

Table 5.7: Performance and comparison between state of the art methods.

Methods	**Replay-Attack**	**CASIA-FASD**
	HTER (%)	
LBP[24], (2012)	13.80	18.20
LBP - TOP [69], (2012)	7.60	10.60
LBP - GLCM [70], (2013)	7.20	-
Motion [71], (2013)	11.70	26.00
Motion + LBP [72], (2013)	5.10	-
Motion Mag [73], (2014)	0.25	14.40
Deep Learning [74], (2015)	2.10	7.34
Fine-Tuned VGG [60], (2017)	4.30	-
DPCNN [60], (2017)	6.10	-
Nonlinear Diffusion CNN [43], (2017)	10.00	-
FASNet (CNN) [61], (2017)	1.20	-
Patch + Depth CNN [44], (2017)	0.72	2.27
(1)(HSV + YCbCr) LBP [28], (2018)	2.90	6.09
Proposed Method	1.09	10.32

As it is displayed in the previous table, most algorithms present a good performance throughout all of the chosen databases, which may raise doubts concerning the necessity of the still ongoing investigation about face presentation attack detection. If there are algorithms that achieve such low error rates then why is PAD an open problem. In spite of these methods achieving high performance rates when following each datasets' set of rules when performing an intra-database test, the same does not apply in the inter-database test, as displayed in Table 5.8.

Table 5.8: Performance and comparison between state of the art methods in an inter-database test [28].

	Replay-Attack (trained on CASIA-FASD)		CASIA-FASD (trained on Replay-Attack)	
Test on:	**Dev**	**Test**	**Train**	**Test**
Methods	HTER (%)			
LBP[24], (2012)	44.9	47.0	57.3	57.9
LBP - TOP[69], (2012)	48.9	50.6	60.0	61.3
Motion[71], (2013)	50.2	50.2	47.7	48.2
Correlation[71], (2013)	47.7	48.3	50.2	50.2
Motion Mag[73], (2014)	50.0	50.2	43.8	50.3
Deep Learning[74], (2015)	48.2	48.8	45.7	45.4
(1) + SVM - RBF[28], (2018)	22.5	20.6	47.5	43.9
(1) + SVM - linear[28], (2018)	17.7	16.7	38.6	37.6
Proposed Method	50.5	49.2	44.5	45.3

This latter table shows that unfortunately all of the state of the art methods, including the proposed solution, perform poorly when displayed with a more challenging robustness test. By completely changing all of the conditions in a database, all algorithms fail, which only shows that most of them would perform poorly in a real world situation.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis presented a novel approach, utilizing deep learning, to detect attacks to facial recognition systems. In order to create an acceptable algorithm, the challenges faced by the community were evaluated, analysing in detail what are the most common type of attacks, how are they performed and what do they take advantage of when breaking in a system. Moreover, an evaluation of the current state of the art framework was made, describing the several existing algorithms and its different approaches when addressing the problem. Techniques were subdivided depending on the type of analysis and a particular focus was given to methods that use machine learning, also explaining why is it widely used nowadays and how it can bring improvements to this specific task. As a result an approach was presented, which adapts to different conditions and variabilities not focusing only on specific attacks but rather in its training, trying to create a robust model.

Throughout the process of developing the algorithm several milestones were achieved, Transfer Learning presented as being a good starting method when the problems are similar, allowing to reuse the original weights to the new task, achieving a high performance with a fast convergence. Either way, even if the problems are not similar or the training process differs, the architecture can be used as a starting point to a new network, as it was shown. Regarding colour analysis, and as it was demonstrated, the RGB colour space has a poor performance when regarding presentation attack detection since there is a high correlation between the three channels. Therefore, other colour spaces should be taken into consideration, preferably ones that take the luminance and chrominance of a picture into consideration as they are better for image recognition, such as YC_bC_r and HSV, where the latter achieved the best performance. Fusing these two colour spaces, either in feature or score level, is possible and it might prove advantageous in algorithms that present some uncertainty, or when the decision of the algorithm is unclear and has a low probability output, which is not the case. Instead of performing the usual single frame analysis, the architecture of the network was altered so that a full video analysis could be performed. For this, the LSTM layer was used and it proved fruitful to the task, it allowed a spatio temporal analysis of a picture, analysing its feature evolution throughout the video. This layer has a high

adaptability to different problems, achieving the best performance in a many to one architecture and with a timestep of 7, in this case.

The suggested method has acceptable results, presenting a higher performance when compared to state of the art methods, showing that deep learning can be a good solution to the problem. During the intra-database test, misclassifications happened mostly when high quality presentation attacks were captured by the sensor, mainly due to the high loss of texture features when downsampling the target image. In order to demonstrate why PAD is still an open problem and there is not an accepted solution by the community, an inter-database test was made. This assessment demonstrated that almost all proposed state of the art methods have a high error rate when presented with completely new capture conditions and/or unexpected types of attacks. In the particular case of the proposed method, the algorithm underperforms in the inter-database test as expected since the testing set differs drastically from its training, presenting a disadvantage of deep learning. Algorithms based on deep learning work well within the training set specifications and variations however, if new data outside the training range emerges, it proves unsuccessful. Presentation attack detection presents an open problem since it has many variations in every condition possible as well as its wide range of attacks.

6.2 Future Work

When regarding inter-database analysis and since the proposed solution fails most often when the target image is downsampled, losing rich features that can identify a PA, an analysis without changing the original image size should be made. In order for this to be possible, the input layer of the network should accept any image size, instead of the expected 224 x 224 pixel face image. This might be possible by using the Spatial Pyramid Pooling (SPP) layer [75] instead of the usual max pooling layer. As previously mentioned, a max pooling layer resizes the number of features in order to reduce their dimensionality, however both input and output size need to be previously determined values. By using a SPP layer, the target size is outputted regardless of the input size, having no input size regulation. Although in theory this seems possible, this layer is still in development showing few practical situations, therefore its viability needs to be verified before it can be applied. This approach would ideally increase the number of richer features in the image since there is no initial resizing needed, leaving the facial region of the image untouched. Other method that might present a solution to the downsampling problem would be to use patches of the target image instead of resizing the entirety of the image. This way, training data would increase and the original quality would be kept.

With respect to the robustness of the algorithm when using a test set that greatly differs from training, the performance decreases in all shown algorithms, presenting to be a deep learning disadvantage in this particular case. One possible solution may be using generative adversarial networks or adversarial neural networks [76], which generates an increasing dataset with several variations. In this scheme, there are two neural networks, one called generator, which generates new data instances based on the training set, while the other, the discriminator, evaluates them for authenticity, trying to assign the correct label to each new instance. So the generator creates new instances trying to fool the discriminator into

accepting them as authentic while they are not, training the discriminator for new, different cases. When comparing to a usual neural network, which outputs a label when analysing the features, an adversarial network generates features taking the label into consideration, this allows to create different combinations of features that can represent a genuine sample or a presentation attack, being based, however, on the training set as well. Unfortunately, this type of networks cannot cover all the possible scenario variations, it may improve the algorithm but probably would not fully fix its miscalculations. These misclassifications are not so easily corrected since the algorithm mostly depends on its training. In order to seclude this, restricting capture conditions should be the most plausible approach, if some of the conditions could be manipulated or controlled, when possible, then the number of various scenarios would drastically decrease which, by consequence, would increase the algorithm's performance to disparate conditions.

Bibliography

[1] J. Galbally, S. Marcel, and J. Fierrez. Biometric Antispoofing Methods: A Survey in Face Recognition - IEEE Journals & Magazine. 2:1–23, 2015. doi: 10.1109/ACCESS.2014.2381273.

[2] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: A tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143, 2006. ISSN 15566013. doi: 10.1109/TIFS.2006.873653.

[3] Biometrics market and industry report (Web), 2014. URL `https://www.ibia.org/`. Last Accessed on 2018-03-27.

[4] E. Dunkley. Facial Recognition in Banking (Web), 2017. URL `https://www.ft.com/content/923fec7c-205c-11e7-b7d3-163f5a7f229c`. Last Accessed on 2018-03-30.

[5] INTERNATIONAL STANDARD ISO / IEC Information technology — Biometric presentation attack detection —. 2016.

[6] S. R. Arashloo and J. Kittler. Face Spoofing Detection Based on Multiple Descriptor Fusion Using Multiscale Dynamic Binarized Statistical Image Features. *IEEE Transactions on Information Forensics and Security*, 10(11):2396–2407, 2015. ISSN 15566013. doi: 10.1109/TIFS.2015.2458700.

[7] Z. Xu, S. Li, and W. Deng. Learning temporal features using LSTM-CNN architecture for face anti-spoofing. *Proceedings - 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015*, pages 141–145, 2016. doi: 10.1109/ACPR.2015.7486482.

[8] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001. ISSN 0018-8670. doi: 10.1147/sj.403.0614.

[9] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Professional Computing, 2003.

[10] V. Vijayan, K. W. Bowyer, P. J. Flynn, D. Huang, L. Chen, M. Hansen, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris. Twins 3D face recognition challenge. *2011 International Joint Conference on Biometrics, IJCB 2011*, 2011. doi: 10.1109/IJCB.2011.6117491.

[11] Jukka Komulainen. *Software-based countermeasures to 2D facial spoofing attacks — Center for Machine Vision and Signal Analysis*. 2015. ISBN 9789526208725.

[12] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A Face Antispoofing Database with Diverse Attacks. *5th IAPR International Conference on Biometrics (ICB'12)*, pages 2–7, 2012.

[13] R. Illusion. Commercial Animation Software (Web). URL `https://www.reallusion.com/crazytalk/`. Last Accessed on 2018-05-16.

[14] Creafx. Artificial Masks (Web). URL `http://www.creafx.com/en/shop/`. Last Accessed on 2018-05-16.

[15] A. Sepas-Moghaddam, F. Pereira, and P. L. Correia. Light Field-Based Face Presentation Attack Detection: Reviewing, Benchmarking and One Step Further. *IEEE Transactions on Information Forensics and Security*, 13(7):1696–1709, 2018. ISSN 15566013. doi: 10.1109/TIFS.2018.2799427.

[16] P. Majaranta and A. Bulling. Advances in Physiological Computing. 2014. doi: 10.1007/978-1-4471-6392-3.

[17] R. Ramachandra and C. Busch. Presentation Attack Detection Methods for Face Recognition Systems. *ACM Computing Surveys*, 50(1):1–37, 2017. ISSN 03600300. doi: 10.1145/3038924.

[18] D. Tang, Z. Zhou, Y. Zhang, and K. Zhang. Face Flashing: a Secure Liveness Detection Protocol based on Light Reflections. (January), 2018.

[19] N. Erdogmus and S. Marcel. Spoofing face recognition with 3D masks. *IEEE Transactions on Information Forensics and Security*, 9(7):1084–1097, 2014. ISSN 15566013. doi: 10.1109/TIFS.2014.2322255.

[20] R. Raghavendra, K. B. Raja, and C. Busch. Presentation attack detection for face recognition using light field camera. *IEEE Transactions on Image Processing*, 24(3):1060–1075, 2015. ISSN 10577149. doi: 10.1109/TIP.2015.2395951.

[21] Z. Zhang, D. Yi, Z. Lei, and S. Z. Li. Face liveness detection by learning multispectral reflectance distributions. *2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011*, pages 436–441, 2011. doi: 10.1109/FG.2011.5771438.

[22] J. Yan, Z. Zhang, Z. Lei, D. Yi, and S. Z. Li. Face liveness detection by exploring multiple scenic clues. *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, pages 188–193, 2012. doi: 10.1109/ICARCV.2012.6485156.

[23] J. Maatta, A. Hadid, and M. Pietikainen. Face spoofing detection from single images using texture and local shape analysis. *Biometrics, IET*, 1(1):3–10, 2012. ISSN 2047-4938. doi: 10.1049/iet-bmt.2011.0009.

[24] I. Chingovska, A. Anjos, and E. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. *International Conference of the Biometrics Special Interest Group*, pages 1–7, 2012. ISSN 1617-5468.

[25] J. Kannala and E. Rahtu. BSIF: Binarized statistical image features. *21st International Conference on Pattern Recognition (ICPR)*, (Icpr):1363–1366, 2012. ISSN 10514651. doi: 10.0/Linux-x86_64.

[26] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5099 LNCS:236–243, 2008. ISSN 03029743. doi: 10.1007/978-3-540-69905-7_27.

[27] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face Spoofing Detection Using Colour Texture Analysis. *Computer Vision-ACCV 2012 Workshops*, 11(8):146–157, 2013. ISSN 1556-6013. doi: 10.1109/TIFS.2016.2555286.

[28] L. Li, P. L. Correia, and A. Hadid. Face recognition under spoofing attacks: countermeasures and research directions. *IET Biometrics*, 7(1):3–14, 2018. ISSN 2047-4938. doi: 10.1049/iet-bmt.2017.0089.

[29] D. A. Shepherd. Color Image Processing and Applications. (July 2015):0–2, 2015. doi: 10.3109/10717544.2013.779332.

[30] G. Kim, S. Eum, J. K. Suhr, D. I. Kim, K. R. Park, and J. Kim. Face liveness detection based on texture and frequency analyses. *Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012*, pages 67–72, 2012. doi: 10.1109/ICB.2012.6199760.

[31] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh. Computationally efficient face spoofing detection with motion magnification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 105–110, 2013. ISSN 21607508. doi: 10.1109/CVPRW.2013.23.

[32] B. Peixoto, C. Michelassi, and A. Rocha. Face liveness detection under bad illumination conditions. *Proceedings - International Conference on Image Processing, ICIP*, pages 3557–3560, 2011. ISSN 15224880. doi: 10.1109/ICIP.2011.6116484.

[33] S. Kim, S. Yu, K. Kim, Y. Ban, and S. Lee. Face Liveness Detection Using Variable Focusing. *International Conference on Biometrics (ICB)*, pages 1–6, 2013. doi: 10.1109/ICB.2013.6613002.

[34] W. Shuigen, C. Zhen, and D. Hua. Motion Detection Based on Temporal Difference Method and Optical Flow field. *2009 Second International Symposium on Electronic Commerce and Security*, (May):85–88, 2009. doi: 10.1109/ISECS.2009.62.

[35] W. Yin, Y. Ming, and L. Tian. A Face Anti-Spoofing Method Based on Optical Flow Field. pages 1333–1338, 2016.

[36] B. Biggio, Z. Akthar, G. Fumera, G. L. Marcialis, and F. Roli. Robustness of multi-modal biometric verification systems under realistic spoofing attacks. *2011 International Joint Conference on Biometrics, IJCB 2011*, 2011. doi: 10.1109/IJCB.2011.6117474.

[37] E. A. Rúa, H. Bredin, C. G. Mateo, G. Chollet, and D. G. Jiménez. Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models. *Pattern Analysis and Applications*, 12(3):271–284, 2009. ISSN 14337541. doi: 10.1007/s10044-008-0121-2.

[38] Occipital. 3D scanning, augmented reality, and more for mobile devices (Web). URL `https://structure.io/`. Last Accessed on 2018-05-22.

[39] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 15730565. doi: 10.1023/A:1022627411411.

[40] C. Li and B. Wang. Fisher Linear Discriminant Analysis. pages 1–6, 2014. ISSN 10459227. doi: 10.1109/NNSP.1999.788121.

[41] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. ISSN 14764687. doi: 10.1038/nature14539.

[42] J. Yang, Z. Lei, and S. Z. Li. Learn Convolutional Neural Network for Face Anti-Spoofing. 2014.

[43] A. Alotaibi and A. Mahmood. Deep face liveness detection based on nonlinear diffusion using convolution neural network. *Signal, Image and Video Processing*, 11(4):713–720, 2017. ISSN 18631711. doi: 10.1007/s11760-016-1014-2.

[44] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face Anti-Spoofing Using Patch and Depth-Based CNNs. *IEEE International Joint Conference on Biometrics*, pages 319–328, 2017.

[45] Y. Rehman, L. Po, and L. Mengyang. Deep learning for face anti-spoofing: An end-to-end approach. *IEEE Signal Processing Magazine*, pages 195–200, 2017.

[46] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518, 2001. ISSN 1063-6919. doi: 10.1109/CVPR.2001.990517.

[47] S. Hochreiter and J. J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1–32, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[48] A. Geitgey. Modern Face Recognition with Deep Learning, 2016. URL `http://tinyurl.com/hak4lcv`. Last Accessed on 2018-09-14.

[49] G. B. Huang and E. Learned-Miller. Labeled faces in the wild : Updates and new reporting procedures. *University of Massachusetts Amherst Technical Report*, 2014.

[50] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, I: 886–893, 2005. ISSN 1063-6919. doi: 10.1109/CVPR.2005.177.

[51] V. Kazemi and J. Sullivan. One Millisecond Face Alignment with an Assemble of Regression Trees. *27th IEEE Conference on Computer Vision and Pattern Recognition*, (August):1867–1874, 2014. ISSN 978-1-4799-5118-5. doi: 10.13140/2.1.1212.2243.

[52] L. Torrey and J. Shavlik. Transfer Learning. *Machine Learning*, pages 1–22, 2009. ISSN 0219-7200. doi: 10.1016/j.jbi.2011.04.009.

[53] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. *Procedings of the British Machine Vision Conference 2015*, (Section 3):41.1–41.12, 2015. ISSN 00313203. doi: 10.5244/C.29.41.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 15731405. doi: 10.1007/s11263-015-0816-y.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012. ISSN 10495258. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014. ISSN 15337928. doi: 10.1214/12-AOS1000.

[57] P. Golik and P. Doetsch. Cross-Entropy vs. Squared Error Training:a Theoretical and Experimental Comparison. *Interspeech, Isca*, 2(2):1756–1760, 2013.

[58] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. pages 1–15, 2014. ISSN 09252312. doi: http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503.

[59] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning Word Vectors for Sentiment Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, 2011. ISSN 9781932432879. doi: 978-1-932432-87-9.

[60] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid. An original face anti-spoofing approach using partial convolutional neural network. *2016 6th International Conference on Image Processing Theory, Tools and Applications, IPTA 2016*, (i), 2017. doi: 10.1109/IPTA.2016.7821013.

[61] O. Lucena, A. Junior, V. Moia, R. Souza, E. Valle, and R. Lotufo. Transfer Learning Using Convolutional Neural Networks for Face Anti-Spoofing. 10317(July), 2017. doi: 10.1007/978-3-319-59876-5.

[62] Colah. Understanding LSTM Networks. URL `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`. Last Accessed on 2018-07-25.

[63] A. Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. URL `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`. Last Accessed on 2018-07-25.

[64] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[65] F. and others Chollet. Keras, 2015. URL `https://keras.io`. Last Accessed on 2018-07-27.

[66] J. M. Samy Bengio. A Statistical Significance Test For Person Authentication. *Proceedings of Odyssey 2004: The Speaker and Language Recognition Workshop*, (2), 2004.

[67] A. Hadid, N. Evans, S. Marcel, and J. Fierrez. Biometrics Systems Under Spoofing Attack: An evaluation methodology and lessons learned. *IEEE Signal Processing Magazine*, 32(5):20–30, 2015. ISSN 10535888. doi: 10.1109/MSP.2015.2437652.

[68] D. Lal, B. Bhushan, and C. Kant. Making Biometric Systems More Robust With Multibiometrics. 5 (1):211–214, 2012.

[69] T. De Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. LBP-TOP based countermeasure against face spoofing attacks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7728 LNCS(PART 1):121–132, 2013. ISSN 03029743. doi: 10.1007/978-3-642-37410-4_11.

[70] M. A. Waris, H. Zhang, I. Ahmad, S. Kiranyaz, and M. Gabbouj. Analysis of textural features for face biometric anti-spoofing. *European Signal Processing Conference*, pages 1–5, 2013. ISSN 22195491.

[71] T. De Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Can face anti-spoofing countermeasures work in a real world scenario? *Proceedings - 2013 International Conference on Biometrics, ICB 2013*, 2013. doi: 10.1109/ICB.2013.6612981.

[72] J. Komulainen, A. Hadid, M. Pietikainen, A. Anjos, and S. Marcel. Complementary countermeasures for detecting scenic face spoofing attacks. *Proceedings - 2013 International Conference on Biometrics, ICB 2013*, 2013. doi: 10.1109/ICB.2013.6612968.

[73] S. Bharadwaj, S. Member, T. I. Dhamecha, and S. Member. Face Anti-spoofing via Motion Magnification and Multifeature Videolet Aggregation. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (ii):1–12, 2014.

[74] D. Menotti, G. Chiachia, and A. Pinto. Deep representations for iris, face, and fingerprint spoofing detection. *International Journal of Professional AL Engineering Students*, VIII(3):138–142, 2017.

[75] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9): 1904–1916, 2015. ISSN 01628828. doi: 10.1109/TPAMI.2015.2389824.

[76] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Adversarial Generative Nets: Neural Network Attacks on State-of-the-Art Face Recognition. 2017.