CONVENTIONAL AND NEURAL ARCHITECTURES FOR BIOMETRIC PRESENTATION ATTACK DETECTION

A Thesis Submitted to the University of Kent

For the Degree of Doctor of Philosophy

In Electronic Engineering

By

Shi Pan

University of Kent

August 2019

Abstract

Facial biometrics, which enable an efficient and reliable method of person recognition, have been growing continuously as an active sub-area of computer vision. Automatic face recognition offers a natural and non-intrusive method for recognising users from their facial characteristics. However, facial recognition systems are vulnerable to presentation attacks (or spoofing attacks) when an attacker attempts to hide their true identity and masquerades as a valid user by misleading the biometric system. Thus, Facial Presentation Attack Detection (Facial PAD) (or facial anti-spoofing) techniques that aim to protect face recognition systems from such attacks, have been attracting more research attention in recent years. Various systems and algorithms have been proposed and evaluated. This thesis explores and compares some novel directions for detecting facial presentation attacks, including traditional features as well as approaches based on deep learning. In particular, different features encapsulating temporal information are developed and explored for describing the dynamic characteristics in presentation attacks. Hand-crafted features, deep neural architectures and their possible extensions are explored for their application in PAD.

The proposed novel traditional features address the problem of modelling distinct representations of presentation attacks in the temporal domain and consider two possible branches: behaviour-level and texture-level temporal information. The behaviour-level feature is developed from a symbolic system that was widely used in psychological studies and automated emotion analysis. Other proposed traditional features aim to capture the distinct differences in image quality, shadings and skin reflections by using dynamic texture descriptors.

This thesis then explores deep learning approaches using different pre-trained neural architectures with the aim of improving detection performance. In doing so, this thesis also explores visualisations of the internal representation of the networks to inform the further development of such approaches for improving performance and suggest possible new directions for future research. These directions include interpretable capability of deep learning approaches for PAD and a fully automatic system design capability in which the network architecture and parameters are

determined by the available data. The interpretable capability can produce justifications for PAD decisions through both natural language and saliency map formats. Such systems can lead to further performance improvement through the use of an attention sub-network by learning from the justifications.

Designing optimum deep neural architectures for PAD is still a complex problem that requires substantial effort from human experts. For this reason, the necessity of producing a system that can automatically design the neural architecture for a particular task is clear. A gradient-based neural architecture search algorithm is explored and extended through the development of different optimisation functions for designing the neural architectures for PAD automatically. These possible extensions of the deep learning approaches for PAD were evaluated using challenging benchmark datasets and the potential of the proposed approaches were demonstrated by comparing with the state-of-the-art techniques and published results.

The proposed methods were evaluated and analysed using publicly available datasets. Results from the experiments demonstrate the usefulness of temporal information and the potential benefits of applying deep learning techniques for presentation attack detection. In particular, the use of explanations for improving usability and performance of deep learning PAD techniques and automatic techniques for the design of PAD neural architectures show considerable promise for future development.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACER	Average Classification Error Rate
APCER	Attack Presentation Classification Error Rate
AUC	Area Under Curve
BPCER	Bona Fide Presentation Classification Error Rate
BSIF	Binarised Statistical Image Feature
CCPAD	Colour Convolutional Presentation Attack Detection Network
CLBP	Colour LBPs
CNN	Convolutional Neural Network
CoALBPs	Co-occurrence of Adjacent LBPs
DACN	Dynamic Attention Convolutional Network
DL	Deep Learning
DMD	Dynamic Model Decomposition
DNN	Deep Neural Network
DTL	Deep Transfer Learning
EAC-Net	Enhancing and Cropping Net
EER	Equal Error Rate
ELU	Exponential Linear Unit
FA	False Acceptance
FAR	False Acceptance Rate
FACN	Frame Attention Convolutional Network
FAUH	Facial Action Unit Histogram
FASAN	Facial Action Signal Analysis Network
FC	Fully Connected
FLOPs	Floating-Point Operations Per second
FR	False Rejection
FRR	False Rejection Rate
GAN	Generative Adversarial Network
HSV	Hue Saturation Value
LBP	Local Binary Patterns

LBPV	LBP Variance
LDP	Local Derivative Pattern
LFW	Labelled Faces in the Wild
LNF	Local Neural Field
MHI	Motion History Image
MHP	Motion History Patterns
ML	Machine Learning
NAS	Neural Architecture Search
PA	Presentation Attack
PAD	Presentation Attack Detection
PAD-NAS	Presentation Attack Detection Neural Architecture Search Network
PAI	Presentation Attack Instrument
PR	Pattern Recognition
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TCoALBP	Temporal CoALBP
TNR	True Negative Rate
TPR	True Positive Rate
TPU	Tensor Processing Unit
TOP	Three Orthogonal Planes

Chapter 1: Introduction

This chapter states the objectives of the program of research, includes definitions of the key concepts and variables, and gives a brief outline of the background and research approach. The aim of the introduction is to contextualise the proposed research.

In this chapter, section 1.1 will outline the background information and the keywords of the research. Then, section 1.2 will provide the purposes and the significance of this research. Finally, section 1.3 will includes an outline of the remaining chapters of the thesis.

1.1 BACKGROUND

Biometric technology deeply changes people's daily lives and stimulates the developments of large-scale identity management solutions. Protecting private information, securing financial activities, and other possible applications rely on a trustworthy authentication system which should also be non-invasive, user friendly, and process efficiently with low costs.

Biometric authentication systems, which aim to recognise personal identity by analysing and measuring biological characteristics (such as fingerprints, irises, facial patterns, etc.) or behaviours (such as voice, etc.), are becoming increasingly common using personal characteristics that are universal and easy to acquire.

Biometric systems overcome some weaknesses of conventional authentication systems (e.g. knowledge-based system using password, token-based system using ID cards) that stem from inappropriate usage. For instance, selecting a complex password may be hard to attack but also hard to remember. Furthermore, some users tend to use the same password for several applications. The token-based systems are vulnerable to the loss of the token and to the token being shared.

Automatic facial recognition systems, on the other hand, overcome most of these weaknesses. Facial biometric systems offer an increasingly effective and accurate authentication experience for users using the latest progress in machine learning and computer vision. Since Sun, et al [1] published 'DeepID' in 2014, the reported

accuracy of published facial recognition systems dramatically increased when tested on published datasets such as LFW (Labeled Faces in the Wild) [2] that incorporate different environmental conditions such as various illuminations, poses, and image quality.

The progress of facial recognition has also stimulated the development of consumable electronic devices, which incorporate facial biometric authentication system. For instance, Apple[3] announced their first device that included a facial biometric authentication system on September 12, 2017. AliPay [4]announced their facial biometric payment system named 'Smile To Pay' in 2017. These developments, which previously were only available for large organisations, bring the benefits of biometric systems to the average citizen.

Alongside the increasing adoption of biometric technologies, the potential threat of sensor-level spoofing or presentation attacks has also increased rapidly. Current facial biometric authentication systems are vulnerable to presentation attacks in which the attackers aim to masquerade as another user and mislead the biometric systems by presenting fake facial biometric information in front of the sensors. Facial presentation attacks are relatively easy to create and hard to detect. The popularity of social networks such as Facebook[5] make high-quality identity-bearing facial information easily available and biometric information can be shared at almost no cost. The developments of low-cost 3D printing technology further decease the cost of creating an attack artefact. For these reasons, facial spoofing detection research has attracted much attention in recent years.

Techniques for protecting biometric systems from presentation attacks are referred to as Presentation Attack Detection (PAD) or anti-spoofing methods[6]. The range and quality of possible artefacts and application environments create particular challenges for PAD. For facial recognition systems, presentation attacks can be categorized by the type of the attack artefacts, including printed papers (paper attack), display screens (video replay attack), and 3D masks (mask attack) [6]. In this thesis, genuine biometric samples from valid users are also referred to as bona-fide class. For developing and evaluating robust PAD algorithms, various public datasets, including various attack types and environmental conditions, are used.

In general, presentation attacks can be recognised by human observers via the material differences between genuine faces and attack artefacts; the different

representations between the non-rigid facial movements and rigid movements for artefacts, and the texture differences between the recaptured images and original images. Thus, some researchers in this area aim to build software-based methods that can detect attacks without costly additional hardware. This research direction is known as software-based PAD or feature-based PAD. One of the key assumptions of current feature-based PAD research is that attacks can be detected by the distorted information that is injected into the sensor data during the spoofing attack by the material of attack artefacts, changing both the spatial and temporal appearance of the data when compared with a bona-fide presentation. PAD research, therefore, has explored both static and dynamic feature-based methods, in which the static methods only aim at detecting the traces of artefacts in the spatial domain and the dynamic methods also aim at such detection in the temporal domain. Dynamic feature-based methods have also been explored in the past, such as using texture differences, motion differences and image quality differences for PAD.

More recent research using Deep Neural Networks (DNNs) has presented new possibilities for PAD without the need for using 'hand-crafted' features. One of the popular ways of using deep learning for PAD is the use of pre-trained DNN features which demonstrate some promising results when evaluated on widely used datasets. However, the opacity of these approaches may be considered as a significant weakness in biometric applications in which particular decisions to deny or grant access to individuals must be justified.

Presentation attacks remain one of the main problems for the existing facial biometric authentication systems, and there is a need for developing better PAD systems that can detect presentation attacks efficiently and can be robust when facing various attack types.

Here is a small glimpse of the next generation of the PAD system from my imagination. The PAD system in the future should be robust for various presentation attacks. It should also be computationally efficient on the mobile platforms (such as mobile phone, AR/VR devices, etc.). It should be trustworthy for different users who have different backgrounds. It should continuously learn from data, which only include a limited number of samples and lake of labels. The research target of this thesis is to push the boundary of current research and try to find a possible route to the better PAD system in the future.

The main objectives of this thesis are as follows:

(a) Explore different features to improve the performance of PAD systems.

(b) Dynamic (time-varying) biometric data provide a greater ability for humans to distinguish whether they come from genuine face presentations [19]. This thesis aims to find some efficient way to use temporal information for PAD.

(c) Deep learning stimulates significant performance improvements in PAD, but it also has some significant disadvantages. This thesis also aims to overcome the "black-box" nature of Deep neural networks. Meanwhile, this thesis aims to build efficient neural networks for PAD without requiring extensive works of designing neural architectures.

1.2 CONTRIBUTIONS OF THE THESIS

An extensive literature review of software-based spoof detection schemes is presented as the first step in this thesis, and the details of the experimental framework used for the evaluation of presentation attack systems are provided.

The main focus of this thesis is the exploration of new features for presentation attack detection to improve performance and initiate new directions for research. The proposed methods, which consider short video frame sequences captured with consumable cameras as input data, can be roughly divided into traditional features and features based on deep learning.

There are four novel traditional features explored in this thesis thatexplore distinct temporal information. Detecting presentation attacks by using temporal information is an intuitive idea in some early research. However, using temporal information requires more computational resources, and high-quality video attacks can still mislead biometric systems in some cases. The proposed methods focus on overcoming the disadvantages of the existing methods. One of the proposed methods considers a symbolic system for facial movements to represent unconscious facial motions.

Other proposed traditional features aim to focus on the temporal texture patterns that are distinct to PAD. These methods aim to provide computationally efficient features for temporal information. First, the Motion-History Patterns are used to

compress the temporal texture changes into one frame and different texture feature descriptors provide final feature vectors. The proposed spatio-temporal template of the Motion-History Patterns can be considered as a novel framework to produce temporal information for PAD which is different from the existing methods using temporal orthogonal planes. Secondly, the temporal texture co-occurrence is observed to be distinct for some presentation attacks. The proposed feature combines the co-occurrence matrix and a widely used local texture descriptor to get feature representations for PA. The third traditional feature uses dynamic textures inspired by a widely used pre-processing method. The proposed feature firstly generates a set of clustered pixels as an intermediate representation of the raw input. The local texture descriptors are then generated for each pixel grid. The final feature is generated from these local texture descriptors by following the bag-of-words approach. These proposed traditional features are evaluated by using some benchmark datasets.

The proposed features based on deep learning are based on two widely used research paradigms in the deep learning area. First, a novel neural architecture is designed and trained with benchmark datasets for presentation attack detection. Deep learning can be considered a representation-learning algorithm based on large-scale data. The limited volume of training datasets in some applications has led to the feature extraction part of some proposed deep neural networks being trained using large-scale datasets for different tasks. The proposed PAD experiments also explore this paradigm and provide some new results by using different datasets and different feature extraction components of some existing deep neural networks.This thesis provides some visualisation experiments to analyse and understand the behaviour of deep neural networks. .

Some ideas from the proposed traditional features have also inspired some of the novel deep learning based approaches proposed in this thesis. The symbolic system, which is considered by a traditional feature, is also modelled with recurrent neural networks, and the performance is improved when evaluated using different datasets. The feature representation for temporal local texture patterns and the discriminative cues for spoofing attacks are explored by using a novel patch-based 3D convolutional neural network, which efficiently extracts the spatio-temporal information. The effectiveness of these proposed deep learning methods is demonstrated by the results

when they are evaluated using benchmark datasets and compared with existing methods.

The rise of deep learning approaches also offers some new possible research directions for PAD. The proposed methods, which extend the current research boundaries, focus on adding some new functionality to PAD systems. First, a PAD system is proposed that can not only detect presentation attacks, but also provide the justification for its decisions using both natural language descriptions and saliency maps. The proposed system can answer questions such as "Why did the system make this decision?". Furthermore, an attention mechanism is proposed for this system that improves performance by learning from these justifications. Second, a Neural Architecture Search (NAS) method is used to automatically propose a neural architecture to suit the training data. This circumvents the need for designing a novel neural architecture for PAD 'by hand' and the experiments for the searched neural architectures provide some encouraging results for benchmarking datasets.

All of the proposed methods are evaluated using benchmarking datasets and compared with state-of-the-art methods. Possible directions for future work are also discussed.

1.3 THESIS OUTLINE

The outline of this thesis is as follows:

Chapter 1 is an introduction to the thesis and presents the general background of the topic, the contribution of the thesis, and a brief summary of the chapter contents.

Chapter 2 first introduces the basic concepts for biometric system and presentation attacks using definitions from the research literature and international standards. A literature survey is then provided to present a review of existing algorithms.

Chapter 3 provides the experimental framework for the thesis, which includes experimental design, pre-processing algorithms, related datasets and evaluation metrics, and a brief view of a workflow for presentation attack detection systems.

Chapter 4, as a main contribution chapter, provides four different novel features for anti-spoofing, and the effectiveness of the proposed features is demonstrated by comparing them with the state-of-the-art methods in the same categories.

Chapter 5, as the second contribution chapter, introduces three novel deep learning presentation attack detection methods that follow different directions and provide new results for the widely used transfer learning paradigm.

Chapter 6, as the third contribution chapter, attempts to extend the current research boundaries by integrating an interpretable capability for a deep learning based anti-spoofing system, and introduces the neural architecture search paradigm to discover an efficient deep neural architecture automatically.

Chapter 7 concludes the thesis with a summary of its findings and contributions. An outline of possible directions for future work is also provided.

Chapter 2: Literature Review

This chapter provides a general introduction to the research literature related to biometric systems and presentation attack detection for facial recognition systems. Basic concepts and definitions from international standards for biometric and facial anti-spoofing are provided in Section 2.1, in which research gaps in this area are also identified to guide the following chapters and highlight the contributions of proposed works. The potential threat of presentation attacks that is accompanies the widely used biometric-based authorization has recently attracted more attention. To detect this kind of attack, researchers have explored various methods, including the use of dedicated hardware and software, aiming to improve detection performance. These techniques are covered in Section 2.2 which after a broad overview focuses on software-based facial anti-spoofing (using data input with Red Green Blue channels (RGB)). Finally, Section 2.3 provides some suggestions as to what kind of methods may be helpful to push the boundary of PAD performance.

2.1 Problem definition and introduction

2.1.1 Biometric and facial biometric systems

Biometrics as an active research area aims to recognise users' identity by detecting biological features (for instance vein patterns , fingerprints, and faces) and behavioural characteristics (such as gait, etc.) [6]. The etymology of the term 'biometric' is 'bios' and 'metric' which mean 'life' and 'to measure' in ancient Greek [7]. Biometrics offers an attractive solution for 'problem of lost and forget' in the traditional security research. In the new paradigm offered by biometrics, "forget about cards and passwords, you are your own key"[8]. Biometrics as a sub-area of Pattern Recognition (PR) and Machine Learning (ML) have been attracting substantial interest from researchers in the security area. Researchers explored various techniques from other related areas (such as image processing, computer vision, speech recognition, etc.) for different applications of biometrics (such as automatic voice and facial recognitions) to provide reliable and efficient methods based on extracting physical and behavioural characteristics from users. Table 2.1 demonstrates the advantages of the biometrics-based authentication systems in comparison to traditional methods.

Table 2.1 Comparison of existing authorization systems

Name	Methods based on	Examples	Properties
Knowledge	Something users know	User ID, Passwords or PIN, etc.	It can be shared with others Many passwords are easy to guess It can be forgotten
Possessions	Something users have	Cards, Keys etc.	It can be shared with others It can be duplicated It can be stolen
Knowledge and Possessions	Something users know and something users have	Card and PIN pair, etc.	It can be shared with others Many PINs are written on the card
Biometrics	Something unique about users	Face, Iris, etc.	Cannot be shared Cannot be stolen Can hardly change

Biometric systems aim to recognise personal identities by using the features automatically generated from users' physical traits or behavioural characteristics. In these systems, one or more biometric characteristics can be fused for better performances, For example, faces or fingerprints can be fused with other biometric characteristics. Before using a biometric system, there is an enrolment stage to collect biometric samples from valid users and generate templates for authentication.

There are two different modes of using a biometric system: *Verification* and *Identification*. *Verification* means matching the associated personal identity with the claimed template and verifying whether a person has same as the identity that they claimed to process. Typically, the *Verification* is a one-to-one matching process, which only accepts the users who have same personal identity as they claim. *Identification* is a one-to-many matching process. In this mode, users do not need to claim who they are, the biometric system will automatically generate a result. Both of these two modes of operation are widely used for different purposes.

Key features of a biometric system, which are important for the robustness and reliability of the system, are based on the following five points[9]:

(1) **Universality**: the proposed biometric characteristic is present for all users.

(2) **Uniqueness**: the proposed biometric characteristic should include unique representations for different users.

(3) **Permanence**: the proposed physical and/or behavioural traits are stable with the passage of time.

(4) **Collectability**: the proposed physical and/or behavioural traits should be measurable.

(5) **Acceptability**: the major users and public should not strongly resist the biometric measuring process.

Based on these considerations, facial recognition is considered an effective biometric technology, because: (a) facial traits can be collected in a non-intrusive way; (b) the collection process can be done at a distance, even without the need for users to cooperate with the system; (c)although some facial characteristics may change with the passage of time, some distinct facial traits still appear to be persistent over long periods of time to make identity recognition possible. Some papers report that facial biometric systems had the second largest market share after fingerprint recognition in 2007 [10]. The *Security Systems Market Trends* predicted that the biometric market will theoretically surpass $51.98 billion in 2023 and the facial biometric systems are considered as a significant growth point due to dramatic increases in the number of consumer products, such as Face ID for iPhone and iPad[11].

The rapid development of processors, portable cameras, and batteries stimulates the growth of consumable devices and creates various scenarios for biometric systems such as boarding control, mobile account authentication, forensics, and intelligent surveillance cameras. Furthermore, the popularity of mobile apps and social networks significantly enlarge the potential demands of biometric authentication systems.

The widespread use of electronic passports[12] in the last 15 years demonstrates the effectiveness of facial biometric systems[13]. However, the facial biometric systems are currently under threat from Presentation Attacks (PA).

2.1.2 Presentation Attacks and Presentation Attack Detection

Facial anti-spoofing is an active research area that aims to protect facial-based biometric systems from a possible type of attack called presentation attack. Attacks threatening biometric systems can be broadly categorised into two types [14]: direct attacks (PAs) and indirect attacks. PA is classified as a direct attack by[14]:

"Presentation to the biometric data capture subsystem with the goal of interfering with the operation of the biometric system."

Figure 2.1 [14]: illustrates possible attack points for a facial biometric system. In this type of system, there are several modules and points that could be the target of an attack (arrows 1 to 8). Presentation attacks are performed at sensor level (arrow 1) without needing to access to the interior of the system. Indirect attacks (arrows 2 to 8) can be performed in areas such as the databases, the matchers, or the communication channels; in this type of attack the attacker needs access to the interior of the system. In this figure, the grey boxes represent the main modules of a facial biometric system, the black arrows indicate the dataflow, and the red arrows indicate the possible attack points. The red arrows 2 to 8 represent indirect attacks that can be prevented by changing the inner processing functions of the biometric system. For instance, the changes of communication channels between modules, different sensors and infrastructures involved in the system, can easily block the indirect attacks which require the attackers to have a detailed understanding of the system. However, presentation attacks do not require substantial knowledge of the system, and it is difficult for the system to detect them.

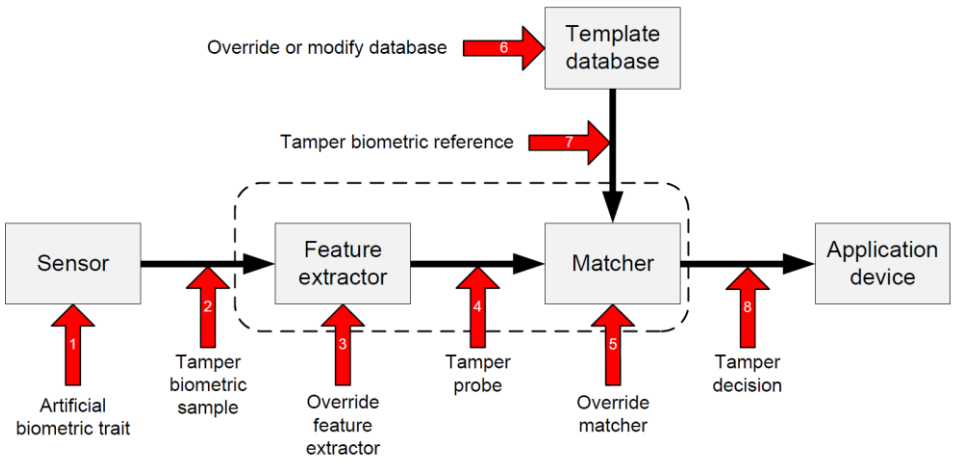


Figure 2.1 Vulnerability of a biometric system. [14]

The advantages of presentation attacks can be summarised as follows [9]: (a) they only need to present fake facial biometric information in front of the sensors; (b) the artefact for the presentation attack is easy to produce at low cost; (c) no extra knowledge about the operational biometric system is required to produce a successful attack; (d) the inner changes to the biometric systems, such as protecting the communication channels or encrypting the biometric templates, will not protect the system from presentation attacks.

Real attack cases have been reported in the past 10 years and demonstrate the potential risk of using facial biometric systems. At the Black Hat Conference in 2009 [15], presentation attacks were simulated for existing facial recognition systems that were widely used on laptops from different manufacturers[16] . The security and vulnerability research team from University of Hanoi showed that spoofing attack can simply subvert different manufacturers' systems (Lenovo's Veriface III, Asus' SmartLogon V1.0.0005, and Toshiba's Face Recognition 2.0.2.32 – each set to its highest security level) by only using some fake images thatincluded facial information from legitimate users.

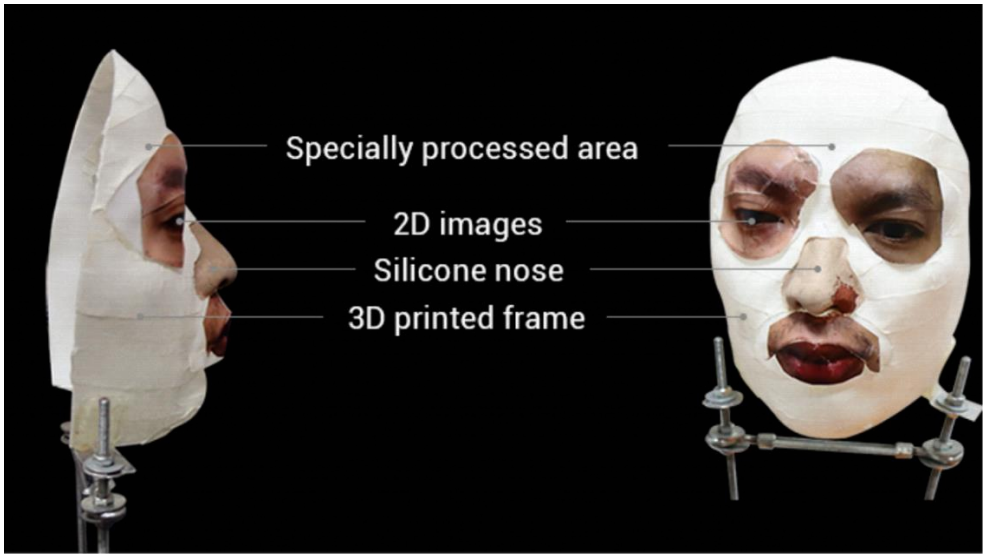


Figure 2.2 The 3D printed mask that fooled an iPhone X. Image: Bkav [17]

FaceID from Apple, which was claimed to be "ultra-secure" was hacked using a 3D printed mask that cost less than $150 only a week after its release [17]. Researchers from the Vietnamese cybersecurity firm Bkav built a 3D mask, which included facial information from valid users, by using a 3D printer (Figure 2.2). According to their report, this was not a sophisticated 3D mask, and some areas in the mask were not even

coloured as human skin. These examples show the threat posed by presentation attacks, and the popularity of social networks increases this threat by decreasing the cost of acquiring high-quality biometric information from valid users.

The process and detailed information about creating attack artefacts can be easily found on various web pages. The price of 3D printers is decreasing to an affordable level for personal usage which also decreases the cost of producing a 3D mask. Attackers may use such facial information without their victims' awareness. These developments have stimulated research in PAD to protect existing and future biometric systems.

Detecting presentation attacks is not as well explored as facial recognition. The term *Presentation Attack Detection* or *liveness detection* are also used in some papers in relation to detecting eye blinks or facial movements. This thesis is more focused on the general area of PAD, and the terms PAD and Facial Anti-Spoofing are used in the following chapters. [14]

In facial biometric research, the main focus of previous research has been given to improve the performance of the verification and identification tasks. Improving the performance of recognition systems in the presence of various environmental factors (such as occlusions, low-resolution, different viewpoints, lighting, etc.) is currently an active area in facial recognition research. In contrast, the security vulnerabilities of facial recognition systems have been studied much less.Furthermore, among the new issues and challenges that have emerged around biometrics, resilience against external threats has drawn a significant level of attention lately.

Presentation attacks arebroadly classified into two types by ISO [14] :

(1) "Biometric imposter, where the subversive biometric capture subject intends to be recognized as an individual other than him/herself."

(2) "Biometric concealer, where the subversive biometric capture subject intends to evade being recognized as any individual known to the system."

By following these definitions, the presentation attack is reported in different ways. Normally, a presentation attack for an authentication system means that, attackers present a biometric artefact of a legitimate user, who has been enrolled in the biometric system. Conversely, a presentation attack for an identification system (in an open set application) means the attacker can conceal his or her identity by presenting

disguised or altered biometric characteristics [18].Using the methods proposed in this thesis, the presentation attack should be detected in both cases.

2.2 Face attack types and artefacts

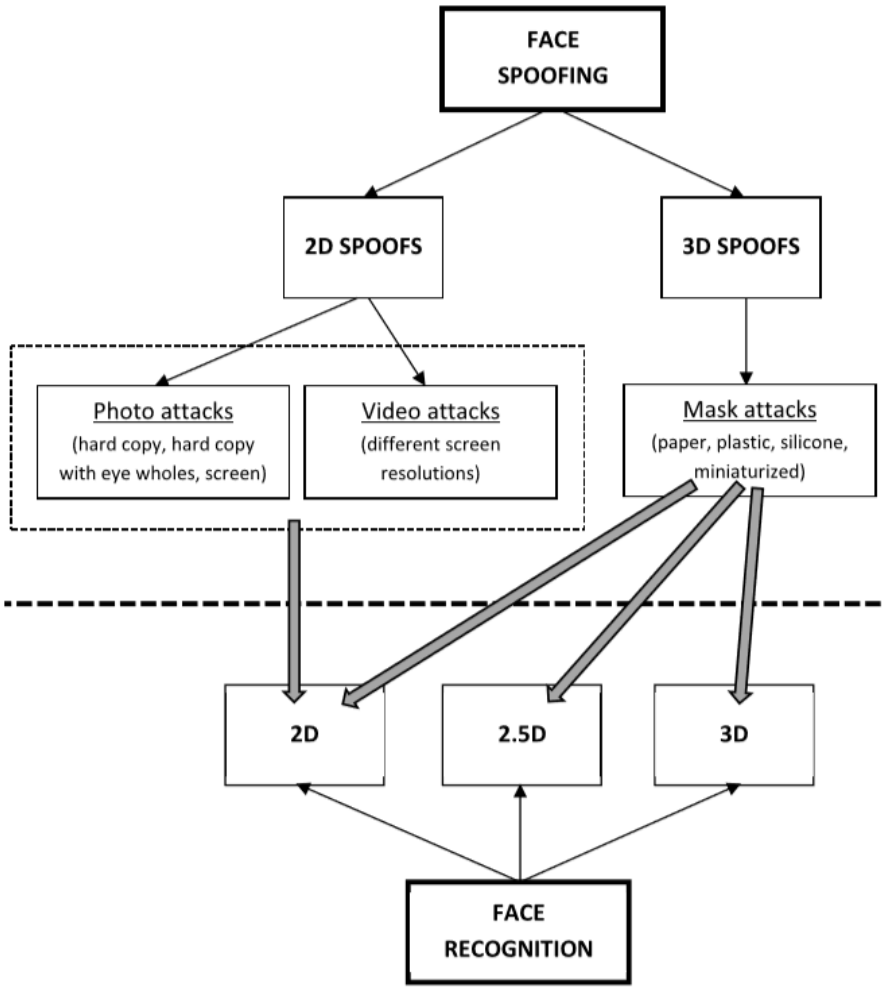


Figure 2.3 General classification of face spoofing (Presentation Attack) techniques studied in the literature. Grey arrows indicate the face recognition technology for which each attack represents a potential threat.[19]

An artefact, which is used for PA, is defined in ISO/IEC JTC1 SC37 documents [14]: as an artificial object for presenting "a copy of biometric characteristics" or "synthetic biometric patterns". The term for the biometric characteristic or object used in a presentation attack is the Presentation Attack Instrument (PAI) [14]. By following this definition, the photo, for example, which includes the biometric information from a valid user, and which is presented in front of a biometric system to mislead the biometric system, could be considered as PAI. From the research literature, the progress of research is constrained by the limited availability of data.

From the literatures, the existing presentation attacks may be classified in one of two groups, as shown in Figure. 2.3.

a) 2D instrument (e.g., photo, video) which may successful mislead the 2D face recognition systems without a PAD subsystem.

b) 3D instrument (e.g., masks) which may successfully attack 2D, 2.5D and 3D face recognition systems.

In some literatures, researchers also classify the attacks into three main types: (1) Photo Attacks (the PA instrument is a photo print with a laser jet printer), (2) Video Attacks(the PA instrument is an electronic display of a photo or video of a face) (3) Mask Attacks (the PA instrument is a 3D face mask).

(a) (b) (c) (d) (e)

Figure 2.4 (a) Bona fide facial image and examples of face artefacts: (b) laser print face artefact; (c) display face photo artefact using an iPad; (d) inkjet print face artefact; (e) 3D face mask.[20]

(a) (b) (c) (d)

Figure 2.5 Illustration of face artefacts generated using the legitimate user photo obtained from a social website: (a) photo from the social website, (b) inkjet print, (c) electronic display, and (d) laser print. [20]

Figures 2.4 and 2.5 show examples of PAIs which are described before. Figure 2.4 shows the attacks and bona fide subjects at an indoor environment, and Figure 2.4 shows the attacks at an outdoor environment. Especially, Figure 2.4 (e) demonstrates examples of 3D face masks produced by ThatMyFace company [21].

The photo attack means that the biometric samples for a fraudulent access attempts are presented by using a photograph. The biometric samples (photograph) may have been taken by the attacker using a digital camera. Sometimes, attackers can get the high-quality biometric sample even from the internet. Especially, the high-

quality facial images can be found at the social networks such as Facebook.[22] . The facial images can then be printed on a paper and displayed in front of the camera. Some recent works also consider to display a static image by using a screen and this attack is named as digital-photo attack [22].

Some researchers explore different sub-categories of the paper attacks[23] which may cut out the eye-position or mouth-position at the paper. Some face movements, such as eye blinking, will still be detected at these sub-categories of the paper attacks.

Video attacks are also named as replay attacks. Attackers use videos in this attack type instead of static face image. This attack category is difficult to detect by using the facial movements such as eye blinking. Different screens from different devices, such as mobile phone, tablet or laptop, are used to show the video in front of the camera. Some particular texture patterns may appear at this attack type and researchers can use these characteristics for detection.

Mask attack uses a 3D mask of the genuine client's face as the PA instruments. The depth cue of facial structures, which was used as a solution of preventing presentation attacks that carried with flat surfaces, became inefficient for detecting presentation attacks. Producing a high-quality 3D face mask is still difficult and relatively expensive now. However, the developments of 3D printing technology may significantly decrease the price of producing a 3D mask.

2.3 REVIEW ON PRESENTATION ATTACK DETECTION

Following a review published in 2015 [19] the existing anti-spoofing algorithms can be categorised as follows (cf. Figure. 2.6): (1) Hardware-based methods, (2) Software-based methods, (3) Score-level approaches.

There is another survey published in 2018 [20] which provided a detailed classification of hardware-based and software-based methods in Figure 2.7. In this work, the hardware-based methods are defined as methods using specially designed hardware; and they considered three sub-categories of hardware-based methods. The software category is slightly complicate which includes static feature-based approaches and dynamic feature-based approaches.

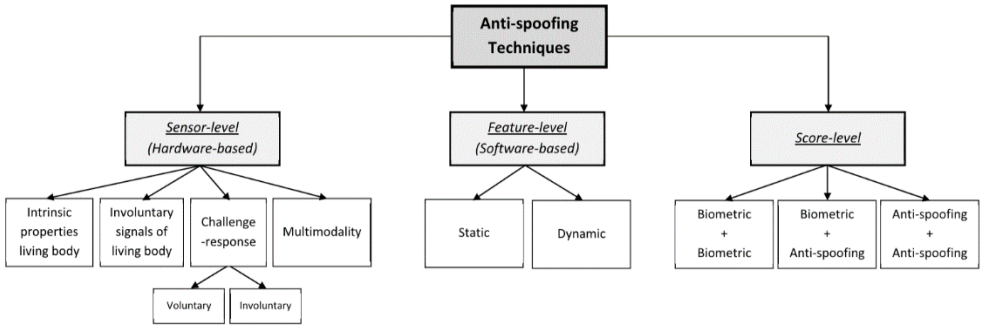


Figure 2.6 Classification of face PAD algorithms. [19]

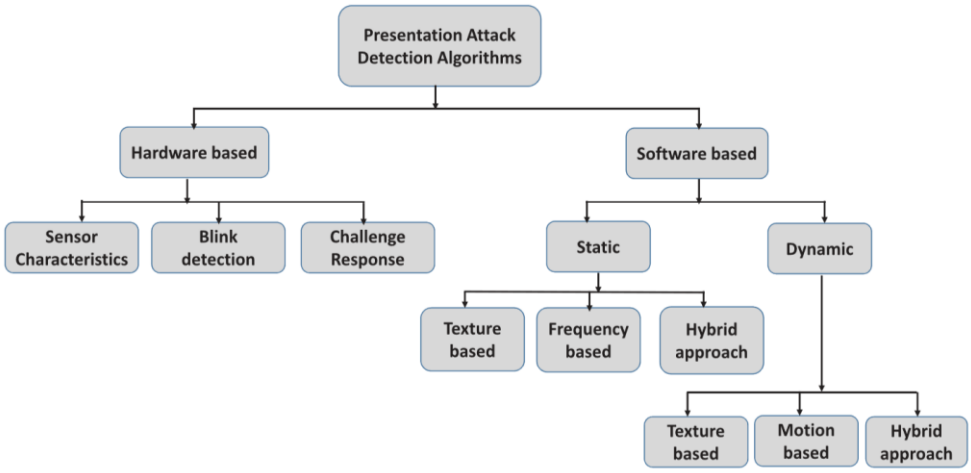


Figure 2.7 Classification of face presentation attack detection algorithms [20]

The advantage of hardware-based approaches includes their lower-computational complexity and higher robustness for different attack types. However, specifically designed hardware implies higher costs. For instance, Apple released iPhone X, which includes a 3D structured light sensor for face recognition. They claimed that iPhoneX is immune to the video attack and paper attack. However, the price of iPhoneX is $999 where the iPhone 8, which only includes an RGB camera, with $699 at 2017. The price difference may be largely due to hardware costs and the necessity of developing robust feature-based approaches for PAD. The proposed works mainly focused on the software-based approaches.

2.3.1 Software-based approaches for PAD

Software-based approaches, which are developed to detect PA, demonstrate their effectiveness with high detection accuracy and low costs. Moreover, these schemes do not necessarily need the cooperation from users and also exclude the need for

specialized hardware. Face images captured from printed photos may visually look very similar to the images captured from live faces. Therefore, a suitable feature space is needed for separating the two classes (live vs. fake face images). The main issue is how to derive such a feature space. The existing methods in this family can be further divided into two main types: (1) static methods and (2) dynamic methods.

2.3.2 Static feature-based approaches for PAD

The static feature based approaches, which are also named as static approaches in the following descriptions, are designed for a single facial image, and, sometimes, are applied to each frame of the video input independently to increase the performance [19], [20]. Generally, the advantages of static approaches could be listed as: good performance, low computation, low cost, etc. The review from Ramachandra et al [20] suggested to further divide the static feature based approaches into three main groups, (1) texture-based approaches, (2) frequency-based approaches, and (3) other approaches. The following descriptions follow their suggestions to demonstrate the traditional feature based methods for the static input.

Texture-based methods for PAD

Texture-based approaches generally analyse the texture differences between human faces and printed faces. These approaches are based on analysing textural patterns in the face image sample. This kind of approaches, which can efficiently discriminate between artefact characteristics such as the presence of pigments (due to printing defects), specular reflection, and shades (due to a display attack), give good results for detecting photo and display artefacts.

The basic observations and assumptions about texture-based approaches can be summarised as modelling small texture differences between real faces and attack artefacts. Also, the difference between these 3D structures may cause different specular reflections and shades. When using 2D cameras to capture faces, the different representation of reflections and shades are represented as different textures. The surface properties of real faces and prints, e.g. pigments, are also represented as different textures. In addition, face prints usually contain printing quality defects that may be detected from texture.

The micro-texture-based methods, which may be inspired by the observations about image quality assessment, can be summarized as emphasizing the texture differences between attacks and genuine faces in the feature space.

Figure 2.8 Examples of two images (a live face and a face print) in the original space and the corresponding LBP images using LBP as a feature space [5]

Figure 2.9 Examples of two images (a live face and a 3D mask) in the original space and the corresponding LBP images using basic LBP as a feature space. [20]

The Local Binary Pattern (LBP) [24] is a texture feature which is robust and computationally efficient, and is widely used for PAD. It is a powerful means of texture description and among its properties in real-world applications are its discriminative power, computational simplicity and tolerance against monotonic gray-scale changes[25]. The LBP method was first explored in Maatta et al. [26] for photo print attacks and was then extended successfully to address replay video attacks [27] on face recognition systems. Local primitives, which are codified by each LBP binary

code, include different types of local texture patterns such as curved edges, spots, flat areas etc. Normally, the occurrences of the LBP codes in the image are collected into a histogram, and the similarity of this histogram can be used for classification.

From the results in [27] , facial images are divided into several local regions from which LBP histograms are extracted and concatenated into an enhanced feature histogram. Also, Chingovska, I et al. [27] demonstrated the importance of using multi-scale LBP, overlapping blocks, and combination of local and holistic descriptions. From their work, the importance of modelling printing quality defects using micro-texture patterns is emphasized.

Ramachandra et al [20] showed some visualisation results for the genuine face and the mask attack.(Figure 2.9) From their results, the 3D mask exhibits different local texture representations as compared to real skin. And, the LBP features are quite successful in capturing these differences. It may be the results behind the success of using LBP for PAD.

Table 2.2 Texture descriptors for PAD

References	Texture descriptors	Attacks
Maatta et al. [26]	LBP, LPQ, Gabor	Photo attack
Chingovska et al. [27]	LBP, tLBP, dLBP, and mLBP	Video Attack
Nesli and Marce [28]	LBP	3D Mask Attack
Kose and Dugelay[29]	LBPV	Photo attack
Raghavendra et al [30]	BRSIF, CSLBP, Constrast LBP	Photo attack
Waris et al[31]	GLCM	Video Attack
Boulkenafet Z, et al. [32]	Colour LBP	Photo and Video Attack

The table 2.2 includes some static approaches using different local texture descriptors. Maatta et al. [26] use LBP for PAD and concatenate the histograms from different LBP variants (namely LBPu2 8,1, LBPu2 8,2, and LBPu2 16,2) to get a single feature vector for classification. This work inspired a lot of local texture-based methods for PAD in the following years. Table 2.2 provide a brief review for local texture-based methods for PAD which select some works that aims to introduce different local texture descriptors and get some good results. Selecting a good texture descriptor plays a substantial role in this category and combining multiple good texture descriptors will further improve the reliability of these methods.

Frequency-based methods for PAD

Texture-based anti-spoofing methods are based on the analysis of the properties such as skin reflectance. Frequency-based algorithms can also be used for this purpose. For instance, Li et al.[33] designed a PAD method, which uses 2D Fourier spectra and show some encouraging results in their private dataset. Their method assumes that PAI may contain fewer high frequency components than the real faces. Figure 2.10 shows the examples derived for a client image (top row) and an imposter image (bottom row). After their work, researchers explored different algorithms to quantify the frequency domain(Discrete Cosine Transforms (DCTs) [34], Difference of Gaussian (DoG) filters [35], and high-frequency components[36]). However, these works are tested at private datasets, which cannot be compared with other methods.

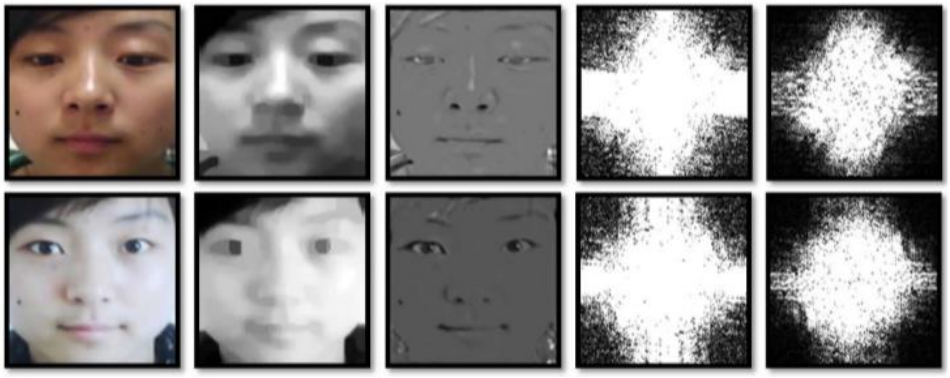


Figure 2.10 Illustration of latent samples for frequency based methods [33].

Tan, et al. [37] tested their frequency-based method at NUAA database, which is a benchmark dataset and includes paper attacks for 15 subjects. In their work, the Lambertian reflectance is used to discriminate the differences between printed faces and real faces. The variational retinex-based method and difference-of-Gaussians (DoG) are used in their method to extract latent reflectance features for PAD.

Other methods for PAD

Also, there are some methods trying to combine texture based static approaches and frequency analysis-based approaches. For instance, Wen et al.[38] provided a feature, which is named as image distortion analysis as shown in Figure 2.11, to

combine four different characteristics for PAD (specular reflection, blurriness, chromatic moment,

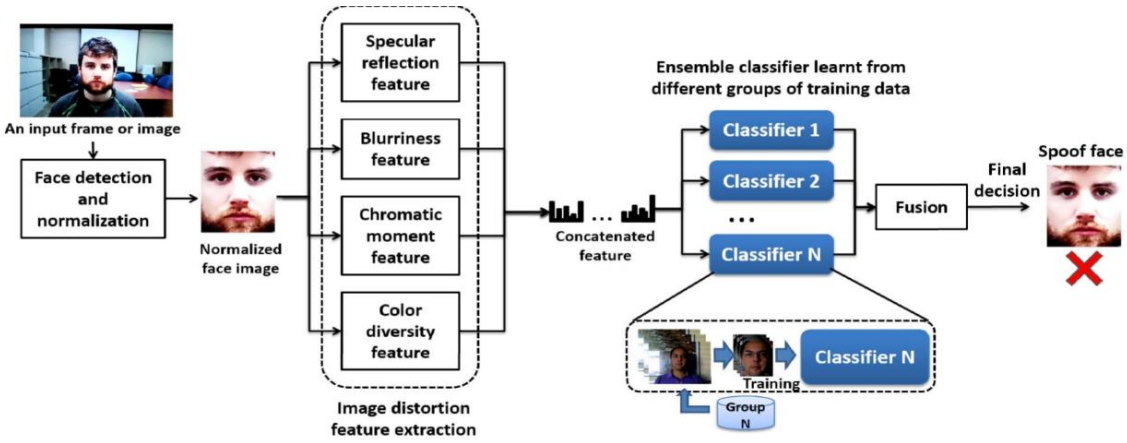


Figure 2.11 Face spoof detection algorithm based on Image Distortion Analysis [38].

and colour diversity). They summarized some disadvantages of the texture-based methods and frequency analysis-based methods: For existing texture-based methods, the facial details, which can differentiate one subject from the other (for the purpose of face recognition), may also be captured by using the local texture descriptors(such as LBP). As a result, genuine faces may be wrongly classified as presentation attacks due to the redundant information from facial details. Some researchers claimed that the features from local texture descriptors may be too person specific[38]. Meanwhile, existing frequency analysis-based methods are highly relying on the selection of camera, photo and screen display. These disadvantages stimulate the development of the dynamic features for PAD.

2.3.3 Dynamic Features for PAD

Dynamic approaches use temporal information, which may also be discriminative for PAD, and usually take more computational effort for the sequence of biometric samples. Existing dynamic approaches can be broadly classified into two types: (1) motion based-approaches, (2) texture-based approaches [19]

Motion-based methods for PAD

The motion-based methods consider the muscles movements in the face or the head movements as the discriminative feature for detecting PAs. The existing methods are effective for various print attacks by exploring the facial movements over video

sequences. In general, they are based on the trajectory analysis of specific face segments. Some researchers[39] involved challenge-response strategies which ask users' cooperation to demonstrate specified facial movements (such as eye blinking, smiling and moving the head/eyes in some directions) in front of the camera. Some researchers claimed that the methods using challenge-response strategy shows a performance decrease when facing the replay attacks[39]. Also, the challenge-response strategies request cooperation from users, which is also considered as a disadvantage for the early motion-based methods.

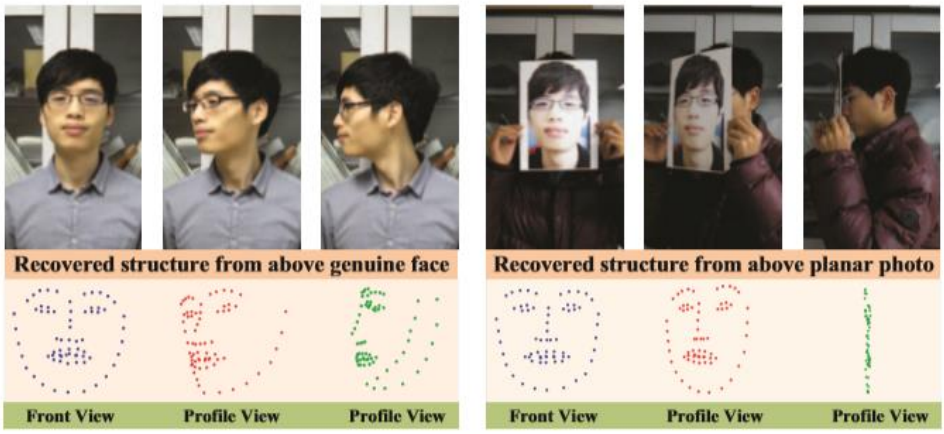


Figure 2.12 A comparison of recovered sparse 3D facial structures between genuine and photo face. There are significant differences between structures recovered from genuine and photo face [40].

Some dynamic feature based methods [40], [41] are designed to detect video-attack without cooperation from users. 3D facial structures have been to reconstruct through the analysis of several 2D image sequences to detect PAs. As Figure 2.12, Wang, et al. [40] built a sparse 3D facial structure, which is generated by using facial landmarks from key frames, to demonstrate the difference between real faces and presentation attacks. They used SVM as the classifier and showd some encouraging results at benchmark datasets. The constant evolution of mobile technology and the smartphone market has brought new possibilities for the 3D facial structure-based PAD methods[41].

With the development of 3D model generation algorithms, creating realistic, textured, 3D facial models that can undermine the security of widely used face authentication solutions has become possible [42]. Accurate 3D facial models can be created by using some publicly accessible photos, which may be easily collected from,

for example, Facebook or Twitter. And by using these high-quality 3D facial models, attackers can easily subvert commercial facial biometric system; even if these apply a challenge-response strategy. These new attack methods raise new requirements for facial PAD. (Figure 2.13)

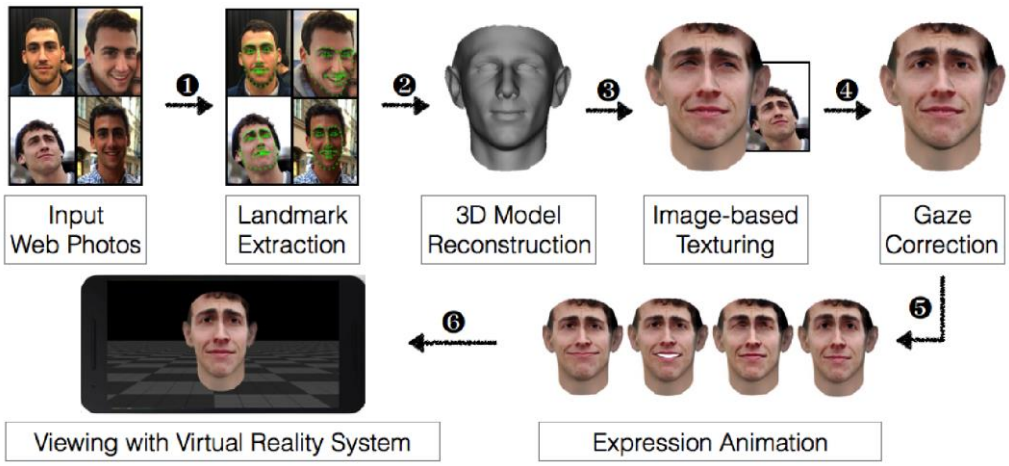


Figure 2.13 Overview of a 3D structure reconstruction based attack method [42].

Dynamic Texture for PAD

Another category for the dynamic features explores the dynamic texture changes across the captured video. Early work in this direction is based on Local Binary Patterns from three orthogonal planes (LBP-TOP) [43] and has demonstrated a reasonable performance on the Replay-Attack database. As illustrated in Figure 2.14, both the spatial domain and the time domain for the entire video sequence are explored in "local texture descriptor-TOP" style. Local texture descriptors from three orthogonal planes are extracted as detection features.

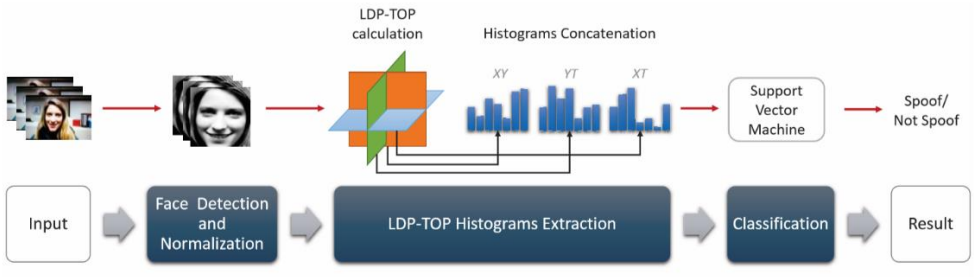


Figure 2.14 Overview of LDP-TOP workflow [44]

Other dynamic texture-based methods have been proposed to improve the detection performance, such as binarized statistical image features on three orthogonal

planes (BSIF-TOP), local phase quantization on three orthogonal planes [45](MLPQ-TOP) and local derivative pattern from three orthogonal planes (LDP-TOP)[44]. These methods use a similar protocol for temporal information. In the first step (face detection and normalization), each video frame is transformed to a grey-scale image and passed through a face detector. The detected faces are then geometrically normalized. In the second step (histograms extraction) local texture operators are applied on three orthogonal planes intersecting at the centre of the XY, XT, and YT direction, where T is the time axis (the frame sequence). The feature vectors from different orthogonal planes are concatenated and fed into SVM for PAD. This spatial-temporal analysis protocol can explore the information of the whole video sequence and show some encouraging results in their experiments.

2.3.4 Deep learning for PAD

Deep Learning offers new research opportunities for the PAD area including new feature extractors based on deep learning and a new learning paradigm. The basic learning paradigm of using deep learning for PAD is designing a novel neural architecture for PAD and training this novel neural architecture with PAD datasets. This learning paradigm is named as “learning from scratch” or “training from the scratch” in this thesis. Researchers have explored several DNN methods for PAD such as convolutional neural network (CNN), recurrent neural networks (RNN), generative adversarial networks (GAN), etc. An alternative approach, based on the transfer learning paradigm (or the network-based transfer learning), uses the feature extraction part of a trained DNN which is designed for other computer vision tasks.

“Hand-crafted” Neural Architectures

Designing a network from scratch is an active area in recent years. Since the success of AlexNet [46], the performance of deep neural networks has been significantly improved (e.g. VGGNet [47] , GoogleNet [48] and ResNet [49]). Some researchers focused on optimising the convolutional operator and developed various convolution operators such as transposed convolution[50], dilated convolution [51], etc. Others focused on developing better activation functions to improve the performance of the neural architectures, (e.g. Rectified Linear Unit (ReLU) [52] and Exponential Linear Unit (ELU) [53]).

Since the success of the VGG16 network[47], many researchers aim to optimize the structure of these successful networks to reach the same performance level but with a smaller size. Before the development of Neural Architecture Search (NAS), some commonly-used approaches included quantizing the weights and/or activations of a baseline CNN model into lower-bit representations[54] or pruning less important filters [55] during or after training. These methods are focused on reducing the computational effort. However, they are tied to a baseline model which they tried to optimize and do not aim to learn novel compositions of CNN operations.

Designing efficient operations and neural cells (or neural blocks in some literatures) by human experts is a popular research direction for deep learning. The task of designing efficiently neural architectures by human experts aims to use deep neural networks on mobile platforms such as the iPhone. This direction has led to some efficient designs: SqueezeNet [56] provides a low-cost convolutional operator that can reduce the number of parameters and computational costs; MobileNet [57] extensively employs depth-wise separable convolutions to minimize computation density; ShuffleNet [58] use pointwise group convolutions and channel shuffle method to decrease the computational cost; MobileNetV2[59] shows the performance with state-of-the-art performance level but only uses mobile-size models by introducing resource-efficient inverted residuals connections and linear bottlenecks into their work. Unfortunately, these hand-crafted models usually take quite significant human efforts to design.

Alotaibi et al.[60] used a non-linear diffusion operator in their pre-processing step and processed images by applying a custom six layers CNN. They tested their proposed method using the Replay-Attack database and provided some results (Half-Total-Error-Rate (HTER)=10%) to demonstrate the potential of using deep learning methods.

Some researchers[61] assumed that using the whole facial area may mislead deep learning models by focusing on the facial structures rather than the texture features which may be more distinct for anti-spoofing. Patch-based methods were used based on this assumption. Y. Atoum et al [61] proposed a patch-based deep convolutional architecture, which models different distinct visual patterns from each facial region to detect spoofing(As Figure 2.15). In order to train this network, they re-organised the training data by using data augmentation methods and a cropping function to generate

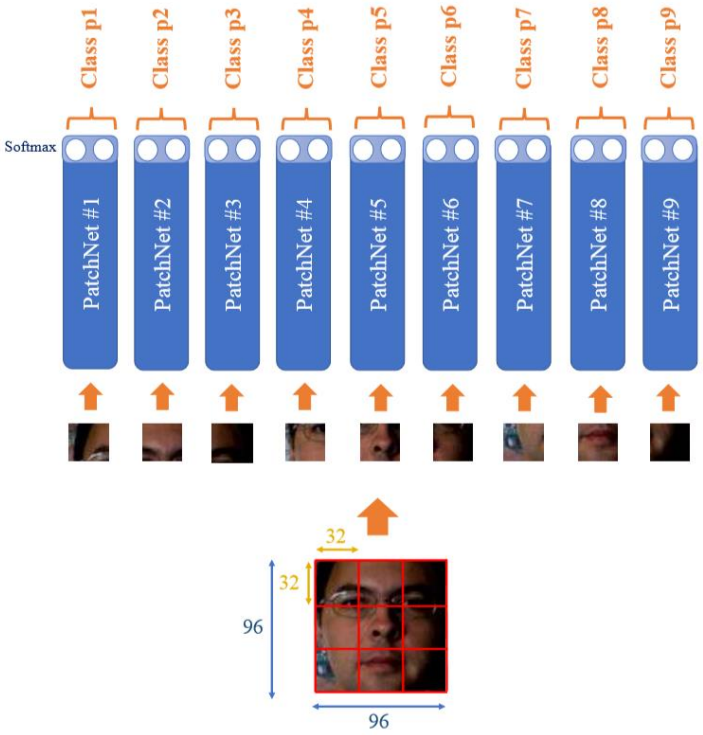


Figure 2.15 Overview of patch-based CNN workflow[61].

facial patches. They assumed that the texture patterns of spoofing attack may appear over the whole facial area and the patch-based method can significantly reduce the computational complexity.

From the analysis in [61], the backpropagation algorithm is easily misled by the texture representation of the facial elements, such as the shape of the eyes or the size of the nose. Their work [61] claimed that the small DNN does not need as much training data as that needed for very deep neural architectures. However, the small network still needs the training data with various conditions and this may still be very hard to achieve with the currently available training datasets. And overfitting is still a problem for their neural networks. Also, the patch-based method may only help the deep neural networks focusing on the globally appearing texture patterns, which means the non-related texture patterns will be ignored, such as the strange eye shape in the paper cut attacks.

Deep learning also extends the possible research directions for PAD. For instance, Liu et al.[62] push the performance boundary for PAD by integrating remote photo-plethysmography and 3D facial model extraction. Their methods are shown in

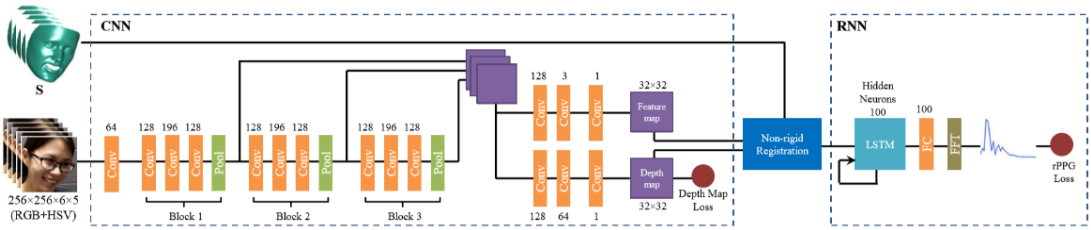


Figure 2.16 Overview of CNN-RNN architecture for rPPG signal. The number of filters are shown on top of each layer, the size of all filters is 3 × 3 with stride 1 for convolutional and 2 for pooling layers. Colour code used: orange=convolution, green=pooling, purple=response map. [62]

Figure 2.16. Remote photo-plethysmography (rPPG) means that the vital signals (such as heart rate) are tracked by using the features extracted from RGB data. This process does not need any physical contact with human skins. Liu et al [62] noticed that: the rPPG signal is detectable by using a deep learning method for PAD. In order to further decrease the effect of facial movements and illumination changes, Liu et al. firstly extracted the dense 3D facial masks for each frame by using the DeFA algorithm [62]. The convolutional blocks proposed by their work consist of 3 convolutional layers, one pooling and one resizing layer. They integrated one exponential linear layer and batch normalization layer after each convolutional block to decrease the risk of overfitting. They also considered the bypass connections structure which is similar with the ResNet [49] structure to help the network to fuse the inner representation and generate robust features. After that, they used non-rigid registration to further decrease the effects of facial movements and different head gestures to get the rPPG signal by using the recurrent neural network. In other words, their model can only learn the selected temporal signals from the activations of the feature maps but ignore the micro facial movements and head gesture difference which may also be distinct in temporal for PAD.

Transfer learning for PAD

Normally, PAD is approached as a typical supervised learning task which assumes that the training and the testing data are in the same feature space or distribution. However, transfer learning does not need learning algorithm to train the model from scratch, even the targeted task is not in the same feature space or distribution.

Some researchers [63] noticed that the training data from similar domains can be used for training PAD systems. For this reason, the transfer learning approach has become popular in deep learning-based PAD research where the selection of a good pre-trained feature extraction network is a key factor.

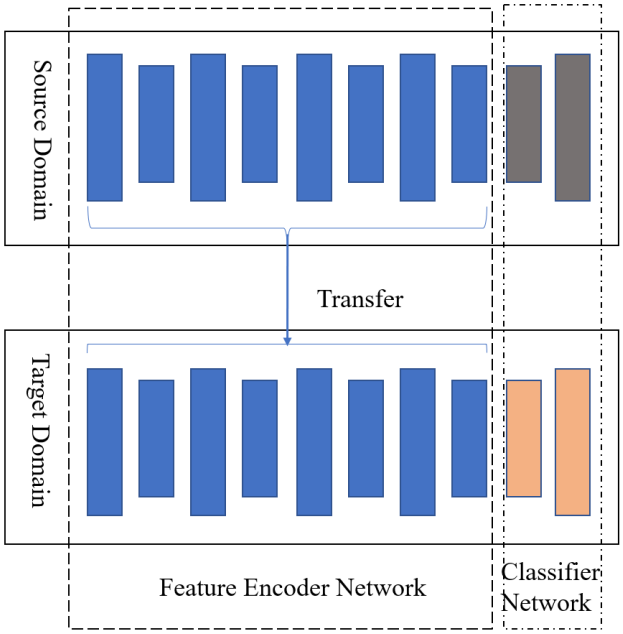


Figure 2.17 Network-based deep transfer learning

Fig. 2.17 demonstrates the basic idea of network-based deep transfer learning. The network can be divided into two parts; the front part is the feature encoder sub-network and the back layers form the classifier. The front-layers which are trained using large datasets such as ImageNet[64] are normally reused to compute intermediate image representations for images in other datasets. The features from CNN can be efficiently transferred for other visual recognition tasks with limited amount of training data.

The transfer learning paradigm normally includes two training stages: Firstly, the parameters from the pre-trained encoder network is trained with a small learning rate and the classifier network is trained with a normal learning rate. Then, the whole network is fine-tuned with a small learning rate during the second training stage.

The learning rate is one of the most important hyper-parameters for deep learning. Bengio, Y.[65] discussed the reasonable ranges for learning rates in their work. The proposed network-based deep transfer learning experiments follow the

suggestions of [65] [66] and set different learning rates during the first training stage for the encoder network and the classifier network. The pre-trained feature encoder network includes some latent knowledge which is learned from the source domain. A lower learning rate can help the network keep the latent knowledge from the source domain. The higher learning rate for the classifier network aims to optimise the randomly initialised parameters. At the second stage, the whole network is fine-tuned with a lower learning rate for a better performance. It has been suggested that this fine-tuning stage can help the neural network escape from local optima [63].

Yang et al. [67] first used CNN for feature extraction in the PAD workflow. They used AlexNet [46] for feature extraction and SVM for classification. They deployed various methods for pre-processing images to vary the bounding boxes sizes, and image qualities used for the CNN. They reported HTER=2.81% for the REPLAY-ATTACK database. However, they only considered the transfer learning for the feature extraction sub-network. The fine-tuning stage for the whole network was excluded in their work.

There are some two-stage transfer learning that has also been used for PAD[68] . They explored a VGGFace network [69]as the feature extraction part in their work which was originally designed for large-scale face recognition applications and trained on a dataset consisting of 2.6 million images from 2622 different individuals. The VGGFace Network is a benchmark CNN architecture and they obtained some good results for some challenging datasets. [68]

Long Short-Term Memory networks (LSTMs)[69], [70] are a development of Recurrent Neural Network (RNN)[71] which is designed to model temporal information and other sequence learning tasks (such as sequence generation, speech recognition and video description). A typical method of using LSTM for video data is to consider a CNN architecture as a feature encoder and connect the CNN to the LSTM layers.

Xu et al. [72] was first to introduce LSTMs for PAD and followed the commonly used method to combine CNNs and LSTMs. They claimed that the CNNs can learn the local texture patterns and the temporal relations between these patterns can be learned by the hidden state of the LSTM unit. However, the LSTMs in Xu et al.'s work[72] discards the spatial locations of the features generated by the CNNs with a

fully connected layer. The spatial location information is important for PAD especially when the model needs to capture the motion cues and the distinct dynamic textures.

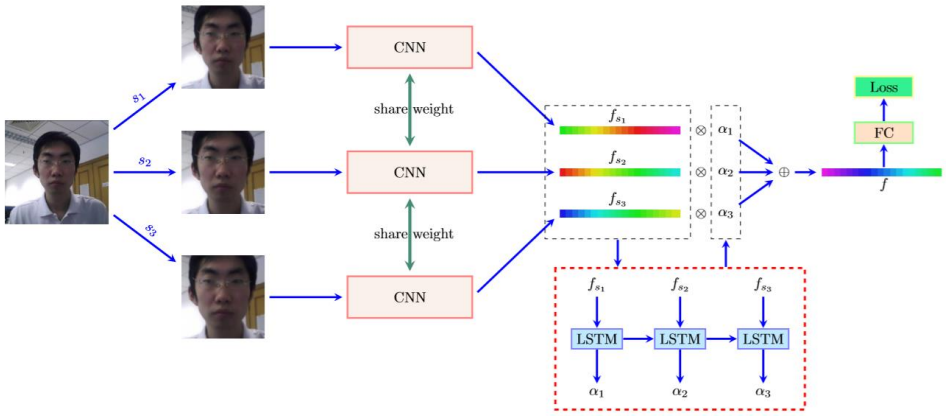


Figure 2.18 Overview of multiscale CNN workflow.[73]

Tu, et al. [74] attempted to overcome these disadvantages by adding a confusion loss layer based on the loss function in LSTM and the loss function in CNN to balance the learning rate of CNN and LSTM. However, their model also has the computational efficiency problem when using LSTM for temporal information. The inner structure of LSTM may need to be optimised for the PAD problem.

Some ideas from research into traditional hand-crafted features may also be used in the development of deep learning approaches for PAD. For instance, the scale difference between genuine and fake faces can be used to increase the performance of PAD models(as Figure 2.18) [73]. Information fusion is an important aspect of multi-scale PAD methods where the LSTM model can be used to provide efficient fusion of multi-scale models. Luo et al. [73] followed this idea and considered the features from different scales as a sequence of input for the LSTM. They assumed that the LSTM can model the inner connections between different scales. Their model can be considered as imitating of "take a closer look for better understanding". The good performance they report may demonstrate that LSTM can adaptively fuse the features from multiple scales. However, single directional LSTM may highly rely on the sequence orders. The bi-directional LSTM may offer better performance as they suggested in their future work.

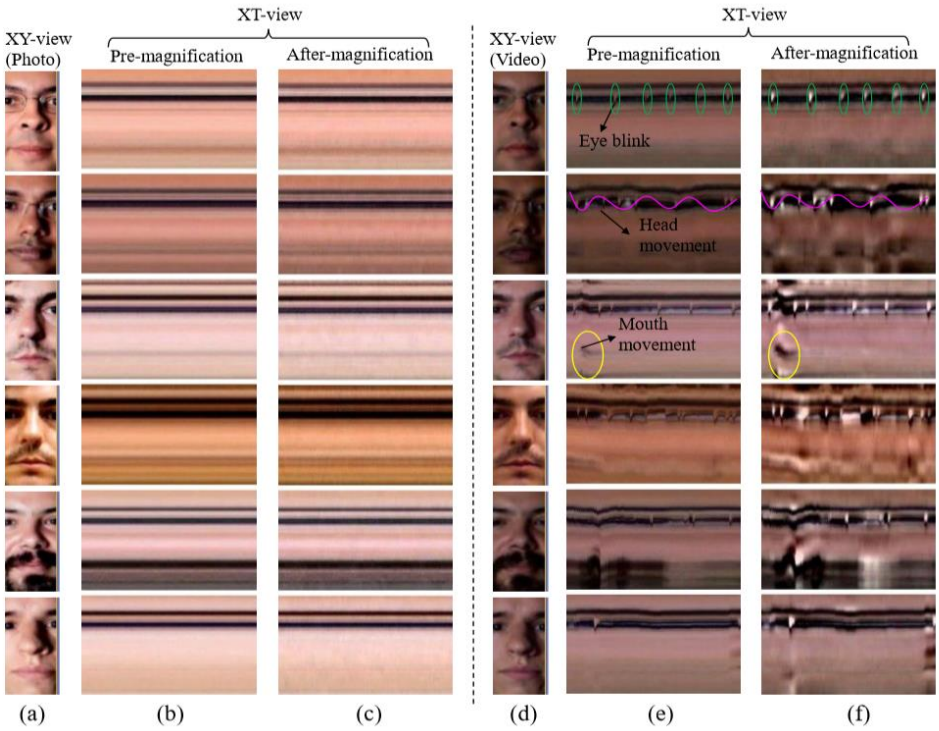


Figure 2.19 Visualisation of Asim et al.'s work using CNN and LSTM (a) shows the video of fixed photos in XY view, (b) and (c) are the corresponding images in XT view without and with magnification, respectively. (d) is the video with dynamic facial expressions in XY view, (e) and (f) represent the corresponding images in XT view without and with magnification, respectively. [75]

Some researchers have attempted to combine traditional features and deep neural networks together. Asim et al.[75] aimed to combine the LBP-TOP and CNN together for better performance. Firstly, a temporal ontological plan was generated from each video for further processing by following the ontological plan generation part of the LBP-TOP algorithm. They then extracted the deep features for PAD by using a typical convolutional neural network (CNNs) for the input data. Then, their method collected the intermediate feature map calculated by the CNNs and used the LBP algorithm to calculate the histograms for the feature maps generated by the 3rd, 4th, 5th convolutional blocks. The generated histograms were concatenated together which is very similar to the LBP-TOP. They also applied a magnification approach, which can enhance facial movements such as eye blinks, to help the liveness signals in the temporal domain to be significant. Their methods were reported to slightly improve on the performance of the traditional features. However, they only used the deep neural network as a new feature encoder and ignored the new learning paradigm offered by these deep architectures. The popularity of the DNNs relies on an assumption that the

intermediate feature before the last classifier layer learns a good feature which can help the model to classify the spoofing attacks. In other words, the DNN learns what are good features directly from the training data. This property may offer a possibility, which can improve the performance without human experts, for future works. The deep learning approach may best focus on this direction rather than simply provide other features for fusion with traditional features.

Neural Architecture Search

Neural Architecture Search is considered as the next step for automating machine learning when researchers and engineers have struggled with the complexity of designing an effective neural network. It has increasingly attracted researchers' attention by outperforming the human-designed neural architectures on some computer vision tasks such as object classification[76].

NASNet[77] started the wave of automated neural architecture search using reinforcement learning in 2016. More recently, with the successful project named AutoML[78], designing neural network automatically instead of relying heavily on human experts attract researchers' attention. Importantly, NASNet has successfully identified architectures that reach performance levels comparable to state-of-the-art human designed architectures for large-scale image classification problems. Considering NAS as a reinforcement learning (RL) problem, the generation of a neural architecture can be treated as the action of intelligent agents, with the action space identical to the search space[77]. The reward function of the RL paradigm is based on the estimation of the performance of designed architectures on unseen data.

As suggested by Elsken, T. et.al.[79], NAS research can be divided into two categories: macro search and micro search. The macro search category aims to find an algorithm to generate the entire neural architecture directly where the micro search strategy arrives at the overall network by stacking together optimum micro neural architecture blocks (also known as cells).

The methods which optimize the entire neural network directly can produce neural architectures automatically but the output of these methods is generally "shallower" than the other DNNs[80]. One of the typical approaches is applying RL algorithm to optimize the policy of searching neural architectures. Some widely-used RL methods (such as Q-learning[81]) is used to train the neural networks, which can

select the connections and configurations of convolutional layers sequentially. These algorithms will build the entire network from the first layer and then generate the next layer until the end of the network. The LSTM is used for selecting the filter shape and number of filters and optimized by using the RL method [82]. Evolutionary algorithms are also considered for NAS [14] where they are used to guide the mutation and recombination of candidate architectures to arrive at optimum architectures. The computational complexity of exploring the search space to generate candidate networks directly is one of the main disadvantages of these approaches. Some researchers have analysed the magnitude of this problem by using a simple approximate calculation method: the volume of the potential search space can be approximated by using the exponent of the total depth in the proposed network. For instance, searching a shallow network with only 12 layers has a search space with 10^{29} possible networks[81].

Researchers, therefore, limit the depth of the proposed neural architecture to constrain the search problem within the limits of feasibility. Another important disadvantage of the Macro search strategy is accuracy limitations due to the necessity to use shallow networks. In contrast, the micro search strategy can achieve better performance with less computational resources by stacking multiple neural cells.

Zoph. et al.[83] proposed a small search space (NASNet search space) for the reusable micro neural architecture. By following this idea, an evolution algorithm was applied by Real et al [84] to search for the optimum architecture for cells with a simple regularization technique. A progressive method was also used to search the micro neural architecture for cells[85]. Wu, et al. [86] tried to improve the efficiency in architecture search by generating the architecture from shallow to complex architecture. This method can reduce the computational effort and time required.

The “differentiable” method[87], which relaxes the discrete architecture space to a continuous one by utilizing gradient-based optimization, is computationally efficient. However, these methods proposed for micro search still take more than one GPU days [87] for searching and the memory cost of the searched neural architecture is not considered as a constraint condition.

Explainable Artificial Intelligence and Generative methods

Explainable Artificial Intelligence (XAI) has attracted significant attention in recent years as a new branch of Machine Learning (ML) research, which aims to improve the transparency of current ML algorithms and decrease the opacity of each decision made by a ML system, [88]. Transparency is especially necessary for PAD due to the need for biometric decisions to be trusted and effectively managed. The explanations for the decisions that are made by a PAD system can justify any unexpected decisions and build trust with users. Also, these explanations can guide the development of a PAD system to improve its performance [89]. Therefore, further research is needed to address the interpretability of the behaviour of current automated biometric systems.

The explanations produced by the interpretable PAD systems can help with enhancing trust, improving performance and helping to detect new patterns of security threats. Also, future biometric systems may be required to provide explanations in order to abide by the law[89]. Interpretable biometric systems have various potential users. For instance, In the event of erroneous decisions, system-generated explanations will help the operators to identify where the responsibility may lie (similar to flight black-box recorders used for investigations). In some applications, such explanations can avoid mistakes by helping human experts rapidly identify and rectify errors to lower the risk of wrong decisions. Finally, the explanations for the wrong decisions from the current biometric systems can inform researchers to design better systems

Different researchers have different understandings of what is meant by explainable artificial intelligence. Visualization of the filters in a CNN are the most direct way to explore patterns hidden within the neural units. The Up-convolutional network [90] was developed to reverse the feature map into an image. In contrast, gradient-based visualization[91] provides a means for understanding the knowledge hidden within the parameters of a CNN. In addition, Ribeiro, et al.[92] defined and analysed the interpretability of each filter.

Modelling the inner connections between the filters in CNNs by using semantic trees or graph models is another possible direction for interpretable capability. Many statistical methods [90] have been proposed to analyse the semantic relation between CNNs' features. In particular, Chattopadhay, et al. [93] have demonstrated that in spite

of their good classification performance, CNNs may encode biased knowledge representations due to dataset bias. However, currently there is no commonly used evaluation methodology to quantitatively measure the effectiveness and accuracy of the explanations produced by such systems.

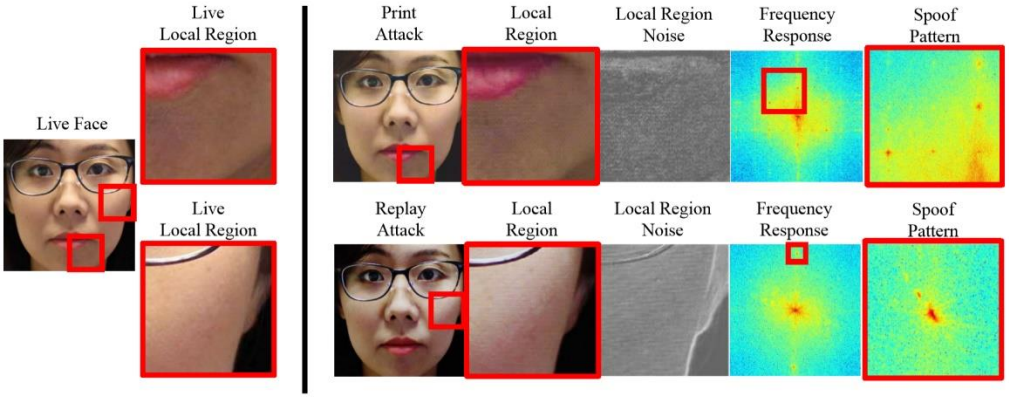


Figure 2.20 Visualisation of spoof signal in De-spoofing paradigm. Left: live face and its local regions. Right: Two registered spoofing faces from print attack and replay attack. The local region, intensity difference, magnitude of 2D FFT, and the local peaks in the frequency domain that indicates the spoof noise pattern is shown. [94]

The latest developments of Generative Adversarial Network (GAN) can also be used for detecting PA and the GAN network can be used to visualise the distribution of spoofing patterns. Amin et al [94] defined a de-spoofing problem by considering the distinct features for the spoofing attacks as the particular texture representation which can be learned by using a deep generative model: The deep generative model is an active sub-area of deep learning. Amin et al.'s work was inspired by the classic image de-X problem (for instance, image denoising and de-blurring) and aimed to learn a mapping function which can transfer the genuine faces into the spoofing face by adding the learned texture representations. The deep generative model is increasingly applied in the de-X areas due to its significant learning capability for the local texture patterns and the impressive generative capability.

Amin et al [94] assumed that deep generative models can transfer a frame with the genuine face into a frame from spoofing attack by using the generative function learned from the training data. They assumed the decomposed noising pattern, which is learned by their model, is the key feature for detecting the PAs. Their processing pipeline is very similar to image de-noising and they named their work as Face De-

spoofing. One of the main contributions is that they visualise the spoofing patterned learned from their model. However, their visualisation experiment can hardly be used as the additional information for training. They argued that the frequency response of their model and the spoofing pattern generated by their model represent the distinct texture patterns of various spoofing attacks. But their visualisation cannot be directly used to optimise their classification model. Furthermore, their network structure is very complicated consisting of a Discriminative Quality Net (DQ Net) and a Visual Quality Net (VQ Net). Training this complex deep architecture directly is not easy due to the gradient vanishing and local optima problems, which are considered as some common disadvantages of such deep generative models.

2.4 SUMMARY

This chapter introduced the basic concepts of Biometrics, Presentation Attacks and PAD. It then gave an overview of various types of presentation attacks including paper attack, video attack and mask attack. After that, a classification of various anti-spoofing techniques was presented according to the biometric system modules in which they are integrated. As this thesis is focused on software-based methods, key previous research on static and dynamic methods were briefly described covering texture-based, frequency-based, motion- based and other approaches. Finally, the rise of deep learning methods as applied to PAD is reviewed and two potential areas for further development, including NAS and explainable Artificial Intelligence (XAI), are highlighted. The datasets and experiments protocols used in the literature are described in Chapter 3 as part of the experimental framework that underpins the thesis.

Chapter 3: Experimental Framework

This chapter outlines the experimental framework that is used in the subsequent chapters such as experiment workflow design, pre-processing algorithms, and related datasets used for the evaluation of facial anti-spoofing techniques. The structure of this chapter is shown as follows: Section 3.1 will discuss the experimental infrastructure used for facial anti-spoofing (or presentation attack detection) algorithms in the following chapters. Section 3.2 will present the usual pre-processing steps (including face detection, face alignment, and facial area segmentation), in which the pre-processing, as an essential part of a PAD system, can help the PAD algorithm to ignore the irrelevant information in the input data. Moreover, the quality of the pre-processing steps, can profoundly affect the final performance of a PAD system. Feature encoding and classification methods are briefly described in Section 3.3. Then, the datasets and evaluation metrics, which are widely concerned in the state-of-the-art approaches, are investigated in Section 3.4. Some of these datasets and evaluation metrics are used in the following contribution chapters for performance comparison. Section 3.5 will conclude this chapter.

3.1 FACIAL ANTI-SPOOFING DETECTION WORKFLOW

The International Organization for Standardization (ISO) provided a standard pipeline for presentation attack detection in2017 that is found in Figure 3.1 [2]. According to this figure, a standard PAD system consists of three important parts: (1)a *PAD Feature Extractor* (2)a *PAD Comparator* and (3) a *Stored PAD Criteria*. In this pipeline, the captured data is fed into the *PAD Feature Extractor* to get the feature representation; and the *PAD Comparator*, which follows the *Stored PAD Criteria*, generates the result by using the extracted features. According to BS ISO/IEC 30107-3:2017 [14], the *PAD comparator* and the *Stored PAD Criteria* are considered as the fundamental parts in the system. By following the definition from the BS ISO/IEC 30107-3:2017 documents, this thesis mainly focused on the *PAD Feature extractor* and the *PAD comparator part*. However, this pipeline only provides a rough description for a PAD system and excludes the recent developments of deep learning and computer vision methods.

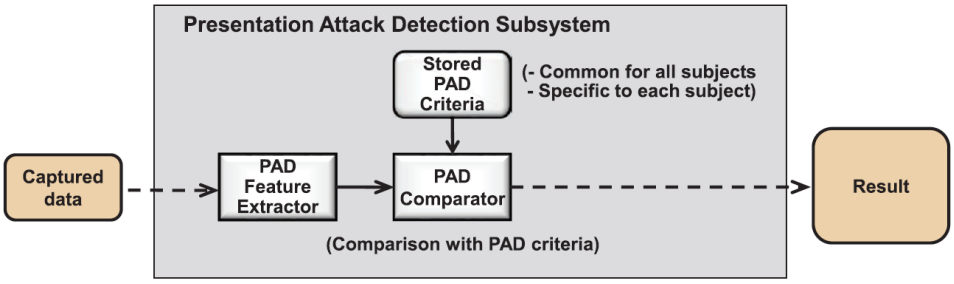


Figure 3.1 Components in a general presentation attack detection subsystem from ISO/IEC 30107 [14]

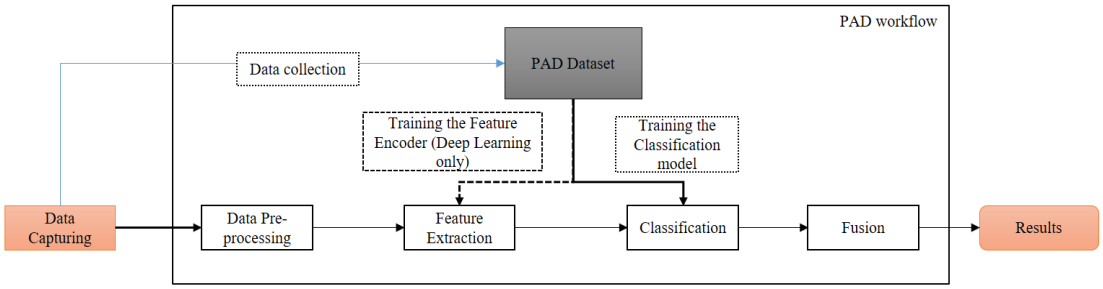


Figure 3.2 Components in the proposed presentation attack detection system. The Feature extractor has been extended to the Data pre-processing and Feature extraction. The PAD comparator has been extended to the Classification and Fusion steps.

In order to make the subsequent description clear, this thesis extends the original pipeline from BS ISO/IEC 30107-3:2017 documents[14], and visualises the proposed pipeline in Figure 3.2. The proposed pipeline of a Facial PAD system can be divided into five fundamental parts: (1) **PAD dataset** (2) **Data pre-processing** (3) **Feature extraction** (4) **Classification** (5) **Fusion**. In the following chapters, a PAD system that consists of these five fundamental parts, is expected to provide a detection result for the biometric system by accessing the captured data.

Data Capturing, which potentially uses multiple sensors and collects biometric samples, is closely related to the selection of the hardware platform and very sensitive to environmental changes. Different image qualities will highly affect the local feature representations that are used to represent the distinct differences between genuine face and attacks [32]. Low-resolution cameras that were used in some early studies can only record facial characters with lower image quality. The low image quality biometric samples influenced some facial details (e.g., skin textures) that became blurry and ineffective for PAD. For this reason, published datasets normally state detailed

information on the hardware used to acquire facial biometric information. This thesis considers various benchmark PAD datasets but only used RGB data as the raw input for the proposed methods. The input of a PAD system is the captured data and the output is the decisions generated by the PAD system in the proposed pipeline. The proposed works are all categorised as the feature-based PAD methods. Thus, the proposed pipeline will not include the possible hardware differences for the data capturing process.

The **PAD dataset** is used to replace **the stored PAD Criteria** in Figure 3.2. The proposed pipeline uses the term "PAD Dataset" for two reasons: (a) the feature-based PAD research is following the supervised learning paradigm, and the Criteria is represented by the label of the dataset; and (b) the performance of the deep learning-based PAD methods highly relies on the quality of their training data. The word "Criteria" may not express the meaning of "dataset" clearly when applying deep learning methods. This thesis considers data and the "criteria" (named as label) as two fundamental parts of a dataset, and consider the **Data Collection** as an independent step to emphasize the importance of datasets. By following this analysis, data augmentation become an important **Data pre-processing** step for the proposed deep learning-based PAD methods which can decrease the risk of overfitting and improve the performance of the proposed methods.

The *PAD Feature Extractor* is extended into two parts: **Data pre-processing** and **Feature Extraction**. The **Data pre-processing** step aims to minimise the effect of environmental changes and the possible noise signals from sensors. This step is also sometimes selected and tuned to maximise the performance score for the proposed algorithms in some literature[32]. A typical pre-processing step may include multiple data pre-processing methods (such as face normalisation, colour space transfer, etc.), and selecting different pre-processing methods follows some basic considerations: (1) the requirements of the feature extraction algorithms, (2) the requirement of decreasing the effect of irrelevant information and (3) the requirement of decreasing the data volume.

In PAD researches, the requirements of the feature extraction algorithm are a significant reason to include the **Data pre-processing** step. Some feature extraction algorithms cannot be used directly for PAD or result in good performance without suitable pre-processing steps. For instance, the traditional Local Binary Patterns (LBP)

feature can only work for one-channel images and needs a pre-processing step to convert RGB images to grey images. Moreover, in this case, a single grey channel discards much information that may be useful for PAD. Some literature [32] suggests the experimental pipeline includes the colour transformations as an important pre-processing step; and claims that the concatenation of the feature vectors from multiple colour spaces may be sensitive to the colour difference between genuine face and spoofing attacks.

Another example of the importance of considering data pre-processing is related to background information. Some researchers [95] claimed that background regions in the frame might include some irrelevant information for PAD. This information does not include any biometric information, but includes some materials that may be similar to the attack artefacts. For instance, the low-quality scenario of the CASIA-FA dataset includes some screen flashing in the background, which may highly affect the robustness of the features as it relies on the temporal frequency difference. For this reason, some algorithms only apply feature extraction on the facial area. Thus, facial detection and facial area cropping are widely considered as an important pre-processing step in the literature. Section 3.2 will produce a detailed description of some commonly considered algorithms such as colour space transformation, image cropping, affine transformation, face detection, face normalisation, facial area cropping, and facial landmark detection. Some proposed works in the following chapters select these pre-processing methods carefully due to the considerable influence of these methods.

After the **Data pre-processing**, the **Feature Extraction** step, which transforms the raw data to a low-dimensional representation (or a feature vector), regularly consumes the most computational resources and produces some distinct characteristics of the biometric samples. Normally, this low-dimensional representation, which is shown as a feature vector, is fed into the classifier to produce the final decision. Creating a distinct feature representation for PAD is the main target for the software-based PAD research. The proposed works in the following chapters, which are motivated by some observations of the PA samples and assumptions from literature, will focus on designing efficient feature extraction methods as the main contribution of this thesis.

The proposed workflow in Figure 3.2 adds a connection between the PAD dataset and the Feature Extraction part that is different from the BS ISO/IEC 30107-3:2017 document[14]. This connection emphasizes the feature extraction part that is trained by using the PAD dataset. From the emergent rising of deep learning, researchers are aware that good features can be learned from the data automatically. Some of the paradigms that comes with deep learning are different from the traditional paradigm that is widely considered for the conventional feature-based PAD approaches. Generally, the **Classification** step is used to generate a judgment about whether a biometric system is under spoofing attack by using the feature vectors from the **Feature Extraction** step. After the feature extraction step, the dimension of the data should be greatly reduced. Some learning algorithms (e.g., kernel-SVM, decision tree, or neural networks) are applied to get a classification result in the conventional feature-based approaches. Moreover, the decision fusion step, as a commonly considered method for the system that includes multiple preliminary classifiers, may be used to further improve the performance in some literature [38]. PAD can be considered a classification task under the supervised learning paradigm; and different scenarios can be defined as different types of the classification problem. For instance, if a PAD system only includes two possible outcomes (genuine access or spoofing attack), PAD is considered to be a binary classification task. Otherwise, the PAD system may attempt to not only detect a spoofing attack, but also classify the specific attack type.

Galbally, et al. [19] suggest a **Fusion** step for the PAD system to further optimise the performance when many different features (or sub-classifiers) are applied in one biometric system. Normally, the performance of a PAD algorithm can be improved by adding a weight for different classifiers or different confidence scoring. Some of these scoring algorithms can improve system performance through training. Different features for PAD may aim for the different distinct characteristics for PAD. A fusion step can help the system avoid the possible overfitting risk. By following their suggestions, the proposed pipeline also consider a fusion step. However, fusing multiple features are a difficult problem. This thesis will not include an independent fusion section; but fusion experiments are considered in some proposed traditional features to demonstrate the effectiveness of the proposed method and the possibility of using fusion step.

The proposed pipeline only considers PAD as a typical supervised classification problem. However, some recent developments [96] and the proposed work in Chapter 6 point out the limitations of this learning paradigm. Zhao, C. et al. [97] claimed that the limited data volume causes the previous PAD system always face the risk of overfitting. The distinct representation for a presentation attack may also be unpredictable when facing different scenarios (such as different screen for replay attack). Some novel presentation attacks, such as the mask attack, evolve rapidly. The desired learning paradigm should help the system produce a robust classifier from a limited number of training samples.

3.2 PRE-PROCESSING ALGORITHMS

In this section some commonly used pre-processing algorithms are described in detail: (1) frame colour space transformation, (2) image cropping, (3) image affine transformation, and (4) face area segmentation and normalisation. These algorithms are also used in the experimental work reported in this thesis.

3.2.1 Colour space transformation

Colour space transformation is widely used as a pre-processing step. Some researchers claimed that the colour difference between the attack video and real face can be distinguished using their naked eyes [32]. However, human eyes are not very sensitive to colour difference [98] . This observation shows the potential benefits of applying various colour model transformation as a pre-processing algorithm.

Multiple local texture features are combined with the colour space transformation in recently proposed PAD algorithms[32]. In some deep learning based PAD approaches, colour model transformation is still considered an important pre-processing step to improve performance[61]. Different material reflectivity between human skins and the PA instruments may be more significant at some colour channels (such as hue, saturation, and lightness) other than the RGB channels. Boulkenafet, et al.[32] attempted to discuss the effect of colour space and the effect of the concatenation of the same feature from different colour spaces. They applied multiple combinations of colour spaces in their work, and concatenating features from different colour channels significantly increased the performance. The proposed methods also consider colour space by following the suggestions of literature [32], [99].

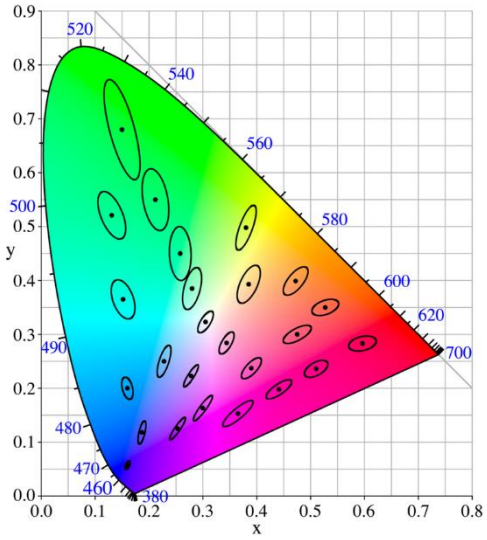


Figure 3.3 Visible colour difference range visualisation [98]

In some proposed experiments in this thesis, the raw input data with RGB channel are transformed to the HSL colour space. r, g, b represent red, green, and blue coordinates of the colour representation at a pixel; those values are real numbers between 0 and 1. In the HSL representation system, h, s, l means Hue, saturation, and lightness. Hue, denoted by *h*, is the measurement to describe a colour that is similar to the three primary colours.[100] :

$$h = \begin{cases} 0^{\circ} & if\ max = min \\ 60^{\circ} \times \frac{g-b}{max-min} + 0^{\circ} & if\ max = r\ and\ g \geq b \\ 60^{\circ} \times \frac{g-b}{max-min} + 360^{\circ} & if\ max = r\ and\ g < b \\ 60^{\circ} \times \frac{g-b}{max-min} + 120^{\circ} & if\ max = g \\ 60^{\circ} \times \frac{g-b}{max-min} + 240^{\circ} & if\ max = b \end{cases} \quad (3.1)$$

In the following proposed works, the numerical representation of colours is always normalised from the range [0,255] to [0,1]. In formula (3.1), *max* denotes the largest pixel value of *r*, *g* and *b* channels and *min* is used to represent the smallest pixel value of these different colour channels. In HSL space, $h \in [0, 360)$ is also named as hue angle, and *s*, $l \in [0,1]$ can be calculated by using formula (3.2) and (3.3) [100]

$$l = \frac{1}{2}(max + min) \quad (3.2)$$

$$s = \begin{cases} 0 & if\ l = 0\ or\ max = min \\ \dfrac{max - min}{max + min} = \dfrac{max - min}{2l} & if\ 0 < l < \dfrac{1}{2} \\ \dfrac{max - min}{2 - (max + \min\ \ )} = \dfrac{max - min}{2 - 2l} & if\ \dfrac{1}{2} < l \end{cases} \quad (3.3)$$

The value of *h* is usually normalized to between 0° and 360° and $h = 0$ is used for *max = min*. [100]

3.2.2 Image cropping and affine transformation

Image cropping and affine transformations are used differently in the traditional feature workflow and the deep learning workflow. For instance, some traditional facial anti-spoofing algorithms [32] claimed that cropping the facial area is considered as the pre-processing step. By applying this pre-processing step, traditional features may be less affected by the non-related information from the background region in the frame. In deep learning, however, researchers considered the cropping algorithms as an efficient way to enlarge the data volume of their training dataset. In this section will introduce both of these usages in detail.

Image cropping and affine transformations, as the essential pre-processing steps in the traditional features for PAD, are widely considered to enhance the proposed feature by reducing noise, or eliminating information from non-related backgrounds. Some researchers have proposed two cropping steps in their pre-processing stage: (1) facial area cropping and (2) cropping facial area into patches (or blocks)[61]. Cropping facial area for PAD can be considered as implementing a "hard attention" method for conventional features to help the method focus on the facial region. Meanwhile, dividing the facial region into n × n patches of same size is another popular way to produce feature vectors in PAD research [61]. Conventional features are applied on these patches, and the final feature vector is the concatenation for the feature vectors from different patches[101].

Dividing the facial area into patches as a pre-processing step, which has been shown to improve the performance of many traditional local features [27], is not an intuitive move. The possible reason for dividing the facial area into patches is that the patches exclude the spatial structure of faces, and help the traditional features focus on the global appeared local texture patterns. Some researchers [102] suggested that the facial structure information makes the classifier focus on the distinct facial

information, which may be more useful for face recognition rather than PAD. By following their suggestions, cropping facial regions into patches helps traditional local features focus on the spoofing texture differences, rather than the facial spatial structures. The classifier can then identify more relevant feature patterns of the presentation attack.

Affine transformations are used to resize the detected facial region into same scale for feature extraction to eliminate effects such as image resolution differences. Sometimes the affine transformation in the traditional feature-based PAD is considered part of facial normalisation.

The cropping and affine transformation, as a part of pre-processing, are used for data argumentation in the deep learning workflow to increase the number of training samples. For instance, Li, et al.[103] enlarged the volume of their training datasets four times by moving the cropping area to four different directions as shown in Figure. 3.4.

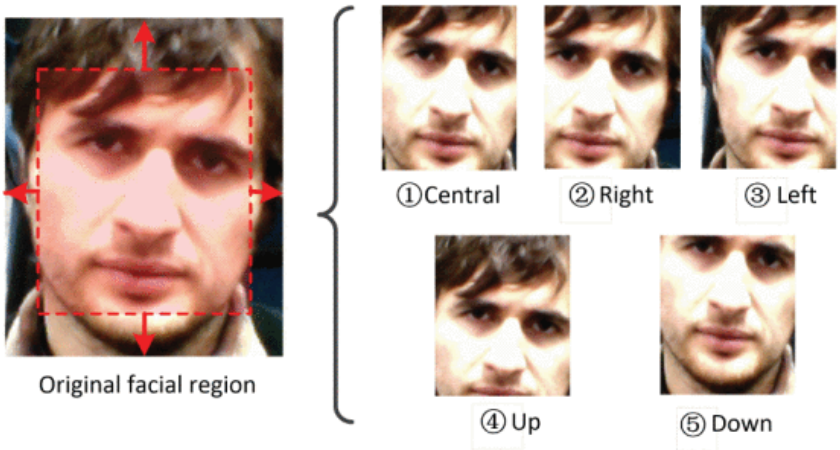


Figure 3.4 Example of data argumentation [103]

Data argumentation may be a fundamental step for deep neural networks, because: (1) Deep neural network is a data dependency algorithm and (2) The PAD datasets include imbalanced classes.

Deep learning is reported as a data dependency algorithm, and the successful application of deep learning algorithms usually requires large training datasets [104]. Most of these have collected of their raw training data from the Internet with relative ease. For instance, Imagenet [64] includes more than 14 million images and the size of this dataset was around 1TB at 2017. Only images are included in that database.

Youtube-8M [105], as one of the famous video datasets that is widely used in the video recognition area, includes 5.6 million videos for 3,862 different classes; and the total length of this dataset is more than 350,000 hours. However, it is difficult to obtain large volume training datasets for biometric facial PAD from publicly available datasets. The privacy considerations associated with biometric data restrict the volume of benchmark datasets. Additionally, each attack type requires a different data collection effort and each data collection efforts requires the involvement of human participants.

The imbalanced data for PAD is another problem, that causes a high risk of bias o for the deep leaning model. For instance, CASIA-FASD[27] includes 50 subjects, and each subject is provided with three genuine records and nine attacks. If researchers consider PAD as a binary classification problem and train their DNN without organising data batches carefully, the training data will be imbalanced and the trained DNN will tend to classify any input data as the presentation attack. This is not an isolated occurrence. Various presentation attack datasets include imbalanced number of samples for each class and insufficient dataset volumes. In this case, the data augmentation step is very necessary for applying deep learning algorithms in the field of PAD.

To overcome the drawback of the limited volume of training data, data augmentation methods are widely considered in literatures[106], [107]. The input frames or images can be cropped, rotated, zoomed in and zoomed out to generate new samples that have same labels as the original data. For instance, an input image size is 256×256 pixels and the desired input size of the neural network is 224×224 pixels. If researchers apply a cropping process as data augmentation step, each original image can generate up to (256-224)×(256-224)=1024 additional samples by cropping a 224×224 from the original frame. This means that the volume of the original dataset can be increased 1,024 times. Not all of this additional data can provide new information to train the neural network. However, the data augmentation still provides a possible way to expand the volume of available training data when the original dataset is limited.

There are many other data augmentation methods in deep learning; but , those methods are not suitable for the facial anti-spoofing study. For instance, adding noise to the original frames is widely applied in many recognition problems. Adding noise

will change the image quality of the raw data[108], and some researchers believe that image quality differences may be considered as a distinct characteristic for some PAs. Ojala, T. el al. [24] suggested that high-quality biometric data may help the biometric system to detect presentation attacks. Adding noise will confuse the deep learning-based PAD methods and mislead the model in the training phase. Meanwhile, data augmentation methods such as random erasure of some part of the original data may obscure important texture differences between genuine and attack presentations. Thus, the proposed experiments in this thesis only consider cropping and affine transformation for data augmentation.

In the proposed experiments, the following methods are applied for data augmentation: (1) rotating the original image within a certain angle range (0-20 degrees), (2) cropping the original images into different sub-images (normally one original frame can be used to generate 50 training samples, which also include the facial area), (3) scaling the original image by applying bilinear interpolation. All of these augmented data included facial areas (but the face was not necessarily centred). These data augmentation steps can be applied together. For instance, an augmented data sample could be rotated 10 degrees, enlarged 20% and cropped around the facial area at the same time.

The angles for rotation and the shear mapping for data augmentation should be selected carefully [40]. Large angles for rotation and shear mapping may change the 2D representation of the 3D facial structures which is not consistent with the real world. For instance, paper and video presentation attacks display the biometric sample on flat attack artefacts, and there are some works that detect PA by estimating the 3D structures from 2D input data [109]. However, rotations with large angles and shear mapping processes, which may both change the 2D representation of the input data, will increase the difficulties of 3D estimation. For this reason, the data augmentation method should be carefully selected for the proposed methods.

In this thesis, the following experiments which use deep transfer learning protocol only applied rotation and cropping to generate training samples. The total number of augmented training samples should not be more than the 1/5 of the total number of training samples. Other experiments, which use deep neural architectures but train the neural network from scratch, should apply all three methods to get a bigger

training set. The total number of augmented training samples should be more than the 1/3 of the total number of training samples.

3.2.3 Face detection and normalisation

Some researchers only consider the facial region in the raw data, as the input to decrease the computational complexity, and avoid the non-related information from the background [32]. There are some more reasons to emphasize the facial region in the input data. First, the facial region will include some important characteristics for PAD. The ISO document [14] considers the behaviour of occluding and misleading the biometric recognition system as a kind of attack that means that: some artefacts of PA will only appear at the facial region [96]. Second, the facial region can be considered as additional label for the learning models. Some recent work[110] provided pixel level labels that emphasized the importance of the facial region in PAD.

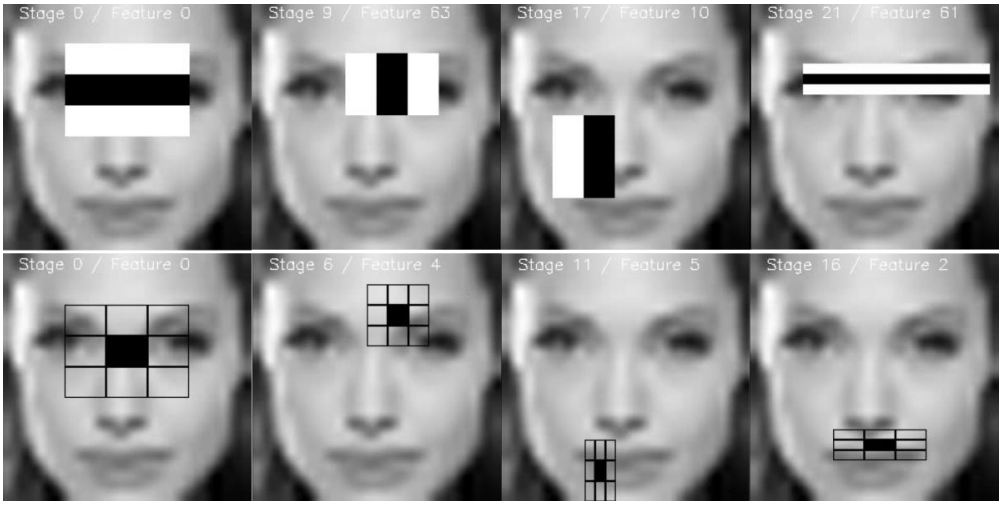


Figure 3.5 Example of Haar-based face detection [111]

Face detection and normalisation emphasize the information from the facial region and filter out non-related information from the background. Considering face detection and normalisation as a pre-processing step can greatly reduce data volume. The computational complexity and the processing speed of a PAD system, therefore, can be improved by processing the facial area only. In a possible workflow from the literature [9] (1) Face detector should be applied at the first step to detect the facial region in the input data. (2) Some facial landmark should be detected to confirm the

head gesture, (3) A normalised facial region should be calculated by using the head gestures; and (4) Some affine transformation method should be applied to get the normalised facial region.

In this pipeline, the face detectors should match the requirements of different feature extractors. There is a commonly used face detection method that is based on the Viola Jones face detector and implemented by OpenCV[112] (The example of applying this face detector can be found at Figure 3.5.). Some proposed novel methods in Chapter 4 use this face detector in the proposed experiments. The Viola Jones face detector has a good recognition rate for frontal faces in a brighter indoor environment and low computational complexity. Unless otherwise stated, the proposed methods in the contribution chapters will use Viola Jones face detector from OpenCV[112].

Moreover, some algorithms for PAD need the selected face detector to provide stable facial region detection in a frame sequence. In the proposed methods, detecting facial action units need the detected facial area to be very stable for different head positions in a continuous frame sequence to generate smooth facial action unit signal. Also, Liu, et al.[62] suggested the use of a face detection method that can extract facial regions precisely between the neighbouring frames in their implementation details. They claimed that the stable facial regions can help their method provide a "pseudo depth mask" for facial region and improve the performance of their PAD system. The proposed experiments in this thesis consider both Viola Jones Face detector and the DNN based face detector to demonstrate the performance of the proposed works.

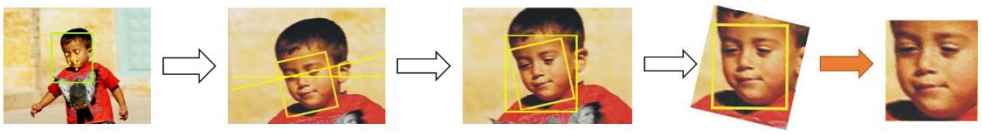


Figure 3.6 Example of face normalisation by using the position of eyes [113]

Once the facial area has been successfully detected, the facial region can be normalised to help the feature extractor produce consistent results. A good face normalisation step can improve the performance of a PAD system by reducing the effect of environmental condition changes (e.g. the effect of head motion and camera shaking). Two different facial normalisation methods have been applied in the

proposed experiments. The first method is based on affine transformations and the positions of eyes. When the face is detected, the eye detector, which is based on the *Haar cascades* pre-training model implemented by OpenCV[112], is used to detect the left and right eyes in the facial area. The rotation angle is calculated by measuring the angle between the line connecting the eyes and the facial bounding box. After this facial normalisation process, the rotated image is resized to the desired scale. Fig 3.6 shows the processing step for this facial normalisation step.

An alternative facial normalisation method [114] for face normalisation relies on detecting 3D facial landmarks. And this method normally has high computational costs. After determining the facial area, a 3D facial pose estimation and a 3D facial landmark detection step are performed by following an end-to-end neural architecture. In [114], researchers introduce the Local Neural Field (LNF) as descriptors for patches, and integrate the non-linearity of Conditional Neural Fields [115] together with the output of Continuous Conditional Random Fields [116] to represent the relationships in both temporal and spatial information. Here, the alternative facial normalisation method [114] can capture complex non-linear relationships between pixel values and extract accurate 3D facial landmark from the 2D raw input data. The proposed experiments can provide better normalisation results by using these facial landmarks.

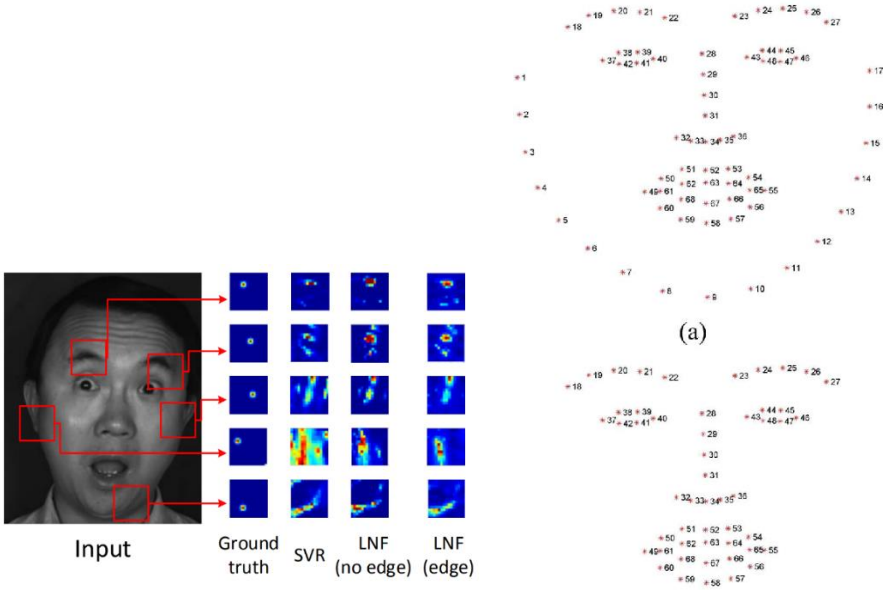


Figure 3.7 Example of facial landmark detection [114]

The proposed experiments tend to choose the pre-processing step with less computational complexity. In the following contribution chapters, only two proposed

methods, which extract facial action unit signals as the intermediate feature for PA, request to calculate 3D facial landmarks for face normalisation. The rest of the proposed methods only need a simple face normalisation step using the position of eyes. Perhaps a precisely normalised facial area could improve the performance of the proposed methods, but a complicated pre-processing step would significantly increase the difficulty of re-implementation.

3.2.4 Summary for Pre-processing

Applying a set of pre-processing steps can further improve the performance of a PAD system. However, too many pre-processing steps will also increase the system complexity. Each pre-processing step will bring new challenges for fine-tuning their parameters; and multiple pre-processing steps may make optimising all of the parameters virtually impossible. For this reason, each pre-processing method should be selected carefully to balance the performance improvements and the difficulty of optimizing the parameters. The challenge of optimizing such parameters for pre-processing and feature extraction is one reason for exploring the deep learning structures in Chapters 5 and 6.

3.3 FEATURE ENCODING AND CLASSIFICATION

Extracting useful information from data and encoding this information as a feature vector are two important parts of the feature extraction step. After this step, classifiers can be applied to obtain a decision about whether the input data is from a genuine user or an attack attempt. How to design a robust feature encoder and how to improve the performance in detecting PA are the main contributions of this thesis. A good feature encoder can produce a feature vector which is robust to multiple environment changes. Also, different PAI types can be classified using an effective feature space. Evaluating the performance of a feature encoder or a complete PAD system is essential for deploying such systems. Depending on the selection of datasets, there are some evaluation metrics which can be selected in different situations for performance evaluation and comparison. The following sub-sections will describe these issues in detail.

3.3.1 Feature Encoder

In this thesis, the term “feature encoder” is used to describe some algorithm or sub-neural network which can map the raw data to a feature vector. As the description in the introduction part, some proposed methods in this thesis are focusing on developing a novel feature or workflow which can detect PA precisely.

The proposed works are normally described by following a storyline: (1) Providing a distinct characteristic for PAD which can be observed and visualized by human experts or the machine itself. This distinct characteristic may follow other researchers’ work (such as the dynamic texture changes), or it follows the observation by some proposed experiments. (2) Generating an assumption from this observation or the visualization results. For instance, the proposed Facial Action Coding Histogram (FACH) assumes that the intensity value of the facial action unit may be a distinct difference between genuine face and spoofing attacks. (3) Providing an algorithm which follows the observation and the assumption generated above. Testing the proposed method by using the widely used datasets and producing a comparison for the proposed with the state-of-the-art methods to demonstrate the effectiveness of the proposed methods.

In the deep learning paradigm, researchers can train their DNN from scratch with PAD dataset and consider the feature extraction part from various pre-trained neural networks which are trained for other tasks (such as image recognition). This thesis also attempts to explore novel DNN methods in Chapters 5 and 6.

3.3.2 Classifier

In general, the problem of facial spoofing detection is defined as a supervised learning problem. As described earlier, two sub-categories are identified for this supervised problem: (1) binarized classification problem, and (2) multi-classification problem. The mission of a classifier is making predictions about the category to which the input data belongs. The multi-classification problem aims to classify the type of attack as well. Moreover, the multi-classification problem can be solved by cascading multiple binarized classifiers. When a PAD system implements multiple features, the results about features or even classifiers can be fused by weighting, voting, etc., to achieve the final performance improvement.

Here, we briefly introduce two widely used classifiers (SVM[117] and NN[118]) which can be used for both binarized and multi-categorised classifications. As a supervised learning problem, the *i*-th sample from dataset is denoted by $x_i$and the label of this sample is y_i and $i \in [1, n]$, where n is the total volume of this dataset. Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper-plane[117]. Being a supervised learning algorithm, the output model of SVM is an optimal hyper-plane which can categorize new samples. The original SVM cannot be applied to high-dimensional data. For this reason, the kernel trick[119] is applied to the original SVM. Here, SVM can be optimised by applying next formula[119]:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0,1-y_i(w\cdot x_i-b))\right]+\lambda\|w\|^2 \tag{3.4}$$

where x_i, y_i represent the labelled training data and *w* is the optimised parameter and λ is the hyper parameter for normalisation part. Here, n is the number of data points and w is the parameter need to be optimised. [119]

Artificial Neural Networks (ANN) [118] is another widely used classifier in PAD. A Neural Network classifier consists of multiple neurons which are arranged as layers. It also aims at mapping an input vector into some output. Theoretically, a deep enough neural networks can fit any mathematical function[104]. The classifier for PAD can be considered as a special function, and Neural Networks are used to learn this function from the training data. Here, ANN is different from DNN. The ANN can only learn how to classify the genuine and attack presentation by using Error Back Propagation algorithm[104]. It is not deep enough to learn the feature representation from the training dataset. And the Stochastic Gradient Descent (SGD)[120] or other terms for optimisers are used to represent the optimisation algorithm for a DNN.

3.4 DATASETS AND EVALUATIONS

There are two important factors to evaluate the PAD algorithms: Datasets and Evaluation Metrics. By using these two factors, researchers can compare their works, and analyses the advantages and disadvantages of their methods. The following subsections provide the widely used evaluation metrics and datasets. The proposed works in the following Chapters use these datasets to train the model and demonstrate the effectiveness by comparing with the state-of-the-art methods.

3.4.1 Evaluation Metrics

In a biometric system, the achievement of a high recognition rate is a basic requirement of the system. In order to understand the distribution of wrong classifications and evaluate the performance of a PAD algorithm, researchers have developed some evaluation metrics to measure performance. These evaluation metrics help the comparison of various methods. Basically, the performance of the system is often measured in terms of rates of these two different errors, False Accept Rate (FAR) and False Reject Rate (FRR):

$$FAR = \frac{FA}{NI} \quad , \quad FRR = \frac{FR}{NC} \tag{3.5}$$

Here, *FA* is the total number of false acceptances made by the system, *FR* is the total number of false rejections, *NC* is the number of client/genuine accesses, and *NI* is the number of impostor/attack accesses. The *FAR* is the false accepted rate and the *FRR* is the false reject rate. An widely used measure metrics combines these two ratios into the Half Total Error Rate (*HTER*), which is represented as follows[14]:

$$HTER = \frac{FAR + FRR}{2} \tag{3.6}$$

Also, the Equal Error Rate (EER) is another widely used evaluation metric which is used to determine a threshold value for its *FAR* and its *FRR*. When *FAR* and *FRR* are equal, the common value is referred to as the Equal Error Rate. The value indicates that the proportion of false acceptances is equal to the proportion of false rejections. It can represent the performance of a PAD algorithm in one number. In this thesis, EER is used for ease of comparison with the state-of-the-art.

More recently, two new metrics have been proposed for the evaluation of PAD systems, namely [14]:(1) Attack Presentation Classification Error Rate (APCER),and (2) Bona fide Presentation Classification Error Rate (BPCER). The APCER for a given Presentation Attack Instrument Species (PAIS) can be calculated as the formula: [14]

$$APCER_{PAIS} = 1 - (\frac{1}{N_{PAIS}}) \sum_{i=1}^{N_{PAIS}} (RES_i) \tag{3.7}$$

where N_{PAIS} is the number of attack presentations for the given presentation attack instrument (PAI) species [14]:. RES_i takes the value 1 if the *i*-th presentation is

classified as an attack presentation and a value of 0 if classified as bona fide presentation.

The BPCER is defined as follows: [14]

$$BPCER = \frac{\sum_{i=1}^{N_{PAIS}}(RES_i)}{N_{BF}} \tag{3.8}$$

where N_{BF} is the number of bona fide samples. RES_i takes the value equals to 1 if the i-th presentation is classified as an attack presentation and value equals to 0 if classified as genuine samples.

However, in the following contribution chapters, the performances of the proposed methods are reported by only using HTER or EER. The main reason behind this is providing a fair comparison with the state-of-the-art methods. Also, various datasets use HETR or EER, which can report performance with a single number, as their default evaluation metrics. The proposed experiments in the contribution chapters merely following the requests of datasets.

3.4.2 Datasets

The quality and the volume of existing datasets can be considered as an important index for the developments of PAD researches. One of the reasons for the rapid developments in this area is that high-quality datasets are published. These benchmark datasets offer comparison of the performance with the existing baseline algorithms to demonstrate the advantages of new algorithms. Also, deep learning-based PAD algorithms as an emergent rising branch of software-based PAD highly rely on the datasets with larger volume and better quality. The importance of datasets is repeatedly emphasized by the developing of the Deep learning algorithms. Therefore, collecting new datasets will always be an important mission in the future research. Here we list some important datasets in the area of PAD. The following datasets, which are widely used in the past several years, offer fair comparison with various state-of-the-art methods. The proposed works in the following Chapters are trained and evaluated by using these datasets.

Most of the datasets aims to collect various scenarios for the paper attack and the video attacks. NUAA Photograph Imposter Database[121] was designed for paper attacks which contains three sessions with changing environmental conditions. They used the camera with resolution of 640 × 480 pixels at 20 fps to record 15 subjects;

and each subject was recorded with 500 images. In order to make the dataset harder for the motion-based algorithms, Tan, et al. [121] requested their subjects try their best to keep their face motionless in front of the camera. When capturing the data, their subjects minimised the facial movements such as the eye-blinking.

Figure 3.8 Example of Idiap REPLAY-ATTACK database[27]

The REPLAY-ATTACK database [27] is another widely used face spoofing dataset which contains various attack behaviours and contains 1300 video clips. There are 50 clients recorded for both real access attempts and 3 different attack behaviours. Two illumination conditions were considered: *controlled* and *adverse*. For each condition, three attack categories were included: (1) *print attacks*, (2) *mobile attacks*, and (3) *highdef attacks*. The *mobile attacks* and *highdef attacks* can both be categorised as video attacks but use different sizes of the screen with different resolutions. They also considered various conditions about whether the attack device is fixed in front of the camera:(1) hand-based attack (the attack devices were held by hand) and (2) fixed-support attacks (the attack devices were fixed on a stand). The Replay-Attack database divides the whole datasets into three subsets, which are: the training set, the development set, and the testing set.

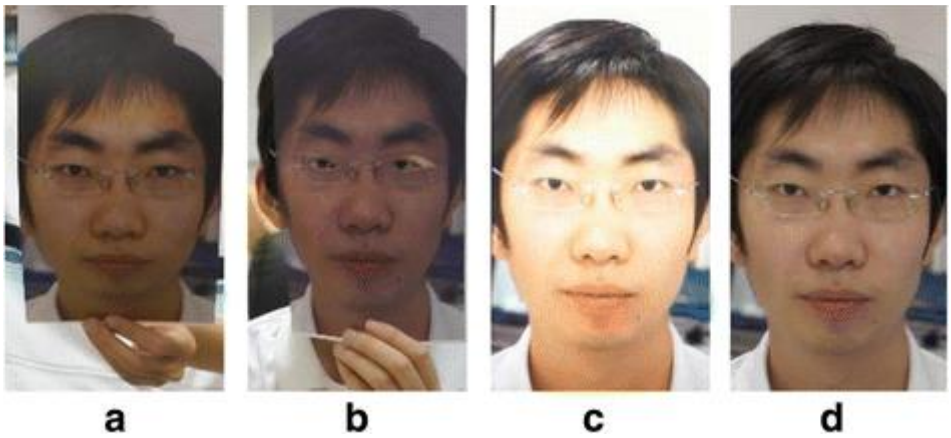


Figure 3.9 Examples of CASIA-FASD database, from left to right: (a) wrapped paper attacks, (b) cut paper attacks, (c) video attack, (d) real face [122]

The CASIA Face Anti-Spoofing database[122] consists of 600 video clips which include both real and spoofing access attempts, totally, there are 50 individuals listed in the dataset, where the spoofing artefacts were produced from high-quality records of genuine faces. Three different attack artefacts are included: warped photo attacks, cut photo attacks, and video attacks. All of them were designed to simulate real attack attempts. For instance, the cut photo attack is a special photo attack, in which a high-quality face is printed on paper, but where the area surrounding the eyes is cut to subvert eye-motion-based spoofing attack detection methods. Three different image resolutions were used in this dataset to simulate different usage conditions, namely low resolution, normal resolution, and high resolution. In their evaluation scenarios, 50 subjects were split into two categories: the training set (20 subjects) and the test set (30 subjects). They also designed seven detailed scenarios which are: (1) l*ow-quality*, (2) *normal-quality* (3) *high-quality,* (4) *warped photo attacks*, (5) *cut photo attacks*, and (6) *video attacks*. The (1), (2), and (3) scenarios are used to test the robustness at different image quality conditions. The (4), (5), and (6) scenarios are used to simulate different attack behaviours. The overall test scenario (7) provides combined performance test results for all attack types and qualities.

The MSU mobile face spoofing database [123], which consists of 280 video recordings of real and fake faces, addresses the challenge of using a low quality mobile camera. They used a built-in camera of MacBook Air 13-inch laptop (640 × 480 pixels) and a front camera of a Google Nexus 5 Android phone (720 ×480 pixels) to capture videos with at least nine seconds duration for all 35 subjects. This dataset includes both video and paper attacks. The high-quality biometric samples were taken by using

a Canon 550D camera and the back camera of an iPhone 5S. There are two screens used to generate video attacks which are an iPad Air screen and an iPhone 5S screen. For the printed attacks, HD pictures (5184×3456 pixels) were printed on A3 paper using an HP colour Laserjet CP6015xh printer. They require two subject-disjoint subsets for training and testing (15 and 20 subjects) in their evaluation protocol.

The OULU-NPU dataset[125] consists of 4950 samples from both genuine and attacks. Totally, 55 subjects were recorded by 6 different cameras under separate conditions. Each condition corresponds to a different combination of illumination and background. This dataset includes print attacks (created using two printers) and video-replay attacks (using two different displays).

Figure 3.10 Example of Rose-Youtu database [124]

The Rose-Youtu [124] dataset consists of a larger number of video clips. This dataset includes 3350 videos from 20 subjects which were recorded by 5 different cameras. They also include different attack types which consist of printed paper attack, video display attack, mask attack and video replay attack.

Figure 3.11 Example of HKBU MARs database [126]

On the other hand, it is also worth considering how much the attacker is going to spend to attack. With the popularity of 3D printing technology, the price of high-precision masks that were originally expensive have become affordable. However, this price is still relatively expensive for creating a data set. So even in recent years, it is still difficult to find a large amount of mask attack data sets. In addition, due to the lack of an open flexible mask dataset, it is still difficult for researchers to evaluate the performance of high-precision flexible material masks.

The HKBU MARs [126] is a dataset for high-quality 3D mask attack, which includes 2 types of 3D masks (6 from Thatsmyface.com and 2 from REAL-F.) and contains 120 videos (36000 frames) recorded from 8 subjects. This dataset uses a

Table 3.1 Datasets for PAD

Datasets	Sensors	Resolution	Attacks	Subjects	Released date
NUAA Impostor Database [121]	Webcam	640×480	Photo Attack	15	2010
Yale-Recaptured Database	Kodak C813 8.2MP Omnia i900, with 5MP	64×64 (following their pre-processing)	Video Attack (LCD)	10	2011
Print-Attack Database	Apple 13-inch MacBook	320×240	Photo Attack (printed); Video Attack	50	2011
Replay-Attack Database [27]	Apple 13-inch MacBook	320×240	Video Attack	50	2012
CASIA-FASD [122]	Sony NEX-5 camera, Two different USB Camera	640×480 1280× 720 1920×1050	Photo Attack (wrap and cut) Video Attack	50	2012
3D Face Mask DB	Kinect 1.0	640×480	3D Mask Attack	17	2013
MSU-MFSD Database [123]	Google Nexus 5 MacBook Air 13-inch	640×480 720×480	Photo Attack Video Attack	35	2015
HKBU MARs [126]	Logeitech C920	1280×720	3D Mask Attack	8	2016
Oulu-NPU Database [125]	Samsung Galaxy S6 edge, etc	1920 ×1080 (HD front camera)	Print Attack Video attacks	55	2017
Rose-Youtu [124]	Hasee Huawei iPad 4 iPhone 5s, etc.	640×480 1280× 720	Print Attack Video attacks	25	2018

Logeitech C920 web-camera (1280×720 resolution) to record their subjects and each video contains 300 frames with 25fps frame rate.

Table 3.1 summarises these widely used datasets by using following factors: (1) What sensor the dataset used to collect the video? (Sensor category in the table) (2) What resolution is used in their dataset? (resolution category in the table) (3) Which attack types the dataset is considered in their work? (4) How many subjects are considered in their dataset? (5) And the released date for this dataset.

From the middle of the 2017 to 2018, various datasets for PAD are released which include more subjects, various type of cameras, various attack types (such as silicon mask attack). However, the proposed works and experiments are nearly finished in that moment. The following descriptions are proposed to demonstrate the main advantages about these new datasets and to show the possible directions in the future. Firstly, the volume of the dataset is significant improved recently. For instance, the Unicamp Visual Attack Database (UVAD)[127] consists of 17,076 bona fide and attack presentation videos corresponding to 404 identities. However, they only consider the video attack in this dataset which is not enough for the robust PAD system. Meanwhile they only consider LBP and Histogram of Oriented Gradients (HOG) in their work and use Area Under Curve (AUC) to measure the performance which means other researchers can hardly to compare with the state-of-the-art methods in their dataset. Secondly, various cameras are considered in one dataset to record different attack types under different situations. For instance, HKBU-MARs consider seven different cameras under six different illumination conditions. Thirdly, multi-model data is considered in the dataset. For instance, the CASIA-SURF dataset[128] collect RGB, Depth, and IR data together. However, the proposed work only considers the RGB data which is easy to get. Finally, some latest dataset considers various attack types. For instance, Liu, Y. et al.[96] consider 13 types presentation attacks in their dataset which is designed for the few-shot facial PAD. However, they released their dataset in 2019, and their definition about few-shot facial PAD protocol is different with the proposed work.

3.5 SUMMARY

This chapter provided a basic experiment pipeline that will be used in the following contribution chapters. The experimental design, benchmark datasets and

evaluation metrics, which are considered in the proposed experiments in the following chapters, were provided as the materials that make the subsequent descriptions understandable and consistent.

Selecting a set of pre-processing algorithms is an essential task for some conventional features; and this chapter also introduced some widely used pre-processing algorithms such as face detection, face normalisation, and colour space transformation. The data augmentation, as an important pre-processing step for the deep learning-based methods, was also introduced in this chapter. The proposed experiments in the following chapters will follow the description in this chapter to select and implement the pre-processing methods.

This thesis also provides technical details for 10 widely used benchmark datasets which were published from 2010 to 2018 and compare these datasets by sensor types, resolutions, attack types, and the number of subjects. Some datasets that are not considered in this thesis are also briefly reviewed at the analyses part in Section 3.4. In the following chapters, some of these datasets and evaluation metrics will be used to demonstrate the effectiveness of the proposed novel PAD methods.

Chapter 4: Novel Traditional features for Presentation Attack Detection

This chapter introduces some novel conventional features for PAD that produced promising performances when evaluated using standard datasets. The motivation for developing these features is presented in Section 4.1. Then, a baseline experiment section is provided in Section 4.2. A novel feature named Facial Action Unit Histogram (FAUH) is described in Section 4.3, based on an encoding system for human facial movements. Then, three novel PAD features based on temporal texture changes are presented in Section 4.4 (Motion History Patterns (MHP),). Finally, some summaries are provided at Section 4.5. Part of this chapter was adapted from the published paper in the List of Publications.

4.1 Motivation

Despite the rising popularity and success of deep learning techniques in many areas of pattern recognition, the interest in developing conventional features for PAD has continued. One reason for this is the need to have a better understanding of the underlying mechanisms involved so that future threats can be better dealt with. It is also important to note the relative performances of conventional and deep learning approaches, especially with limited training data volumes.

As mentioned in Chapter 3, an important reason for the dramatic growth of deep learning in recent years is that DNNs provide feature encoders which have significantly advanced the state-of-the-art boundary through the use of large amounts of training data. Moreover, a DNN-based feature encoder can be trained with commonly available large datasets. And these feature encoder networks are transferable between different applications with a relatively short training time and a relatively small datasets for some computer vision tasks.

The challenge of using deep learning approaches in PAD research is the absence of published datasets with sufficiently large volumes of data. This situation requires a more careful examination of the characteristics of the PAD problem. This thesis explores the possibility that deep learning techniques can provide better performance

for PAD, but this would require a more sophisticated optimisation and design of the PAD system.

In order to better understand the nature and constraints of PAD, the proposed methods began with the exploration of conventional but novel features. This research began through observing data characteristics and making some assumptions about the nature of data. Then, some novel conventional features were designed and evaluated by using published datasets. And the results are compared with baseline algorithms and the state-of-the-art methods. These explorations in turn guided the design of DNN-based PAD methods in Chapters 5 and 6.

For this reason, the following experiment descriptions will follow the traditional processing steps: observing characteristics of PAD dataset, proposing conjectures and assumptions, designing traditional feature encoder algorithms, and evaluating the performance of the designed features to verify the assumptions.

Dynamic (time-varying) biometric data provide allows humans to distinguish whether they come from genuine face presentations with more confidence [19]. However, as stated in Chapter 2, some of the best performing features in published evaluations are often designed for static biometric samples. For example, Boulkenafet et al. [32] used the combination of colour space transformation and traditional texture feature descriptors and achieved a high-performance level on many widely-used datasets, by only using static biometric data. While temporal information may be useful for PAD; work continues towards finding an efficient way to use this information. This thesis presents some new features focused on using temporal information efficiently to detect presentation attacks.

Some researchers believed that Convolutional Neural Networks (CNNs) can model texture features (such as edges and texture patterns) in their first two or three convolutional layers [50] . And they can model some object level features (such as object parts) in their last two or three convolutional layers[129]. We use texture level features to represent such features as it can hardly be described by using human language (e.g. various texture patterns). And we use object level features to represent those features which are likely to be complete object parts and can be easily described using human description.

Firstly, some baseline experiments are presented in Section 4.2.

4.2 Baseline experiments

Before introducing the novel algorithms proposed in this thesis, a commonly used algorithm is presented to establish a performance baseline and to demonstrate the experiment workflow using a traditional feature. This will establish some basic concepts that will be used in subsequent experiments. Here PAD is formulated as a binary classification problem. In subsequent sections, different formulations with more classes are used depending on the application.

There is a long history of using Local Binary Patterns (LBP) as an efficient feature descriptor in the PAD area. Here, we follow Boulkenafet et al.'s [32] to obtain the feature vector by modelling the micro textures using LBP.

Firstly, one of the common observations in PAD is that texture difference between genuine presentations and spoofing attacks is often easily observed by visual examination. Although, these face images captured from presentation attacks may look very similar to the images captured from live faces, they do include some significant differences which can be used to detect attacks. The reason behind this phenomenon is the real human faces and attack artefacts reflect light in different ways. And real human faces are a complex non-rigid 3D objects whereas a photograph is a planar rigid object. This may cause different specular reflections and representations of shades. The texture representations for PA instruments may also be considered as some significant characteristics for PAD. Furthermore, presentation attack images may have different image quality due to different recapture conditions. All of these observations can be modelled by a good micro texture detector.

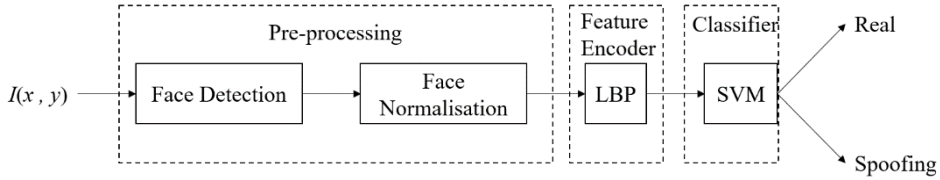


Figure 4.1 Block Diagram of the baseline method

A powerful texture descriptor is selected for this baseline experiment. The LBP texture descriptor, introduced by Ojala et al.[24] [8], is defined as a grey-scale invariant texture descriptor. It is derived from a mapping function of local

neighbourhoods for each pixel. The relations of local pixel patterns are modelled by a binary code; and a histogram of these binary code is generated as the feature vector. LBP is a powerful texture descriptor and is widely used for various applications due to its computational efficiency.

The original LBP operator [24] forms labels for the image pixels by thresholding the 3×3 neighbourhood of each pixel with the centre value and considering the result as a binary number. The histogram of these $2^8 = 256$ different labels can then be used as a texture descriptor.

The LBP has been extended to use different sizes of neighbourhoods set by using a circular neighbourhood and bilinearly interpolating values at non-integer pixel coordinates. [24]. This method allows any radius and number of pixels in the neighbourhood set. The notation (*P*, *R*) is generally used for pixel neighbourhoods to refer to *P* sampling points on a circle of radius *R*. The calculation of the LBP codes can be formulated as follows:

$$LBP_{P,R} = \Sigma_{p=0}^{P-1} Sig(g_p - g_c) * 2^p \tag{4.1}$$

$$Sig(Z) = \begin{cases} 1 & if\ Z > 0 \\ 0 & otherwise \end{cases} \tag{4.2}$$

where g_c corresponds to the gray value of the central pixel (x_c, y_c), g_p refers to gray values of *P* equally spaced pixels on a circle of radius *R*, and *Sig*() defines a thresholding function as (4.2).

Two pre-processing algorithms are considered in these experiments: facial cropping and facial normalisation. The implementation follows the descriptions in Chapter 3. Once the feature vector is computed by $LBP_{8,1}$, we use a nonlinear SVM classifier with a radial basis function kernel for determining whether the input image corresponds to a genuine face presentation or not. The SVM classifier is first trained using a set of positive (genuine presentations) and negative (attack) samples. The performance of this baseline algorithm is reported by EER for comparison. LibSVM Library [9] is used for SVM implementation in all experiments. The spoofing detection module takes only about 10.5ms in average to process an image on a MacBook pro (2012) using un-optimized MATLAB code.

Table 4.1 Performance of The LBP As Baseline Feature For Multiple Dataset

Datasets	**NUAA [121]**	**REPLAY-ATTACK database [27]**	**CASIA-FASD [122]**	**MSU-MFSD [123]**	**HKBU MARs [126]**	**Rose-Youtu [124]**
EER (%)	12.90%	16.10%	24.80%	14.70%	55.8%	27.7%

Performance results for various datasets are reported in Table 4.1. The following observations can be made from these results: (1) The worst result at HKBU MARs demonstrate that high-quality masks cannot be easily detected by using static texture descriptors. As mentioned in Chapter 2, temporal information can easily overcome this problem. The temporal features may be computationally complex and not represent competitive performance for paper and video attacks. But they have a good potential for mask attack detection. Also, there may be room for improvement for temporal features for paper and video attacks. (2) The performance differences at different datasets also demonstrate that the type of camera used for image capture may highly affect the performance of the baseline algorithm. Thus, it is important to evaluate any novel feature using multiple datasets. This principle will be adopted in the evaluations reported in the following chapters.

4.3 OBJECT LEVEL TEMPORAL FEATURE: FACIAL ACTION UNIT HISTOGRAM (FAUH)

The idea about designing an object level feature was inspired by some challenge response algorithms and some PAD algorithms detecting unconscious facial movements in the literature. Some researchers in this area observed that paper attackers and mask attackers cannot easily follow the request of the facial anti-spoofing system to complete some basic facial movements. For instance, the attackers using original paper attack method cannot respond to the request of biometric system to “blink eyes”. And some other researchers focused on detecting additional facial movements to detect presentation attacks. In some early work, researchers, also using challenge response approaches, asked users to give a smile, turn their face etc. Such techniques are especially effective for detecting artefacts such as photographs on printed-paper. [122]

However, such simple challenge-response techniques may experience a significant performance drop with video playback attacks. And the problem of usability may also be an issue for some challenge-response techniques, as these techniques may take more time and require more effort from users.

Facial movements can be categorised into conscious movements and unconscious movements [19]. As described in Chapter 2, there are multiple unconscious facial movements that can be detected within the face area, such as lips movements, eye blink movements, eye ball movements. The idea of FAUH as PAD features starts with modelling the relationship between the facial spoofing attack and some unconscious facial movements.

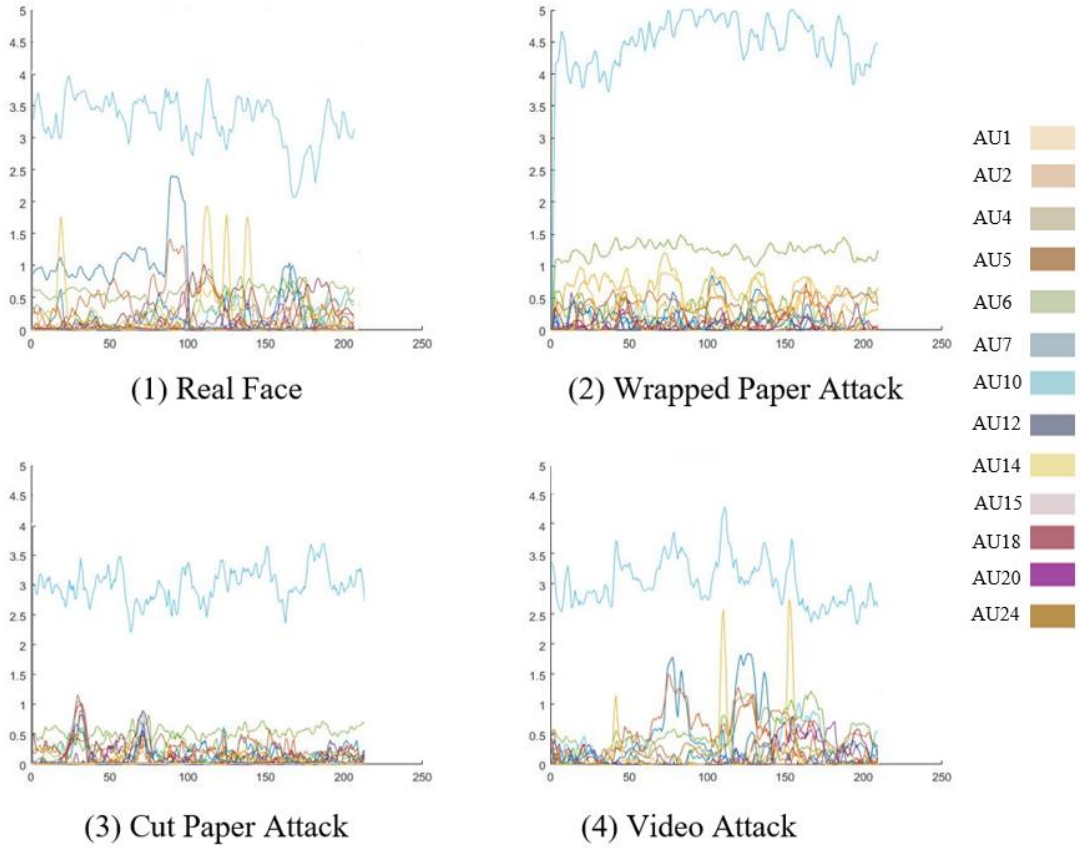


Figure 4.2 Example of AU signal visualization for different attack types. The x-axis represents different frame numbers of a video sequence and the y-axis represents the intensity value of the AUs at that frame. In this figure, different colours are used to distinguish different AU signals.

The key challenges in pursuit of these goals are (1) How to get a reliable symbolic representation of the variety of facial movements? (2) How to model the relationship of this representation and facial spoofing detection?

The Facial Action Coding System (FACS), suggested by the Carl-Herman Hjorstjö [130] in 1963, is designed to represent facial muscular activity for emotion analysis in psychological research. Ekman, P., et al [131] used this concept in an attempt to observe and interpret facial representation for automatic emotion recognition. Until now, FACS-based approaches remain popular for facial emotion analysis, avatar generation etc. [131]. The facial Action Units (AUs) may be categorised as additive or non-additive, depending on whether the triggering of an AU implies the triggering of other AUs [132] . Attack behaviour may be easier to classify by combining multiple AUs in the additive group as it is more difficult to construct an attack that ensures co-activation of multiple AUs. This is one of the reasons we chose groups of AU signals to develop our features.

Ekman,P et al. [131] defined 46 AUs through the observation of facial structure and muscle movement. Mihai Gavrilescu [132] suggested that not all AUs are effective at individual recognition and facial anti-spoofing. Cohn et al 's work [133] suggests that 13 AUs are stable over time and distinctive in identification (AU1, AU2, AU4, AU5, AU6, AU7, AU10, AU12, AU14, AU15, AU18, AU20, AU24). Here, AU1 means inner brow raised; AU2 shows outer brow raised; AU4 demonstrate brow lowered; AU5 means Upper lid raised; AU6 means cheek raised; AU7 shows lid tightened; AU10 shows upper lip raised; AU12 shows lip corner pulled; AU14 means dimpled appeared; AU 15 means lip corner depressed; AU18 means lip puckered; AU20 shows lip stretched; and AU24 shows lip pressed.

The facial expression recordings and the individual identification tests of 85 people show the potential of using facial movements in a person recognition system[133]. Mihai Gavrilescu [132] extended Cohn et al.'s study by analysing micro-expressions and introduced AUs to facial anti-spoofing usage. Gavrilescu suggested that a micro-expression-based personal identification system may be harder to subvert due to correlations between AUs signal. Using the taxonomy of [134], the system in[132] can be categorised as a person-specific face anti-spoofing approach. However, the accuracy of a person-specific face anti-spoofing approach is limited by the smaller size of the training data available for each person. Also, facial anti-spoofing is a sub-function of Mihai Gavrilescu 's [132] work, which requires the person recognition system to also work with FACS. This work, also requires information regarding the vertical distance from eyes (E) to brows (EB) (E–B distance), from mouth peripherals

(MP) to eyes (E) (MP–E distance) and from cheek internal extremities (CIE) to mouth centre (MC) as part of the feature vector.

The FAUH algorithm proposed in this thesis is based on the idea of using FACS for facial spoofing detection and attempts to provide a more general framework that is not person-specific and independent of a facial person recognition system. The second key problem identified above is to model the relationship between the facial movements' symbolic representation and the facial spoofing detection. Before we start to think how to model this relationship, an initial visualisation experiment is helpful to illustrate the potential of FACS for facial spoofing detection.

Firstly, the proposed visualisation experiment selects a fixed frame length as a hyper-parameter to cope with the different video lengths in the different datasets.(In this thesis, "parameters" indicate the trainable parameters and "hyper-parameters" indicate the pre-defined parameters, which are selected by researchers and can be optimised in the experiments.) Then, the proposed visualisation experiment feeds the fixed-length frame sets to a Facial Action Unit Detector which includes a pre-trained end-to-end model to extract the multiple Facial Action Unit labels from each frame. For each Action Unit signal within one frame, the end-to-end model will provide the classification results about which Action Unit exists and provide the intensity scores about this AU for each frame. Then, the intensity scores for AUs are concatenated and we name this temporal representation as the facial action units intensity signal in the following descriptions.

The visualisation experiment is to show the visible differences of the temporal intensity signal between spoofing attack and genuine presentations. Figure 4.2 presents examples of Facial Action Unit intensity signals and illustrates their potential capability to distinguish between genuine presentations and spoofing attack attempts. In this figure, the x-axis represents different frames and the y-axis represents the intensity values of AUs at that frame. Different colours are used to distinguish different AU signals. It is easy to identify that the intensity level of facial action unit signals for a real face are different from that for fake face presentations. For video replay attacks, which have the greater potential for subverting facial movement-based spoofing detection, there is an indication of some identifiable differences with genuine presentations as shown in Figure 4.2. The proposed method is based on the assumption

that the distribution of AU intensity signals can represent some significant difference between real presentation attempts and spoofing attacks.

Methodology

The proposed method can be defined as follows in Figure 4.3. G is defined as the index set of all the AUs, which include *N* elements. S is the selected subset of G and j is any element belonging to S.

$$i \in \mathrm{G} = \{1,2,\dots,\mathrm{N}\} \tag{4.3}$$

$$j \in \mathrm{S} \subseteq \mathrm{G}, |S| = n, n \in [1, N] \tag{4.4}$$

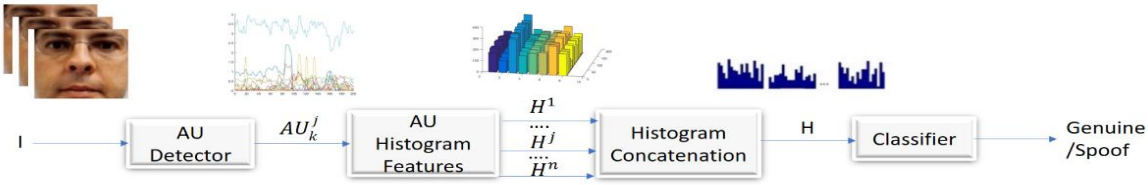


Figure 4.3 Block Diagram of the FAUH method

The mapping function *f* represents any AU detector which can extract AU_k^j from the *k*-th frame. Here, AU_k^j is the intensity value of the *j*-th AU at the *k*-th frame, where *K* is total number of frames of input video.

$$f \xrightarrow{AU\ detector,j} AU_k^j \quad where\ AU_k^j \in [0,5], \quad k \in [0,K] \tag{4.5}$$

The *hist*(A, B) function is then used to calculate a *B*-bins histogram H_B^j from the input data array A.

$$H_B^j = hist\left(AU_k^j, B\right) \tag{4.6}$$

The proposed feature H is generated by concatenating all the calculated H_B^j.

$$\mathrm{H} = H_B^1 \parallel H_B^2 \parallel \cdots \parallel H_B^j \parallel \cdots \parallel H_B^n \tag{4.7}$$

Where if p, q are column vectors $p^T \parallel q^T = (p_1, \dots p_m) \parallel (q_1, \dots q_r) = (p_1, \dots p_m, q_1, \dots q_r)$. To model the temporal AU signal, we firstly consider the histogram function which is widely used in the traditional features such as LBPs. We named the proposed novel feature Facial Action Units Histogram (FAUH) to

encapsulate this information for the detection of biometric presentation attacks without the need for active user cooperation. Here we provide a block diagram to show the processing workflow. In *Fig. 4.3*, video frames I were used to generate Facial Action Unit signals by using an AU detector. Then, the AU histogram was calculated by a histogram function $H_B^j = hist(AU_k^j, B)$. Finally, the histograms H_B^j for different AUs were concatenated to form a feature vector *H*.

Experiments and results for FAUH

The accuracy of AU signals is important in the proposed approach. Two different AU detectors are used to evaluate the influence of this accuracy: OpenFace project C++ open source implementation [135] and Temporal-based Action Unit Detection (TAUD) [136]. To assess the effectiveness of the proposed facial spoofing detection method two presentation attack detection datasets are used: the CASIA-FASD dataset [122] and the Replay-Attack dataset[27]. These two datasets are the most widely used datasets for presentation attack detection, which contain several recordings of the real client accesses and recordings of various spoofing attack attempts. They offer a fair comparison with the state-of-the-art approaches to show the effectiveness of the proposed method.

Table 4.2 CASIA-FASD overall test results with different AU selections

	OpenFace (EER)	**TAUD (EER)**
G1	42.37%	48.71%
G2	39.11%	49.25%
G4	36.77%	44.13%
G1+G2	37.89%	47.82%
G1+G2+G3	35.91%	N/A
G1+G4	26.41%	43.39%
G2+G4	25.56%	41.21%
G1+G2+G3+G4	21.11%	41.89%

The AU detector from the OpenFace project is used to estimate facial action unit signals from the image sequences using the pre-trained Constrained Local Neural Field (CLNF) for facial landmark detection[137]. In our experiment, the OpenFace AU detector was able to estimate 18 AU active signals and 16 AU intensity signals in real-time (20-30 fps) without any GPU support. The TAUD Action Unit detector [136] ,

also uses a pre-trained model which was trained and evaluated using a subset of the SEMAINE database[138]. This dataset is much smaller than the one used to train the OpenFace AU detector [135]. The TAUD AU detector can only estimate 7 AU signals. Both of the implementations can work on a single CPU machine without GPU support in real-time.

In our experiment, we define some AU groups to study the efficiency of different AU locations. To make our description clear, we define G1 to represent an AU subset which includes AUs around the brow (AU1-4), G2 to represent an AU subset which is related to blinks and eye-lid movements (AU5-7, AU41, AU45), G3 to represent an AU subset around the nose (AU9) and G4 to describe an AU subset related with the lip and the cheek (AU10-28).

There are more than 50 action units which can be detected in theory and the selected subset of Action Units in our experiment includes 18 AU active signals and 16 AU intensity signals. When we faced with this situation, some questions naturally arise: which FAUs subset is more related with facial spoofing detection? And why they are so representative of the spoofing behaviour? To answer these questions, we define different groups of AUs and also run feature selection algorithms in the experiments. The definition of groups, based on the location of different facial action unit, has potential to make the decision of this facial anti-spoofing system explainable. The intensity value of AUs needs to undergo a discretization process to calculate histograms for better performance.

Table 4.3 CASIA-FASD test results in terms of EER (%) at different Scenarios:(1) low quality, (2) normal quality and (3) high-quality (4) warped photo attacks, (5) cut photo attacks, (6) video attack, and (7)overall test

	1	**2**	**3**	**4**	**5**	**6**	**7**
LBP [122]	16.5	17.2	23.4	25.1	17.6	26.7	25.0
FAUH	22.1	20.7	21.4	16.3	17.1	28.5	21.11

Table 4.4 Replay-Attack DB overall test

	Dev (EER)	Test (HTER)
LBP [122]	17.9	13.7
FAUH	11.6	12.9

This section discretises the continuous AU intensity values into B=8 bins. In the experiment, a Support Vector Machine (SVM) with the RBF kernel is used as the classifier for comparison with other published results using the CASIA-FASD[122] and Replay-Attack datasets[27] offering a fair comparison with the state-of-the-art methods. The experiments use a four-fold subject-disjoint cross-validation protocol using the CASIA-FASD training set due to the absence of a development subset for this dataset[122]. The performance is reported by using the Equal Error Rate (EER) on the test set. The evaluation protocols of the Replay-Attack database[27] require producing the EER on the development set and the Half Total Error Rate (HTER) on the test set. The usefulness of different facial action units and groupings is evaluated by the pre-defined CASIA-FASD overall test[122]. The results of different facial action units and groupings are reported in Table 4.2. From this table, the AU features using the OpenFace detector are seen to result in better system performance. This suggests that the accuracy of AU detector can affect the final performance of spoofing detection. In general, AUs around the eye-lid (G2) and the AUs around the lip (G4) are more sensitive to spoofing behaviour. Table 4.2 also suggests that performance can be improved by combining different AUs into groups. Table 4.3 and Table 4.4 show results for the CASIA-FASD dataset[122] and Replay-Attack datasets[27].

The grey-scale LBP is used as a baseline algorithm for comparison. From Table 4.3, it is easy to notice that the Action Unit feature represents better results for warped photo attacks and cut photo attacks. And the proposed FAUH shows better results at overall tests (scenarios 7), which is an encouraging result to demonstrate the potential of using facial action unit signals.

Table 4.5 Comparison with the state-of-the-art at CASIA-FASD and Replay-Attack DB overall test

	CASIA-FASD (EER)	*Replay-Attack DB (HTER)*
LBP-baseline	25.0	13.7
DMD	21.8	3.8
Motion-Meg	14.4	0
FAUH(Proposed)	21.1	12.9

Table 4.5 represents the results of the comparison between the proposed method and some state-of-the-art methods, which include the initial results of different attempts on the CASIA-FASD [122] and the Replay-Attack datasets[27].

Table 4.5 shows the potential capability of AU signals for attack detection by comparison with base-line LBP features and other dynamic approaches. Here, the proposed method produces better result than baseline LBP and the Dynamic Mode Decomposition (DMD) [139] algorithm for the CASIA-FASD[122] dataset. The Motion-Meg algorithm [140] shows better results for both datasets. However, the proposed method is only an initial exploration of the potential of AU signals. Further refinements and optimisation can be performed that may enhance the accuracy of an AU-based approach for presentation attack detection.

4.4 Texture level temporal feature

The temporal texture changes may also provide significant information to distinguish between genuine presentations and spoofing attacks, despite any noise/distortion that may be introduced by signal capture or the nature of attack artefacts. Following this basic assumption, many published works have focused on using particular texture features (e.g. modelling Moiré patterns) for detecting presentation attacks.

Detecting facial spoofing attacks from static texture patterns is a fast and low-cost strategy. However, these static anti-spoofing approaches may be less accurate than the methods using temporal information for detecting mask attacks as they ignore temporal correlations between frames. Dynamic anti-spoofing schemes are designed to exploit spatial and temporal information together. However, such approaches require the higher computational complexity due to the data volumes associated with video processing.

In the following sections we present some novel PAD features incorporating low-level spatio-temporal information. These are motion history and motion energy images, super-pixel clustering and spatio-temporal co-occurrence matrices. The background literature for each of these are presented in subsequent sub-sections, followed by their description and evaluation.

4.4.1 Motion History Patterns (MHP)

In this section, the proposed novel time-based PAD algorithm, which is named as Motion History Patterns (MHP), combines Motion History Image (MHI) as primary features and two local texture descriptors as secondary features for PAD. In general,

presentation attacks can be recognised by human observers via the material differences between genuine faces and attack artefacts; the different representations between the non-rigid facial movements and rigid movements for artefacts; and the texture differences between the recaptured images and original images [19]. From the literature, PAD related temporal changes can be identified with some local texture descriptors by establishing an algorithm which can transfer temporal differences to texture patterns. Some previous works (such as LBP-TOP [43] and texture co-occurrence patterns) have demonstrated the feasibility of this general approach. The proposed work develops this idea to model temporal changes but also explores different ways to create time-related texture difference patterns.

Moreover, the proposed method is also focusing on exploring temporal texture differences and local texture co-relations between frames. For instance, in [19], the moiré pattern is considered as a significant indicator for detecting video attacks. However, moiré patterns may not be visible in every frame of the video. Some evaluation datasets, even with higher video quality, may still contain frames where moiré patterns are not visible. The disappearance of the moiré patterns makes the modelling of temporal local textures difficult, especially for shorter presentations (video sequences). However, these temporal texture changes (such as the appearance and disappearance of moiré patterns) can be easily enhanced and identified by using the frame difference method [141]. Furthermore, these temporal texture differences for PAD could appear in almost any location within a frame. Inspired by these facts, the proposed method is focusing on exploring temporal texture changes and transferring these changes into spatial texture patterns.

The frame difference algorithm as initially concerned by researchers [141] can only represent texture changes between two selected frames. However, not all the desired dynamic texture changes will appear between two selected frames. And applying the frame difference algorithm for each frame will produce a frame difference image sequence, thus enlarging the volume of data that needs to be processed. Furthermore, the texture changes between two frames may not be significant enough for PAD as in many cases the frame difference will not include significant temporal texture changes (such as moiré patterns). Also, object movements (such as body movements and facial movements) will also represent large pixel value changes that are not necessarily significant for PAD.

To overcome the limitations of the frame difference algorithm, the Motion History Image (MHI) is introduced to provide primary features for PAD-related temporal texture changes which are combined with a local texture descriptor (such as the LBP) to produce secondary features for PAD. According to Bobick and Davis, object movements can be decomposed using MHI by describing where the motion appeared and how the object moves. In this way the texture changes caused by object movements can be collected for recognition. One of the advantages of their idea is that the desired object movements and texture changes may be compressed and encoded into the spatial texture changes within a single frame[142].

There are two steps to produce an MHI. In the first step, the binary Motion Energy Image (MEI) is created and transformed into a Binary Motion Region (BMR) mask to represent the spatial relationship for the motions that have occurred in the image sequence. These BMR masks encapsulate temporal texture changes which have different characteristics for different presentation attack categories. For instance, paper attacks may include significant texture changes caused by different movement trajectories between faces and attack artefacts. These trajectories will be represented by different spatial locations in the sequence of MEI. The motion regions in MEI include the information about the motion-shapes and the spatial distribution of motions. The particular shapes of the motion texture patterns such as moiré pattern will be enhanced in the MEI. In the second step, the BMR mask sequence is compressed to generate the MHI by calculating a function of motion density at each pixel location. The intensity value of each pixel is a function of the motion at that pixel position. The original MHI algorithm can only be applied for fixed cameras. The data from hand-held cameras would need an optical flow algorithm as an additional pre-processing step. [142]

The proposed spatio-temporal primary feature construction consists of two parts: (1) The spatial component of the feature is normally the first image in the frame sequence which is used as the first image for calculating the MHI. (2) The temporal component of the proposed feature is the MHI itself. Then, the secondary features are explored to produce the final feature vectors.

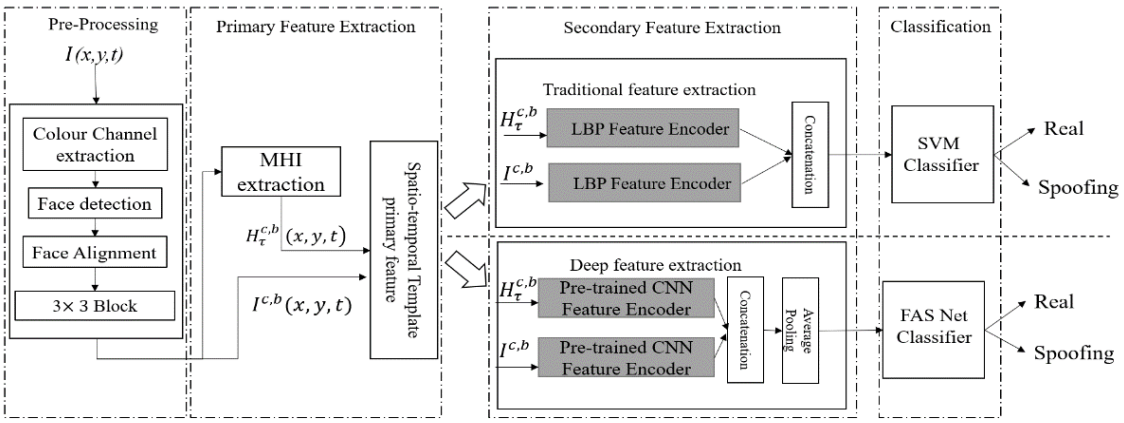


Figure 4.4 Experimental workflow for MHP(LBP) and MHP(CNN)

Methodology for MHP

The overall workflow of the proposed experiments exploring two different secondary feature extractors is presented in Fig. 4.4. For each frame sequence multiple colour channels (HSV and grey scale) are extracted and the algorithms will be applied on all of these colour channels. Then, the proposed method detects the facial area and performs face alignment by using eye positions. After that, the cropped facial area is divided into 3x3 blocks. For each of the blocks at each colour channel, a block sequence is formed for calculating MHI. This sequence, together with the first frame of the video are used as the primary spatio-temporal feature. Then, the local texture descriptors for the spatial texture and motion history texture are calculated separately for each of the block sequences. The final feature vector is constructed by the concatenation of multiple local texture descriptors. The proposed experimental workflow considers two secondary feature extractors separately and uses two different classifiers for different feature extractors. The system consisting of the LBP as the secondary feature extractor and SVM classifier is a traditional classification approach. The alternative system consisting of a pre-trained CNN and a FAS Net classifier has elements of a deep learning approach.

Bobick and Davis in [143] first proposed a representation and recognition method that decomposed motion-based recognition by first describing where there is motion (the spatial pattern) and then describing how the object is moving. They presented the construction of a binary MEI or binary motion region (BMR), which represents where motion has occurred in an image sequence [143], [144] . The MEI describes two things: the motion-shape and the spatial distribution of a motion. Next,

an MHI is generated. Intensity of each pixel in the MHI is a function of motion density at that location. One of the advantages of the MHI representation is that a range of times may been coded in a single frame, and in this way, the MHI spans the time scale of movements.

The MHI [144] can be considered as a temporal template, a vector-valued image where each component of each pixel is some function of the motion at that pixel position. The MHI, $H_\tau(x, y, t)$ can be computed using an update function $\Psi(x, y, t)$ [144]:

$$H_\tau(x, y, t) = \begin{cases} \tau & if\ \Psi(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t-1) - \delta) & otherwise \end{cases} \quad (4.8)$$

Where (x, y) represent the spatial location of movement and t shows the time point. $\Psi(x, y, t)$ is an indicator function to represent whether important temporal texture changes (moiré patterns) or object movements (e.g. head motion) are present in the current video frame. The temporal extent of the texture changes and movements is represented by the duration τ. The δ denotes the decay, which is used toreduce the influence of earlier texture. Each new video frame will call this update function to calculate the correlated Motion History Image as the temporal feature. The indicator function $\Psi(x, y, t)$ is calculated from a binarized frame difference image using a threshold ξ [144]:

$$\Psi(x, y, t) = \begin{cases} 1 & if\ D(x, y, t) \geq \xi \\ 0 & otherwise \end{cases} \quad (4.9)$$

Where $D(x, y, t) = |I(x, y, t) - I(x, y, t \pm \Delta)|$ And $I(x, y, t)$ is the intensity value of pixel location with coordinate (x, y) at the t-th frame of the image sequence. When the desired MHI is extracted from a frame sequence, the local texture descriptors are calculated for PAD. Then, the MHI and original frame are used together to form the spatio-temporal feature as described below.

In order to show the discriminative capability of the proposed spatio-temporal feature, two widely-used algorithms (LBP and CNN) for texture feature processing are are considered as the secondary feature in the proposed workflow for PAD. In the system using the traditional feature extraction method, the proposed workflow considers LBP as the secondary feature to describe texture patterns in the original frame and the MHI. The LBPs are widely used [122] for PAD as a highly discriminative texture descriptor for the proposed spatio-temporal feature. For each

pixel in the proposed feature, LBP can be defined with (4.1) and (4.2). Various unified LBP descriptors for different colour channels and blocks are concatenated to construct the final feature vector.

In the alternative system using convolutional neural network to extract the secondary feature, the proposed method uses a pre-trained CNN to generate feature descriptors. The convolutional neural network is a well-known algorithm for texture feature extraction. As shown by Lucena et al. [145], applying a pre-trained CNN using the transfer learning paradigm can improve the robustness of features and avoids the overfitting problem for PAD. Thus, the proposed workflow also uses the feature extraction part of a pre-trained CNN. The full architecture of the selected pre-trained DNNs should be designed for large-scale image classification problems. The efficiency of such deep neural architectures is demonstrated by the competitive performance score for the image classification competitions such as the ImageNet competition [64]. The transfer learning paradigm reuses the feature extraction part of the pre-trained network and trains a new classifier on the target domain by transferring the expressive feature space which is learned by the pre-trained neural network[146].

The transfer learning paradigm of a pre-trained deep neural architecture can be defined as the following steps: (1) The pre-trained DNNs, which include the deep neural architectures and all of the parameters in the networks, should be divided into feature extractor parts and classification parts by following [145]. The original classification layers should be replaced by a new classification sub-network. This new classification sub-network is initialised and trained for presentation attack detection. (2) The new network with pre-trained feature extraction part and replaced classification part is trained using presentation attack datasets with different learning rates. The classification sub-network can follow the suggested learning rate of [145]. But the pre-trained feature extraction part should start with the lower learning rate than the classification sub-network. (3) The whole network is fine-tuned with a low learning rate to get better performance.

In the proposed alternative system using CNN to produce the secondary feature, the initial frame of the video sequence and the related MHI are fed into the feature extractor network. Then, the feature vectors from the frame and the related MHI are concatenated as the feature descriptor for the proposed spatio-temporal feature. One average pooling layer is applied to combine various colour channels and image blocks

to generate the feature descriptor. And this final feature vector is then fed into the classification sub-network.

Experiments and results for MHP(LBP) and MHP(CNN)

To assess the effectiveness of the proposed facial spoofing detection method two presentation attack detection datasets are used: the CASIA-FASD dataset [122] and the Replay-Attack dataset[27] . These two datasets are the most widely used datasets for presentation attack detection, which contain several recordings of the genuine client accesses and recordings of various spoofing attack attempts. They offer a fair comparison with the state-of-the-art approaches to show the effectiveness of the proposed method.

The pre-processing for the proposed workflow is important. The fusion of multiple colour spaces is a commonly used approach to enhance performance. The proposed method follows Boulkenafet et al's work [32] to consider multiple colour spaces to explore the colour texture information for PAD. The reason behind this is that the character of the artefact may be more visible in the local uniform areas (e.g. cheeks) . For this reason, the proposed method crops the facial area into 3×3 patches after face alignment and face normalisation. The final feature vector is the concatenation of multiple colour channels and all of the cropped patches.

In the proposed system using LBP to produce secondary feature, a Support Vector Machine (SVM) with the RBF kernel is used as the classifier for comparison with other works using the CASIA-FASD[122] and Replay-Attack datasets[27]. The alternative system using CNN includes two important factors: the pre-trained feature extractor network and the classifier network.

The selection of the pre-trained feature extractor network is important for the proposed alternative system using CNN as the secondary feature. In this system, the pre-trained VGG16 [47] network is considered as a texture feature extractor network in our implementation. The original VGG 16 network, which includes 16 convolutional layers with 3×3 kernel size, is a 2D convolutional neural network for the ImageNet competition[64]. They use ReLU[52] as activation function and 3 dense layers (or fully connected layers (FC)) for classification. The original dense layers are removed for transfer learning. The classifier network is another important factor for the proposed method. The proposed method follows Lucena et al. [145]'s suggestion

which uses a new classification sub-network consisting of one flattened layer, one dense layer with ReLU activation, one dropout layer [147], and one dense layer with a sigmoid activation function. In the training stage, the classification sub-network is optimized by using the initial learning rate of 10^{-4}. The pre-trained feature extractor network is optimized by using the initial learning rate of 10^{-6}. The last dense layer uses a sigmoid activation function.

These experiments use a four-fold subject-disjoint cross-validation using the CASIA-FASD[122] training set due to the absence of a development subset for this dataset. The performance is reported by using the Equal Error Rate (EER) on the test set. The evaluation protocols of the Replay-Attack database[27] require producing the EER on the development set and the Half Total Error Rate (HTER) on the test set.

Table 4.6 Effect of different hyper-parameters for MHI-LBP

Hyper-parameters	**Replay-Attack DB (HTER)**
τ=15 δ=30 gray channel	19.4
τ=15 δ=10 gray channel	13.1
τ=15 δ=10 conbined with spatial LBP at gray channel	7.4
τ=15 δ=10 spatial LBP with RGB colour space	4.3
τ=15 δ=10 spatial LBP with RGB_HSV colour space	3.9

The initial experiment is used to explore the effectiveness of the different hyper-parameters τ and δ in Table 4.6 for the proposed MHP(LBP) feature. Here, the term "hyper-parameter" is used to indicate the parameters which is tuned by researchers and the "parameter" is used to indicate the trainable parameters. From this table, the proposed MHP(LBP) reach the best performance when $\tau = 15$ and $\delta = 10$. The LBP feature vector from different colour channels are concatenated together and the performance is further improved by using this strategy. It is important to notice that different colour channels may need different selections of τ and δ for best performance. And different datasets may need different selection of hyper-parameters to reach the global optimum point. However, the proposed experiment only explores the combinations in the Table 4.6 due to the time limitation.

Table 4.7 Comparison with the state-of-the-art LBP-based PAD methods on CASIA-FASD and Replay-Attack DB overall test

Method	CASIA-FA (EER)	Replay-Attack DB (HTER)
LBP-baseline[122]	25.0	13.7
Colour LBP[32]	2.1	3.5
LBP-TOP[43]	10.6	7.6
Proposed MHP(LBP)	4.8	3.9

Table 4.8 comparison with the state-of-the-art DNN-based PAD methods on CASIA-FASD and Replay-Attack DB overall test (The * means the performance score from our implementation following the referenced work)

Method	CASIA-FA (EER)	Replay-Attack DB (HTER)
YangNet[67]	6.2	2.6
FASNet*[145]	8.6*	3.9*
CNN+LSTM[72]	5.8*	6.3*
CNN- LBP-TOP[75]	8.0	4.7
Proposed MHP(CNN)	6.0	4.5

The proposed method is compared with 7 other approaches which include baseline LBP [122], Colour LBP [32], LBP-TOP [43], CNN [67], FASNet [145], CNN+LSTM [72], and CNN- LBP-TOP [75]. We firstly compared the LBP-based methods in Table 4.7. The implementation detail of baseline LBP has followed the CASIA-FASD protocol[122]. Then, Colour LBP [32] can be considered as the representative method of static feature-based PAD algorithms. The LBP-TOP [43] is also designed for the spatio-temporal texture changes which use LBP as the texture descriptor. From Table 4.7, the proposed MHP(LBP) is seen to provide good performance scores for both datasets. Although the Colour LBP [32] as a static-texture method represents better performance than the proposed method, the proposed MHP(LBP) shows a better performance when compared with LBP-TOP using LBP as the texture descriptor.

Then CNN-based methods are compared at Table 4.8. The CNN [67] is the first published work using convolutional neural network for PAD. The FASNet [145] also uses a pre-trained VGG16[47] as the feature encoder and use the transfer learning paradigm to fine-tuning their networks. It demonstrates the effectiveness of the

proposed feature. The original FASNet do not train and test their algorithm on the CASIA-FASD dataset[122]. We follow their paper and re-implement their algorithm for the CASIA-FASD dataset. The CNN+LSTM [72] method represents the effectiveness of the end-to-end neural network for spatio-temporal texture changes. We also re-implement their work for the comparison on the Replay-Attack Dataset[27]. The CNN-LBP-TOP [75] as a hybrid method which combines traditional features and DNNs is also considered for the comparison.

The proposed MHP(CNN) provides the second best performance for the CASIA-FASD [122] when compared to the listed CNN-based methods. It demonstrates the effectiveness of the proposed spatio-temporal feature. On the Replay Attack Dataset, CNN [67] showed the best performance. However, it includes multiple data augmentation stages and is trained from scratch. Some have claimed that these are overfitting their training data [75]. The proposed method uses the pre-trained CNN architecture to overcome the overfitting problem. Moreover, the proposed spatio-temporal feature outperforms those proposed in [72] and [75]'s when evaluated on the Replay-Attack dataset. [27].

4.4.2 Temporal Co-occurrence Local Binary Patterns

In this subsection, we will briefly describe the literature related with the Local Binary Patterns feature and provide details about the proposed novel feature Temporal Co-occurrence Adjacent Local Binary Patterns (TCoALBP). Local texture descriptors have been used in Feature-Level spoofing attack detection. The Local Binary Patterns (LBP) descriptor proposed by Ojala et al. [24] has been used extensively by researchers for spoofing attack detection, in part due to its computational efficiency [122]. Various extensions of the LBP are also introduced to improve its performance. For instance, the usability of colour extensions of LBP (CLBP) by modelling the colour characteristics of spoofing artefacts is explored in [32] . These extensions are categorised as static features due to their direct use of texture descriptors on a single biometric sample. Some researchers try to combine LBP with temporal information. The main challenge of using LBP directly to extract temporal information is that the original LBP can only process 2D texture information.

However, a 3D extension of the 2D Local Binary Pattern can be envisaged where the third dimension is time as represented by the sequence of video frames. There are

two challenges in the design of a temporal extension of the LBP feature. Firstly, there is the problem of defining a meaningful time-series extension of the LBP descriptor. Secondly, there is the problem of coping with the large data volume inherent in video processing. Selecting a set of neighbouring points in 3D can be considered as an equidistant sampling problem on a sphere, which may still be difficult to implement.

Moreover, while it may be easy to design encoding for the sequence of neighbouring points, it may still be hard to prove its effectiveness for distinguishing between different textures. Zhao and Pietikainen's proposed the Volume LBP (VLBP), in an attempt to extend 2D LBP to 3D volume data [148]. They suggest their approach can be used in both video and RGB-D images. An alternative approach proposed in [149] encodes 3D local texture in a video sequence by sampling the neighbouring points defined on the surface of a ball using the Uniform LBP[122]. However, this method may encode different textures with the same binary code. For the data volume problem, researchers have considered data selection methods as solutions. The data volume of a video cube is related to the video length/duration. de Freitas Pereira et al. [43] followed the data selection approach, which selects three orthogonal planes X-Y, X-T, and Y-T for a simple implementation named LBP-TOP, which compresses temporal-related data by generating X-T and Y-T orthogonal planes.

Inspired by LBP-TOP, a number of other three orthogonal planes have been investigated. However, the disadvantage of using only three orthogonal planes is that some crucial information may be missed. The orthogonal planes are selected at the middle of frames and may thus miss some crucial information such as eye blinks. The co-occurrence adjacent LBP (CoALBP) [150] was originally designed for texture pattern recognition, and was used for facial spoofing attack detection as a colour texture descriptor in [32].

Methodology

For any frame I in the frame sequence, $I\ (x, y)$ represents the pixel value located at (x, y). The LBP is defined using (4.1) and (4.2). The Uniform function is defined in Ojala et al's work [24] as (4.10):

$$U\left(LBP_{P,R}\right) = \Sigma_{p=1}^{P-1}\left|Sig\left(g_{p-1} - g_c\right) - Sig\left(g_p - g_c\right)\right| + \left|Sig(g_{P-1} - g_c) - Sig(g_0 - g_c)\right| \tag{4.10}$$

where P is the number of sampling points in a circular neighbourhood set of radius R centred at (x, y); and g_p indicates the pixel value of the p-th point on this neighbourhood. The pixel value of the central points g_c is used as a threshold for g_p. The Uniform LBP $LBP_{P,R}^{U2}(x,y)$ only considers the binary patterns which $U\left(LBP_{P,R}\right)$ <2. For instance, the "00001111" and "00111100" are uniform patterns. The Uniform function reduces the number of valid LBP codes and therefore reduces the dimension of the LBP descriptor.

The co-occurrence adjacent LBP (CoALBP) is defined using (4.11), (4.12), and (4.13) [150].

$$H(k) = \Sigma_{x=1}^{M-2}\Sigma_{y=1}^{N-2}\delta\left(g\left(LBP_{P,R}(x,y),a\right),k\right) \tag{4.11}$$

$$where\ k\epsilon[0, Np \times Np - 1]\ and\ a\epsilon A$$

$$\delta(u,v) = \begin{cases} 1 & if\ u = v \\ 0 & otherwise \end{cases} \tag{4.12}$$

$$A = \{(0, \nabla)^T, (\nabla, 0)^T, (\nabla, \nabla)^T, (-\nabla, \nabla)^T\} \tag{4.13}$$

In (4.11), A is a transition vector set. Each element of A, a, represents a transition/displacement between two locations in space. The parameter ∇ means the distance between these two locations and $\nabla_x, \nabla_y, \nabla_t$ has been used to demonstrate the distance at different dimensions. The function $g\left(LBP_{P,R}(x,y),a\right)$ returns a binary code by concatenating the $LBP_{P,R}(x,y)$ and its adjacent pattern addressed by a. $\delta(u,v)$ function is the function used to generate a histogram as the feature vector. In [150] the length of the feature vector is $Np \times Np \times 4$, where $Np \times Np$ is the number of all possible combinations of spatially adjacent patterns and there are four elements in A. If an implementation uses the original LBP with P =8, Np is 256. The feature length is then 256×256×4=262144 for one image [24].

In order to utilise the possible patterns of texture co-occurrence across time as well as space, we extend the transition vector set A to a 3D space (A_{3D}) by adding the temporal dimension.

Additionally, we calculate the Uniform LBP $LBP_{P,R}^{U2}(x,y)$ instead of using the LBP to decrease the dimension of histogram. The function $g\left(LBP_{P,R}^{U2}(x,y),a_{3D}\right)$ concatenates the binary codes of $LBP_{P,R}^{U2}(x,y)$ and updates the histogram. The length of the feature vector should be $Np^{u2} \times Np^{u2} \times h$, where h is the number of elements

in A_{3D} and $Np^{u2} \times Np^{u2}$ is the number of all possible combinations of spatially adjacent uniform patterns. The feature vector $H(k)$can then be calculated using (4.14) and (4.15):

$$H(k) = \Sigma_{x=1}^{M-2}\Sigma_{y=1}^{N-2}\delta\left(g\left(LBP_{P,R}^{U2}(x,y), a_{3D}\right), k\right) \quad (4.14)$$

$$where\ k\epsilon[0, Np^{u2} \times Np^{u2} \times h - 1]\ \ and\ a_{3D} \in A_{3D}$$

$$A_{3D} = \left\{(\nabla_x, \nabla_y, \nabla_t)^T \middle| \nabla_x, \nabla_y, \nabla_t \in \boldsymbol{Z}\right\} \quad (4.15)$$

Implementation details

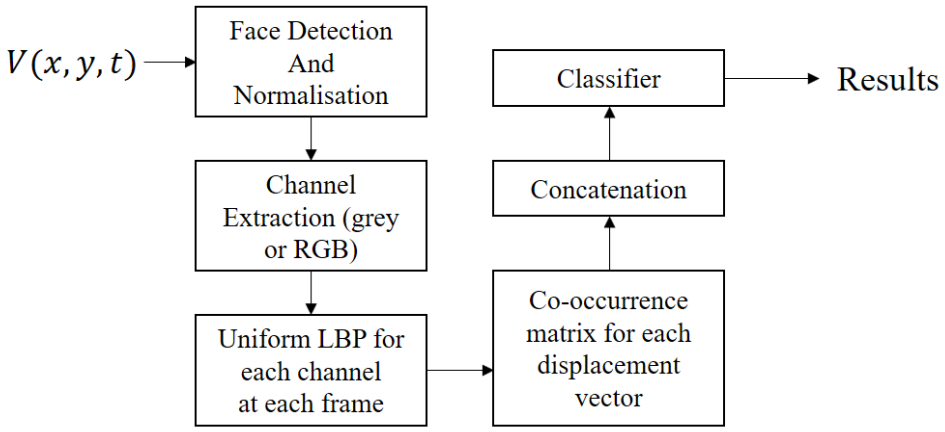


Figure 4.5 Temporal Co-occurrence Adjacent Local Binary Pattern (TCoALBP) workflow

Figure 4.5 is a block-diagram of the proposed system that was evaluated. The face areas are detected and normalised using facial landmarks to decrease the effect of the background as the description in Chapter 3. Then, the frames are divided into R, G and B channels. For each channel, the Uniform LBPs are calculated for each face area. Then, the feature histogram is calculated by concatenating histograms. Moreover, the number of co-occurrence matrices in our implementation are determined by the number of distinct elements in *A*. A different selection of *A* will cause a different feature size, which is fixed to 7 in the evaluated implementation to balance the speed and the performance of the proposed feature. For the TCoALBP (RGB) in our experiment, the final feature vector H is generated by formula (4.16), whereH_R, H_G,

and H_B are TCoALBP feature vectors for red, green and blue channels of the input videos respectively.

$$\mathrm{H} = \{H_R, H_G, H_B\} \tag{4.16}$$

The classifier is a Support Vector Machine (SVM) with an RBF Kernel which has also been used by other researchers with whom results are compared.

Datasets and performance metrics

There are three widely used anti-spoofing benchmarking datasets which were used to evaluate the effectiveness of the proposed anti-spoofing algorithm: the CASIA -FASD dataset [122], the Replay-Attack database[27], and MSU MFSD dataset[123]. All of these datasets include some recordings of genuine client access attempts and various presentation attacks. The pre-defined evaluation protocol for each dataset was followed for a fair evaluation and comparison with the state-of-the-art.

Various environmental conditions can affect performance; i.e., different image qualities, different distances between the face and the camera, different face angles, and background changes. The pre-defined scenarios at CASIA-FASD [122]dataset is used for a detailed performance test at various conditions and attack types.

The CASIA-FASD[122] and MSU MFSD datasets[123] do not contain a development set for fine-tuning of parameters. Thus, a four-fold subject-disjoint cross-validation was used on the training set to train the classifier and fine-tune parameters. After that, the experiments evaluate the performance by calculating the Equal Error Rate (EER) on the test set [15]. The Replay-Attack database[27] contains a development subset for parameter fine-tuning. The experiments follow their protocol to produce the EER on the development set and the Half Total Error Rate (HTER) on the test set [16]. To make the error trade-off clearly, the True Positive Rate (TPR) were reported where False Accept Rates (FAR) are fixed to 0.01 and 0.1.

The effectiveness and robustness of the proposed system using the TCoALBP features have been tested for different attack types and environment conditions using the CASIA-FASD[122], Replay-Attack, and MSU-MFSD databases[123] and the results are reported in this section and compared with some state-of-the-art techniques. Additionally, the selection of different parameters are tested for the different datasets.

Parameter optimization

Three different parameter selection experiments were conducted to improve the final result and show some properties of the proposed method. All of them provide results using the pre-defined overall test in the protocol for each dataset by implementing TCoALBP with different parameters. All tables include three different columns of results which are EER for CASIA-FASD overall test[122], HTER for Replay-Attack overall test [27], and EER for MSU MFSD overall test[123].

Firstly, the neighbourhood location is an important parameter for TCoALBP. Table 4.9 includes the result of TCoALBP ((P, R,∇_x, ∇_y, ∇_t)=(4,1,1,1,1)) for grey-scale video input with 30 frames but with different *A* sets. The first row of the *A* set in the table only contains spatial correlation with $\nabla_t = 0$, which can be considered as CoALBP. The second row of set *A* includes two parts: (1) neighbour sub-set only including spatial displacements (2) neighbour sub-set only including temporal displacement. The third row of set A includes neighbours with both spatial and temporal displacements. Clearly including both spatial and temporal displacements improves performance. The following experiments will follow the best results of the Table 4.9.

Table 4.9 Performance for TCoALBP for different *A* sets (grey-scale video, 30 frames and parameters are fixed to (P,R,∇_x, ∇_y, ∇_t)=(4,1,1,1,1)).

A	**CASIA (EER)**	**R-A (HTER)**	**MSU MFSD (EER)**
{(1,0,0),(0,1,0),(-1,0,0),(1,1,0), (-1,1,0), (1,-1,0),(0,-1,0)}	12.1%	11.8%	18.7%
{(1,0,0),(0,1,0),(1,1,0),(-1,1,0),(1,-1,0), (-1,0,0),(0,0,1)}	10.16%	7.01%	20.01%
{(1,0,1),(0,1,1),(-1,0,1),(1,1,1),(-1,1,1), (1,-1,1),(0,-1,1)}	8.69%	6.07%	16.60%

Secondly, the video duration is considered as a parameter in this paper. Table 4.9 shows performance at different video durations, where TCoALBP((P,R,∇_x, ∇_y, ∇_t)=(4,1,1,1,1)) using grey-scale video as input. Generally, a

longer video duration can improve system performance especially between 30 frames and 60 frames.

Table 4.10 Performance for TCoALBP on different frame numbers and grey-scale video (parameters are fixed to ((P,R,∇_x, ∇_y, ∇_t)=(4,1,1,1,1)) DFN means different frame numbers

DFN*	CASIA-FASD (EER)	R-A (HTER)	MSU MFSD (EER)
5	23.33%	18.08%	34.35%
10	16.67%	15.97%	27.54%
15	15.42%	11.44%	27.74%
20	13.53%	9.81%	24.01%
25	11.15%	7.33%	15.77%
30	8.69%	6.07%	16.60%
60	7.96%	**5.32%**	14.29%
100	8.02%	5.94%	**12.33%**
All frames	**7.62%**	5.88%	14.97%

The uniform LBP with different parameters (*P*, *R*) = {(4,1), (8,1), (4,2)} is explored to test the impact of different radius *R* and the different number of sampling points *P* with different ∇ sets. The magnitude of displacement $\nabla_x, \nabla_y, \nabla_t$ can be roughly split into three subtypes: (1) ∇>2R, (2) ∇=2R, (3) ∇<2R. From the result of these three subtypes, the selection of $\nabla_x, \nabla_y, \nabla_t$ does appear to affect system performance, $\nabla_t \geq$ 2R slightly improves the system performance. Also, optimizing the (P,R,∇_x, ∇_y, ∇_t) parameter set may not be a convex optimization problem. Thus the following experiments will use (P,R,∇_x, ∇_y, ∇_t)=(4,1,1,1,3) as a fixed parameter set.

Intra-dataset results and comparison

Table 4.11 CASIA-FA test results in terms of EER (%) at different Scenarios:(1) low quality, (2) normal quality and (3) high-quality (4) warped photo attacks, (5) cut photo attacks, (6) video attack, and (7) overall test.

Scenarios / Features	1	2	3	4	5	6	7
LBP-baseline	16.5	17.2	23.4	25.1	17.6	26.7	25.0
CoALBP(grey)	16	15.2	14.6	13.7	14.6	17.3	14.9
TCoALBP(grey)	9.7	**8.1**	8.9	10.3	9.1	8.4	8.69
TCoALBP(RGB)	**5.7**	7.3	**6.6**	**8.1**	**6.9**	**7.1**	**6.71**

Table 4.12 Replay-Attack test result in terms of EER (%) and HTER (%)

	EER (%)	**HTER (%)**
LBP^{U2}	17.9	13.7
CoALBP(grey)	12.9	16.7
CoALBP(RGB)	6.2	8.0
TCoALBP(grey)	2.4	5.7
TCoALBP(RGB)	**0.1**	**0.6**

Table 4.13 Comparisons with the state-of-the-art.

	CASIA-FA (EER %)	**Replay-Attack (HTER %)**	**MSU MFSD (EER%)**
CoALBP(RGB)	11.1	8.0	17.7
DMD[139]	21.8	3.8	N/A
Motion-meg[140]	14.4	**0.0**	N/A
LBP-TOP[43]	10.6	N/A	N/A
LDP-TOP	8.9	1.7	N/A
CNN[67]	1.1	0.8	N/A
Multi-scale LBP(RGB)	10.7	5.1	11.7
Proposed method	6.71	0.6	10.07

Tables 4.11 and 4.12 provide the results of the grey-scale TCoALBP descriptor (TCoALBP (grey)), the RGB channel concatenated TCoALBP descriptor (TCoALBP (RGB)), the grey-scale CoALBP (CoALBP (grey)) [150], and the grey-scale LBP[122]. All of these descriptors use fine-tuned parameters for improving performance. The grey-scale LBP and CoALBP (grey) are provided as baseline results for comparison. For some static local texture descriptors, colour channels are believed to provide more information than the grey-scale image [32]. Thus, we design the TCoALBP (RGB), which divide the video cube into separate RBG colour channels in order to compute TCoALBP on different channels independently and concatenate the resulting feature vectors from different colour channels before classification.

Table 4.11 also shows the performance results of different scenarios in CASIA-FASD [122]. Results presented in Table 4.11 and 4.12 suggest that the TCoALBP features can significantly improve the system performance by combining temporal information and static texture information. Comparing the results of the original LBP [24], [122] and TCoALBP (RGB), the proposed method shows 65.2% performance improvements for the CASIA-FASD, and 92.7% improvements by combining

temporal information and static texture information. Also, TCoALBP (RGB) shows an improved performance on the CASIA-FASD by 41.6% respectively compared with the grey-scale CoALBP.

Table 4.13 shows the results of the comparison between the proposed method and some state-of-the-art methods, which contains the best results for the dynamic features attempting to use temporal information: DMD[139], Motion-meg[140], and LBP-TOP [43]. The proposed method shows very competitive results on the challenging CASIA-FASD [122] and the Replay-Attack database[27]. Some approaches included in Table 4.13 report better results than the proposed method for some of the datasets. However, the proposed approach outperforms these methods in other datasets. For instance, a CNN-based method [67] is reported to have a very competitive result for the CASIA-FASD dataset. However, the proposed method produces better results than that which is reported in [67] for the Replay-Attack dataset[27].

The effectiveness of different temporal texture representations was studied by extracting TCoALBP features from grey-channel image sequences as well as RGB colour channel image sequences. On CASIA-FASD [122], the result of TCoALBP (RGB) feature reaches the state-of-the-art level. Furthermore, in the intra-database evaluation, TCoALBP (RGB) feature shows very promising generalisation capabilities. The TCoALBP algorithm requires the optimisation of several parameters for different datasets to reach the best performance. Also, the inclusion of colour information did not result in a significant performance improvement in the experiments.

4.4.3 Super pixel-LBP for PAD

This thesis also provides a novel workflow named super-pixel LBP for PAD which consists of a super-pixel extractor and a local texture descriptor for detecting PA. This proposed method is inspired by a widely used pre-processing method: cropping facial area to an $n \times n$ sub-area. By following this motivation, the proposed workflow combines some existing algorithms to produce a novel feature representation for PAD and the effectiveness of this proposed feature is evaluated by using multiple benchmark datasets for the comparisons with the-state-of-the-art methods. The good

results demonstrate the effectiveness of the proposed method. In the following sub-sections, the motivation about the proposed super-pixel LBP algorithm is provided firstly. Then, the literature related with the super-pixel algorithm will be explored and the details for the proposed super-pixel LBP feature will then be provided.

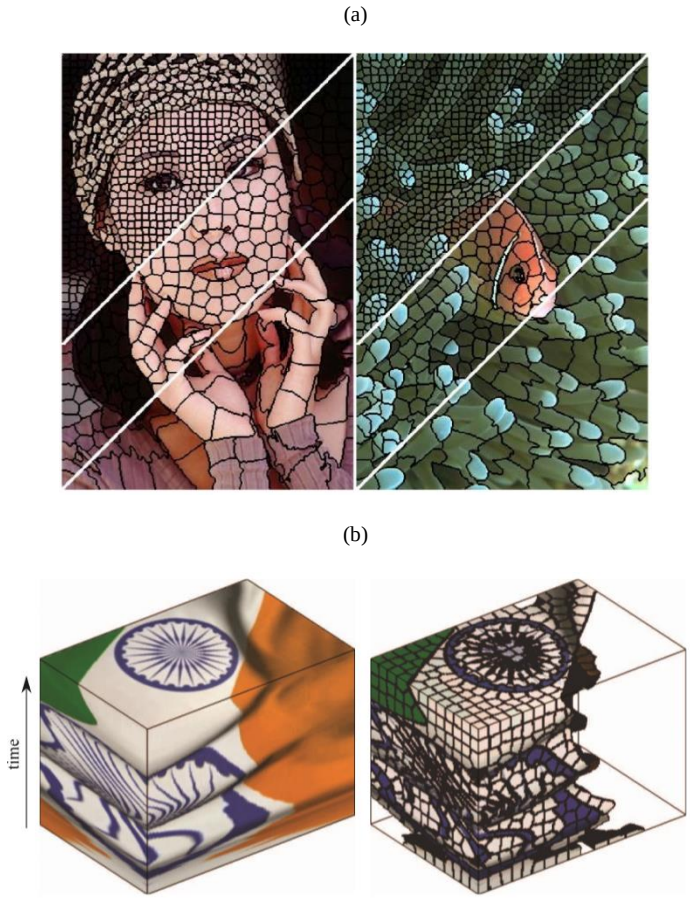


Figure 4.6 Examples of super-pixel segmentation by using SLIC algorithm: (a)The image segmentation example with size 64, 256, and 1,024 pixels,(b) video segmentation(3D) segmentation example with size 64 [151]

Motivation and a brief review for super-pixel algorithms

The concept of the super-pixel is a kind of pixel-grid which is firstly introduced by Ren, X. and Malik, J. [152] at 2003, which is designed as an computationally efficient perceptually meaningful underlying representation for the frame input. The super-pixel algorithm can re-organise the raw input frames into various over-segmented pixel grids (or named as pixel groups) by following the nearest neighbour principle. Fig 4.6 provides an example of using super-pixel algorithm to segment the input image and videos. Generally, the raw input data will be considered as a group of

pixels which include some atomic regions and can be used as meaningful underlying representation. The initial super-pixel only considered the neighbours at two dimension and then some researchers extended their method for the video data which is also named as super-voxels segmentation [151] Fig. 4.6 is an example for the 3D super-pixel segmentation from the paper [151].

(a) (b)

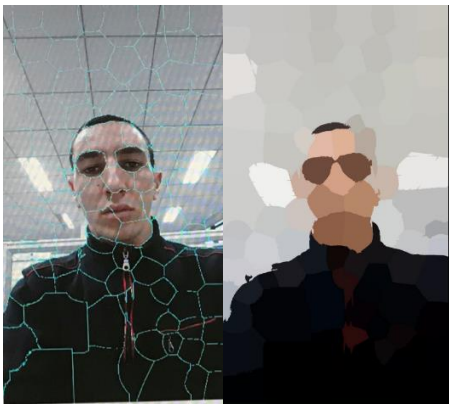


Figure 4.7 Examples of super-pixel segmentation for PAD. (a) is the original frame with the segmentation boundary (b) is the super-pixel segmentation and visualised with the clustering centre.

As the description in Chapter 3 and previous sections in Chapter 4, researchers consider an pre-processing algorithm which crops facial area into $N \times N$ blocks to help the feature descriptors focusing on the local texture representation (such as moiré pattern) rather than the facial information. By observing the example of video attacks, the distinct texture characteristics of artefact will not affect the representation of the object boundary where the object boundary can be considered as the representation of the non-related information for facial structures.

Inspired by this pre-processing step, the proposed work needs an algorithm which can also divide the facial area into various blocks to abandon the non-related facial structure information. The super-pixel method is one of the possible ways to reach these requirements which is generally considered as an unsupervised method to produce image segmentation. Figure 4.7 shows an example of applying super-pixel method to the PAD data.

Ren, X. and Malik, J. [152] firstly introduced the concept of super-pixel for the computer vision research. The existing methods to produce super-pixel can be broadly categorized as the graph-based methods and the gradient ascent methods by following

the suggestions of Achanta, et al. [151]. The graph-based algorithms consider the pixels in the raw data as the nodes of a graph where the similarity score between neighbouring pixels is represented as the weight of the edge in the graph. For instance, Veksler et al. [153] produced a graph based super pixel algorithm by fusing overlapping image patches in the graph model. The compact super pixel generation is named as GCa10 in their work. The gradient based methods produce a pixel cluster and refine this cluster iteratively. For instance, Achanta, et al. [151] represent each pixel as a 5-dimensional position and move the cluster centres to the lowest gradient position. By following the guide of the gradient, the cluster will close to the optimum position iteratively. There are various algorithms to produce super-pixel in the literatures but the proposed work only needs a simple and efficient algorithm to produce super-pixels. Thus, the proposed work selects one of the famous super-pixel algorithms named Simple Linear Iterative Clustering(SLIC) [151] due to the computational efficiency and the robustness at different benchmark datasets. Figure 4.7 provides an example of applying SLIC algorithm to an sample of the video attack from the OULU dataset[125].

Another important part of the proposed method is the Bag-of-Words (BoW) model which was successfully applied for various computer vision tasks [154]. The basic idea of the BoW algorithm is to produce a visual vocabulary which is generated by clustering the local features. The chaotic feature vector set is encoded to an intermediate representation and this representation can be fed into classifier such as SVM for any classification tasks. Various works are published to improve the performance of the BoW method. For instance, different clustering algorithms are considered such as mean-shift [155], K-means[156], etc. The proposed work considers the BoW model as a simple way to mapping a set of super-pixel LBP descriptors to the final feature vector to make the performance better for PAD.

Methodology

The proposed super-pixel LBP method consists of 4 important parts: super-pixel segmentation, local texture descriptors extraction for each segmented super-pixel, codebook for BoW method and classification. Figure 4.8 visualises this proposed workflow as a block diagram. There are two steps in the proposed workflow. For the first step, the most important part is to produce a codebook for the training dataset. In this step, frames selected from the whole training dataset with all possible categories

are firstly used to generate a set of super-pixel LBP features. Then, the mean shift clustering method is used to generate a codebook for BoW algorithm [157]. At the second step, the codebook generator is used to produce the final feature vector for classifier. The proposed novel workflow combines different traditional features by following the motivation and the details about this workflow are proposed in the following paragraphs.

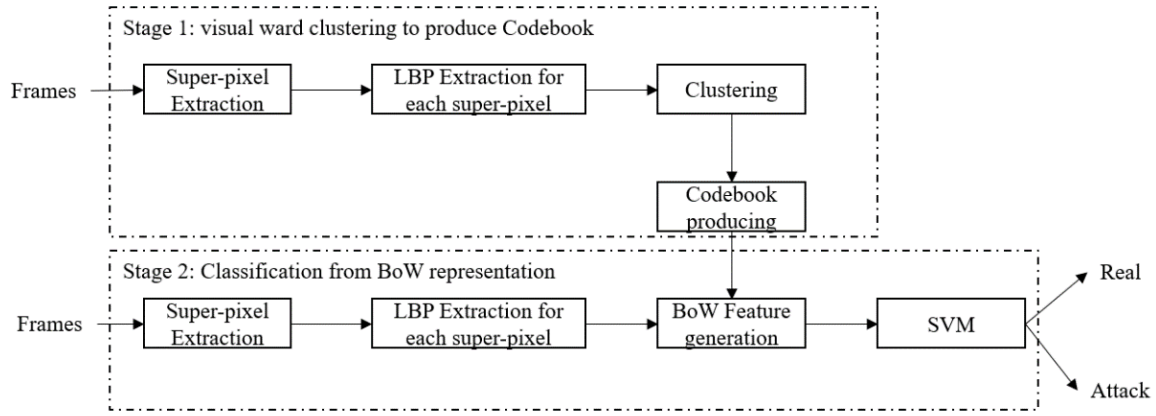


Figure 4.8 Super-pixel Local Binary Pattern for PAD workflow

In order to get the final feature, the proposed method firstly applied a widely used super-pixel algorithm named simple linear iterative clustering (SLIC) [151] for an efficient super-pixel segmentation. For any input image, SLIC algorithm is initialised with k cluster centres. These clusters are updated by using residual error E between the new cluster centre locations and previous cluster centre locations. The update step repeats iteratively to reach the converges points and the implementation from [151]suggests the iterative number should be no more than 10. The proposed work flow considers the maximum iteration number as 5 in the following experiments to decrease the computational complexity.

After the super-pixel segmentation, the raw data will be divided into a set of super-pixels. The clustering centre of the super-pixels are considered as the centre of rectangle grid which is used to generate LBP feature descriptors. Each rectangle is considered as the same length $\mathrm{c} = S - \mu$. To make it simple, the proposed experiments set $\mu = 1$. The LBP feature extractor is applied for each of these rectangles to get a set of feature descriptor for the input data. LBP is widely used for texture description and has good performance in texture classification which is robust with illumination changes and position changes. Thus, the frame input will be transferred into a set of

LBP descriptor with low computational complexity. The proposed workflow can also consider the frame sequence as the input data to measure the temporal difference between real faces and presentation attacks by simply change 2D SLIC and 2DLBP to 3D version. For 3D SLIC[151], $D_s = d_{lab} + \frac{m}{S} d_{xyz}$ where z represents the temporal direction. The superpixel center is used to locate a pixel cube with length $c = S - \mu$ .And the 3D LBP can be calculated by following the suggestion from [158] to get a set of texture descriptors for the next step.

The combination of super-pixel method and LBP descriptor will generate a set of feature representation. Concatenating these feature representations directly will cause the curse of dimensionality. For this reason, the proposed method considers the BoW method to mapping the feature set to a fixed length feature vector for classification. Firstly, the extracted feature sets F for the data samples, which are generated from the combination of super-pixel method and LBP algorithm, are collected together to get $F = \{F_n\}$ for clustering. The mean shift clustering algorithm is then applied for this feature set to get a codebook $C = \{C_{cl}\}_{cl=1}^{cl\_num}$ which consists of the most typical features. These typical features are represented as the Cluster centres and the number of the cluster centres is noticed by cl_num. The proposed workflow considers two colour spaces (RGB and HSV space) to generate two codebooks C_{RGB} and C_{HSV} . By using these two codebooks, the distinct colour difference for PA artefacts will be emphasized by following the suggestions from[32].

The BoW feature extraction uses the codebooks generated with clustering algorithm to get the proposed feature for classification. When training the classifier, the input data will be firstly fed into the super-pixel extraction block. Then, a set of LBP descriptors will be calculated for each rectangle which is centred by the super-pixel clustering centre. The codeword will be generated from the LBP descriptor set by using the nearest Euclidean distance and the appearance times of different codewords is used to generate a histogram. The classifier is then trained to detect spoofing attack by using this histogram as the final feature vector.

Experiment and evaluation

To assess the effectiveness of the proposed facial spoofing detection method two presentation attack detection datasets are used: the CASIA-FASD dataset [27] and the Replay-Attack dataset[27].In the experiment, a SVM with the RBF kernel is used as

the classifier. SLIC algorithm simply use the recommended parameter of [158] and set the initial *k*=30 for 2D super-pixel and *k*=40 for 3D super-pixel. Also, LBP algorithm will set *P*=8 and *R*=1 as default parameters. The number of cluster centres cl_num is fixed to 500 to avoid the dimensional curse. These experiments use four-fold subject-disjoint cross-validation using the CASIA-FASD training set due to the absence of a development subset for this dataset. The performance is reported by using the Equal Error Rate (EER) on the test set. The evaluation protocols of the Replay-Attack database require producing the EER on the development set and the Half Total Error Rate (HTER) on the test set[27] .

The proposed implementation also considers some implementation detail to improve the efficiency and accuracy of the detection. Firstly, the Haar face detector and the HoG detection methods are combined to segment the facial area. Two Colour spaces (RGB and HSV) are considered in the proposed experiments. The 2D super-pixel LBP considers the first frame of the input frame sequence as the input data and 3D super-pixel LBP feature considers the following 20 frames from the first frame as the input data.

Table 4.14 Comparison with the state-of-the-art at CASIA-FASD and Replay-Attack DB overall test.

	CASIA-FA (EER)	**Replay-Attack DB (HTER)**
LBP-baseline [24], [122]	25.0	13.7
DMD [139]	21.8	3.8
MotionMeg [140]	14.4	0
LBP-TOP [43]	10.6	N/A
Proposed 2D Super-pixel LBP (RGB)	7.4	5.5
Proposed 2D Super-pixel LBP (RGB+HSV)	6.6	4.1
Proposed 3D Super-pixel LBP (RGB+HSV)	5.2	2.7

Table 4.14 shows the results of the comparison between the proposed method and some state-of-the-art methods, which contains the best results for the dynamic features attempting to use temporal information: DMD[139], Motion-meg[140], and LBP-TOP [43]. The proposed comparison considers LBP [24], [122] as the baseline of the selected dataset to demonstrate the performance improvements of the proposed

workflow. Then, the DMD [139] Motion-meg [140] and LBP-TOP [43] methods are considered to demonstrate the effectiveness of the proposed methods as a temporal related feature.

Table 4.14 includes three results for the proposed workflow. The proposed 2D super-pixle LBP (RGB) and proposed 2D super-pixle LBP (RGB+ HSV) demonstrate the proposed workflow by only considering 2D SLIC and 2D LBP in the feature extraction part. The proposed 3D super-pixel LBP consider 3D super-pixel segmentation and 3D LBP [158] for code word generation. From this table, the proposed method shows very competitive results on the challenging CASIA-FASD [122] and the Replay-Attack database[27]. The 3D super-pixel LBP gives better performance at both datasets. The performance difference between the proposed 2D super-pixel LBP (RGB) and the Proposed 2D super-pixel LBP (RGB+ HSV) demonstrate the effect of combining different colour spaces. Thus, the proposed 3D super-pixel LBP directly apply two colour channels for better performances.

4.5 SUMMARY

In this chapter, we present four features for PAD based on conventional pattern recognition approaches and focused on the incorporation of temporal information for PAD.

Table 4.15 Performance of the proposed features for multiple dataset

Datasets (EER (%))	REPLAY ATTACK	CASIA FASD	MSU MFSD
LBP-baseline[24], [122]	16.10	24.80	14.70
LBP-TOP[43]	7.9*	10.00	N/A
Colour-LBP [32]	0.40	3.20	3.50
FAUH	12.90	21.10	N/A
MHI-LBP	3.90	**4.80**	N/A
MHI-CNN	4.50	6.00	N/A
TCoALBP	**0.60**	6.71	10.07
2D super-pixel LBP	4.10	6.60	**7.67**
3D super-pixel LBP	6.70	5.20	9.20

The performances of the proposed methods are very encouraging when following the intra-test protocols on the three benchmark datasets. Table 4.15 provides a comparison for the various proposed methods. The performance is represented with the Equal Error Rate to make the table clear. The local binary pattern is selected as the baseline algorithm at various benchmark datasets, and Table 4.15 also considers this algorithm as the baseline method to demonstrate the performance improvements of the proposed methods.

The LBP-TOP [43] is considered a representative traditional feature using temporal information for PAD. The original LBP-TOP was only evaluated with CASIA-FASD dataset[122], and the proposed experiment provide the evaluation results with Replay-Attack dataset[27] by following the implementation detail of the original paper. The Colour LBP[32] was published at the end of 2016 andwhich represented the best results with multiple datasets at that time. The proposed traditional features were investigated and evaluated the same year. The Colour LBP is a static feature and the proposed methods focus on the temporal information for PAD. By comparing them with the Colour LBP[32], the table demonstrates that all of the proposed traditional methods have some encouraging results when compared with the state-of-the-art methods.

In Table 4.15, the highlighted performance score is the best score when only considering the proposed traditional methods. The TCoALBP method as a temporal feature represented best results at Replay-Attack dataset[27] while the MHI-LBP feature demonstrated the best results with CASIA-FASD[122]. And the 2D super-pixel LBP showed the best results with MSU-MFSD[123].

The contributions in this chapter include the following：

（1） The novel FAUH feature is constructed based on FACS which provides a common symbolic description system for PA. The feature is based on high-level semantic information and shows potential as a new type of temporal-based feature for PAD.

（2） The proposed new MHP constructs MHIs as a description for temporal texture changes and uses LBP to produce feature vectors. The MHP offers a method to consider temporal texture changes with a low computational complexity and it can be considered as a new framework for temporal-

based PAD. This method approaches the performance of state-of-theart when evaluated using multiple data sets.

（3） A novel feature for biometric presentation attack detection is proposed using temporal texture co-occurrence in a video sequence of facial images. The effectiveness of different temporal texture representations was studied by extracting TCoALBP features from grey-channel image sequences as well as RGB colour channel image sequences. Extensive experiments showed good results on three challenging spoofing detection databases. On the CASIA-FASD, the result of TCoALBP (RGB) feature reaches the state-of-the-art level. Furthermore, in the intra-database evaluation, TCoALBP (RGB) feature shows very promising generalisation capabilities.

（4） A novel feature named super-pixel descriptor, which produces super-pixel for spatial information and super-voxel for spatiotemporal information, is designed for PAD task. The super-pixel extraction method, or its 3D extension named super-voxel, is used as the primary feature in the proposed workflow to get the clustered pixel group. LBP and BoW algorithm are used as the secondary feature to generate the final feature vector for classification. Some encouraging results in the benchmark datasets show the potential of the proposed workflow.

Chapter 5: Deep Learning Approaches for PAD

In this chapter, some facial spoofing attack detection methods based on deep learning approaches are explored. In Chapter 2 and Chapter 4, we covered various traditional hand-crafted features for PAD. Some progress has been made by introducing novel hand-crafted features and promising results have been obtained. However, the performance of such systems, trained with a particular dataset, will significantly drop when tested with a different dataset, even when the same attack type is used. To overcome such performance limitations, deep learning techniques have been gaining ground for PAD. This chapter is concerned with developing new approaches that apply deep learning techniques for PAD.

Section 5.1 introduces the motivations for the set of experiments that follow. Section 5.2 introduces a DNN architecture that can process temporal information for PAD. Section 5.4 introduces two methods for applying such a DNN architecture for PAD. Then Section 5.3 then applies some visualisation techniques to gain more insight on the operation of the DNN. The results from these studies are then used for the following work in Chapter 6.

5.1 Motivation

Deep learning is an emerging and rapidly developing branch of machine learning that learns the representation of the target data by stacking multiple processing layers.[104]. Traditional machine learning techniques require considerable efforts of careful design and complicated engineering for the feature extractor that can transform the raw data (such as the pixel values of frames of a video) into an appropriate internal representation (the feature vector) for classification or detection. However, these human-designed internal representations are seldom fully invariant to the range of possible environmental conditions of the input data (such as illumination or camera changes). Deep learning offers a possible way to overcome the limitations of such traditional approaches to feature extraction.

Presentation attack detection research has followed the supervised learning paradigm by training a learning system that can produce an output as a vector of

classification scores by minimising training errors. Traditional PAD algorithms have relied on sophisticated hyper-parameter optimisation to maximise their performance for particular evaluation datasets. Some drops in performances are therefore expected when applying a traditional PAD method to different diatasets without hyper-parameter tuning. The hyper-parameters would need to be selected to adapt the algorithm for different environmental and application conditions. This requirement for adapting the algorithms for each application is an impediment to the wider adoption of biometric systems.

The rise of deep learning has dramatically changed the PAD research by learning suitable intermediate representations from the training data.

(1) Deep Learning can learn good features automatically, but designing a good deep architecture is difficult. Researchers can train a novel neural architecture from scratch by following a common procedure rather than using specialist domain knowledge [67] [104]. But designing such a deep neural architecture requires expertise in of deep neural networks.

(2) Deep Transfer Learning can easily use the implicit knowledge from a source domain with insufficient training data in the target domain. But the selection of the feature extraction subnetwork from different pre-trained deep neural networks is a difficult problem. For instance, VGG16 [47] network, which is trained using the ImageNet dataset[64], could be used for deep transfer learning in PAD. However, the deep transfer learning with VGG16 pre-trained network does not give the best results at various datasets, contrary to researchers' expectations. The reason behind this may be that the VGG16 is not a good neural network for deep transfer learning. Additionally, the reason may be that the researchers did not find a good way of using the pre-trained neural networks for PAD.

(3) DNNs also offer some new ways to process temporal information [72]. However, the deep neural network is computationally expensive, and the existing methods, which use Recurrent Neural Networks (RNN) for temporal information, may not be the best way of processing temporal information.

Several recent papers have claimed to produce good performance in PAD using DNN based methods. This chapter focuses on exploring the main factors that influence the performance of a DNN-based PAD. The deep transfer learning paradigm is first

introduced and some widely used pre-trained neural networks are considered as robust feature extraction methods to demonstrate the potential of the DNN-based methods. The performance scores from experiments using the deep transfer learning paradigm are also considered as the baseline for comparison with subsequent proposed methods. Section 5.2. presents pre-trained feature extractors which have good performance and generalisation capability. These DNN-based PAD algorithms, however, do not provide the best performance in all cases and demonstrate that more efforts may be needed to improve the design of deep neural architectures for PAD.

Some visualisation algorithms and justification generation algorithms are explored to analyse the inner mechanisms behind the behaviour of PAD systems generated by deep transfer learning. Generally, deep learning is a black-box algorithm. Here, some basic visualisation examples are first explored in Section 5.3.1. Then, some "interpretable visualisation" is provided for analysing the basic principle of design for a neural architecture for PAD.

Some ideas that are used in the conventional features inChapter 4, are also used to design deep neural architectures in this chapter. Section 5.4.1 provides a neural architecture for the temporal intensity signal from facial action unit system to detect PAs. And the motion textures are modelled by using a patch-based CNN, which is inspired by a commonly used pre-processing step: dividing the facial area into $M \times M$ sub-blocks. The 3D CNN is considered for the patches sequence and the small temporal texture difference could be recognised by using 3D CNN.

5.2 Convolutional neural network for PAD

Deep learning is a popular method for representation learning which has a very strong dependence on the large scale of training data. The distinct performance improvement by using deep learning methods relies on the capability of learning latent patterns. Various widely considered PAD datasets, which are collected for testing and evaluating the conventional features, can hardly be used to train a deep neural architecture from scratch directly. And the limited training data will significant increase the overfitting risk for a deep neural network. This section will provide two possible solution for this problem: (1) deep transfer learning and (2) training a neural architecture from scratch with less trainable parameters.

The deep transfer learning paradigms is explored in literature which demonstrated that a pre-trained CNN could be transferred to PAD without much fine-tuning [146]. The proposed deep transfer learning experiments introduce some latest pre-trained neural architectures for transfer learning paradigm and evaluate the proposed deep transfer learning methods at various benchmark datasets.

"Training a neural network from scratch", which means researchers design and train a neural network without transfer learning, is another popular way of using deep learning methods. However, the deep neural networks are "data hunger". The proposed architecture should include less trainable parameters than the normal deep neural networks to decrease the risk of overfitting. For this reason, the proposed Colour Convolutional Presentation Attack Detection Network (CCPADNet) considers the network structure of MobileNet [57], which introduces two modified convolutional operators to decrease the volume of trainable parameters, and a Colour sub-network, which can be trained independently with PAD dataset. This Colour sub-network aims to learn a colour space transfer function for spoofing detection. And the global average pooling and residual connections are introduced to increase the performance.

Some basic concepts of deep learning techniques are introduced in the section in order that the details in the following experiments are demonstrated clearly. A novel neural architecture for PAD is proposed to achieve better performance. To simplify the problem, PAD is considered as a binary classification problem in this section. In subsequent sections, this constraint will be relaxed. Both of these proposed methods are evaluated at 5 widely used datasets and data argumentation is considered as a pre-processing step to enlarge the volume of training data.

5.2.1 Deep Transfer Learning for PAD (DTL-PAD)

The limited volume of datasets will restrict the performance of the deep neural networks[146]. And the insufficient training data is becoming a distinct problem for training a deep architecture to detect presentation attacks. However, collecting training data for PAD is expensive and complex. Thus, the proposed method follows [159] and considers multiple pre-trained neural networks for the deep transfer learning paradigm to detect facial presentation attacks.

Deep transfer learning relaxes the independent and identically distributed (iid) assumption and allows a pre-trained neural network to be quickly transferred for a different task. [146] .

To make the description clear, the domain $\mathcal{D}$, which consists of the probability distribution $P(X)$ and the feature space $\mathcal{X}$, is defined as a set for a supervised learning task $\mathcal{T} = \{y, f(x)\}$. In this description, $X=\{x_1, x_2, \dots, x_n\} \in \mathcal{X}$ denotes a subset of $\mathcal{X}$ and $\mathcal{T}$ aims to learn a mapping function which represent the hidden conditional distribution $P(y|x)$.

Transfer learning paradigm considers two domains: target domain $\mathcal{D}_t$ and source domain $\mathcal{D}_s$. Given a learning task $\mathcal{T}_t$ on the target domain $\mathcal{D}_t$, transfer learning paradigm aims to achieve a good predictive performance for function $f_{\mathcal{T}}(.)$ on the learning the task $\mathcal{T}_t$ by transferring latent knowledge from $\mathcal{D}_s$.($\mathcal{D}_s \neq \mathcal{D}_t$ and/or $\mathcal{T}_s \neq \mathcal{T}_t$) In the most case of the deep transfer learning, the size of $\mathcal{D}_s$ is much larger than the size of $\mathcal{D}_t$.

By following these definitions, deep transfer learning can be described as a tuple $\langle \mathcal{D}_s, \mathcal{T}_s, \mathcal{D}_t, \mathcal{T}_t, f_{\mathcal{T}}(.) \rangle$ with 5 elements, where $f_{\mathcal{T}}(.)$ is a non-linear mapping function that deep neural networks aims to learn. Tan, C. et al. [146] give a name (Network-based Deep Transfer Learning) for the paradigm which reuses the partial network that has been pre-trained in the source domain and transfer the partial network as a part of the DNN that is proposed for the target domain. The network structures and trained parameters should be both included in this process.

In the experiments reported in this chapter, the source task is general image classification task $\mathcal{T}_{IMG}$ for the image classification domain $\mathcal{D}_{IMG}$. The presentation attack detection is the target domain $\mathcal{D}_{PAD}$with the target task $\mathcal{T}_{PAD}$. The proposed experiments follow the definition of the network-based deep transfer learning and consider the task as $\langle \mathcal{D}_{IMG}, \mathcal{T}_{IMG}, \mathcal{D}_{PAD}, \mathcal{T}_{PAD}, f_{\mathcal{T}}(.) \rangle$.

The network for deep transfer learning consist of two parts: (1) the front part is the feature encoder sub-network (the language-independent feature transform [160]) and (2) the back layers form the classifier. The feature encoder sub-network, which is trained using large datasets such as ImageNet[64], is reused to compute intermediate feature representation in this paradigm.

Generally, the transfer learning paradigm needs a learned mapping function $f_S(.)$ which is trained for the task $\mathcal{T}_s$ in source domain $\mathcal{D}_s$ by using a large volume of training data. Then, the feature encoder sub-network of $f_S(.)$ is connected with a new classifier network to learn a new mapping function $f_{\mathcal{T}}(.)$ at the target domain. The training time for the new mapping function at the targeted domain is significantly decreased due to the latent knowledge in the feature encoder sub-network of $f_S(.)$. For this reason, the investigation reported in this thesis started with exploring deep transfer learning-based PAD. [146]

The feature encoder sub-network for computer vision tasks consists of convolution layers from the pre-trained networks; and sometimes it includes one flatten layer [161] to reform the output of convolutional layers as a feature vector. The classification network, which sometimes only consists of one or two fully connected (FC) layers, is similar with the Neural Network classifier in some papers exploring the conventional features.

Selecting good pre-trained networks is an important mission for the transfer learning paradigm especially when applying transfer learning for PAD. Yosinski, J., et al. [162] suggest that some weights in the pre-trained neural network may not influence in-domain accuracy but influence the transferability. They suggest that LeNet, AlexNet, VGG, Inception, ResNet are identified as good choices in network-based deep transfer learning.

Some works [163] from model compression area suggest these popular neural networks include some "redundant weight" and compress these weights will not affect the performance on the testing set. However, Yosinski, J., et al. [162] also point out that the redundancy of weights is suitable for the transfer learning. The proposed experiments explore the feature encoder sub-network from three widely used deep neural network (VGG-16[47], ResNet50[49], and NAS-large networks[10]) by following the suggestion from Yosinski, J., et al [162]. The following experiments choose VGG-16 network as feature encoder sub-network due to its good transferability.

After loading the pre-trained model, each frame from PAD evaluation dataset is fed into the pre-processing pipeline: (1) The facial area in each frame was detected and cropped as the first step. (2) The facial landmark data provided by the dataset (if

applicable) are used in this step to avoid possible detection errors. (3) Then, the cropped facial area is normalised and resized to fit the input size of the pre-trained network. For instance, VGG16 pre-trained network requires the form of the input data to be 224*224*3 pixels. The resized input data is normalised to be within the range [0,1] by following the suggestion of TensorFlow platform [164]. In the PAD dataset, different frame sequences may have different number of frames. The proposed experiments randomly select some frames from each frame sequence by using the random model provided by Python 2.7 to decrease the similarity of training data and speed up the feature extraction and classification.

The Classifier Network have same neural architecture, which include two fully connected layers, for the proposed deep transfer learning experiments. However, the size of features from different pre-trained neural networks are different. In order to provide a useful comparison of these feature encoder networks, the proposed transfer learning experiments resize these feature tensors to same shape to fit the input configuration of Classifier Network. For instance, the length of the feature vector will be different when it is generated by different pre-trained feature encoder networks. The proposed method selects a fixed length and uses the additional pooling layer [165] to adjust the feature vectors from different encoder networks.

The proposed DTL-PAD algorithm follows the same experimental setup. The whole network is trained for 200 epochs in total. In the first 50 epochs, the trainable parameters in the feature extraction network is fixed. The trainable parameters within the classification network are optimised by using a SGD algorithm [200], which is initialized with an initial learning rate of 0.03. Then, the learning rate will down to 1×10^{-3} by following a cosine schedule. The following fine-tuning stage takes 150 epochs with a small learning rate (1×10^{-5}). The input data will use Viola-Jones face detector from OpenCV [112] in the pre-processing stage. And all of the input data should be normalised to range [0,1].

Table 5.1 provides the performance about the proposed DTL-PAD and the state-of-the-art methods of deep learning for PAD. Yang-Net [67] is the first work which considers convolutional neural architecture to detect PAD. Lucena. et al. [145] also consider transfer learning paradigm and use the feature extraction part of the pre-trained VGG16 network in their FASNet. The proposed DTL-PAD (VGG16) gives

better performance when comparing with FASNet. The reason may be the fine-tuning stage, which is suggested by [162], can improve the system performance. The

Table 5.1 Performance of the Deep Transfer Learning for PAD at multiple datasets

Datasets (EER%)	NUAA	REPLAY ATTACK	CASIA-FASD	MSU-MFSD	HKBU MARs	Rose-Youtu
FAS-Net [145]	N/A	10.0	N/A	N/A	N/A	N/A
Yang-Net [67]	N/A	2.4	5.0	N/A	N/A	N/A
DTL-PAD (VGG16)	3.6	8.4	7.1	16.0	39.7	15.4
DTL-PAD (ResNet)	4.9	5.7	6.3	11.4	33.1	14.8
DTL-PAD (NAS)	2.5	9.4	8.0	14.3	35.0	18.5

proposed DTL-PAD (ResNet) and DTL-PAD (NAS) demonstrate better performance in some dataset. However, the transfer learning based approaches not represent better performance than the Yang-Net [67] which is a neural architecture designed for PAD.

By analysing these results, there are two points, which may be important for the following experiments: (1) A neural architecture, which is designed and trained for PAD, may represent better performance than only using the transfer learning paradigm. (2) Some works may need to be done to transferring a pre-trained neural network for PAD with better performance.

5.2.2 Colour Convolutional Presentation Attack Detection Network (CCPAD-Net)

From the previous results, designing some novel neural architectures for PAD is attractive for researchers due to the performance improvements and lower computational costs. Presentation Attacks have some distinct characteristics which may need researchers to design some novel neural architecture (such as colour space differences, motion texture differences, etc.). The proposed Colour Convolutional Presentation Attack Detection Network (CCPAD-Net) aims to get better performance by designing a novel neural architecture for the colour differences.

Methodology

The proposed network consists of three main sub-networks: (1) Colour Space network (2) Feature Encoder network and (3) Classifier Net. The Colour Space Projection network aims to generate a mapping function which can transfer the input

image to a specified colour space. The Feature Encoder network aims to produce a feature vector that can represent the distinct texture difference for presentation attack

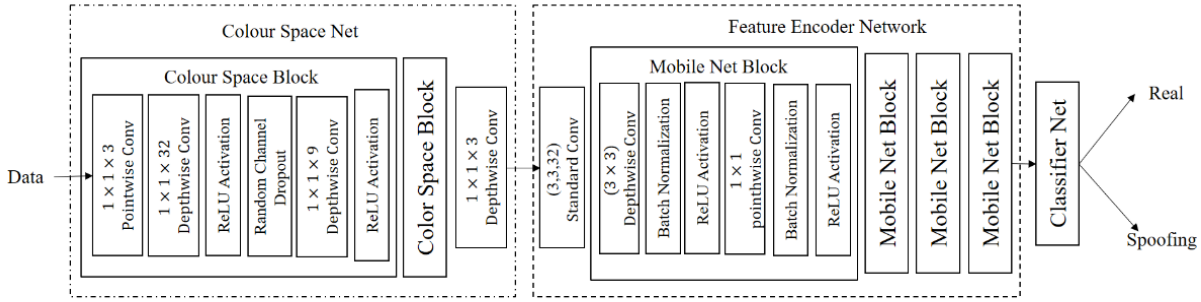


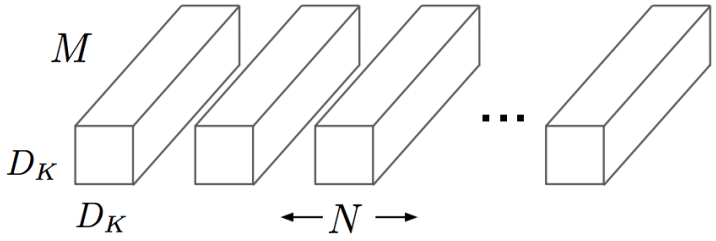
Figure 5.1 The architecture overview of the proposed CCPAD-Net

detection. And the Classifier network is used to generate the final decision. The overview of this proposed algorithm can be found at Fig. 5.1.

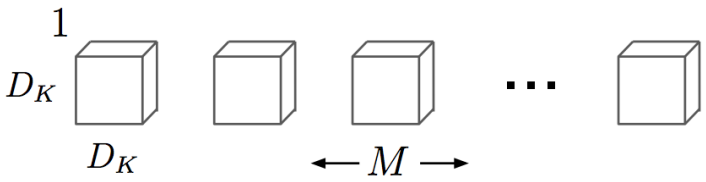
In the proposed pipeline, the RGB face image is fed into the Colour Space Network which maps the input image into a learned colour space. Then, the output of the Colour Space Network is fed into a standard convolutional layer to generate the intermediate feature maps. After that, the intermediate feature maps are sent into the Feature Encoder Network which consist of 4 convolutional blocks. Each of these blocks follows the suggestions of [57] to generate the discriminative feature vector for classification. Finally, the feature vector is fed into the Classification Net which consists of an average pooling layer [165], and a fully connected layer with Softmax activation function [166] to produce the predict label for the current data input.

The proposed Colour Space network aims to learn a mapping function to project the raw input data into a colour space where the spoofing attack will represent distinct difference. The input of Colour Space network is the raw data and the output of the Colour Space Network is the 3-channel feature map which has the same width and height as the original frame. Colour space transformation function should calculate the transferred colour representation in pixel wise. Thus, the proposed neural architecture considers depth-wise convolution and point-wise convolution [57] rather than the original convolutional operator. Both of these convolution methods are visualised in Figure 5.2.

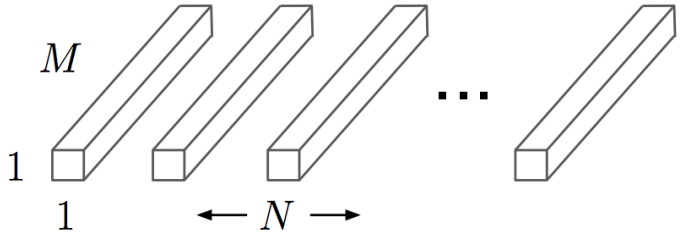
In the proposed Colour Space Blocks, the point-wise convolution operator [57] is firstly applied to merge the pixel value from different colour channels. Then, the depth-wise separable convolutions [57] are followed, and use various filters to factorize the output of point-wise convolution layer into different channels. The point-



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 5.2 Different convolution methods. From top to bottom, (a) illustrates the standard convolution method, (b) shows the depth-wise convolution method, and (c) shows the point-wise convolution method. [57]

wise convolution layer and depth-wise convolution layer can highly decrease the computational complexity and the volume of the trainable parameters.

A standard convolutional layer takes an input feature map **F** with shape $D_F \times D_F \times M$ and the output feature map **G** of this standard convolutional layer is a $D_G \times D_G \times N$. Here, D_F is the spatial width and height of the filter size and the following description only consider the square input feature map to make the

description simple and understandable. And D_G is the size of output feature map. M indicates the number of input channels (input depth) and N is used to represent the number of output channel (output depth).

The standard convolutional layer is parameterized by a convolution kernel $\boldsymbol{K}$ of size $D_K \times D_K \times M \times N$ where $D_K$is the spatial dimension of the kernel assumed to be square. The output feature map for standard convolution, assuming stride one pixel for input with padding [167] is computed as formula(5.1) [57]:

$$\boldsymbol{G}_{k,l,n} = \sum_{i,j,m} \boldsymbol{K}_{i,j,m,n} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \tag{5.1}$$

Standard convolutions have the computational cost of:

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{5.2}$$

where the computational cost depends multiplicatively on the number of input channels M, the number of output channels N the kernel size $D_K \times D_K$ and the feature map size $D_F \times D_F$.

Depth-wise separable convolution is considered to break the interaction between the number of output channels and the size of the kernel. The original convolution operation combines features to produce the representation for next layer. The filtering and combination steps in the original convolution operation can be divided into two steps by using depth-wise separable convolutions. The computational cost will be reduced by following the suggestion of [57].

Depth-wise separable convolutions consist of two layers: depth-wise convolutions and point-wise convolutions. The depth-wise convolutions only consider a single filter for each input channel. A simple1×1convolution, which is named as Point-wise convolution, is then used to generate a linear combination of the output of the depth-wise layer. Depth-wise convolution with one filter per input channel (input depth) can be written as

$$\widehat{\boldsymbol{G}}_{k,l,n} = \sum_{i,j,m} \widehat{\boldsymbol{K}}_{i,j,m,n} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \tag{5.3}$$

where $\widehat{\boldsymbol{K}}$ is the depth-wise convolutional kernel of size$D_K \times D_K \times M$ where the *m*-th filter in $\widehat{\boldsymbol{K}}$ is applied to the *m*-th channel in **F** to produce the *m*-th channel of the filtered output feature map $\widehat{\boldsymbol{G}}$.

To decrease the computational cost is another important reason to integrate the depth-wise separable convolution. The computational cost of the depth-wise separable convolutions can be represented by

$$D_K \times D_K \times M \times D_F \times D_F \quad (5.4)$$

In [57] , they demonstrate between 8 to 9 times less computation than standard convolutions for only a small reduction in accuracy.

The proposed neural architecture, which contains 32 convolution filters with 1×1 filter size in Colour Space network, uses depth-wise convolution and point wise convolution in a different way. The depth-wise convolution operation is used to produce non-linear projection for single colour channel and the point-wise convolution operation is used to fuse multiple channels. This Colour Space network is firstly trained by a generated dataset.

The generated dataset is designed for learning a function to transfer RGB channels into some colour channels that is meaningful for PAD. Z Boulkenafet et al. [32] used Chi-square distance to measure the similarity of texture patterns in different colour channels. They suggested that the textures in the Cb channel of the YCbCr colour space are discriminative for the video attack and the texture pattern in the Cr channel is representative for the printed attack. They also reported that the best performance score is achieved by combining the HSV colour space and the YCbCr colour space. From their work, the Y channel in the YCbCr colour space has a similar meaning to the Value or Brightness channel in the HSV colour space. Thus, the proposed workflow generates a separate training set for the Colour Space network. This dataset considers the original RGB frames from the training dataset as the input. The label of this datasets is the combination of some colour channels generated from the original frame. The proposed method selects S channel from HSV colour space; Cb and Cr channels from the YCbCr colour space in the following experiments. By using this generated dataset, the Colour Space Network can be trained separately.

The proposed method also considers Convolutional blocks from MobileNet [57] to decrease the total number of trainable parameters. The Feature Extraction part of the proposed CCPAD-Net method consists of one original convolutional layer and 4 MobileNet Blocks to extract the feature representation.

Each convolution layer is followed by a batch normalisation layer [168] to decrease the risk of overfitting. The ReLU[52] nonlinear activation function is used in the proposed neural architecture with the exception of the final fully-connected layer which uses SoftMax activation function [166] for classification.

The skip-connections, following ResNet [49], are also used in the Feature extraction part in the proposed neural architecture. ResNets add a skip-connection and the gradient can flow directly through the skip-connection from later layers to the earlier layers. The neural architecture for PAD may also need this feature to bypass some layers and feed the discriminative texture representations from low level layers directly into the high-level layers. The visualisation experiments in section 5.3.2 also show this phenomenon.

Experiment

Data augmentation methods are applied in this experiment by following the descriptions in Chapter 3. The quality and volume of the training data determine the performance of the DNN-based methods. In the absence of sufficient training data, data augmentation may be necessary for training a deep neural network from scratch to detect presentation attacks. The extra training dataset ,which is generated for training the Colour Space Network, can also be considered as a part of data argumentation processing.

Extensive experiments were conducted on 6 challenging datasets and the results for the proposed CCPAD-Net is listed in the Table 5.2. Three methods are compared with the proposed methods: (1)Colour LBP [32] also considers colour space and uses a widely used conventional texture feature descriptor as the secondary feature to generate the detection results. (2) Yang, et al. [67] and Li, et al. [103] design novel neural architectures and train their neural network from scratch by using PAD dataset. The proposed experiment also explores a hybrid architecture which combine the proposed Colour Space Net and the feature extraction part of the VGG16 pre-trained neural network[47].

Table 5.2 Performance of the CNN as baseline feature for multiple datasets

Datasets (EER%)	**NUAA**	**REPLAY-ATTACK**	**CASIA-FASD**	**MSU-MFSD**	**HKBU MARs**	**Rose-Youtu**
Colour LBP[32]	N/A	0.0	3.2	3.5	N/A	N/A

Yang-Net [67]	N/A	4.32	6.25	N/A	N/A	N/A
3DCNN [103]	N/A	0.3	1.4	0.0	N/A	7.0
Colour Space Net (VGG16)	0.3	**0.6**	1.5	3.2	**19.7**	**7.9**
CCPAD-Net	**0.1**	0.8	**1.1**	**2.7**	20.5	8.3

From this table, the proposed Colour Space Net(VGG16) and CCPAD-Net demonstrate better performance than the Colour LBP method[32] which shows the effectiveness of the deep learning based methods. The proposed experiments represent better results than Yang, J. et al. [67]'s work which also explores 2D CNN and train their neural network from scratch by using PAD dataset. These results demonstrate the effectiveness of the proposed neural architectures. Some results from Li, et al. [103]'work shows better results in some datasets. However, their work consider temporal information by using 3D CNN and a complicate "model generalisation" step to improve their performance. This suggests that future work for applying DNN to detect PA needs to pay more attention on design of the network, which can fit the special needs of presentation attack detection, rather than simply use the transfer learning paradigm.

5.3 VISUALISATION AND ANALYSIS OF DNN-BASED PAD

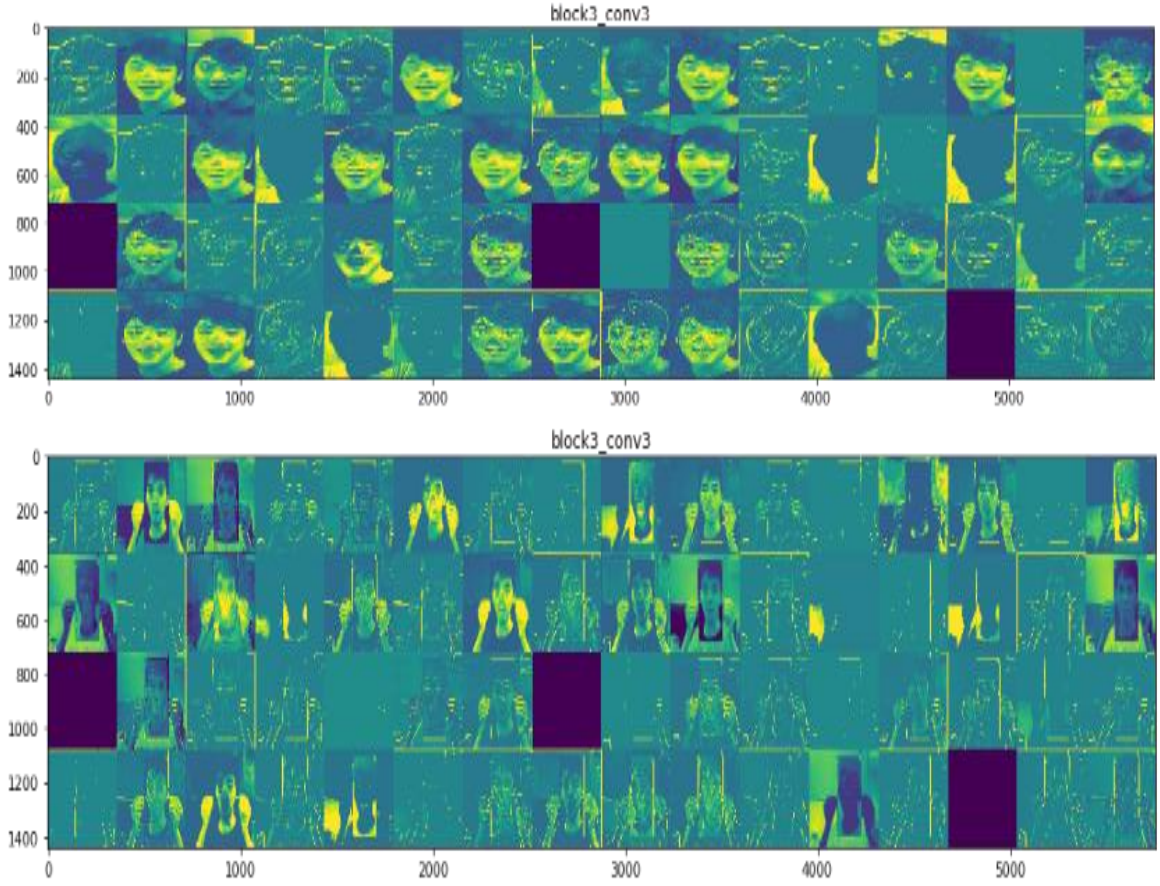
After exploring the DTL-PAD and CCPAD-Net methods, there are two questions naturally comes out: (1) What latent knowledge, that is learned from the training data, can be considered as the distinct difference between genuine and attacks? (2) Which region in the input raw data is important for the current decision? To answer these two questions, this section firstly provides some visualisation for the deep learning based PAD architecture. Then, some interpretable algorithms from explainable artificial intelligence (XAI) are explored to provide deeper understanding for the inner mechanism of the deep learning based PAD methods. The proposed experiments visualise the intermediate representation within CNN structures to explain the spatial or temporal correlations behind the decisions. The study in this chapter will guide the development of the new systems presented in Chapter 6.

5.3.1 Basic visualisation

The proposed visualisation experiments start with simple visualisation methods: visualise the activation of different convolutional blocks in DTL-PAD(VGG16). The

visualisation results are presented as visual images in Fig. 5.3. By following the suggestions in [50] , the initial visualisation experiment is to visualise the intermediate output from each convolutional block.

Fig. 5.3 shows the results of visualisations for different convolutional blocks. The feature extraction subnetwork for VGG16 architecture[47] consists of 5 convolutional blocks. The initial visualisation experiment selects block 3 to demonstrate the difference between genuine face and attacks in the perspective of VGG16. Block 3 in the VGG16 network is the middle convolutional block. Some researchers[169] claimed that the lower level of neural layers will focus on the texture level features and the higher level will focus on the object level. Selecting the middle



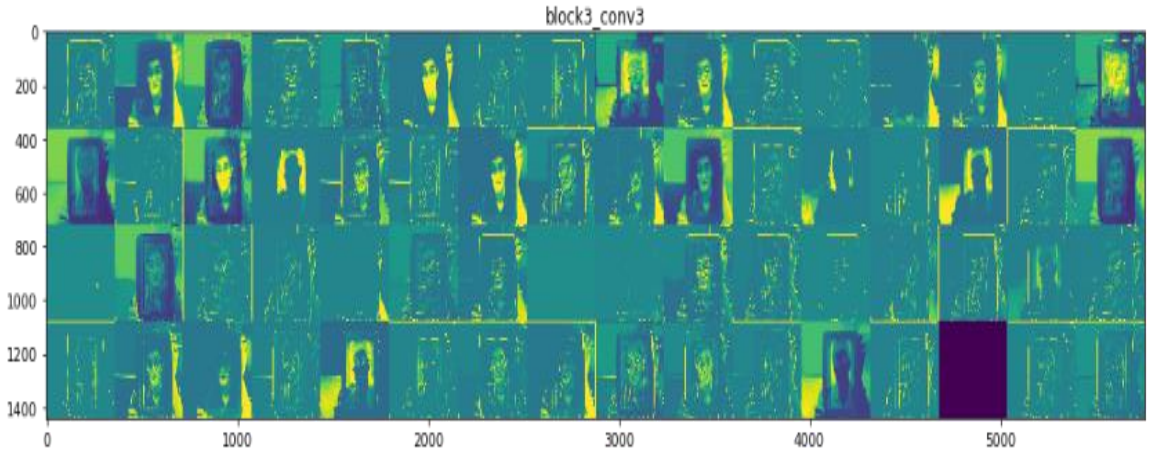


Figure 5.3 Visualization of DNN layer's response map for different spoofing attack types.(From top to bottom, the visualisation of Block 3 convolutional 3 layer at VGG 16 for real access, the paper attack, and the video attack visualisations.)

convolutional block for visualisation aims to show the activation difference in the texture level, which is represented as different pixel intensity value in activation image, and to show the visible difference in the object level for human eyes. Fig 5.3 considers the output from real face, paper attack, and video attack. And the same position, which is represented as row *x* and column *y*, at (a), (b) and (c), indicate the output for different attack types from same filter.

From Fig 5.3, it can be easily noticed that even lower-level convolutional filters have some significant response areas where the spoofing artefacts are different from real faces. It means some texture difference is very significant for detecting presentation attacks. Some filters will represent high activation value (high pixel intensity value in the image) in some region. For instance, the results from the filter (row 1, column 2 at (a) (b) and (c)) shows some regions with high activation values for real face texture but relatively low activation values for paper attack and video attack. These outputs from convolutional layers represent some semantic meaning for PAD. They also illustrate the potential of DNN for spoofing detection.

However, this simple visualising is not enough for the deep learning based methods. By exploring the literature of interpretable artificial intelligence, we realised that the interpretation of existing neural network may be very important for developing new and better deep architectures. From the explanations provided by some interpretable algorithms, researchers can understand the problem in a new way: based on perspectives of neural networks which are learned from large datasets. Deep

learning methods learns the latent model from large volumes of data which may not accessible for human beings. The new insights bring from these explanations could inform and support future research.

The interpretation capability may be much more important for biometric research and application than other areas. Generally, people need some reasons when the decision can highly affect their life. For example, explanations of treatment decisions between patients and doctors are often considered necessary; however the treatment decisions produced by using deep learning system can hardly be explained [170].

As a security technology, a biometric system may need not only to provide the decision but also the reason behind this decision to convince its users. When the wrong rejection or wrong acceptance has occurred, system managers need the reason behind this failure to optimise the system. It may indeed be necessary to know not only the decision made regarding genuine and fake presentations from the system but also the reason behind it. For instance, E-payment applications, based on facial biometric technologies are now widely used. Each wrong accept decision may lead to a significant property loss to the user. The explanations of the system can help people to identify who have the responsibility to compensate this property loss as the "black box" in the airplane.

Moreover, in some security scenarios, an explainable algorithm can improve their performance immediately by simply introducing human experts into the processing loop to make the final decision [171]. It can further decrease the risk of false acceptance and false rejection. When the system can provide some reasons for generating each important decision, the human experts involved in the daily maintenance of the system can quickly provide a decision based on their knowledge about trusting the system or not. Thus, the wrong decision from the system can easily be corrected by human experts. A black-box system cannot do this. In the proposed architecture, a natural language generator was added to provide more understandable explanations.

5.3.2 Interpretable visualisation

From the initial visualisation experiment, some regions with high pixel intensity value may have semantic meaning which are reported as conventional features (such

as gamut, texture, foreground-background, and so on). These visualised activation map from the intermediate convolutional blocks could partially explain the relatively good generalisation capability of deep neural networks. However, the initial visualisation experiment still cannot answer the question like: Which region in the input raw data is important for the current decision?

One of the important reasons is that researchers cannot locate areas that are significant for PAD at the pixel level. For instance, researchers cannot connect the final results to a low-level texture pattern and/or high-level object parts. For these reasons, two visual explanation algorithms are explored and applied for the PAD problem in the proposed experiment to visually highlight the reason behind decisions made by deep-learning based PAD systems. Some analysis will also be provided to answer the question such as: whether the deep learning based PAD system correctly identify the location of the PAD artefacts or signatures in the image.

Partial Oculus Sensitivity Map

Partial Oculus Sensitivity Map [172] show the spatial importance of the input data for the current decision by systematically occluding different portions of the input image, and monitoring the difference for the decision output. It is a simple but efficient method which can clearly demonstrate the spatial importance for a particular region model. Especially, the output probability score will drops significantly when the important object is occluded.

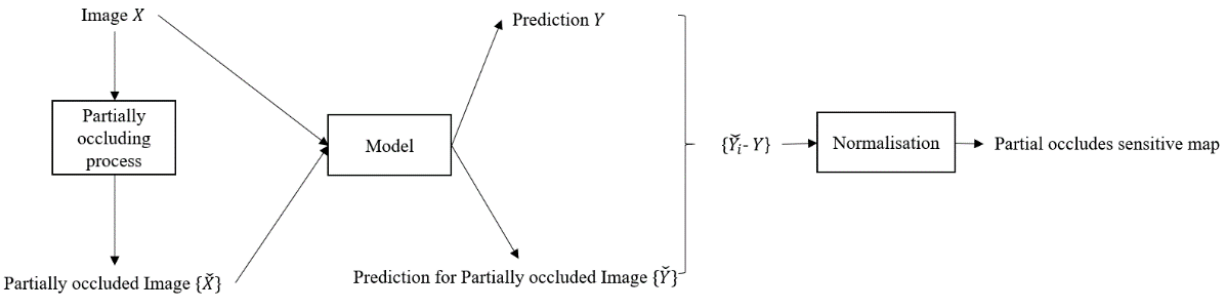


Figure 5.4 Workflow for the partial occluded sensitivity map

Fig 5.4 visualises the workflow of producing a Partial Oculus Sensitivity Map. For an input image ,the model will provide a probability prediction Y. The partially occluding process is then used to generate the partially occluded images $\{\check{X}\}$. The

difference between the probability output for X and $\{\check{X}\}$ will then be calculated by using $\{\check{Y}_i - Y\}$ where $\check{Y}_i$ indicates the probability output for$\{\check{X}\}$ generated by the model. Then, a normalisation function is applied to normalise the probability difference to the range [0,1] as the Partial Oculus Sensitivity Map.

The partial oculus process firstly divides each frame into an $A \times A$ blocks. Then, the algorithm selects a block and use a grey block to replace the original image in this region. This replaced image can be named as partially occluded image. The partially occluded images $\{\check{X}\}$ can be calculated by enumerating all possible blocks. The example of the partial oculus processing and the example of the Partial Oculus Sensitivity Map can be found at Fig 5.5.

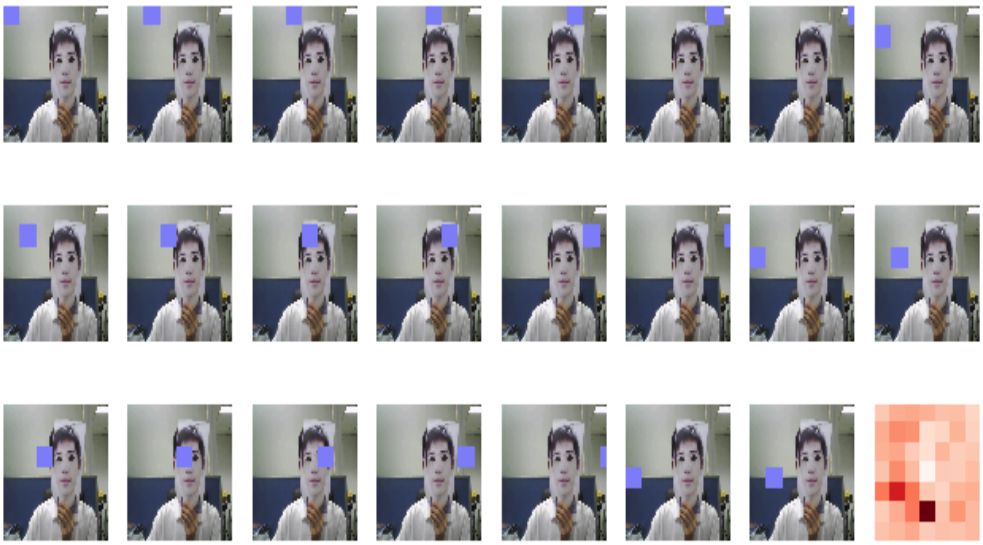


Figure 5.5 Visualization of partial occluded and heatmap example

GRAD-CAM

Another visualisation algorithm, which is explored to provide the visual explanation for the current decision from the deep learning based PAD system, is named as Gradient-weighted Class Activation Mapping (Grad-CAM) [173]. Grad-CAM also aims to provide a spatial importance visualisation map but uses the gradient from the deep neural network to calculate. The Grad-CAM is a "class discriminative localization map" which is calculated for one category each time. Normally, researchers only calculate Grad-CAM for the ground truth category. For a Grad-CAM $L^c_{Grad-CAM} \in \mathbb{R}^{u \times v}$ of width u and height v, the prediction of the score for class c, y^c,

and the feature maps A^k of a convolutional layer are used to calculate the gradient flow $\frac{\partial y^c}{\partial A^k}$.

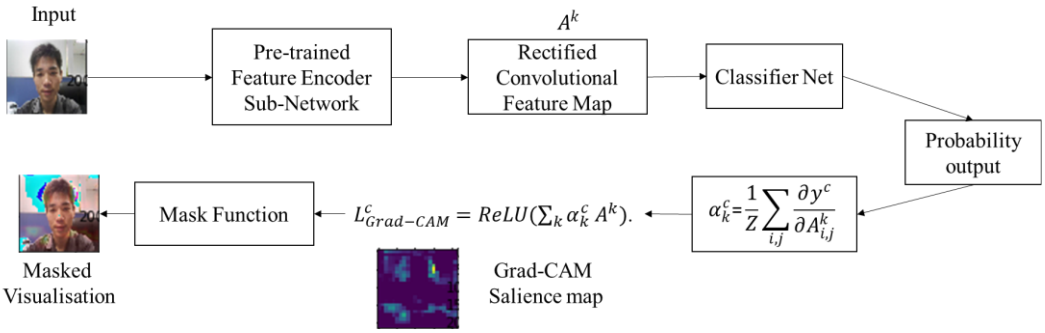
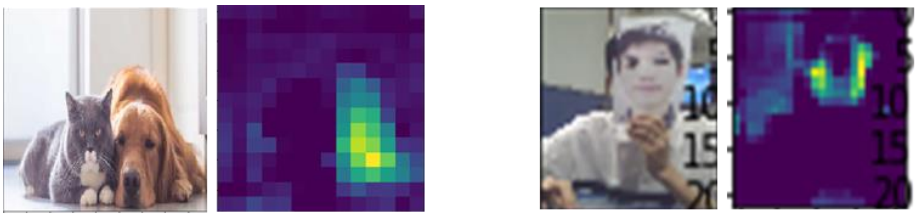


Figure 5.6 Workflow for the Grad-CAM saliency map

This gradient flow can be calculated by using the global average-pooling [165]to obtain the neuron importance weights $\alpha_k^c = \frac{1}{Z}\sum_{i,j}\frac{\partial y^c}{\partial A_{i,j}^k}$ by following the suggestion of [174]. This neural importance weight α_k^c can be understood as the partial linearization of the deep network in Selvaraju, R. R., et al [173]'s work to get the "importance" of the feature map k for the ground truth category. The weighted combination of forward



(a) Grad-CAM for dog detection

(b) Grad-CAM for PA detection

Figure 5.7 The Grad-CAM saliency map for object detection and for PAD

activation map which is followed by a ReLU activation function[52] to obtain the Grad-CAM map as $L^c_{Grad-CAM} = ReLU(\sum_k \alpha_k^c A^k)$. The result of this algorithm is a coarse heat-map of the same size as the convolutional feature maps (14 × 14 in the case of last convolutional layers of VGG). The proposed experiments only interested in the features that have positive influence on the ground truth. And the input frames in the following visualisation experiment are selected from CASIA-FASD [122].

Visualisation Results

Figure 5.8 shows the results of using two visualisation approaches for the different categories on the examples from CASIA-FASD [122]. In the proposed

experiment, the feature extraction part of the VGG16 network[47], which consists of 5 convolutional blocks, is selected as the pre-trained feature extraction sub-network. In the proposed saliency maps, the brighter colour demonstrates the importance of the spatial location. The generated saliency maps are enlarged by using linear interpolation method [175] to fit the size of the original input image. Some open source codes for Grad-CAM are used in the proposed visualisation experiments, which are released by Selvaraju, R. R., et al. [173] on the torch platform [176]. The original source code for Grad-CAM requires a separate training stage and this training stage can be replaced by simply calculating the gradient for the additional global average pooling layer at the classifier network[174]. The proposed experiment follows the suggestions of Chattopadhay, et al. [174] to accelerate the Grad-CAM method. To generate the masked frame, a generated saliency map is firstly normalised to range [0,1] by divide the maximum value within this saliency map. Then, the normalised saliency map is zoomed to [0,255] to fit the valid pixel intensity value. The proposed experiment uses applyColorMap() function from OpenCV[177] to transfer the generated saliency maps into 3 channels and the masked image is visualised by using the default function provided by Keras [178].

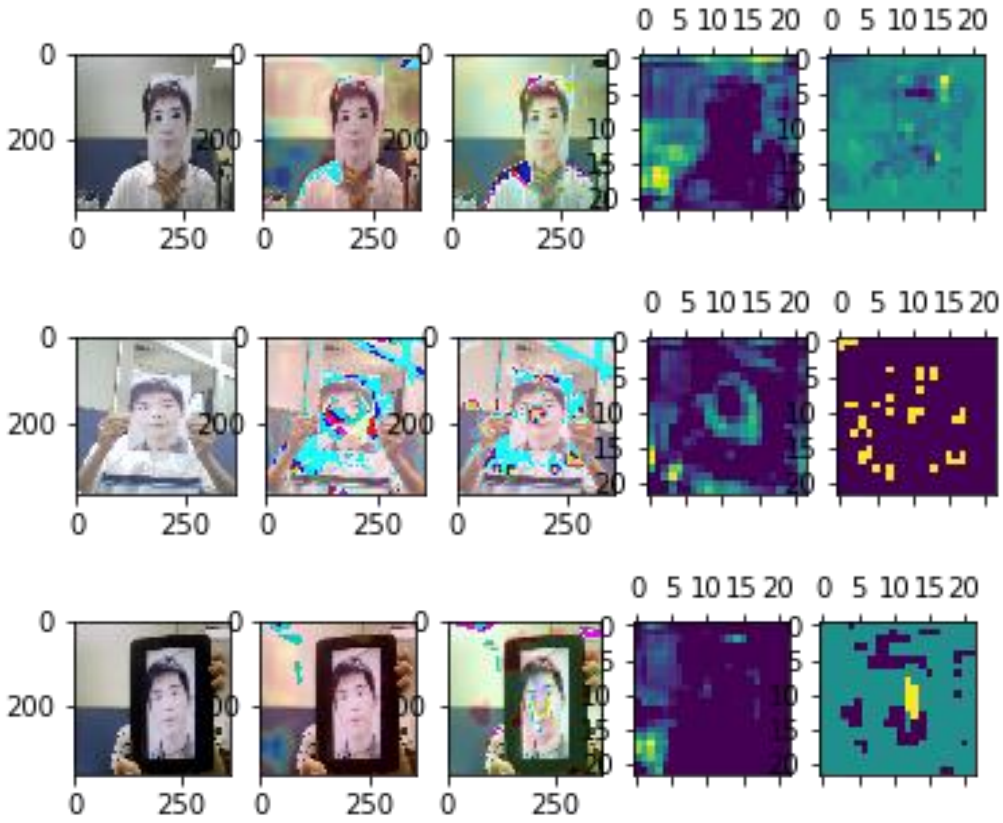


Figure 5.8 Visualization of grad-CAM salience map and partial oculus salience map. Each row of this figure is two masked visualisation and two salience map visualisations for different attack types of the CASIA-FA dataset. (From top to bottom, the category name is real, paper attack, cut paper attack, video attack.) From right to left, different columns represent different visualisations (the sequence is original frame, grad-CAM soft masked frame, partial occluded soft masked frame, grad-CAM salience map, partial occluded salience map).

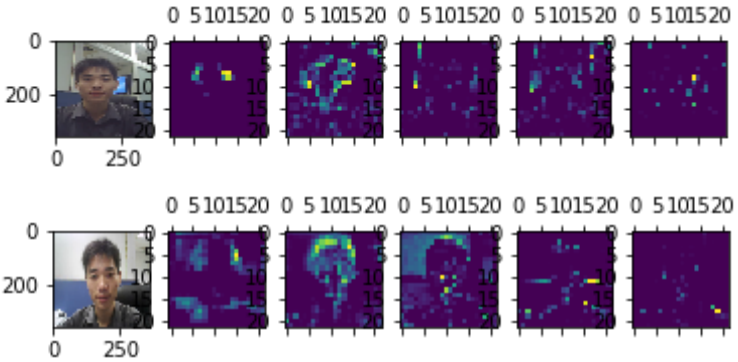
From Figure 5.8, the numerical difference makes the saliency maps generated by different algorithms looks different. Meanwhile, the visualisation for masked image demonstrates that different visualisation algorithms may consider the same spatial location. It can be considered as the evidence about that the deep neural network "focus" on some particular spatial location. And the texture patterns in these particular spatial locations may be considered as the most suspicious representation, or the most evidential feature representation for presentation attacks. For this reason, the justifications generated by grad-CAM algorithm[174] could be used, as some additional information from a computational efficiency approach, to train the neural network for further improving the performance of the neural networks.

The hierarchical structure of deep neural networks is believed as a possible reason of its efficiency[104]. In VGG16 network[47], some convolutional layers and

pooling layers [165]are considered as a “block” of convolutional layers. To demonstrate the relationship between the intermediate results of different convolutional blocks and final decisions, the proposed experiment visualise the important spatial locations generated by different convolutional blocks. In this case, the algorithm of generating the Partial Oculus Sensitivity Maps cannot fit the requirement, which need the algorithm demonstrate the relationship between the intermediate output and the final output. The visualisation results for different convolutional blocks by using grad-CAM method[174] is presented at figure 5.8. And the results from different categories is listed at different rows.

The visualisation results at figure 5.9 only show the original frame and the saliency maps generated for different convolutional blocks. (From right to left, different column represents different layers (the sequence is block1_conv2, block2_conv2, block3_conv3, block4_conv3, block5_conv3 in VGG-16)). The brightness of the saliency map indicates the importance of the spatial location for the ground truth category. The word “focus”, in the following descriptions, will be used to demonstrate the spatial location which have high brightness value in the generated saliency map.

From this visualisation, different convolutional blocks may “focus” on different spatial locations. This phenomenon may suggest two points: (1) different convolution blocks may “focus” on the texture patterns at different semantical levels as the visual cortex in human brains[104]. (2) The presentation attacks, as some literatures mentioned [20], include different distinct feature representations at both texture level and object level.



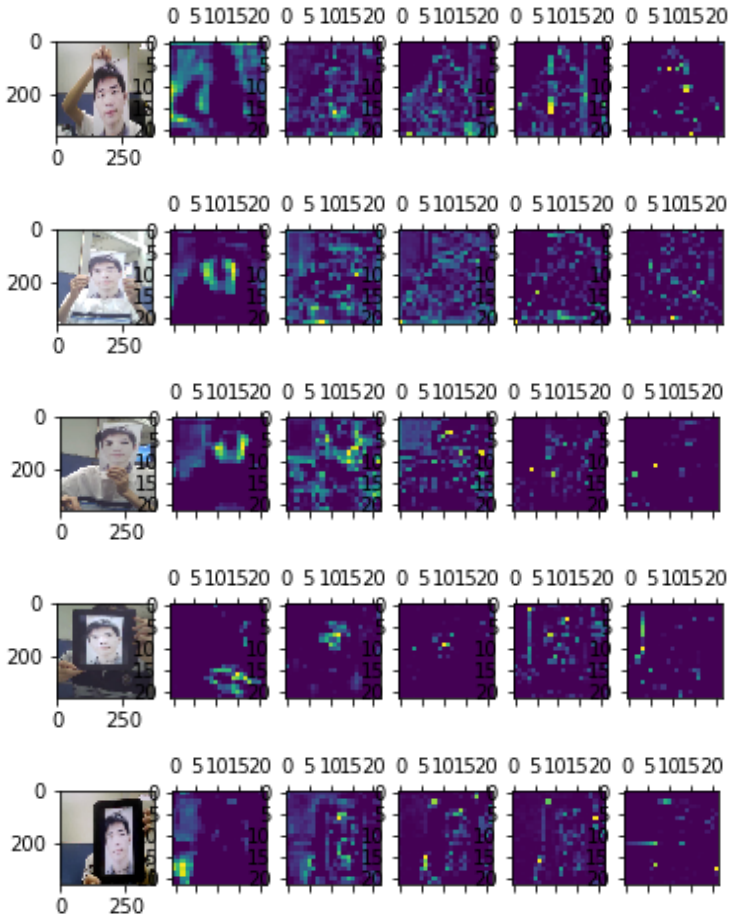


Figure 5.9 Visualization of grad-CAM with different depth of VGG-16. Each row of Fig 5 is a grad-CAM heatmap visualisation for different types. (From top to bottom, the category name is real (low quality), real(middle quality), paper(low quality), paper(middle quality), cut paper, video attack(low quality), video attack(middle quality),) From right to left, different column represent different layers (the sequence is block1_conv2, block2_conv2, block3_conv3, block4_conv3, block5_conv3 in VGG-16)

This visualisation results may partially explained the results shown at Table 5.1. The DTL-PAD (ResNet) and DTL-PAD(NAS) both include the residual connections [49], which include the Residual learning block, as the shortcut connections with gating functions, between two convolutional blocks. The residual connections in DTL-PAD (ResNet) and DTL-PAD(NAS) may help the neural networks use the information at both texture level and the object level for PAD.

Even in the VGG16, the networks may also “focused” on both texture level and the object level. The block 5, which is the last convolutional block in VGG16 before the classification layers, is considered to “focus” on some interesting regions in the proposed experiments. Researchers normally believes that the “focused” region in the

last convolutional layer may include some discriminative representation for the classification[104]. By visualising the results in figure 5.8 and figure 5.9, the deep neural network for PAD not only "focus" on the PA instruments but also consider some regions from background. Some examples (such as the saliency map at the second column and last row in figure 5.9) are focusing on some area that has pure white or grey colour in the background. A possible reason is that the region with pure colour may decrease the probability of detecting the texture representations such as the moiré pattern.

Some objects (such as human hands, the black band of iPad, etc) are also "focused" by the deep neural network which is similar with some assumptions from earlier conventional features[19]. These background-based anti-spoofing research has not been actively followed up in recent years and the deep neural networks seems learn some similar "knowledge" from the training data automatically[19]. The "focusing" regions, that include some significant object parts(such as the black edge of an iPad used for photo/video replay) , may be the reason behind the increased robustness for the deep neural network when different image qualities are included in the testing dataset.

However, the deep neural network is not perfect. The predicted spoofing detection probability score is expected to show a significantly drop when the facial area is blocked. The proposed experiment considers $\varphi = \frac{1}{M}\sum((P_{normal} - P_{blocked})/P_{normal})$ as an index to show the influence for blocking the face, where P_{normal} indicates the predicted probability output from deep neural network for the original frame; the $P_{blocked}$ means the output for the frames which replace the detected facial region with a grey square. *M* is the total number of the image samples. If the original prediction P_{normal} =0.9 for the ground truth category, and the $P_{blocked}$ =0.1, the probability score should significantly dropped ($\varphi = 0.89$). However, the φ is 0.47 at CASIA-FASD[122] and φ is 0.53 at Replay-Attack dataset[27]. It means the deep neural networks may consider some regions, which is not within the region of PA instruments, as the support evidence for the decision. For instance, the most important part should be the screen area for the video attack. It may be difficult to believe that this kind of deep neural networks is trustworthy for the security applications.

Moreover, the disadvantage of using the pre-trained feature extraction sub-network from VGG16 may also be obvious from the visualisation experiments: the convolutional blocks from VGG-16 does not include the Residual learning block to help the neural network consider the information from texture level directly.

5.4 NEURAL NETWORKS FOR TEMPORAL INFORMATION PROCESSING

In this section, two novel deep neural architectures for PAD using temporal information as well as spatial information are introduced in detail. Firstly, the neural architecture, which uses facial action units to detect PA, is introduced and named as Facial Action Signal Analysis Network (FASAN). FASAN improves the way of processing temporal facial action unit signals by utilising a recurrent neural network. Then, the Facial Temporal Cube Network, which considers a 3D convolutional neural network to produce the dynamic textures for detecting PA, is presented and evaluated.

5.4.1 Facial Action Signal Analysis Network (FASAN)

In Chapter 4, the idea of using the facial action unit coding system for presentation attack detection was introduced and its potential for PAD was explored. The proposed FAUH performed well when evaluated using two well-known datasets by using a histogram of activation to model the temporal changes. However, the histogram approach discards much of the temporal information available from the facial action unit intensity signals. To overcome this limitation, we analysed the selection of action unit groups to build the FAUH to minimise the effect of loss of information. While this remedy works to some extend it still does not fully utilise all the available information.

With the rising of the DNNs, modelling temporal information by using neural networks has also become possible. One of the commonly used methods is considering the Recurrent Neural Networks (RNNs) which was explored by Hochreiter and Schmidhuber in 1997[179]. Currently, RNNs are a family of neural networks for processing variable-length sequential data. A RNNs maintains a recurrent hidden state matrix, whose activation value at the current time step, is dependent on the previous time step.

Long Short-Term Memory Networks (LSTMs), which is a special kind of RNNs, has been introduced by[179] [180] in order to overcome the problem of long-time

dependency for the original RNN. In the original RNNs, a long input sequences will be difficult to learn due to the problem of vanishing/exploding gradient [181]. The gate mechanism is introduced for this problem and the LSTMs is one of the widely used replacement versions of RNNs.

LSTMs is considered for PAD in some recent works. For instance, Xu et al. [72] integrated a LSTM sub-network into their deep neural architecture to extract features from frame sequence. However, they only considered LSTMs as a simple way to generate a feature vector from the intermediate feature outputs that is generated by convolutional layers. And their proposed neural architecture is computational expensive. The proposed CNN-LSTM architecture also considers LSTM for temporal information. However, the facial action unit intensity signal is a low dimensional signal which can highly decrease the computational costs of using LSTMs.

In the following paragraphs, the proposed neural architecture, which jointly use CNN and LSTMs, is introduced in detail. Then, the LSTM network and facial action unit extractor are briefly introduced. The implementation detail and experimental results are presented at last part.

Methodology

In the proposed workflow, as shown in Fig. 5.10, the image sequence is firstly fed into the Action Unit Detector (AU Detector) to extract the action unit intensity signal AU_k^j as the descriptions in Chapter 4. Then the action unit intensity signal is fed into the Temporal Compress Network (TCN) to generate the temporal feature f_{AU}. The overall sum for the action unit intensity signal is used to select two important frames for extracting spatial information by using convolutional neural networks. Finally, a classification network is used to generate the final decision about whether the input frame sequence includes presentation attacks.

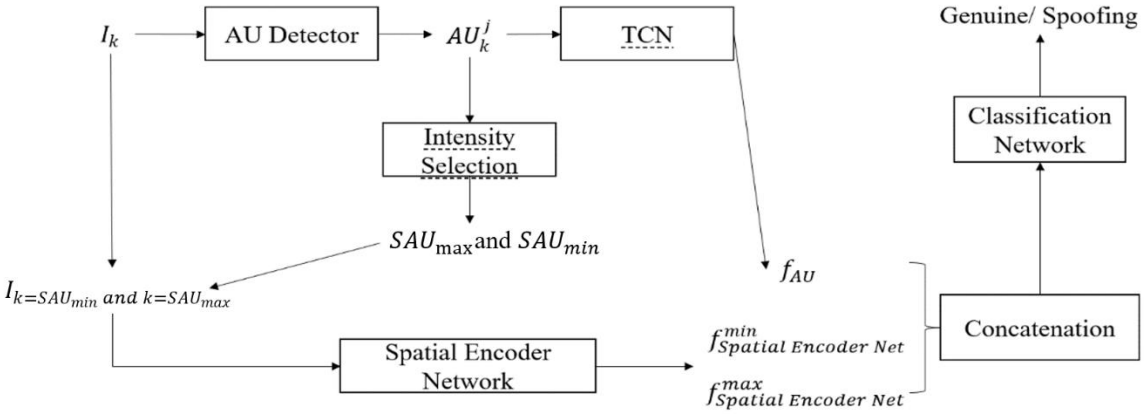


Figure 5.10 FASAN system block diagram

The action unit intensity signals, which is extracted from the input frame sequence, are defined by following the definitions in Chapter 4. Set *G* is the index set of all the AUs, which include *N* elements. *S* is the selected subset of *G* and *j* is any element belonging to *S*. Here, AU_k^j is the intensity value of the *j*-th AU at the *k*-th frame, where *K* is the total number of frames of input video. Then, the facial action unit detector can be used to extract AU_k^j from the *k*-th frame as following:

$$I_k \xrightarrow{AU\ Detector} AU_k^j \tag{5.5}$$

Here, Long Short-Term Memory Networks (LSTMs) is used in the FASAN to replace the simple histogram method in FAUH, which is mentioned in Chapter 4. In FASAN, three fully connected LSTM layers and two dropout layers are used to generate the Temporal Compress Network (TCN) for the action unit intensity signal.

$$AU_k^j \xrightarrow{\text{TCN}} f_{AU} \tag{5.6}$$

The TCN used here consists of the input layer, the two sequence-to-sequence LSTM layers, two dropout layers with a probability of 0.2 and a many-to-one LSTM layer, as shown in Figure 5-11. The dropout layers are used to reduce risk of overfitting. The combination of LSTM and dropout layers are used to learn features from action unit intensity signals to produce a fixed length feature vector f_{AU} for PAD.

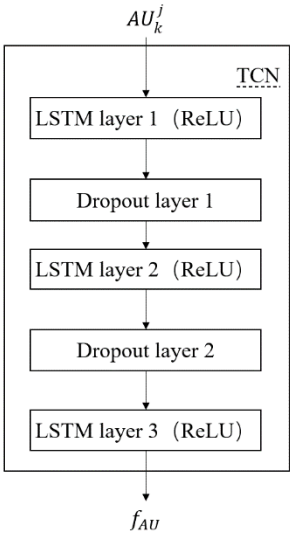


Figure 5.11 TCN architecture.

There are many reasons for selecting LSTM in the proposed TCN architecture. Firstly, LSTM can be optimised by using gradient descent algorithms. The histogram method, which is considered in FAUH, is a conventional feature extraction algorithm, which cannot be optimised when training the classifier net. Secondly, the gate mechanism of LSTM can help the proposed neural architecture learn the important temporal information from facial action unit intensity signals.

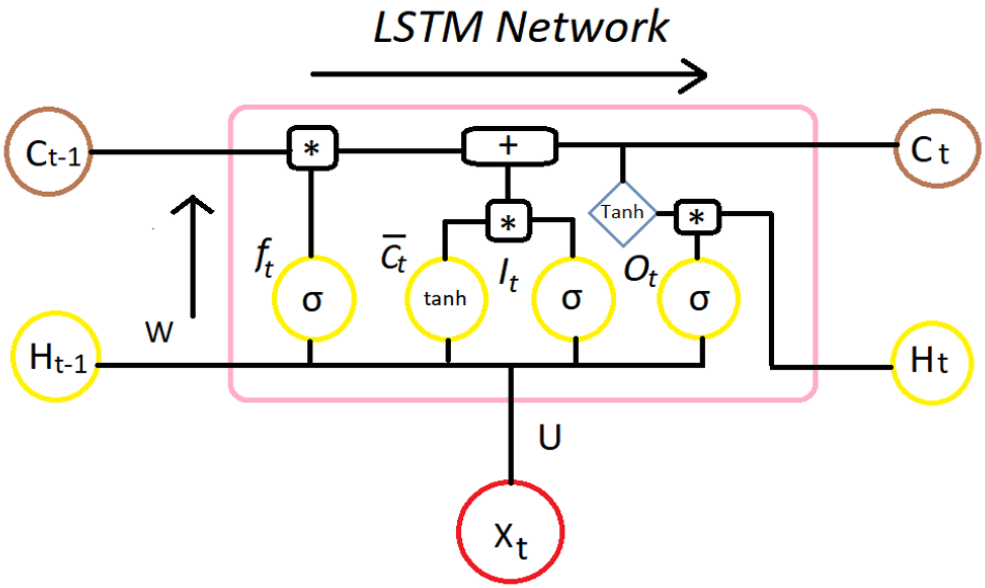


Figure 5.12 LSTM cell architecture [182]

As mentioned above, the gates mechanism, which decide whether remember the current input information in the hidden state matrix, is the most important part of the LSTM network. LSTM define a cell state matrix for "remembering" or "forgetting" the information at previous time steps by using three gate functions. Fig 5.12 [182] used a horizontal line on the top to demonstrate the changes for cell state during time step *t-1* to *t* (from C_{t-1} to C_t).

The first gate is a *forget gate* to decide what information to throw away from the cell state, this decision made by a neural layer with *sigmoid* activation function [180]:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5.7}$$

The second gate is an input gate which consists of a neural layer with *sigmoid* activation function to decide which values will be updated, and the a neural layer with *tanh* activation function which creates a vector of new updated values as described in (5.8) and (5.9) [180]:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5.8}$$

$$\widetilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5.9}$$

Then the cell state updated from the output of equations (5.7), (5.8), and (5.9) by[180]:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \tag{5.10}$$

Finally, the output of the current state will be calculated based on the updated cell state and a neural layer with *sigmoid* activation function which decides what parts of the cell state will be the final output as described in equations (5.11) and (5.12) [180]:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5.11}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{5.12}$$

σ indicate the sigmoid activation function which squashes numbers into the range (0,1), $\tanh()$ is hyperbolic tangent activation function which squashes numbers into the range (-1,1), W_f, W_i, W_c, W_o are the weight matrices, x_t is the input vector, h_{t-1} denotes the past hidden state and b_f, b_i, b_c, b_o are bias vectors. [180]

The proposed TCN architecture uses the two LSTM layers containing 64 neurons with ReLU activation function[52]. ReLU activation function ReLU(x) can decrease the computational complexity of LSTM.

$$\mathrm{ReLU}(x) = \max(0, x) \tag{5.13}$$

The third LSTM layer, as the many-to-one layer in TCN, contains 32 neurons and also uses the ReLU activation function. By applying the ReLU activation for the LSTM cell, the formula (5.12) are replaced as (5.14). The proposed method only calculates h_t with ReLU to decrease the risk of gradient exploding[180].

$$h_t = o_t \cdot \mathrm{ReLU}(C_t) \tag{5.14}$$

The proposed FASAN also consider the spatial information by using CNN with the guide of AU intensity signal. In the proposed FASAN architecture, the neural network only considers the spatial textures from two important frames when k=SAU_{max}or k= SAU_{min}.

$$SAU_{max} = k \; when \; maximum \; value = \left(\sum_{j=1}^{j=n} AU_k^j \right) \tag{5.15}$$

$$SAU_{min} = k \; when \; minimun \; value = \left(\sum_{j=1}^{j=n} AU_k^j \right) \tag{5.16}$$

The feature vectors f_{spatial} for the selected frames are generated by the feature encoder part of the selected pre-trained convolutional neural network.:

$$I_{k=SAU_{min} \; or \; k=SAU_{max}} \xrightarrow{\text{spatial feature encoder network}} f_{\text{spatial}}^{\min \, or \, max} \tag{5.17}$$

The proposed feature *F* is generated by concatenation:

$$F = f_{Spatial \; Encoder \; Net}^{min} \parallel f_{Spatial \; Encoder \; Net}^{max} \parallel f_{AU} \tag{5.18}$$

Implementation details

The proposed facial action unit intensity system, which partially include 46 AUs in the original FAUS, follows the suggestions in Chapter 4 and suggests 12 AUs, which are stable over time, for the proposed FASAN. (AU1, AU2, AU4, AU6, AU7, AU10,

AU12, AU14, AU15, AU18, AU20, AU24). Also, from the analysis of FAUH in Chapter 4, the AU5, which does not show a significance for PA classification, is excluded in the proposed FASAN.

The Enhancing and Cropping Net (EAC-Net) structure from Li, Wei et al. [183]'s work are used to learn both features-enhancing and region-cropping functions effectively, and to detect facial action units. And the pre-trained model available from [183] is used. As EAC-Net uses the VGG 16 pre-trained network[47] to extract features, the proposed FASAN apply the VGG16 as our spatial feature encoder network to decrease the processing time and apply a flatten layer after the VGG16 pre-trained network to get the feature vector.

Finally, a classification network with two fully-connect dense layers is used. The first dense layer also uses ReLU as activation function. The second dense layer uses a sigmoid activation function.

Experimental Results

Extensive experiments are conducted on two very challenging datasets: CASIA-FASD[122] and REPLAY-ATTACK[27] which are publicly available. Competitive detection scores were obtained when the proposed system was compared with other state-of-art techniques. In the experiments, the proposed results follow test protocols of the two datasets to make fair comparisons with recent works. To report the performance, the HTER on the test set and EER on the development set is used for Replay-Attack dataset[27]. Since the CAISA-FASD lacks a pre-defined validation set, the training dataset is divided into four folds and the results for CASIA-FASD[122] are reported in terms of EER. Results are given in terms of EER computed on development set and the Half Total Error Rate (HTER) on test dataset.

Table 5.3 CASIA-FASD test results in terms of EER (%) at different Scenarios:(1) low quality, (2) normal quality and (3) high-quality (4) warped photo attacks, (5) cut photo attacks, (6) video attack, and (7)overall test

(EER%)	**1**	**2**	**3**	**4**	**5**	**6**	**7**
LBP	16.5	17.2	23.4	25.1	17.6	26.7	25.0
FAUH	22.1	20.7	21.4	16.3	17.1	28.5	21.11
DTL-PAD (VGG16)	**5.9**	5.2	8.5	4.1	6.7	**9.3**	7.1
FASAN	6.4	**3.8**	**4.3**	**2.6**	**3.2**	11.5	**4.3**

Table 5.4 Replay-Attack DB overall test

	Dev Set (EER%)	**Test Set (HTER%)**
LBP	17.9	13.7
FAUH	11.6	12.9
DLT-PAD(VGG16)	8.4	7.7
FASAN	**2.0**	**3.1**

Table 5.3 and Table 5.4 provide a performance comparison of the proposed FASAN system with fine-tuned VGG16 , baseline LBP[24], and FAUH. The FASAN system is seen to produce better results for the normal-quality scenario, cut paper attack scenario, and wrapped attack scenario. Also, the proposed feature resulted in better performance with the Replay Attack dataset.

Table 5.5 represents the results of the comparison between the proposed method and some state-of-the-art methods, which include the initial results of different attempts on the CASIA-FASD[122] and the Replay-Attack datasets[27]. The FASAN system produced better performance than the FAUH in all of these tests. From Table 5.5, it may be noticed that FASAN produces better performance than DPCNN[68] and LSTM + CNN [72] but is worse than 3DCNN [103]. However, 3DCNN[103] includes much more trainable parameters in their network. It needs more training time, processing time, and computational resource than FASAN.

Table 5.5 Comparison with the state-of-the-art at CASIA-FASD and Replay-Attack DB overall test ("*" indicate the performance implemented by ourselves)

	CASIA-FA (EER%)	**Replay-Attack DB (HTER%)**
LBP[122]	25.0	13.7
DPCNN[68]	5.4	6.1
LSTM + CNN[72]	7.6*	5.93
3DCNN[103]	**1.4**	**0.3**
FAUH (Chapter 4)	21.1	12.9
FASAN(Proposed)	4.3	3.1

5.4.2 Temporal Local Texture Network with 3D CNN

As established in Chapter 4, the temporal texture changes can also possess discriminative characteristics for presentation attack detection. In this section, the proposed Patch-3DCNN is introduced in detail and demonstrate the potential of using 3D-CNNs for PAD from the encouraging results.

Some deep convolutional neural architectures, such as the proposed DTL-PAD or CCPAD-Net, consider the static texture representations for detecting PA. However, some discriminative cues for spoofing attacks are related to temporal texture changes but the standard 2D CNN cannot explore temporal information efficiently. In this section, a novel patch-based 3D CNN method is proposed to address the issues and some experiments are conducted on two widely used datasets to explore its effectiveness.

From the published literature, the standard 2D convolutional neural networks have proven its effectiveness by successfully outperforming other learning algorithms in PAD. However, recent datasets for the evaluation of PAD algorithms include video clips (or frame sequences) as a source of PAD sensor data. One of the disadvantages of applying standard 2D CNN to video sequences is the potential loss of temporal information between frames. Such information can provide discriminative cues such as unexpected motions etc. The previous work on temporal features for PAD shows the necessity of exploring both spatial and temporal dimensions.

However, extracting spatio-temporal information efficiently for PAD is still a challenging problem. Learning temporal features which are distinct for PAD would become more difficult, as the data dependency for the larger neural network could be even more pronounced. In this section, the proposed experiment explores a 3D CNN architecture for PAD due to the effectiveness of 3D CNN at other computer vision tasks[184] . The proposed 3D CNN architecture processes the distinct cues for presentation attack associated with both spatial and temporal variations. Data augmentation is used to decrease the impact of the data dependency problem associated with DNNs.

Meanwhile, a CNN+LSTM architecture as a widely used strategy to process spatio-temporal information can hardly explore this temporal correlated information efficiently due to the integration of the pooling and dropout layers in the convolutional

blocks[72]. The pooling layers [165] and dropout layers are used to decrease the risk of overfitting problem and help the network build an "information bottleneck" to filter important information. However, some texture representations which may be distinct in the temporal dimension are not vary significant in the spatial domain and will also be filtered by using this strategy.

Methodology

The proposed method consists of two processing flows: random patch flow and down sampled facial flow, to get the characteristics for PAD from both texture representation level and object part level. The pipeline of the proposed method is presented in Fig. 5.13.

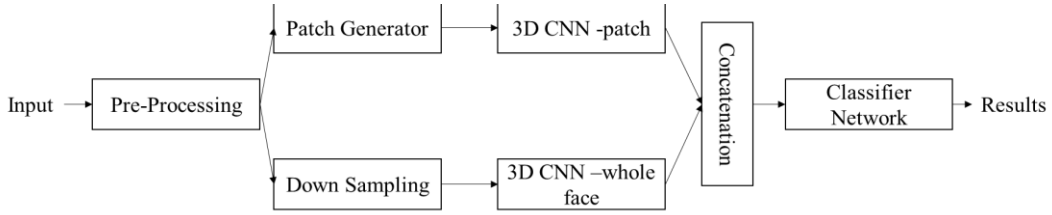


Figure 5.13 3D patch based facial anti-spoofing pipeline

After the pre-processing steps, which has been mentioned in Chapter 3, the frame sequence is transferred into a facial area sequence where the appeared face is detected and normalised by using facial landmarks. Then, the Patch Generator is used to divide each normalised frame into $M \times M$ patches. Thus, the normalised frame sequence is used to generate $M \times M$ cubes. The proposed *3D CNN-patch* uses these spatio-temporal cubes as the input of the network and produces the feature vector as the output.

The input of the *3D CNN-whole face* is generated by using the down sampling algorithm to fit the scale of the input layer in the 3D CNN architecture. The input of the classifier network is the concatenated feature vector consisting of the output of the 3D CNN for the patches and the 3D CNN for the whole face. The final feature vector is calculated using the flatten operator, which is normally considered as a core operator on various DNN platforms[178].

There are multiple motivations to use patches for training the DNNs. Firstly, the patch-based method can significantly increase the volume and the diversity of the

training dataset. The existing presentation attack datasets only include a very limited number of samples for training. For instance, the CASIA-FASD[122] only include 20 training subjects and for each subject it includes 12 short video clips. The overfitting problem could be a major risk for the DNN models trained by using these data directly. Secondly, the resize method, which may lead to the abandonment of some discriminative texture changes, is widely used for deep neural networks to fit the input scale that is defined by the first convolutional layer of the pre-trained neural networks. In contrast, using local patches can maintain the native resolution of the original face images, and thus preserve the discriminative ability. The texture difference and the colour representation difference which may be distinct for the presentation attack detection are believed to appear in the entire facial region. These distinct representations may be significant when using patches as the input of proposed neural network.

Adding the sub-network for the whole facial region which is down sampled before feeding into the neural network is another important design aspect in the proposed pipeline to produce features at "high-semantic level". The proposed down sampling function works in both spatial and temporal dimensions. The normalised facial area is resized to the same input size of the 3D-CNN for the entire facial region as the down sampling process for the spatial dimension. And the frames are selected at intervals of *G* frames (The proposed experiments fix *G*=30 to decrease the implementation complexity).

The 3D CNN offers a possible solution for the difficulty of processing temporal information in PAD. In 3D convolution operation, filters (made up of weights) are moved spatially as well as temporally, performing dot products at each spatial-temporal position in the input.

A standard 2D convolution which is widely used in DNNs can be represented [169]as an operator to extract features from local neighbourhood sets. The result of convolutional operation, which integrate the additive bias is fed into a non-linear activation function. Thus, the activation value at position (x, y) in the jth feature map in the ith layer is denoted as v_{ij}^{xy} which can be represented as (5.19) [169]:

$$v_{ij}^{xy} = \varphi\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right)$$

(5.19)

where $\varphi(.)$ is the non-linear activation function, b_{ij} denote the bias for the feature map. m represents the index value over the set of feature maps in the previous layer which is connected to the current feature map. P_i and Q_i are used to represent the height and width of the kernel which are normally equal to each other in the implementation. And w_{ijm}^{pq} denotes the value at the position (p, q) of the kernel that is connected to the *m*-th feature map.

Thus, the 3D convolution can be extended from the 2D convolution operator by following the suggestion of [184] to calculate the feature map from both spatial and temporal dimensions. The 3D kernels are used to convolving the spatio-temporal cube which is formed by stacking a frame sequence. The 3D convolutional operator is represented by the formula (5.20):

$$v_{ij}^{xyz} = \varphi\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \tag{5.20}$$

where R_i is added to represent the size of the 3D kernel in the temporal dimension. w_{ijm}^{pqr} denotes the activation value at position (p, q, r) and connected to the m-th feature map in the $(i-1)$th layer. Similar with 2D convolutional blocks, a 3D pooling layer is subsequently connected to the 3D convolution operator to reduce the size of the feature map. Tran, et al.[185] demonstrated that the smaller 3D convolutional kernels may improve the performance of the video classification. Also, Li, et al. [68] noted a similar phenomenon when adopting a smaller receptive field for the 3D convolutional neural networks. Therefore, the proposed 3D CNN architecture adopts a spatio-temporal receptive field size of $3 \times 3 \times 3$.

Experiential detail and results

There are three widely used anti-spoofing benchmarking datasets which were used to evaluate the effectiveness of the proposed anti-spoofing algorithm: the CASIA-FASD[122], the Replay-Attack database[27], and MSU MFSD dataset[123]. All of these datasets include some recordings of genuine client access attempts and various presentation attacks.

Table 5.6 compares the performance of the proposed method with selected deep learning methods for spoofing detection. Here, FASAN [145], which use VGG16 as

the pre-trained feature extraction network, can be considered as a typical 2D CNN method in this table. Table 5.6 also considers the proposed Yang-Net[67] to demonstrate the effectiveness of directly applying 3D CNN for temporal information.

Table 5.6 Comparison with the state-of-the-art at CASIA-FASD and Replay-Attack DB overall test('*' indicates the performance score which follows the reference and implemented by ourselves)

	CASIA-FA (EER%)	**Replay-Attack DB (HTER%)**	**MSU (EER%)**
Yang-Net [67]	9.94*	8.4*	5.8*
FASAN [145]	4.3	3.1	N/A
3DCNN[103]	3.02	0.3	0
3DCNN[186]	6.4	0.0	4.8
Patch-3DCNN (Proposed)	4.2	0.1	1.7

3DCNN[103] and 3DCNN [186] are two works which also consider 3DCNN as the part of their architecture. The proposed Patch-3DCNN is different with their work. In summary, Li et al. [103] and Gan et al. [186] only considered the entire face as the input of their architecture. Although they selected different kernel size of their convolutional operation, their method can be considered as the same approach. The proposed method considers a *3DCNN-Patch* subnetwork, which considers the divided cubes rather than the entire facial region as the input. The final feature vector is generated by concatenation the output of the *3DCNN-Patch* subnetwork and *3DCNN-whole face* subnetwork. From the table, 3D-CNN[103] reaches the best result for the MSU-MFSD dataset[123]. However, the proposed system represent the better performance than the results from Le et al. [103] for the Replay-Attack dataset[27]. The results from Gan et al. [186] reach EER=0.0% at Replay-Attack dataset, but the proposed Patch-3DCNN shows better performance at both CASIA-FASD[122] and MSU-MFSD dataset[123].

5.5 SUMMARY

In this chapter, 6 different studies of the application of DNNs for PAD are explored. In exploring different pre-trained neural architecture, the potential of DNN-based methods is demonstrated by the good experiment results. The novel Colour Convolutional PAD Network is then designed for the PAD task and trained using only the PAD datasets. Some visualisation experiments are firstly explored and demonstrate the inner mechanism of using deep neural architectures for PAD. By analysing the

visualisation results from these visual explanation algorithms, ideas and insights about how to improve the baseline protocols are generated for further exploration in Chapter 6. Furthermore, the idea of facial action unit signals and motion texture chances are considered to design the novel deep neural architectures in this section.

Table 5.7 Performance of the DNN based feature for multiple datasets (BPTM* means the best performance of the proposed traditional methods)

Datasets EER(%)	**NUAA**	**REPLAY-ATTACK database**	**CASIA-FASD**	**MSU-MFSD**	**HKBU MARs**	**Rose-Youtu**
BPTM*	N/A	0.60	4.80	7.67	N/A	N/A
DTL-PAD (VGG16)	3.6	8.4	7.1	16.0	39.7	15.4
DTL-PAD (ResNet)	4.9	5.7	6.3	11.4	33.1	14.8
DTL-PAD (NAS)	2.5	9.4	8.0	14.3	35.0	18.5
Colour Space Net(VGG16)	0.3	0.6	1.5	3.2	19.7	7.9
CCPAD-Net	**0.1**	0.8	1.1	2.7	20.5	8.3
FASAN	N/A	2.0	4.3	N/A	N/A	N/A
Patch-3DCNN	N/A	**0.1**	4.2	1.7	N/A	N/A

Table 5.7 demonstrates the performance comparison for deep learning based methods with 7 benchmark datasets. The first row of Table 5.7 is the best results of the proposed traditional features in Table 4.15. The DTL-PAD(VGG16) also applies the transfer learning paradigm for PAD and considers the feature extraction part of the VGG16 pre-trained network[47] as the feature encoder network. As the description in the previous chapters, some of the proposed methods use the same pre-trained feature encoder network but trained in different ways. By comparing proposed methods with the DTL-PAD(VGG16), Table 5.7 can clearly demonstrate the performance improvements of the proposed methods.

The DTL-PAD (VGG16), DTL-PAD (ResNet) and DTL-PAD(NAS) demonstrate the performance differences when using transfer learning paradigm but with different pre-trained networks for feature extraction. The Colour Space Net+VGG16 and CCPAD-Net consider same Colour Space Net but with different overall design in the proposed experiment.

The Colour Space Net+VGG16 connect the proposed Colour Space Net with a pre-trained VGG16 network and the CCPAD-Net is trained from scratch. In table 5.7, the highlighted performance score is the best score when consider the proposed traditional and deep learning-based methods. The CCPAD-Net represent the best result at NUAA dataset[121]. The Patch-3DCNN demonstrate the best performance score at REPLAY-ATTACK dataset[27].

The contributions of the present chapter are listed as follows:

(1) Multiple pre-trained neural architectures (such as VGG16, ResNet50, DenseNet) are explored by using the deep transfer learning paradigm for PAD tasks. The experiment results for the different DTL-PAD networks can be used to show the better choice of pre-trained backbone network in the future usage. The promising results from the DTL-PAD networks demonstrate the effectiveness of using deep architectures for industry applications. Additionally, this thesis considers the results from DTL-PAD(VGG16) networks as baseline results to demonstrate the effectiveness of the proposed methods.

(2) A novel deep neural architecture named CCPAD-Net is designed for PAD task and trained only using PAD datasets. This neural architecture includes a Colour Space Network, which is designed to learn a colour transfer function. This Colour Space Network, which can be trained separately and decrease the overfitting risk of using a small dataset, follows the observations and assumptions in the PAD research area. The encouraging results from multiple benchmark datasets demonstrate the necessity of designing the neural architectures for PAD rather than only using deep transfer learning paradigm.

(3) Some visualisation experiments are first explored for PAD tasks to show the inner mechanism of deep neural networks. From these visualisation experiments, the necessity of producing a PAD system, which can justify each decision generated by the system, is becoming clear as a key motivation for the following work in Chapter 6. Also, the necessity of using high-speed connection between convolutional blocks is emphasized in this Chapter.

(4) A novel neural architecture named FASAN is designed and tested at various benchmark datasets. This neural architecture also uses the facial action unit system, which is firstly introduced in Chapter 4, but provides better performance by using deep neural architectures for temporal information. The encouraging performance

at different datasets shows the potential of using facial action intensity signal and Recurrent Neural Networks.

A novel 3D convolutional deep architecture is provided by following the idea of dividing facial region into patches. The sequence of these patches is used to generate a spatio-temporal cube as the input of *3D CNN-patch* subnetwork. The whole face region is resized and the sequence of resized facial region is generated as the spatio-temporal cube for *3D CNN-whole face* subnetwork. The proposed 3D convolutional deep architecture is designed for both texture level and object level. The results compared with other 3D CNN methods show the effectiveness of the proposed architectures.

Chapter 6: Deep Learning for PAD

The use of deep learning techniques can not only lead to improved performance but also offer new possible research directions for PAD. Based on the results and analysis described in Chapter 5, this chapter explores the interpretable capability of PAD systems using deep learning techniques and the automatic design of deep learning-based PAD systems. The methods for XAI and NAS have been proven to have considerable value in other fields of research but their benefits have not yet been explored and evaluated in the context of PAD. In this chapter, two novel techniques are proposed by integrating the new features into the PAD system. First, the motivations for the proposed developments are introduced in Section 6.1. Then, an attention guided PAD system, which can justify its decisions and learn from the generated justifications is proposed and evaluated in Section 6.2. A novel PAD system, which only needs the labelled data to automatically generate deep neural architectures is introduced in Section 6.3 and evaluated using commonly used public datasets.

6.1 Motivation

Deep learning stimulates significant performance improvements in PAD, and brings new concepts and paradigms that offer new directions for future research. The proposed methods in this chapter demonstrate the potential benefits of these new concepts and paradigms by exploring two possible directions: (1) Explainable presentation attack detection (XPAD) and (2) Neural architecture search for presentation attack detection (NAS-PAD).

One of the main debatable topics in deep learning is its "black-box" nature, which means it is not clear how the network generates its output. Making the system answer the question "What is the reason behind this decision?" is the first possible direction to improve PAD systems in the future. In previous studies, a PAD system was only designed to detect the presentation attack. The output of the system is its (a) decision whether the input data is an attack and (b) determination of the type of the attack (such as paper attack, video attack, etc). However, these systems cannot justify the decision made from the input data.

A novel PAD system, which can explain its behaviour to users, is the first contribution of this chapter. In this system, naturally, the interpretation information is used to further guide the optimization of the system performance. The proposed method also addresses the research question: can the additional explanation information, generated by the proposed interpretation algorithm, be used to make the neural network learn a more robust feature encoder? The details about the proposed model for this direction are provided in Section 6.2.

The second part of this chapter is on efficient neural architectures searching for PAD. Neural architectures designed by human experts have been widely used for PAD in the past years. In this chapter, the proposed novel method goes one step further to automatically generate the neural architecture based on the training data. The design of a usable architecture for the neural network includes several choices of key components, the structure of the CNN layers and the selection of the high-speed connections between layers. Optimising the hyper-parameters of a DNN is also a challenging task. Given the magnitude of this task, researchers have declared "deep learning is new alchemy"[187] . In order to solve this problem, recently, a new end-to-end method has been developed which can design deep neural network structures using NAS. Human expert input is not needed to design or to fine-tune the structures and connections between different layers. Generally, all possible neural architectures are considered in the search space, and the NAS model will select the best model for any particular application. This idea extends the end-to-end trend promoted by the deep learning paradigm. This approach is explored for its application to PAD in Section 6.3.

6.2 Learning from explanations

Despite the high performance achieved using DNNs for PAD, the inability to justify their decisions is a significant drawback given the usability and security requirements of many biometric applications. In this section, an attention-guided convolutional neural network for spoofing detection is presented, which can learn from additional information produced by a visual analysis of DNNs. In particular, the proposed approach utilises both spatial and temporal information to detect facial spoofing behaviours and provides both visual and natural language explanations for each decision to answer the questions such as “How the system makes its decisions?”. Furthermore, the proposed framework can learn from such explanations to improve

the system performance. This is evaluated with different experimental setups and results using the CASIA-FASD[122], Replay Attack, MSU-MFSD[123] and HKBU MARs datasets suggest its effectiveness.

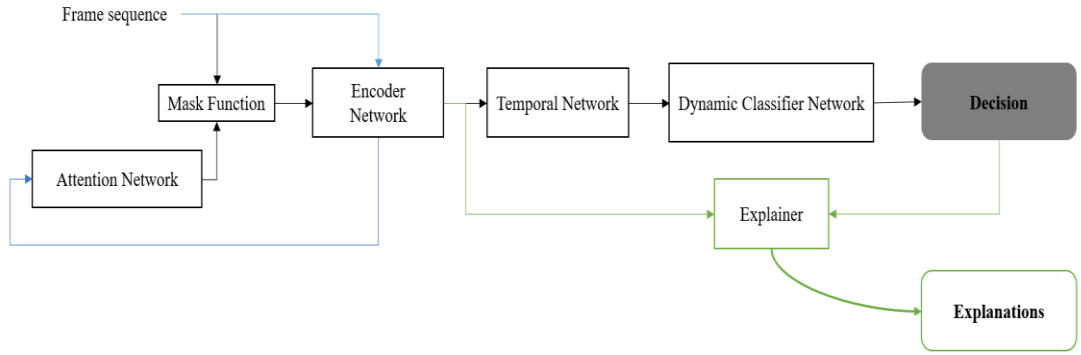


Figure 6.1 System block diagram of the proposed Dynamic Attention Convolutional Network (DACN).

In the following paragraphs, the term "explanation" is firstly defined to make the description clear. Then, the proposed system which uses DNNs to encapsulate both temporal and spatial texture changes, will be introduced sequentially (see Figure 6.1): (1) Attention generation (Blue line): Firstly, the frame sequence is fed into the Encoder Network to get the feature representation of each frames. Then the attention maps will be generated by the Attention Network from these original feature representations. (2) Decision generation (Back line): the masked frame, which is generated by the Mask Function, is fed into the Encoder Network as well. Then the Temporal Network is used to encapsulate time-related information. The Dynamic Classifier Network is then used to provide the final decision about the input frame sequence. (3) Explanation generation (Green line): The Explainer function is used to provide an explanation for current decisions. These training stages in the proposed system are designed to improve the detection accuracy by using the generated explanations as additional information.

The basic idea of the proposed work is aiming to provide an interpretable framework which generates explanations and can learn from these explanations to improve its decisions. However, there is no commonly used definition about what is the "explanation" for a deep learning system. Some have used the feature relevance scores which are calculated by using the gradient flow from each decision to measure the influence of spatial importance[174]. Also, terms such as the Class Activation Map or the saliency map have been used[90]. The proposed framework partially follows the

definition in [174] to calculate the explanation from gradient flow and consider multiple forms in the "Explainer" function for different using scenarios: (E1) feature relevance score (E2) spatial saliency map (E3) natural language explanations. Here, (E1) and (E2) are used to guide the training process in Stage 2 as additional information. Explanations with form (E2) and (E3) can help the human users to understand the reason behind each decision.

6.2.1 DNN-based facial anti-spoofing detection

The proposed framework is based on the deep learning paradigm (Stage 1 in Figure 6.2). For PAD, the input, denoted by $X = \{X_i | i \in [1, N]\}$, is a set of video clips where each clip is a set of frames $X_i = \{I_j | j \in [1, M]\}$, and the desired output, $Y = \{Y_i | i \in [1, N]\}$, is the set of decisions. The number of decision classes is represented by *C* which includes genuine presentations and different attack modalities. *N* is the number of video clips in the dataset, and *M* is the number of frames of each clip. Here we use $\check{Y}$ to represent the predicted output of the model. A deep learning model, with θ_f, θ_c, as trainable parameters, can be represented by equation (6.1):

$$X \xrightarrow{F^f(X;\theta_f)} E \xrightarrow{F^c(E;\theta_c)} \check{Y} \tag{6.1}$$

where $E = \{E_i | i \in (0, N]\}$ is the feature representation of the whole dataset generated by the feature extraction sub-network $\mathrm{F^f(X;\ \theta_f)}$, where $E_i = \{e_t | t \in (0, M]\}$ is used to represent the feature encoding of one video clip. The Encoder Network $\mathrm{F^f(X;\ \theta_f)}$ and the Classifier Network $\mathrm{F^c(E;\theta_c)}$ can be designed specifically for PAD and trained from scratch only using a PAD dataset. Alternatively, these two sub-networks can also follow the transfer learning paradigm for better generalisation capability. In the proposed experiment, the feature extraction part of a pretrained network based on ImageNet[64] is transferred for PAD by following the suggestions of [145] to demonstrate the performance improvements of the proposed framework.

The proposed framework adds the following elements to achieve better performance than the basic deep learning paradigm. (1) The Explainer function $Explan(.)$ is added to provide the explanation of each decision. (2) The attention network $\mathrm{Attention(e_j)}$ is introduced to improve performance by learning the location of significant regions in the image. It is initially trained by using the explanations

calculated by the Explainer function $Explan(.)$ (3) The Temporal Network$Temporal(.)$ is used to model the temporally correlated information and generate a feature vector for each video clip. (4) The Dynamic Classifier Network (Figure 3) is used to generate the final decision about spoofing attacks. The Dynamic Classifier Network is used when the Temporal Network is included in the system. The proposed framework with only the (Frame) Attention Network is referred to as the Frame Attention Convolutional Network (FACN). And the full framework including the Temporal Network is referred to as the Dynamic Attention Convolutional Network (DACN).

The proposed pipeline includes an explanation generator to produce explanations for each decision. A natural language explanation is generated by using $\xi(\check{Y}, exp, Q, L) = l$ for the current decision, where Q represents a question set and L represents the most relevant human language answer set. Here, l indicates natural language explanations for the decisions made to accompany visual explanations s. A set of explanations in the form of questions and answers is provided in Table 6.1. The system can provide the readable explanations which include two explanation forms (E2) and (E3).

The proposed framework has three training stages as shown in Figures 6.2 and 6.3. In Figure 6.2, Stage 1 aims to get a typical DNN architecture for classification. The proposed workflow follows the deep transfer learning protocol to train the classifier network. The fine-tuning stage will then be applied to train both feature extraction network and the classifier network with lower learning rate. Stage 2 has two phases: In 2a, the Attention Network is firstly trained with the pair of the encoded frame and the spatial explanation from the Explainer function. Then, the Spatial Attention Convolutional Network is trained using new data. The green lines indicate the explanation generation process. The orange lines indicate the training steps to learn with explanations. The yellow line is used to indicate the original frame and the features generated from the original frame.

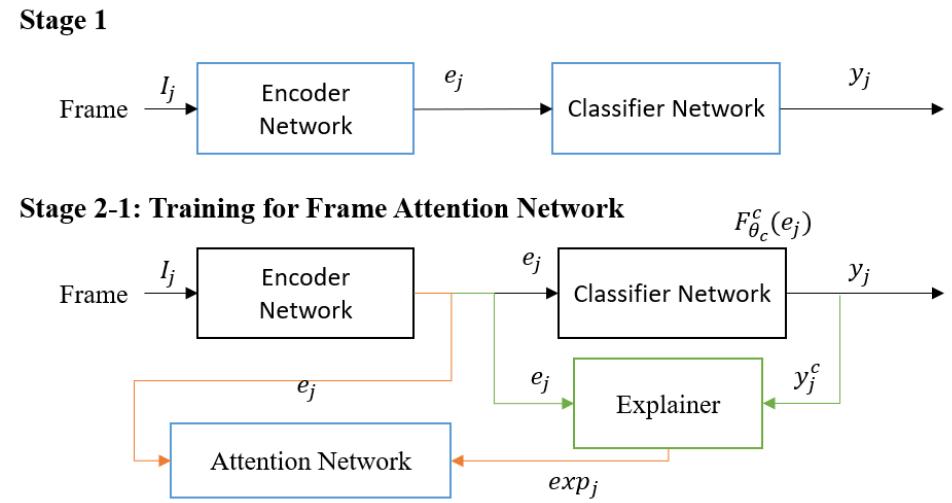


Figure 6.2 First two training stages: (Blue boxes indicate the sub-network(s) that will be trained in each stage.

The second training stage includes two phases: Stage 2a Training for the Attention Network, $\mathrm{Attention}(e_j)$, and Stage 2b Training for the Spatial Attention Convolutional Network (SACN). During this training stage, the Attention Network is initialised for faster convergence [188]. The parameters of the Encoder Network and the Classification Network are shared from the Stage 1 and fixed in the Stage 2a. In Stage 2a, the Attention Network is trained by a generated dataset which consist of the feature encoding e_j for a randomly selected set of frames I_j from each video. Every video clip in the training dataset will provide m randomly selected frames for this training where $0 < m < M$.

These encoded features are the input of the Attention Network. Then, the labels of these encoded frames e_j are provided by the Explainer Function as exp_j which

support the current decision of the Classifier Network. The Explainer Function in the proposed framework is selected to emphasize the important spatial locations for predicting presentation attacks. The visual support for the current decisions is the saliency map (E2) which is converted from E1 to visualise the significant regions in the original frame. The Attention Network $a_j = Attention(e_j)$ consists of 2 fully connected dense layers, one with the ReLU [52] activation function and the other with the Tanh activation function. When the Attention Network is trained, the Stage 2b will commence the training of the SACN. The attention mask a_j will be applied to the original frames by using element-wise multiplication to get the masked frame I_j^*. Then, the new encoded features e_j^* are calculated to get the prediction y_j^*. At Stage 2b the whole SACN is trained using a lower learning rate compared to that used to train the Attention Network for fine-tuning to improved performance.

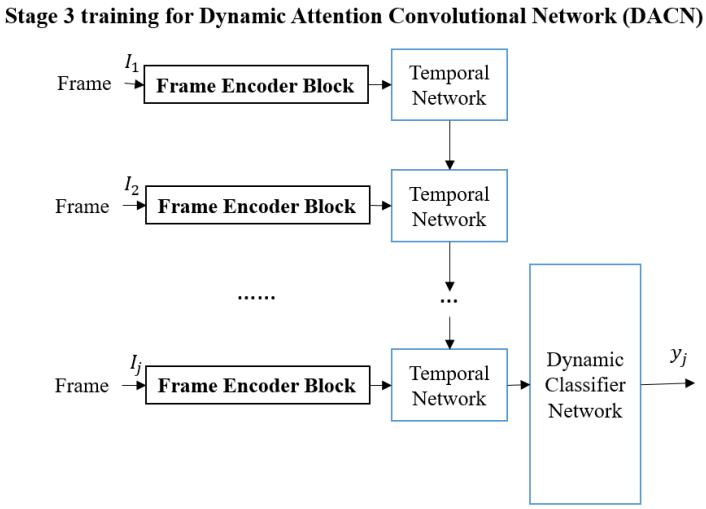


Figure 6.3 The third training stage: (Blue boxes indicate the sub-network that will be trained.) Stage 3 is used to train the Dynamic Attention Convolutional Network (DACN).

The third stage (shown in Figure 3) is used to train the Temporal Network. The deep architecture in Stage 3 is named as the Dynamic Attention Convolutional Network (DACN) to emphasize the usage of temporal information. Each video in the training set will be used to train the Temporal Network $Temporal(a_i, e_i)$ which consists of two LSTM layers to obtain a fixed length feature for each video. The Temporal Network is used to select the significant information in the video.

The original LSTM only considers the temporal relation between frames. However, the importance of a frame comes not only from the temporal relationship with their neighbours but also from the spatial texture changes. For this reason, we change the forget gate function to $f_t^1 = \sigma_g(W_f e_t + U_f h_{t-1} + V_f^* a_t + b_f)$ (the superscript is used to indicate the layer of LSTM.) where the $\sigma_g(.)$ is a sigmoid activation function, W_f, U_f, V_f^* denote the trainable parameters. h_{t-1} is the hidden state of the last time step and b_f is the bias. Here, the attention map a_t, which is the output of the Attention Network $Attention(e_j)$, is included in the control function *C* of the forget gate. And the cell state function is also changed to integrate input features e_t from the Encoder Network, spatial attention heatmap a_t and the hidden state of LSTM, h^{t-1} as: $C_t^1 = \tanh(W_1^t h^{t-1} + U_1^t e_t + V_1^t a_t + b)$. The output of LSTM will be fed into a new classifier with two dense layers using the ReLU activation function[52].

Providing explanations for each decision is the key feature of the proposed architecture. The justification provided by the Explainer function consists of two parts: spatial importance explanation and temporal importance explanation. The temporal importance explanation selects the most important frame in the sequence. And the spatial importance explanation emphasizes the important regions in that frame. The temporal importance explanation is calculated by the $exp^t = \max \sum(f_t^n + i_t^n)$ to select the time step in which the cell state of LSTM has been maximally changed. In a short frame sequence, the proposed method considers the frame, which changes the cell state of LSTM most, as the most important frame in this sequence. The proposed work follows [17] to calculate the spatial importance by using gradient flow of the last convolutional layer. This spatial importance explanation is directly used to train the Attention Network $Attention(.)$ at Stage 2. For the SACN, the Explainer function can be represented as: $exp_j = Explan(e_j^*, y_j^*)$ where the exp_j and related frame is also used to generate the saliency map (E2) to provide a transparent justification in the proposed architecture.

6.2.2 Implementation detail and experiments design

This section describes the experimental design and implementation details used to evaluate the proposed framework. The results of the experiments are also presented.

Four commonly used face spoofing detection databases were used for performance evaluations: (1) Replay-Attack dataset [27] , (2) CASIA-FASD [122] (3) MSU mobile face spoofing database [123], and (4) HKBU MARs Dataset [4].

For the proposed DNNs, the feature extraction part of the pre-trained VGG-16 network is considered as the Encoder Network. The Classifier Network with two fully connected layers and ReLU activation function[52] is trained by using transfer learning in training Stage 1. The Encoder Network (VGG16) is fixed to start with and the Classifier Network is trained with learning rate of 0.001. Then, the Encoder Network (VGG16) is fine-tuned but using a lower learning rate of 10^{-7}at Stage 1. In the proposed implementation, Lucena, et al.'s work [145] has been followed to fine-tune the VGG16 network.

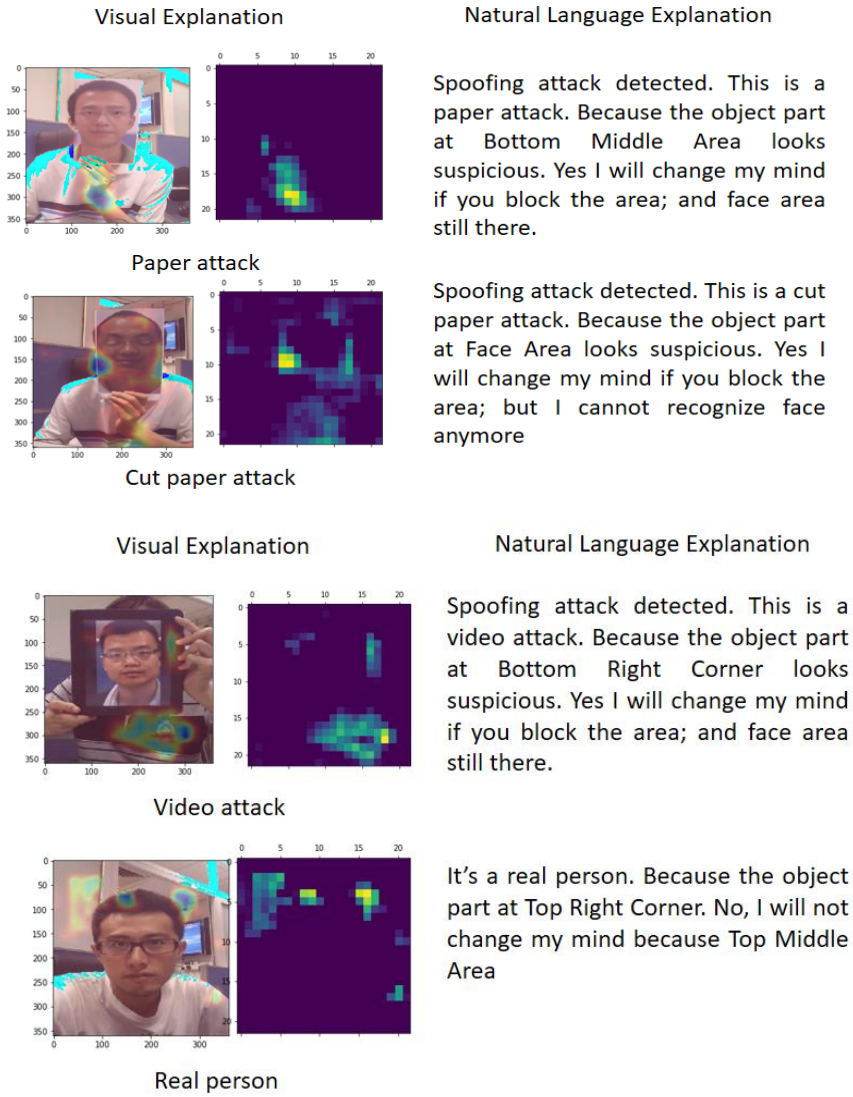


Figure 6.4 Explanation examples generated by the model for different attack types. From top to bottom, the explanation examples are generated for paper attack, cut paper attack, video attack and real face. In each case, the system provide saliency map and

heat-map (left) as visual justification for the decisions and a short paragraph (right) as the natural language explanations

The Grad-CAM[173] algorithm is selected to provide spatial explanation in the proposed framework. In the training Stage2, it was also used to provide additional training information for the Attention Network. As the PAD datasets used in the following experiments do not have pixel-level labels or natural language sentence labels to train a neural network-based natural language generator, Marietto et al's work has been considered to develop a natural language generator [189] in the proposed implementation as in this approach no extra-training data is needed for the natural language generator. In the proposed implementation, the natural language generator selects answers from a pre-defined answer set. The question set and the example answers used can be found in Table 6.1. Four different questions as the question set Q were included and the natural language generator can generate the result l by selecting the most relevant answer from result templates L and fulfil the information from the value of exp. Examples for both visual and natural language explanations can be found in Figure 6.4.

Table 6.1 Example of question answering part

	Question (Q)	**Answer set (L)**	**Answer Example (l)**
1	Is this a spoofing attack?	Spoofing attack detected/It's a real person	{Spoofing attack detected}
2	What kind of spoofing attack?	Real Face / Paper Attack/ Video Attack/ Mask Attack	This is a {paper attack}
3	Why you think this is a spoofing attack?	Face Area/ Top Left Corner/ Top Middle Area/ Top Right Corner/ Left Middle Area/ Right Middle Area/ Centre Area/ Bottom Left Corner/ Bottom Right Corner/ Bottom Middle Area	Because the object part at {face area} looks suspicious
4	If I block that area, will you change your mind?	No, I will not change my mind because{}./ Yes I will change my mind if you block the area; but I cannot recognize face anymore/Yes I will change my mind if you block the area; and face area still there.	{Yes, but I cannot recognize face anymore}.

The Replay-Attack database [27] is divided into three sub-sets: training set, development set and testing set. The feature encoder network is fine-tuned with 60%

of the training set; the attention network is trained using the rest of the training set. The Equal Error Rate (EER) for the development set is reported and used to determine the threshold to obtain the Half Total Error Rate (HTER) on the test set. For CASIA-FASD[122], and MSU databases[123], the Feature Encoder Network is fine-tuned with 50% of the training set and the Attention Network is trained by the rest of the training set. Then EER is evaluated for the test set following the protocols defined in [190]. All of the results are listed in the Tables 6.2 and 6.3.

6.2.3 Experiments results

Table 6.2 Test results for different VGG-16 depths

	Replay-Attack DB	**CASIA-FASD**
	EER (%)	**EER (%)**
VGG16-blocks 1-3	25.64	28.71
FACN(block 1-3)	12.42	16.84
VGG16-block 1-4	14.73	18.01
FACN(block 1-4)	8.30	9.47
VGG16-block 1-5	9.73	10.88
Fine-tuned	8.40	9.94
FACN	**0.20**	**4.12**
DACN	**0. 37**	**1.00**

Table 6.3 Performance comparison ('*' indicates the performance score which follows the reference and implemented by ourselves)

	CASIA (EER %)	Replay-Attack (HTER %)		MSU (EER %)	HKBU MARs (EER %)
VGG16-CNN[145]	9.94*	8.40*	4.30*	5.80*	28.00*
VGG-16-AD [191]	-	-	-	6.72*	**11.79**
CNN+LSTM [192]	5.17	3.66*	4.87*	7.43*	31.20*****
DPCNN [68]	4.5	2.9	6.1	-	-
LBP-CNN [75]	2.5	0.6	1.3		
3DCNN[103]	1.40	0.30	1.20	**0.00**	-
FACN(Proposed)	3.02	**0.20**	2.07	1.67	23.70
DACN(Proposed)	**1.00**	0.37	1.53	0.20	13.51

In Table 6.2, we present the effect of the depth of the Encoder Network using the Replay-Attack[27] and CASIA-FASD [122] in terms of Equal Error Rate (EER). There is a clear trend that can be identified: based on the results, the deeper networks can provide better results and the inclusion of the Attention Network can improve the performance further. This effect of the Attention Network may be similar to facial area cropping which is a widely used pre-processing method. Facial cropping can itself be

considered as a hard attention method to help features focus on key areas as identified by human experts.

Table 6.3 compares the performance of the proposed method with selected deep learning methods in spoofing detection. Lucena, et al.[145]used the same encoder network as ours and can be considered as providing the performance baseline of Table 6.3. Firstly, the proposed workflow uses the same pre-trained feature encoder network as the previously published work[145], [191] and DPCNN [68]. The performance improvements of the proposed FACN compared with [145] demonstrate the effectiveness of using the Attention Network. The VGG-16-AD [191] also significantly improves the performance of the pre-trained VGG16 model for the 3D mask attack detection by selecting significant areas within frames. However, their method is only designed for the 3D mask attack detection and represents worse performance than [145] on the MSU dataset[123]. Secondly, Tu, et al.[192] also attempt to use both temporal and spatial information in their deep architecture. 3DCNN [103]reaches the best result for the R-A and MSU-MFSD datasets[123]. However, the proposed DACN system achieves the best performance for the CASIA-FASD [122]. Thirdly, a hybrid algorithm is presented in [75] which combines LBP[24] and DNNs. This used to be a popular way to use DNNs which only consider DNNs as a robust feature extractor. However, the proposed method which consists of only deep neural networks shows better performance through learning from explanations. These comparisons demonstrate the effectiveness of the proposed approach.

This section explores an attention-based method which uses the VGG-16 pre-trained network[47] as the feature encoder, and implements the Grad-CAM method[173] to guide the training step of the attention network for presentation attack detection in face recognition biometric systems. The proposed framework performs well on multiple benchmarking datasets and reaches a performance level similar to the state-of-the-art.

Furthermore, the proposed method, which provides explanations with both visual and natural language forms, allows the proposed system to be more transparent and trustworthy for users. Additionally, the explanations are incorporated within the proposed algorithm for training the spatial attention stage and results in a measurable improvements of detection performance.

6.3 PAD using neural architecture search

Neural Architecture Search (NAS), which aims to automatically propose a neural architecture to suit the training data, is another important barrier breaking development in the field of Deep Learning. As present, designing a deep neural architecture for PAD is complicated and time-consuming work, which requires the background knowledge of both PAD and DL fields. Moreover, balancing the computational complexity and performance manually is more difficult even for deep learning experts. The NAS approaches offer a possible solution to these problems, which can design an efficient, yet still accurate models by automatically searching deep neural architectures under the constraint of the optimisation function.

From previous works on NAS, some promising results for the image classification task showed the effectiveness of the neural architectures produced by NAS methods. Some recent works applied NAS for various computer vision tasks and the encouraging results from these works inspired the proposed method to extend NAS for PAD. In this section, a novel neural architecture for PAD, which is designed using NAS method, is introduced to overcome the disadvantages of human-designed neural architectures and demonstrate the potential benefits of using NAS approaches. Three key factors for the proposed method are introduced in detail: (a) the search space, (b) the optimization strategy, and (c) the performance estimation strategy.

The proposed NAS method reduces the time costs of processing by selecting an operator set and relaxing the discrete selection for possible operators to a continuous searching space. The computational cost of a PAD method is an important property. The proposed work provides a novel optimisation function to constraint the searching process and balance the computational cost and the performance needs.

The experiments on widely used datasets will be provided to demonstrate the effectiveness of the neural architecture generated by the proposed NAS methods. And these encouraging results demonstrate the potential of using NAS for future deployments of PAD systems.

6.3.1 Methodology

The proposed work in this section addresses the challenge of searching efficient deep neural architectures for facial PAD. It aims to automatically generate a specific deep neural network architecture that can detect various facial presentation attacks

effectively. In the following paragraphs, the overall workflow of the proposed method is introduced and two training stages for the proposed PAD-NAS network are provided in detail. Then, the details of the NAS search methods such as the search spaces, the search strategy and the modified loss function for the PAD will be introduced to show the detail of the proposed method. Finally, some implementation details will be introduced in the last part of this sub-section.

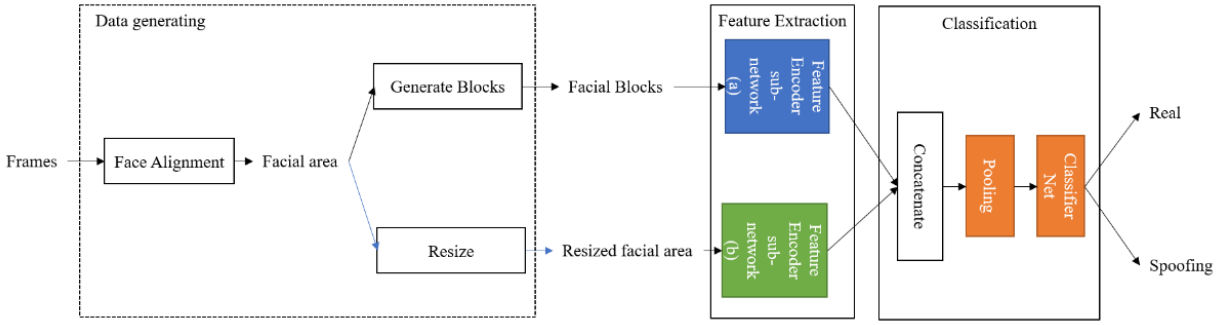


Figure 6.5 Workflow of the proposed PAD-NAS network

Briefly, the workflow for the proposed PAD-NAS neural network consists of three parts: (a)Data generation, (2) Feature extraction, and (3) Classification. The **Data generating** part starts with recognizing and normalising the facial regions from input data by using face alignment method. These facial regions will be divided into 3×3 blocks as the previous descriptions for the **Feature Encoder sub-Network (a);** and the original facial regions will be resized for the **Feature Encoder sub-Network (b)**. The outputs of Feature Encoder sub-Network (a) and (b) are concatenated and a pooling layer[165] is used to decrease the dimension of the feature vector after concatenation. Finally, the Classifier Net provides the final result about whether the input data is a presentation attack.

The proposed PAD-NAS neural network relies on the performance of a feature extraction sub-network (a) and (b). The proposed approach, detailed below, does not use a transfer learning paradigm as some methods in Chapter 5; but directly searches neural cells which will be stacked to produce a neural architecture for feature extraction.

Briefly, two training stages (I and II), which is visualised in Fig. 6.6, is used to get the proposed PAD-NAS neural network: The **Training stage I** is used to search efficient neural cells for different inputs generated from **Data generating process** and

the **Training stage II** is used to train the **Classifier Net** for the proposed PAD-NAS neural network.

In **Training stage I**, each facial region and the whole face area, which is recognized and cropped in the pre-processing step, will be used to search the best neural architecture by using the proposed NAS method. Same NAS algorithm is considered, but **Dataset for Blocks** is used to search the neural cells for blocks; and **Dataset for Whole Faces** is considered to search the neural cells for Feature Encoder sub-network(b). The **Training stage II** is used for training the Classifier Net to provide the final PAD results. The Classifier Net for Blocks, Classifier Net for Faces, and the Classifier Net in the second training stage use same neural architectures, but trained with different dataset. The Classifier Net for Blocks and Classifier Net for Faces will not be used in the proposed PAD-NAS neural network.

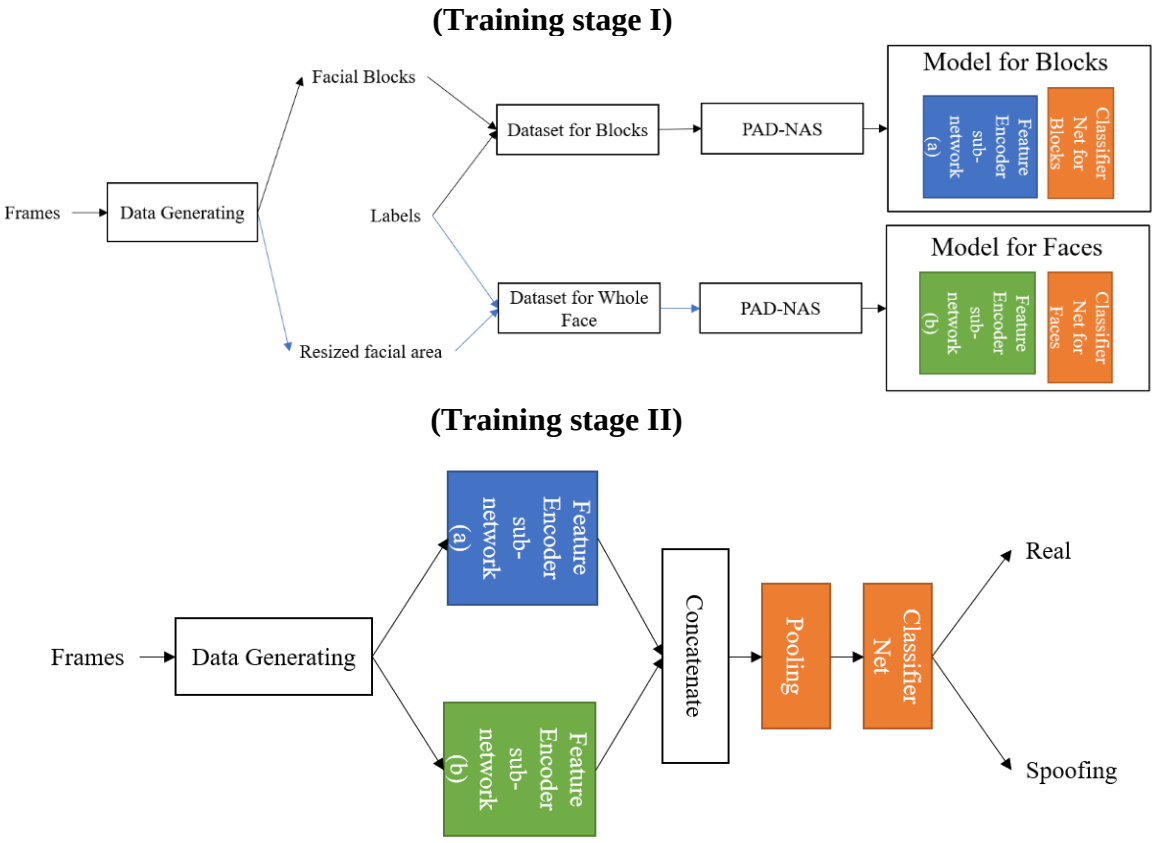


Figure 6.6 Two training stages for the proposed PAD-NAS network

Searching neural architectures from candidate for PAD

The NAS process can be summarised as Figure 6.7. The search strategy selects some operators from an operator candidate set and produces a neural cell which can

be considered as a directed acyclic graph. Then, N searched neural cells will be stacked and a Classifier Net is added to the end of the stacked neural cells to get the generated neural architecture. After testing the performance of this generated neural architecture, the performance estimation strategy will be used to produce a performance estimation for current neural architecture. And the search strategy will be optimized by following the guidience of the performance estimation.

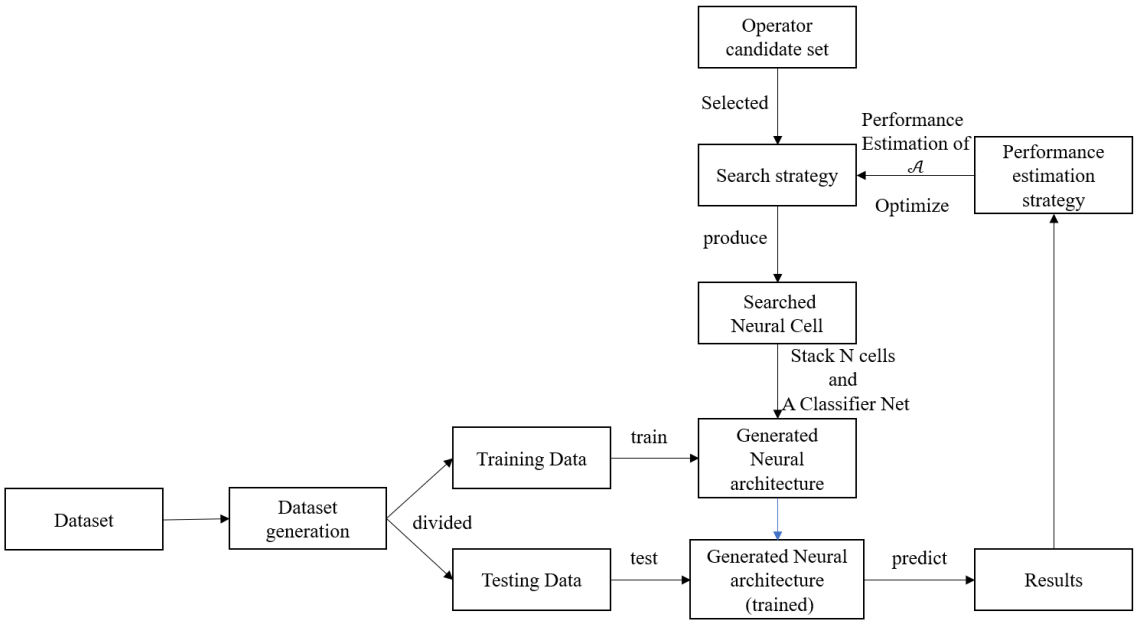


Figure 6.7 Workflow of searching neural architectures

Motivated by hand-crafted architectures consisting of repeated structures [47], the proposed NAS method search neural cells rather than searching entire architectures directly. Each learned cell is a directed acyclic graph consisting of an ordered sequence of N nodes. Each node is selected from a set of operators. The learned neural cell could be stacked to generate a deep neural network. The proposed method follows the design of the high-level structure of well-known manually designed architecture [49] and uses cells within such architectures.

The reason for searching cells rather than entire architectures is that training a macro NAS could take more than 1000 GPU hours[77]. To decrease the computational cost of the searching process, the proposed method follows the idea of [77] which

assumes that the final architecture consists of the sub-networks with the same structure.

The proposed method uses $\mathcal{C} = \{e^{i,j} | 1 \leq i < j \leq n\}$ to indicate a cell with n nodes. A directed edge is denoted as $e^{i,j}$ which is from the i-th node to the j-th node. The k-th candidate operator o_k comes from a pre-defined operator set $\mathcal{O} = \{o_1, \ldots, o_K | 1 \leq k \leq K\}$. Thus, the output of the j-th node and the i-th node is represented by I_j and I_i. If the j-th node only has one input edge from the i-th node, the output can be defined as[193]:

$$I_j = \sum_{i<j} o^{i,j}(I_i) \tag{6.2}$$

The operation calculated from the i-th node to the j-th node is represented by $o^{i,j}(.)$ which is a linear combination of all of the elements of the selected operator set $\mathcal{O}$. It can be denoted as Eq (6.3) by following[193].

$$o^{i,j}(I_i) = \sum_{k=1}^{K} \alpha_k^{i,j} \cdot o_k(I_i) \quad \text{s.t.} \ \alpha_k^{i,j} \in \{0.1\} \tag{6.3}$$

The $\alpha_k^{i,j}$ are used to represent binary parameters to select whether an operator will be used in a particular edge. And, s.t. stand for "subject to" to represent the formula under the constraint of $\alpha_k^{i,j} \in \{0.1\}$. Then, the neural architecture can be represented by a binary set by following[193]:

$$\mathcal{A} = \{\alpha_k^{i,j} | 1 \leq k \leq K, 1 \leq i < j \leq n\} \tag{6.4}$$

In the proposed work, each neural architecture consisting of some candidate operators will be transferred to an unique architecture code $\mathcal{A}$ and the proposed NAS will also consider $\sum_{k=1}^{K} \alpha_k^{i,j} = 1$ as a constraint to confirm that only one operator will be chosen on each edge. The operator selection problem for each edge can be considered as a classification problem [194]. Thus, the NAS can be formulated as

$$\min_{\alpha \in \mathcal{A}} \min_{w_a} \mathcal{L}(a, w_a) \tag{6.5}$$

For the various architectures $\alpha \in \mathcal{A}$ with trainable parameter w_a, the neural architecture search is selecting an optimal architecture that can achieve the minimal loss $\mathcal{L}(a, w_a)$. [193]

Operator candidate set for PAD

The scale of a search space for neural cells, which includes the number of the possible configurations, is related to (1) The number of nodes for the directed acyclic graph (2) the number of candidate operators (e.g. different type of convolution layers, different pooling layers, etc.).

By considering a neural cell as a directed acyclic graph, the selection of a fixed node number for the neural cell is considered as a hyper-parameter in the proposed method to limit the scale of the searching space. The selection of candidate operators will affect the computational complexity for the generated neural architectures. In the proposed NAS for PAD experiments, each neural cell will include two input nodes, one output node, and 4 normal nodes. Meanwhile, the proposed method considers the point-wise convolution operator and depth-wise separate convolution operator as the candidate convolution operator to replace the original convolution operation.

Various convolutional operators can be considered in the operator set in the literature. However, the proposed NAS process aims to balance the computational cost and the performance in the generated neural architectures. Selecting two modified convolution operators from some recent work (MobileNet [57]) can decrease the computational cost for the generated neural architecture. Also, the dilated convolution operator [59] is considered as the operator candidate to decrease the model size and allow the generated neural architecture can learn a generalizable expressive feature space. To further decrease the size of the operator set, the ReLU activation function[52] is integrated into the depth-wise convolution operator and the dilated convolution operator. The kernel size of the convolution operators are fixed to 3×3 and 5×5 pixels as suggested in [195].

Two pooling operators (3x3 average pooling and 3x3 max pooling) are used in the candidate operator set by following the suggestion of Barret Zoph et al.'s work[77]. After selecting the candidate operator set, the possible structure of a cell can be represented by the directed acyclic graphs.

Continuous Search Space

The continues search space, which allows the search strategy optimized by using gradient information, is important for the proposed NAS method. Selecting an operator for each edge of the neural cell is a discrete problem that cannot produce gradient flow to optimise the search strategy. The proposed NAS process follows the method of [193] to convert the discrete neural architecture searching problem into a continuous optimisation problem. Assuming there is an optimal neural architecture $\check{\mathcal{A}}$ for the PAD task, the proposed NAS method aims to approximating this optimal model by relaxing the categorical choice of an operator to a continuous probability representation. Thus, the choice of a set of operators can be formulated as (6.6) [196]:

$$\begin{aligned} \tilde{o}^{i,j}(I_i) &= \sum_{k=1}^{K} f(p_k^{i,j}) \cdot o_k(I_i) \\ s.t. \quad & \sum_{k=1}^{K} p_k^{i,j} = 1 \\ & p_k^{i,j} \geq 0, \forall 1 \leq k \leq K \\ & f\left(p_k^{i,j}\right) \in \{0.1\}, \forall 1 \leq k \leq K \end{aligned} \tag{6.6}$$

Formula (6.6) considers $p_k^{i,j}$ to denote the probability score of selecting the *k*-th operator in $\mathcal{O}$ on the edge $e^{i,j}$. Then $f(.)$ is used for mapping this probability score to a binary code. However, formula (6.6) still not differentiable.

The proposed method follows the suggestion in [197] and uses the Gumbel-Max[198] trick to re-formulate the estimation of $\alpha_k^{i,j}$. This process aims to enable the gradient information comes from back-propagation to optimise the search strategy. The standard Gumbel random variables can be sampled from the Gumbel distribution:

$$G = -\log(-\log(X)) \tag{6.7}$$

where X is used to represent the independent variable $X \sim U[0,1]$. Then, the discrete variable $\tilde{\alpha}_k^{i,j}$ can be sampled by (6.8):

$$\tilde{\alpha}_k^{i,j} \approx \mathcal{L} = \arg \max_{k \in \{1,\ldots,K\}} \log(p_k^{i,j}) + G_k \tag{6.8}$$

where $\{G_k\}_{k<K}$ is a sequence of standard Gumbel random variables. To make argmax operation continuous, the estimation function can be re-formulated as the Gumbel-Softmax (GS) estimation[199]:

$$\tilde{\alpha}_k^{i,j} \approx \tilde{\mathcal{L}}_k = \frac{\exp((\log(p_k^{i,j}) + G_k)/\tau)}{\sum_{k=1}^{K} \exp((\log(p_k^{i,j}) + G_k)/\tau))} \tag{6.9}$$

where τ is used to denote a temperature parameter to control $[\tilde{\mathcal{L}}_1, \ldots, \tilde{\mathcal{L}}_k, \ldots, \tilde{\mathcal{L}}_K]$ set and make this set approaching to a one-hot vector ($\tau \to 0$) or a discrete uniform distribution ($\tau \to +\infty$). $\tilde{\mathcal{L}}_k$ indicates that the probability score for the *k*-th operator $p_k^{i,j}$ is the maximal value in the vector. By using Gumbel-Softmax estimation, the search strategy is relaxed to a differentiable function.

Optimisation function

For PAD, the desired neural architecture should not only be accurate and effective for detecting presentation attacks but also achieve this with a minimum computation effort for widest deployment potential. In order to achieve this goal, the proposed method provides a modified lose function as following:

$$\mathcal{L}(a, w_a) = CE(a, w_a) + \lambda \cdot \sum c^{i,j} \tag{6.10}$$

where $CE(a, w_a)$is used to represent the cross-entropy for the searched neural architecture and $c^{i,j}$ is used to represent the computational complexity for the selected operator from node *i* to node *j*. λ is a parameter to balance these two parts and for the proposed method it is fixed at $\lambda = 0.5$. The effectiveness is represented by the gradient. Here, the computational complexity of an operator is represented by adding the number of trainable parameters in this layer and the floating-point operations per second (FLOPs) for the forward propagation. For instance, a convolutional layer with 3×3@10 convolutional kernels for an image which includes 5×5 pixels as input data, the number of trainable parameters is $3 \times 3 \times 10 = 90$. Meanwhile, doing the convolution operation once requires 3×3=9 times multiplication operations and 3×3-1 times add operation. And the input data needs the convolution operation to be performed $(5 - 3 + 1) \times (5 - 3 + 1) = 9$ times for each kernel. The $c^{i,j} = \zeta \cdot$

$(\text{parameter number} + \text{operation times}) = 0.001 \times 180 = 0.18$ when ζ is fixed to 0.001 in the proposed experiment to map the proposed score within range [0,1]. Thus, the proposed method will optimize the computational costs and cell architectures simultaneously by using the formula (6.10).

Reduction cell to accelerate the NAS

Some researchers claimed that there are two kind of cells that need to be searched in NAS process[197]: normal cell and reduction cell. Some deep neural networks designed by human experts show that the reduction layers, such as the max pooling layer in VGG net[47] and the stride connection layer in ResNet [49], can improve the generalization capability of DNNs. The normal cell generally consists of convolution layers to extract information from raw input. The reduction cell is designed to reduce the spatial dimension and makes the whole network efficient. By following the suggestion of[197], the proposed NAS also considers these two different cells to improve the performance and generalization capability for the generated neural architecture.

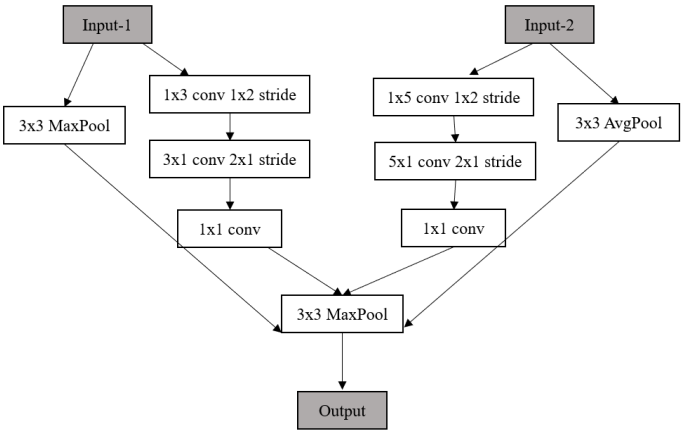


Figure 6.8 The reduction cell designed for the proposed neural architecture search method. 1x3 conv 1x2 stride indicates a convolutional layer with 1 by 3 kernel and 1 by 2 stride; 1x5 conv 1x2 stride indicates a convolutional layer with 1 by 5 kernel and 1 by 2 stride

However, searching two different cells are difficult with computational cost[197]. The proposed method follows the suggestion of [197] and introduces a simple reduce cell which is designed by human expert. This simple reduced cell also selects operators from the candidate operator set and the Fig 6.8 shows this reduced cell. In the proposed Model for Blocks and Model for Faces in Figure 6.6, the

generated neural architecture is designed as Figure 6.9, where the reduction cell will be added after Normal cell blocks.

6.3.2 Implementation for PAD

Data augmentation techniques are introduced here for the NAS to further improve the volume of the training set. In the training stage I, the proposed Model for Blocks and Model for Faces follows the neural architecture in Fig.6.9 with N=4 and uses a fixed 3X3 convolution operator as the input layer. The simple reduction cells designed via human hand are stacked after each normal cell blocks to reduce the dimension of the feature vector. The Classifier Net consists of a dense layer and a SoftMax layer[166]. As mentioned before, the Classifier Net for Blocks, Classifier Net for Faces, and the Classifier Net in the second training stage share the same neural architecture but trained for different data. Since CASIA-FASD[122] and Replay-Attack [27] are video datasets, the proposed experiment consider 10 random frames from each video to extract facial region and train the model for faces. Each facial region can provide 9 face blocks and the proposed experiment randomly select 4 blocks to generate the dataset for blocks. For the images in MSU-MFSD[123], we extract 64 blocks from each live face region, and eight blocks are randomly selected from each face region.

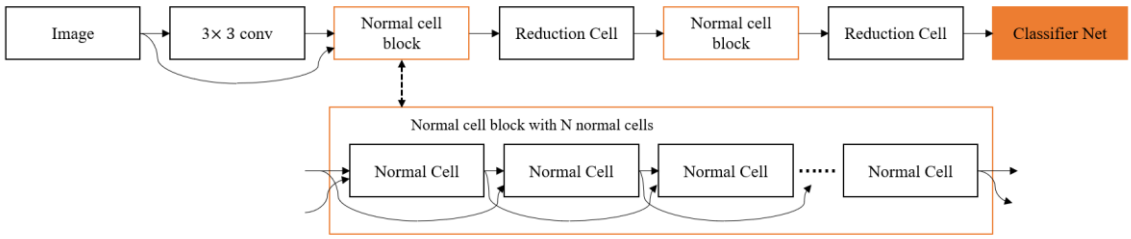


Figure 6.9 The generated neural architecture where the normal cell block includes N cells.

In the training stage II, the Feature Encoder Sub-network (a) and (b) is not trainable. The proposed experiment only allows the Classifier Net learn from gradient which is different from the previous transfer learning paradigm.

The candidate operator set has 8 different functions as (1) identity, (2) zeroize, (3) 3x3 depth-wise separate conv, (4) 3x3 dilated depth-wise separate conv, (5) 5x5 depth-wise separate conv, (6) 5x5 dilated depth-wise separate conv, (7) 3x3 average pooling, (8) 3x3 max pooling.

The NAS is trained for 200 epochs in total. The parameter within the neural network is optimised by using a SGD algorithm [200], which is initialized with an initial learning rate of 0.03. Then, the learning rate will down to 1×10^{-3} by following a cosine schedule. The momentum is set as 0.9 and the weight decay of 3×10^{-4}. The neural structure is searched with the Adam optimization algorithm [201] with the same learning rate with SGD and the weight decay of 1×10^{-3}.τ is initialized as 10 and is linearly reduced to 1. Following[193], the proposed method runs the NAS algorithm 10 times with different random seeds and considers the best cell for the following experiments.

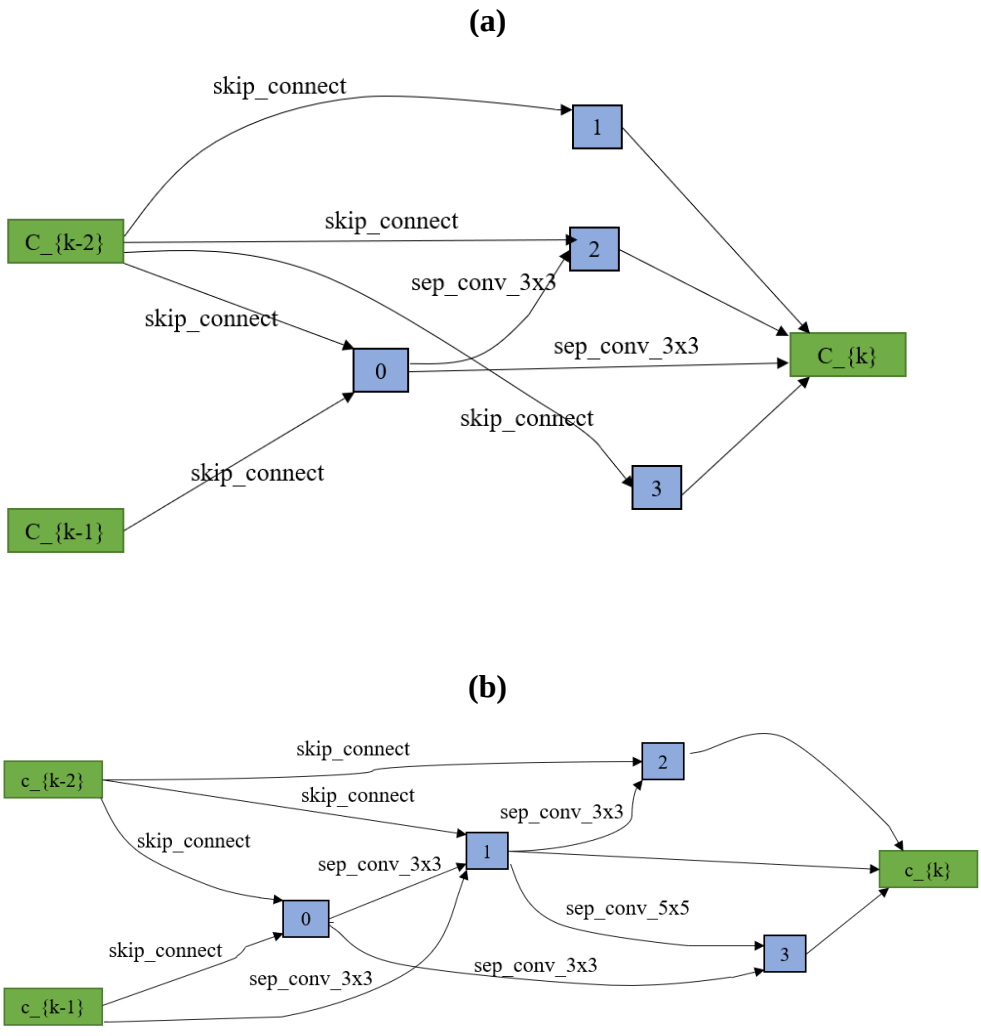


Figure 6.10 Cell discovery by the proposed neural architecture search method. Each neural cell will have two inputs: one from the previous cell, and another from the residual connection. The green blocks are used to demonstrate the start and the end point of a neural cell. Each connection between two nodes is an operator selected from the operator set.

6.3.3 Experiments

In this section, we describe the experimental design and implementation details used to evaluate the proposed NAS framework. The results of the experiments are also presented.

Three face spoofing detection databases were used for performance evaluations: (1) REPLAY-ATTACK dataset[27] , (2) CASIA-FASD[122], (3) MSU mobile face spoofing database[123]. The proposed experiments follow the protocol associated with each of the three databases. For each database, we use the training set to learn the CNN models and the testing set for evaluation in terms of EER and HTER. The Replay-Attack database [27] provides a development set which is only used as a validation set during training to ensure convergence of the network.

The process for discovering computational cells is presented in Figure 6.10. The automatically discovered cells are complex but can achieve better performance than the human-designed networks. The PAD-NAS network provides some encouraging results at three benchmark datasets which can be found at Table 6.4.

There are three baseline results which are considered in this table: (1) LBP baseline[24] which is widely considered as the baseline method in multiple datasets (2)The BPT represent the best performance score provided by the proposed traditional features in this thesis, and (3) the DTL-PAD(VGG16) is a neural architecture which follows the transfer learning paradigm and use the pre-trained feature extraction part of the VGG16 network. The proposed PAD-NAS network shows better the performance at all three datasets when comparing with the baseline methods. This suggests the NAS is a possible direction for future PAD research.

Table 6.4 Performance Comparison For PAD-NAS (BPT* indicate the best performance of the proposed traditional features)

	CASIA (EER %)	**Replay-Attack (EER %)**	**MSU (EER %)**
LBP(baseline) [24]	24.8	16.1	14.7
BPT*	4.8	0.6	7.6
DTL-PAD(VGG16) (baseline)	7.1	8.4	16.0
P&D-CNN[61]	2.6	0.7	0.35
Patch-3DCNN(Chapter 5)	4.2	0.1	1.7
DTL-PAD(NAS) (Chapter 5)	8.0	9.4	14.3
PAD-NAS (proposed)	2.3	**0.4**	1.9

The Patch-3DCNN and P&D-CNN[61] also consider both facial area and facial blocks as the proposed PAD-NAS network. The proposed PAD-NAS network get the best results at CASIA-FASD [122]when comparing with these two methods. P&D-CNN[61] shows better performance score at Replay-Attack dataset[27] and MSU dataset[123]. However, P&D-CNN needs to train a network to estimate 3D facial structure from the raw data which highly increase the computational cost of their method.

The DTL-PAD(NAS) is a neural architecture which follows the transfer learning paradigm and considers the pre-trained feature extraction part from NASNet [77]. NASNet also use neural architecture search method to design neural architecture. However, their neural architecture is searched for image classification task. The proposed PAD-NAS network shows better performance when comparing with DTL-PAD(NAS) method.

Table 6.4 summarizes the test errors of PAD-NAS compared with other approaches. As can be seen from the table, NAS-based encoder network successfully found architectures that outperformed other models for the same dataset. The performance scores demonstrate the effectiveness of the proposed PAD-NAS method.

6.4 SUMMARY

In this chapter, two different extensions for DNN-based PAD were introduced. First part of the chapter presented the experiments and results on interpretable PAD and the second part of the chapter presents automatic network design for PAD. These ideas were implemented and evaluated using frequently used public datasets.

The main contributions of this chapter are listed as follows:

(1) The traditional PAD approach may be extended to provide explainable decisions for presentation attack detection. An attention-based method which uses the VGG-16 pre-trained network as the feature encoder has been shown to be effective in this regard. This attention network is trained by using the explanations generated by the Grad-CAM method to show an improvement in performance. A natural language generation approach for explainable PAD was also explored which can help non-expert users to understand the decisions of a PAD system.

(2) An approach for the automatic design of deep neural structures for PAD tasks was presented and explored. NAS is firstly introduced to discover neural architectures for PAD. The learned architecture is quite flexible as it may be scaled in terms of computational cost and parameters to easily address a variety of applications. The performance of the resulting model is as good as the human-designed models in the proposed experiments.

Table 6.5 demonstrates the performance comparison for deep learning based methods at 4 benchmark datasets. The first column of the table 6.5 is the best results of the proposed traditional features. The DTL-PAD(VGG16) in the second row also applies transfer learning paradigm for PAD and consider the feature extraction part of the VGG16 pre-trained network as the feature encoder network. The BPT gives the best performance score provided by the proposed traditional features in this thesis

Table 6.5 Performance of the DNN based feature for multiple datasets (BPT* indicate the best performance of the proposed traditional features)

Datasets **EER(%)**	**REPLAY-ATTACK**	**CASIA-FASD**	**MSU-MFSD**	**HKBU MARs**
BPT	0.60	4.80	7.67	N/A
DTL-PAD(VGG16) (baseline)	8.4	7.1	16.0	39.7
FACN	**0.2**	3.02	1.67	23.70
DACN	0.37	**1.0**	**0.2**	**13.51**
NAS-PAD	0.4	2.3	1.9	16.10

As described in the previous chapters, some of the proposed methods use the same pre-trained feature encoder network but trained in different ways. By comparing the proposed methods with the DTP-PAD(VGG16), Table 6.4 can clearly demonstrate the performance improvements of the proposed methods. The attention mechanism and the “learning from explanation” pipeline highly improved the performance in this table. And the PAD-NAS network also shows encouraging results in these benchmark datasets.

In Table 6.5, the highlighted performance score is the best score when consider the proposed traditional and deep learning-based methods. The DACN method shows the best results at HKBU MARs dataset, CASIA-FASD[122] and MSU-MFSD[123]. Although the NAS-PAD does not reach the best performance when comparing with

the proposed methods, the searched architecture still demonstrates the effectiveness and the potential of the neural architecture search.

Chapter 7: Conclusions and Further Work

The objective of this thesis is to address the facial presentation attack detection task by exploring both conventional and deep neural network based methods. In this chapter, the proposed methods, conclusions and the possible directions for future works will be presented to summarise the contributions of the proposed works. In Section 7.1, the contributions of the proposed work and experiments are demonstrated. Then, the future work based on the existing results is suggested in Section 7.2

7.1 CONTRIBUTIONS

The main contributions of this thesis have been the development and evaluation of novel features for facial presentation attack detection. In order to have a clear understanding of PAD, existing methods and evaluation protocols, a comprehensive survey for software-based facial presentation attack detection, which includes both traditional features and deep learning features, is provided. After a brief analysis of the existing methods, an experimental framework is presented in detail where the benchmarking datasets and the evaluation metrics are also presented.

After the descriptions of the experimental framework, three main chapters are provided to demonstrate the contributions of this thesis. The details of the proposed methods are also listed in these chapters. Distinct differences between genuine faces and presentation attacks have been detected using temporal information such as facial motion patterns, moiré patterns, shading differences and specular reflections, as reported in the research literature. The initial point of the proposed methods is providing a feature which can efficiently explore temporal differences for presentation attack detection. To achieve this goal, the thesis explores traditional features along two directions: (1) **Unconscious facial movements** are distinct for various presentation attacks and the proposed *FAUH* method uses the facial action coding system, which is a symbolic system for possible facial motions, and extracts a temporal-related feature for detecting PAs. (2) **Temporal texture changes** also contain useful information for presentation attack detection but processing temporal information is computationally expensive. Three traditional methods are proposed to achieve a balance between performance and computational costs. The *Motion History Patterns* combine the

Motion History Image (MHI) as primary features and two local texture descriptors as secondary features to detect presentation attacks. The *TCoALBP* extends the original LBP algorithm as a spatio-temporal texture descriptor for video data. This novel algorithm captures and summarises dynamic textural characteristics of a video sequence by encoding the co-occurrence of local texture features both in space and across time, as contained in a sequence of video frames. The *Super-pixel texture pattern* segments the raw input as a set of super-pixels by using the clustering algorithm and generates the final feature vector from the codebook representation of the local texture representations for each super-pixel. All of these three proposed methods are focusing on the temporal texture patterns which are discriminative for different presentation attacks.

The emergence of deep learning techniques offers some new opportunities for PAD research. One of these is the possibility of obtaining robust features for the classification of presentation attacks. Here two widely used learning paradigms in deep learning are explored:

(1) **Transfer learning paradigm** uses the feature extraction part of various pre-trained deep neural networks and trains a new classifier sub-network that follows the feature extraction network. Then, a fine-tuning stage is applied to further improve the performance of the overall deep neural network. The pre-trained feature extraction networks are normally trained with large datasets for other computer vision tasks such as object recognition.

(2) **Learning from scratch paradigm** only uses PA datasets as the training data and design some novel neural architectures for the facial anti-spoofing task. The proposed CCPAD-Net provides a novel neural architecture, which can detect presentation attacks efficiently, and a Colour Space Net is designed, which can be trained separately, to decrease the risk of overfitting.

The spatio-temporal information is also considered to guide the designing of the proposed neural architectures: (a) *FASAN* follows the assumptions and analysis of *FAUH* but employs a novel neural architecture, which combines the LSTMs and CNNs for the proposed FAU temporal intensity signals. The proposed FASAN shows a clear performances improvement when compared with the FAUH and demonstrates the effectiveness of DNN-based methods for processing temporal information. (b) The *3D Temporal Local Texture Network* uses the distinct motion cues for presentation attacks

associated with both spatial and temporal variations and explores a novel 3D convolutional neural architecture that is used to model the distinct texture correlations between frames.

Other significant contributions of this thesis are concerned with improving the performance of the PAD system as well as making the system (1) have the capability to justify its decisions and (2) have the capability to automatically select the neural architectures for PAD without the need for human design. Despite the high performance achieved using DNNs, the inability to justify decisions is a significant drawback given the usability and security requirements of many biometric applications. The proposed *learning from explanations* approach utilises both spatial and temporal information to detect facial spoofing behaviours and provides both visual and natural language explanations for each decision to answer the questions such as "Why the system makes this decision?". An attention-guided subnetwork is designed in the proposed work to learn from justifications provided by the system as an additional information to further improve performance. The proposed *NAS-PAD* is considered as a logical next step for automating PAD when researchers and engineers are struggling with the complexity of designing an effective neural network. This proposed method can learn an efficient neural architecture by searching the structures of the neural cells. The full network for PAD is created by stacking the searched cells.

As the description in the introduction chapter, this thesis aims to push the boundary of existed PAD researches and to find a possible route to the next generation of the PAD system in my imagination. For this reason, both conventional and neural architectures are explored to improve the performance of PAD systems. Some neural architectures are trained by using additional information, which is extracted by using domain knowledge. Some neural architectures are designed or searched for the platform with low computational capabilities. Moreover, some contributions demonstrate the potential and the possibilities of the explainable PAD system. I hope these works can help people to build better PAD system in the future. In summary, contribution chapters explored various conventional and neural architectures to improve the performance of PAD systems in this thesis. Temporal information was used efficiently in multiple experiments (such as Motion History Patterns, FASAN, etc.) and showed the potential benefits of using Temporal information. Meanwhile, the explainable-PAD system was designed to open the black-box of DNNs, and the

NAS-PAD system offers a possible way to design a deep neural network for PAD without human experts.

7.2 FUTURE WORK

Developing traditional features using temporal information for PAD may still be a very interesting topic in the near feature due to their computational efficiency property. Especially for the biometric systems within the robotic system, such as Quadrone[202] and other possible robot platforms, the restricted computational resources and limited battery capacity justifies the use of traditional features in some scenarios.

For the traditional features explored in this thesis, more experiments may be needed to optimise the parameters such as different colour channels. For future work, the effect of different temporal and spatial displacements for the proposed traditional features will be studied using larger and more challenging datasets. Also, heuristic search algorithms may be explored for optimizing parameter sets of the proposed methods. The proposed FAUH feature can be optimised by selecting different combinations of action units, different temporal durations, and different facial action unit recognisers with better performance. The Motion History Pattern can be tested with different secondary features or some secondary features can even be fused for better performance. The optimum temporal duration for each MHI, the type of colour channels and the parameters such as the threshold for each MHI can also be optimised for different attack types. More experiments are needed to optimise TCoALBP feature by selecting better combinations of different colour channels, choosing better hyper-parameters for LBP descriptors, and choosing different 3D local texture descriptors.

Recent developments of edge computing and tensor processing units (TPU) will accelerate the developments of deep learning based methods for mobile devices. The deep learning computational methods are becoming more sophisticated and powerful at the same time. The data dependence characteristics of these techniques will push researchers to collect more data. Some recent work [97] has started to use meta-learning approaches for PAD to achieve good performance with very limited training data.

The future direction for the methods based on deep learning may include the exploration of small networks for mobile platforms and large networks for server

platforms. For the mobile platforms the deep learning models should be smaller and incorporate efficient operations designed for the mobile computing. The proposed optimisation function in NAS-PAD can provide a better way to balance performance and computational complexity.

Designing a neural architecture to detect particular characteristics, such as moiré pattern or skin reflections, may also be a possible research direction in the future. This direction may affect both small networks for mobile platforms and large networks for server platforms. Neural architectures may include multiple sub-networks, which can be trained separately with additional information.

Interpretability of detection decisions will be important for the next generation PAD system, especially, for large neural networks running at server platforms. “Learning from explanation” may need a better explanation generation approach. The learning process of the “learning from explanation” can also be optimised, such as through exploring different optimisation functions, to help the proposed system achieve better performance.

References

[1] Y. Sun, Y. Chen, X. Wang, and X. Tang. "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, vol. 2, pp.1988-1996, 2014.

[2] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in Workshop on Faces in "Real-life" images: Detection, Alignment and Recognition, Marseille, France, Oct 2008.

[3] A. Etzioni, "Apple: Good business, poor citizen?" *J. Bus. Ethics,* vol. 151, issue 1, pp. 1-11, 2018.

[4] H. Ortiz, "An anthropology of chinese digital payment systems, wechat pay and alipay," in 2019, [Online]. Available: https://hal.archives-ouvertes.fr/hal-02194865/. [Accessed: 28-Nov-2019].

[5] N. B. Ellison, C. Steinfield and C. Lampe, "The benefits of Facebook "friends:" Social capital and college students' use of online social network sites," *Journal of Computer-Mediated Communication,* vol. 12, No.4, pp. 1143-1168, 2007.

[6] A. K. Jain, P. Flynn and A. A. Ross, *Handbook of Biometrics.* Springer Science & Business Media, 2007.

[7] D. Maltoni, M. S. Nixon and S.Z. Li, *Handbook of Fingerprint Recognition.* Springer Science & Business Media, 2009.

[8] A. K. Jain, A. Ross and S. Pankanti. "Biometrics: a tool for information security," *IEEE Transactions on Information Forensics and Security,* vol. 1, *(2),* pp. 125-143, 2006.

[9] S. Marcel, M. S. Nixon and S. Z. Li, *Handbook of Biometric Anti-Spoofing.* New York: Springer. vol 1, 2014.

[10] G. Költzsch. "Biometrics–market segments and applications," *Journal of Business Economics and Management,* Vol. VIII, No. 2, pp. 119-122, 2007.

[11] T. Caldwell. "Market report: border biometrics," *Biometric Technology Today,* vol. 2015, No. 5, pp. 5-11, 2015.

[12] G. M. Ezovski and S. E. Watkins. "The electronic passport and the future of government-issued RFID-based identification," in *2007 IEEE International Conference on RFID,* Grapevine, Texas, USA, March, 2007.

[13] H. Siringoringo and H. M. Valentine. "Electronic passport system acceptance: an empirical study from Indonesia," *International Journal of Electronic Governance,* vol. 10, No.3, pp. 261-275, 2018.

[14] International Organization for Standardization, "Information Technology – Biometric presentation attack detection – Part 3: Testing and reporting," *JTC 1/SC 37, Geneva, Switzerland,* vol. ISO/IEC FDIS 30107-3:2017, 2017.

[15] N. M. Duc and B. Q. Minh. "Your face is not your password face authentication bypassing lenovo–asus–toshiba," *Black Hat Briefings,* vol. 4, pp. 158, 2009.

[16] D. J. Bernstein, C. Tung, C. Chitchanok, H. Andreas, L. Tanja, N. Ruben and V. Christine. "How to manipulate curve standards: A white paper for the black hat. " in *International Conference on Research in Security Standardisation,* Toyko, Japan, Springer, Cham, pp 109-139, 2015.

[17] A. Greenberg. "Hackers just broke the iPhone X's face ID using a 3D-printed mask," *Wired,* 2017.

[18] Z. Akhtar, *et al.*, "Biometrics: In search of identity and security (Q & A)," *IEEE Multimedia,* pp 1-1, 2017.

[19] J. Galbally, S. Marcel and J. Fierrez, "Biometric antispoofing methods: A survey in face recognition," *IEEE Access,* vol. 2, pp. 1530-1552, 2014.

[20] R. Ramachandra and C. Busch, "Presentation attack detection methods for face recognition systems: A comprehensive survey," *ACM Computing Surveys (CSUR),* vol. 50, No.1, pp.1-37, 2017.

[21] *ThatsMyFace*.in 2018. [Online] Available: https://www.thatsmyface.com. [Accessed: 20-Dec-2018]

[22] N. Evans. *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection.* Springer, 2019.

[23] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi and S.Z. Li. "A face antispoofing database with diverse attacks," in *2012 5th IAPR International Conference on Biometrics (ICB),* New Delhi, India pp. 26-31, 2012.

[24] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* Vol. 24, No.7, pp. 971-987, 2002.

[25] T. Ahonen, A. Hadid and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* Vol. 28, No. 12, pp. 2037-2041, 2006.

[26] J. Määttä, A. Hadid and M. Pietikäinen, "Face spoofing detection from single images using texture and local shape analysis," *IET Biometrics,* Vol. 1, Issue 1, pp. 3-10, 2012.

[27] I. Chingovska, A. Anjos and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG-Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG),* pp. 1-7, Sep, 2012.

[28] N. Erdogmus and S. Marcel, "Spoofing face recognition with 3D masks," *IEEE Transactions on Information Forensics and Security,* vol. 9, No. 7, pp. 1084-1097, Jul 2014.

[29] N. Kose and J. Dugelay, "Classification of captured and recaptured images to detect photograph spoofing," in *2012 International Conference on Informatics, Electronics & Vision (ICIEV)* Dhaka, Bangladesh, May 2012, pp. 1027-1032.

[30] R. Raghavendra, K.B. Raja, S. Venkatesh, F.A. Cheikh and C. Busch. "On the vulnerability of extended multispectral face recognition systems towards presentation attacks," in *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA),* New Delhi, India, Feb, 2017, pp.1-8.

[31] M. Waris, H. Zhang, I. Ahmad, S. Kiranyaz and M. Gabbouj. "Analysis of textural features for face biometric anti-spoofing," in *21st European Signal Processing Conference (EUSIPCO 2013),* Marrakech, Morocco, Sep, 2013, pp. 1-5.

[32] Z. Boulkenafet, J. Komulainen and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security,* vol. 11, No. 8, pp. 1818-1830, Aug, 2016.

[33] J. Li, Y. Wang, T. Tan, and A. K. Jain. "Live face detection based on the analysis of fourier spectra," in *Biometric Technology for Human Identification,* Proc of SPIE, Vol 5404, pp296-303, 2004.

[34] M. H. Teja. "Real-time live face detection using face template matching and DCT energy analysis," in *2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR),* Dalian, China, Oct, 2011, pp. 342-346.

[35] Z. Zhang, D. Yi, Z. Lei, and S.Z. Li. "Face liveness detection by learning multispectral reflectance distributions." in *International Conference on Face and Gestures,* Santa Barbara, USA, Mar, 2011, pp. 436-441.

[36] J. Peng and P. P. Chan. "Face liveness detection for combating the spoofing attack in face recognition," in *2014 International Conference on Wavelet Analysis and Pattern Recognition,* Lanzhou, China, Jul, 2014, pp. 176-181.

[37] X. Tan, *et al.*, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision,* Crete, Greece, 2010, pp. 504-517.

[38] D. Wen, H. Han and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security,* vol. 10, Issue 4, pp. 746-761, Apr, 2015.

[39] J. Komulainen, Z. Boulkenafet and Z. Akhtar, "Review of face presentation attack detection competitions," in *Handbook of Biometric Anti-Spoofing* Springer, Cham, pp. 291-317, 2019.

[40] T. Wang, J. Yang, Z. Lei, S. Liao, and S. Z. Li. "Face liveness detection using 3D structure recovered from a single camera," in *2013 International Conference on Biometrics (ICB)*, Madrid, Spain, Jun, 2013, pp. 1-6.

[41] K. T. Nguyen, C. Zitzmann, F. Retraint, A. Delahaies, F. Morain-Nicolier, and H.P. Nguyen. "Face spoofing detection for smartphones using a 3D reconstruction and the motion sensors." in *International Conference on Systems Security and Privacy ICISSP*, Funchal, Madeira, Portugal, 2018, pp. 286-291.

[42] Y. Xu, T. Price, J.M. Frahm, and F. Monrose. "Virtual u: Defeating face liveness detection by building virtual models from your public photos," in *25th Security Symposium (Security 16)*, Austin, TX, USA, 2016, pp. 497-512.

[43] T. de Freitas Pereira, A. Anjos, J.M. De Martino, and S, Marcel. "LBP− TOP based countermeasure against face spoofing attacks," in Workshop of 2012 *Asian Conference on Computer Vision*, Daejeon, Korea, Nov, 2012, pp.121-132.

[44] Q. Phan, D. T. Dang-Nguyen, G. Boato, and F.G. De Natale. "Face spoofing detection using LDP-TOP," in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, Arizona, Sep, 2016, p404-408.

[45] S. R. Arashloo, J. Kittler and W. Christmas, "Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features," *IEEE Transactions on Information Forensics and Security*, vol. 10, Issue 11 , pp. 2396-2407, Nov, 2015.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and A.C. Berg. "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, Issue 3, pp. 211-252, Dec, 2015.

[47] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition," *arXiv Preprint arXiv:1409.1556*, 2014.

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and A, Rabinovich. "Going deeper with convolutions," in *2015 7th International Conference on Games & Virtual Worlds for Serious Applications (VS-Games)*, Skovde, Sweden, 2015, pp.1-5.

[49] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA, Jun, 2016, pp.779-778,.

[50] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, Zürich, Switzerland, 2014, pp. 818-833.

[51] F. Yu and V. Koltun. "Multi-scale context aggregation by dilated convolutions," *arXiv Preprint arXiv:1511.07122*, 2015.

[52] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, 2010, pp.807-814.

[53] D. Clevert, T. Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv Preprint arXiv:1511.07289,* 2015.

[54] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision,* Amsterdam, Netherlands, Oct, 2016, pp. 525-542.

[55] S. Lin, R. Ji, Y. Li, Y. Wu, F. Huang and B. Zhang. "Accelerating convolutional networks via global & dynamic filter pruning." in *International Joint Conference on Artificial Intelligence* (*IJCAI),* Stockholm, Sweden, Jul, 2018, pp. 2425-2432.

[56] F. N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally and K. Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv Preprint arXiv:1602.07360,* 2016.

[57] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand and H. Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv Preprint arXiv:1704.04861,* 2017.

[58] X. Zhang, X. Zhou, M. Lin and J. Sun. "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *the IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah, USA, Jun,2018, pp.6848-6856.

[59] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks," in *the IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah, USA, Jun, 2018, pp. 4510-4520.

[60] A. Alotaibi and A. Mahmood. "Deep face liveness detection based on nonlinear diffusion using convolution neural network," *Signal, Image and Video Processing,* vol. 11, Issue 4, pp. 713-720, May, 2017.

[61] Y. Atoum., Y. Liu, A. Jourabloo and X. Liu. "Face anti-spoofing using patch and depth-based CNNs," in *2017 IEEE International Joint Conference on Biometrics (IJCB),* Denver, Colorado, USA, Oct,2017, pp. 319-328.

[62] Y. Liu, A. Jourabloo and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *the IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah, USA, Jun, 2018, pp. 389-398.

[63] K. Weiss, T. M. Khoshgoftaar and D. Wang, "A survey of transfer learning," *Journal of Big Data,* vol. 3, issue 1, pp. 9, 2016.

[64] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and Fei-Fei Li. "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition,* Miami, Florida, USA, Jun, 2009, pp.248-255.

[65] Y. Bengio. "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, Springer Verlag, p437-478, 2012.

[66] T. M. Breuel. "The effects of hyperparameters on SGD training of neural networks," *arXiv Preprint arXiv:1508.02788,* 2015.

[67] J. Yang, Z. Lei and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv Preprint arXiv:1408.5601,* 2014.

[68] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li and A. Hadid. "An original face anti-spoofing approach using partial convolutional neural network," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA),* Oulu, Finland, Dec, 2016, pp.1-6.

[69] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep face recognition." in *British Machine Vision Conference (BMVC),* Swansea, UK, Sep,2015, Vol 1, No. 3, pp.6.

[70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, Issue 8, pp. 1735-1780, 1997.

[71] H. Yu, J. Wang, Z. Huang, Y. Yang and W. Xu. "Video paragraph captioning using hierarchical recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, Nevada. USA, pp. 4584-4593 Jun, 2016.

[72] Z. Xu, S. Li and W. Deng, "Learning temporal features using LSTM-CNN architecture for face anti-spoofing," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR),* Kuala Lumpur, Malaysia, 2015, pp.141-145.

[73] S. Luo, M. Kan, S. Wu, X. Chen and S. Shan. "Face anti-spoofing with multi-scale information," in *2018 24th International Conference on Pattern Recognition (ICPR),* Beijing, China, Aug, 2018, pp. 3402-3407.

[74] X. Tu *et al*, "Enhance the Motion Cues for Face Anti-Spoofing using CNN-LSTM Architecture," *arXiv Preprint arXiv:1901.05635,* 2019.

[75] M. Asim, Z. Ming and M. Y. Javed, "CNN based spatio-temporal feature extraction for face anti-spoofing," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC),* Chengdu, China, Jun, 2017, pp. 234-238.

[76] R. Shin, C. Packer and D. Song. "Differentiable neural network architecture search," in *ICLR Workshop,* Vancouver, Canada, 2018.

[77] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le. "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah, USA, 2018, pp.8697-8710.

[78] I. Guyon, K. Bennett, G. Cawley, H. J, Escalante, S. Escalera, T.K. Ho and E, Viegas. "Design of the 2015 chalearn automl challenge," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, Jul, 2015, pp.1-8.

[79] T. Elsken, J. H. Metzen and F. Hutter, "Neural Architecture Search: A Survey." *Journal of Machine Learning Research*, vol. 20(55), pp. 1-21, 2019.

[80] H. Cai. T, Chen., W, Zhang., Y, Yu., and J, Wang., "Efficient architecture search by network transformation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, Palo Alto, California USA, 2018.

[81] B. Baker *et al*, "Designing neural network architectures using reinforcement learning," *arXiv Preprint arXiv:1611.02167*, 2016.

[82] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv Preprint arXiv:1611.01578*, 2016.

[83] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le. "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah. USA, 2018, pp. 8697-8710.

[84] E. Real, A. Aggarwal, Y. Huang and Q.V. Le. "Regularized evolution for image classifier architecture search," in *the AAAI Conference on Artificial Intelligence*, Palo Alto, California USA, 2019, Vol. 33, pp. 4780-4789.

[85] C. Liu, *et al*, "Progressive neural architecture search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 19-34.

[86] B. Wu, *et al*, "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 10734-10742 2019.

[87] H. Liu, K. Simonyan and Y. Yang, "Darts: Differentiable architecture search," *arXiv Preprint arXiv:1806.09055*, 2018.

[88] D. Gunning, "Explainable artificial intelligence (xai): Technical report defense advanced research projects agency darpa-baa-16-53," *DARPA, Arlington, USA*, 2016.

[89] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138-52160, 2018.

[90] C. Xu *et al*, "UP-CNN: Un-pooling augmented convolutional neural network," *Pattern Recognition. Letter*. Vol. 119, p34-40Mar, 2019.

[91] R. R. Selvaraju *et al*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, pp. 618-626, 2017.

[92] M. T. Ribeiro, S. Singh and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Francisco, USA, Aug, 2016, pp. 1135-1144.

[93] A. Chattopadhay, A. Sarkar, P. Howlader and V.N. Balasubramanian. "Grad-cam: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV),* Lake Tahoe, USA, Mar, 2018, pp. 839-847.

[94] A. Jourabloo, Y. Liu and X. Liu, "Face de-spoofing: Anti-spoofing via noise modeling," in *Proceedings of the European Conference on Computer Vision (ECCV),* Munich, Germany, 2018, pp. 290-306.

[95] L. Li, P. L. Correia and A. Hadid, "Face recognition under spoofing attacks: countermeasures and research directions," *IET Biometrics,* vol. 7, *(1),* pp. 3-14, 2017.

[96] Y. Liu, J. Stehouwer, A. Jourabloo and X. Liu. "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* Long Beach, California, USA, 2019, pp. 4680-4689.

[97] C. Zhao *et al*, "Meta Anti-spoofing: Learning to Learn in Face Anti-spoofing," *arXiv Preprint arXiv:1904.12490,* 2019.

[98] D. L. MacAdam, "Color Measurement: Theme and Variations".Vol. 15(4), pp.321-321, *Springer* 2013.

[99] F. Chen *et al*, "Face liveness detection: fusing colour texture feature and deep feature," *IET Biometrics,* Vol 8(6), pp. 369-377, 2019.

[100] M. Fairchild, "Color appearance models: CIECAM02 and beyond," in *Tutorial Notes, IS&T/SID, Tekstilec;* Vol. 60 Issue 2, p97-106, 2004.

[101] A. Agarwal, R. Singh and M. Vatsa, "Face anti-spoofing using haralick features," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS),* Buffalo, New York, USA, 2016, pp.1-6.

[102] S. Bhattacharjee *et al*, "Recent advances in face presentation attack detection," in *Handbook of Biometric Anti-Spoofing* Anonymous, pp. 207-228, 2019.

[103] H. Li *et al*, "Learning generalized deep feature representation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security,* vol. 13*(10),* pp. 2639-2652, 2018.

[104] Y. LeCun, Y. Bengio and G. Hinton. "Deep learning," *Nature,* vol. 521*(7553),* pp. 436, 2015.

[105] S. Abu-El-Haija *et al.,* "Youtube-8m: A large-scale video classification benchmark," *arXiv Preprint arXiv:1609.08675,* 2016.

[106] K. Patel, H. Han and A. K. Jain, "Cross-database face antispoofing with robust feature representation," in *Chinese Conference on Biometric Recognition,* Chengdu China, Oct, 2016, pp. 611-619.

[107] M. Frid-Adar *et al*, "Synthetic data augmentation using GAN for improved liver lesion classification," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018),* Washington DC, USA, 2018, pp. 289-293.

[108] Q. Xie, Z. Dai, E. Hovy, M. T. Luong, and Q.V. Le. "Unsupervised data augmentation," *arXiv Preprint arXiv:1904.12848,* 2019.

[109] Y. Liu, A. Jourabloo and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *the IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah. USA, 2018, pp.389-398.

[110] A. George and S. Marcel, "Deep pixel-wise binary supervision for face presentation attack detection," in *International Conference on Biometrics,* Crete, Greece, 2019, pp. 1-8.

[111] T. Mita, T. Kaneko and O. Hori, "Joint haar-like features for face detection," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Beijing, China,* 2005, Volume 1, pp.1619-1626.

[112] G. Bradski and A. Kaehler, "OpenCV," *Dr.Dobb's Journal of Software Tools,* vol. 3, 2000.

[113] H. Jee, S. Jung and J. Yoo, "Liveness detection for embedded face recognition system," *International Journal of Biological and Medical Sciences,* vol. 1*(4),* pp. 235-238, 2006.

[114] T. Baltrusaitis, P. Robinson and L. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proceedings of the IEEE International Conference on Computer Vision Workshops,* Sydney, Australia, 2013.

[115] J. Peng, L. Bo and J. Xu, "Conditional neural fields," in *Advances in Neural Information Processing Systems,* Vancouver B.C., Canada, 2009, pp. 1419-1427.

[116] T. Qin *et al*, "Global ranking using continuous conditional random fields," in *Advances in Neural Information Processing Systems,* Vancouver B.C., Canada, 2009, pp. 1281-1288.

[117] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2*(3),* pp. 27, 2011.

[118] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Comput.,* vol. 3*(4),* pp. 461-483, 1991.

[119] M. Hu, Y. Chen and J. T. Kwok, "Building sparse multiple-kernel SVM classifiers," *IEEE Trans. Neural Networks,* vol. 20*(5),* pp. 827-839, 2009.

[120] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Limassol, Cyprus, Anonymous, 2010, pp. 177-186.

[121] X. Tan, Y. Li, J. Liu and L. Jiang. "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *European Conference on Computer Vision*, Crete, Greece, pp. 504-517, Berlin, Heidelberg, 2010.

[122] Z. Zhang *et al*, "A face antispoofing database with diverse attacks," in *2012 5th IAPR International Conference on Biometrics (ICB)*, New Delhi, India, 2012, pp. 26-31.

[123] K. Patel, H. Han and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11 *(10)*, pp. 2268-2283, 2016.

[124] H. Li *et al*, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13*(7)*, pp. 1794-1809, 2018.

[125] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng and A. Hadid. "OULU-NPU: A mobile face presentation attack database with real-world variations," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, Washington DC, USA, 2017, pp. 612-618.

[126] S. Liu *et al*, "A 3D mask face anti-spoofing database with real world variations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, Nevada, USA, pp. 100-106, 2016.

[127] A. Pinto *et al*, "Using visual rhythms for detecting video-based facial spoof attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10*(5)*, pp. 1025-1038, 2015.

[128] S. Zhang *et al*, "A dataset and benchmark for large-scale multi-modal face anti-spoofing," arXiv preprint arXiv:1812.00408.

[129] A. Mordvintsev, C. Olah and M. Tyka, "Inceptionism: Going deeper into neural networks," 2015, [online] Available: http://googleresearch.blogspot.com/2015/06/ inceptionism-going-deeper-into-neural.html. [Accessed: 2019-12-04]..

[130] C. Hjortsjö, *Man's Face and Mimic Language.* Sweden:Studentlitteratur, 1969.

[131] P. Ekman and W. V. Friesen, *Facial Action Coding System: Investigator's Guide.* 1978.

[132] M. Gavrilescu, "Study on using individual differences in facial expressions for a face recognition system immune to spoofing attacks," *IET Biometrics*, vol. 5*(3)*, pp. 236-242, 2016.

[133] J. F. Cohn, K. Schmidt, R. Gross and P. Ekman. "Individual differences in facial expression: Stability over time, relation to self-reported emotion, and ability to inform

person identification," in *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, Pittsburgh, USA, 2002, pp. 491-499.

[134] J. Yang *et al*, "Person-specific face antispoofing with subject domain adaptation," *IEEE Transactions on Information Forensics and Security*, vol. 10*(4)*, pp. 797-809, 2015.

[135] T. Baltrušaitis, M. Mahmoud and P. Robinson, "Cross-dataset learning and person-specific normalisation for automatic action unit detection," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, 2015, Vol. 6, pp. 1-6.

[136] B. Jiang, M. F. Valstar and M. Pantic, "Action unit detection using sparse appearance descriptors in space-time video volumes," in *Face and Gesture*, Santa Barbara, California, USA, 2011, pp. 314-321.

[137] A. Zadeh *et al.*, "Convolutional experts constrained local model for 3d facial landmark detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2519-2528.

[138] G. McKeown *et al.*, "The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent," *IEEE Transactions on Affective Computing*, vol. 3*(1)*, pp. 5-17, 2011.

[139] S. Tirunagari *et al.*, "Detection of face spoofing using visual dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 10*(4)*, pp. 762-777, 2015.

[140] S. Bharadwaj *et al.*, "Computationally efficient face spoofing detection with motion magnification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Oregon, Portland, 2013, pp. 105-110.

[141] N. Singla, "Motion detection based on frame difference method," *International Journal of Information & Computation Technology*, vol. 4*(15)*, pp. 1559-1565, 2014.

[142] M. A. R. Ahad, *et al.*, "Motion history image: its variants and applications," *Mach Vision Appl*, vol. 23 *(2)*, pp. 255-281, 2012.

[143] A. Bobick and J. Davis, "An appearance-based representation of action," in *Proceedings of 13th International Conference on Pattern Recognition*, Gold Coast in Australia, 1996, Vol. 1, pp. 307-312.

[144] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol 1(3)*, pp. 257-267, 2001.

[145] O. Lucena and R, Lotufo. "Transfer learning using convolutional neural networks for face anti-spoofing," in *International Conference Image Analysis and Recognition*, Montreal, QC, Canada, 2017, pp. 27-34.

[146] C. Tan *et al*, "A survey on deep transfer learning," in *International Conference on Artificial Neural Networks*, Rhodes, Greece, Oct, 2018, pp. 270-279.

[147] P. Baldi and P. J. Sadowski, "Understanding dropout," in *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, 2013, pp. 2814-2822.

[148] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using volume local binary patterns," in *Dynamical Vision*, Beijing, China, 2006, pp. 165-177.

[149] D. Tiwari and V. Tyagi, "Dynamic texture recognition based on completed volume local binary pattern," *Multidimension. Syst. Signal Process.*, vol. 27*(2)*, pp. 563-575, 2016.

[150] R. Nosaka, C. H. Suryanto and K. Fukui, "Rotation invariant co-occurrence among adjacent LBPs," in *Asian Conference on Computer Vision*, Daejeon, Korea, 2012, pp. 15-25.

[151] R. Achanta *et al*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34*(11)*, pp. 2274-2282, 2012.

[152] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV-2003*, Nice, France, 2003, Vol 1, pp. 10-17.

[153] O. Veksler, Y. Boykov and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in *European Conference on Computer Vision*, Crete, Greece, 2010, pp. 211-224.

[154] T. Li *et al*, "Contextual bag-of-words for visual categorization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21*(4)*, pp. 381-392, 2010.

[155] B. Leibe, A. Leonardis and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77*(1-3)*, pp. 259-289, 2008.

[156] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, USA, 2006, Vol. 2, pp. 2161-2168.

[157] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol 5*, pp. 603-619, 2002.

[158] J. Banerjee *et al*, "3D LBP-based rotationally invariant region description," in *Asian Conference on Computer Vision*, Daejeon, Korea, 2012, pp. 26-37.

[159] R. Raghavendra *et al*, "Transferable deep-CNN features for detecting digital and print-scanned morphed face images," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, Hawaii, USA, 2017, pp. 1822-1830.

[160] J. Huang *et al*, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013, pp. 7304-7308.

[161] C. Chen *et al*, "Vehicle type recognition based on multi-branch and multi-layer features," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2017, pp. 2038-2041.

[162] J. Yosinski *et al*, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, Montreal, Canada, 2014, Vol 1, pp. 3320-3328.

[163] Y. He *et al*, "Amc: Automl for model compression and acceleration on mobile devices," in *the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 784-800.

[164] M. Abadi *et al*, "Tensorflow: A system for large-scale machine learning," in *12th Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, 2016, Vol 1, pp. 265-283.

[165] Y. Boureau, J. Ponce and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, 2010, Vol 1, pp. 111-118.

[166] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv Preprint arXiv:1702.05659*, 2017.

[167] H. A. Al-Barazanchi, H. Qassim and A. Verma, "Novel CNN architecture with residual learning and deep supervision for large-scale scene image categorization," in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, USA, 2016, Vol 1,pp. 1-7.

[168] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv Preprint arXiv:1502.03167*, 2015.

[169] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, 2012, Vol 1, pp. 1097-1105.

[170] C. Olah *et al.*, "The building blocks of interpretability," *Distill*, vol. 3*(3)*, pp. e10, 2018.

[171] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?" *Brain Informatics*, vol. 3*(2)*, pp. 119-131, 2016.

[172] R. Henderson and R. Rothe, "Picasso: A modular framework for visualizing the learning process of neural network image classifiers," *arXiv Preprint arXiv:1705.05627*, 2017.

[173] R. R. Selvaraju, *et al*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision,* Venice, Italy, 2017, pp. 618-626.

[174] A. Chattopadhay, *et al.*, "Grad-cam: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV),* Lake Tahoe, USA, Mar, 2018, pp. 839-847.

[175] Y. Wang, *et al.*, "Optimized scale-and-stretch for image resizing," in *ACM Transactions on Graphics (TOG),* ACM, Vol. 27, No. 5, pp. 118. 2008.

[176] Paszke, Adam, *et al.* "PyTorch: An imperative style, high-performance deep learning library." *Advances in Neural Information Processing Systems.* Vancouver, Canada, 2019, pp. 8024-8035.

[177] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, Inc. 2008.

[178] A. Gulli and S. Pal, *Deep Learning with Keras.* Packt Publishing Ltd, 2017.

[179] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9*(8),* pp. 1735-1780, 1997.

[180] H. Sak, A. Senior and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," arXiv preprint arXiv:1402.1128.

[181] T. N. Sainath *et al*, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Queensland, Australia, 2015, Vol 1, pp. 4580-4584.

[182] A. A. Ismail, T. Wood and H. C. Bravo, "Improving Long-Horizon Forecasts with Expectation-Biased LSTM Networks," *arXiv Preprint arXiv:1804.06776,* 2018.

[183] W. Li *et al*, "Eac-net: Deep nets with enhancing and cropping for facial action unit detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40*(11),* pp. 2583-2596, 2018.

[184] G. Varol, I. Laptev and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40*(6),* pp. 1510-1517, 2017.

[185] D. Tran *et al.*, "Learning spatiotemporal features with 3d convolutional networks," *arXiv Preprint: arXiv:1412.0767.*

[186] J. Gan *et al.*, "3d convolutional neural network based on face anti-spoofing," in *2017 2nd International Conference on Multimedia and Image Processing (ICMIP),* Wuhan, China, March, 2017, Vol 1, pp. 1-5.

[187] Hutson, M. *AI researchers allege that machine learning is alchemy*. Science (May.3,2018);[Online].https://www.sciencemag.org/news/2018/05/ai-researchersallege-machine-learning-alchemy [Accessed: 28-Nov-2019]

[188] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, 2010, Vol 1, pp. 249-256.

[189] M. S. Satu and M. H. Parvez, "Review of integrated applications with aiml based chatbot," in *2015 International Conference on Computer and Information Engineering (ICCIE)*, 2015, pp. 87-90,.

[190] L. Viganò and D. Magazzeni, "Explainable security," *arXiv Preprint arXiv:1807.04178,* 2018.

[191] R. Shao, X. Lan and P. C. Yuen, "Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3D mask face anti-spoofing," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Denver, Colorado, USA, 2017, pp. 748-755.

[192] X. Tu and Y. Fang, "Ultra-deep neural network for face anti-spoofing," in *International Conference on Neural Information Processing*, Guangzhou, China, pp. 686-695, 2017.

[193] H. Liu, K. Simonyan and Y. Yang, "Darts: Differentiable architecture search," *arXiv Preprint arXiv:1806.09055,* 2018.

[194] S. Xie *et al*, "SNAS: stochastic neural architecture search," *arXiv Preprint arXiv:1812.09926,* 2018.

[195] C. Liu *et al*, "Progressive neural architecture search," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, Vol 1, pp. 19-34.

[196] R. Shin, C. Packer and D. Song, "Differentiable neural network architecture search," In *ICLR Workshop, Vancouver, BC, Canada,* 2018.

[197] B. Wu *et al*, "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, California, USA*, 2019, pp. 10734-10742.

[198] E. J. Gumbel, "Statistical theory of extreme values and some practical applications," *NBS Applied Mathematics Series,* vol. 33, 1954.

[199] C. J. Maddison, A. Mnih and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv Preprint arXiv:1611.00712,* 2016.

[200] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv Preprint arXiv:1609.04747,* 2016.

[201] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Preprint arXiv:1412.6980,* 2014.

[202] V. Thai *et al*, "Detection, tracking and classification of aircraft and drones in digital towers using machine learning on motion patterns," in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS),* Athens, Greece, 2019, pp. 1-8.

Appendix: Papers Published

Conference Paper

Pan, S. and Deravi, F., 2019, spatio-temporal texture feature for presentation attack detection in biometric systems In *2019 Eighth International Conference on Emerging Security Technologies (EST)* . IEEE

Pan, S. and Deravi, F., 2018, January. Facial biometric presentation attack detection using temporal texture co-occurrence. In *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)* (pp. 1-7). IEEE.

Pan, S. and Deravi, F., 2017, September. Facial action units for presentation attack detection. In *2017 Seventh International Conference on Emerging Security Technologies (EST)* (pp. 62-67). IEEE.