Improving the generalization capability of face spoofing detection

Li, Haoliang

2018

Li, H. (2018). Improving the generalization capability of face spoofing detection. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/89790>

<https://doi.org/10.32657/10220/46392>

Improving the Generalization Capability of Face Spoofing Detection

Haoliang Li

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2018

To my beloved parents, grandparents and Happy Cat

Acknowledgments

First and foremost, I would like to express my sincerest gratitude to my supervisor Professor Alex Chichung Kot, who has supported me throughout my Ph.D. study with his enormous patience, knowledge, invaluable insight and his tough but constructive criticisms. My Ph.D. study cannot succeed without his guidance and support. Everytime I discussed with him, I can gain tremendously knowledge not only on the technical knowledge but also the way how to be a good researcher. I am thankful that I can be a student under his supervision and I am sure this experience will benefit the rest of my life.

I would like to express my thanks to Professor Shiqi Wang who I closely worked with during my Ph.D. study for his guidance of my research work and his encouragement when I suffered from frustration and exhaustion. I have also been fortunate to collaborate with Professor Sinno Jialin Pan. I would like to thank him for his many ideas and suggestions.

I would also like to acknowledge the fund and Ph.D. grant from Nanyang Technological University (NTU), Singapore. I am honored to be studying in NTU ROSE lab. ROSE lab has continuously organized research seminars and invited renowned researchers around the world to give seminars and workshops, which have benefit all of us. I appreciate ROSE for providing excellent environment and computing facilities for my research.

During my Ph.D. study, I enjoyed the time spending with my colleagues in ROSE

lab including Dr. Yan Wang, Dr. Sheng Li, Dr. Khosro Bahrami, Huijing Zhan, Ze Lu, Devadeep Shyam, Renjie Wan, Jun Liu, Junlin Hu, Tianyi Zhang, Tan Yu. I am grateful to Dr. Dennis Sng, Mr. Joseph Lim, Ms. Qian Wang and Ms Ooy Mei Chai, who have patiently provided me with administrative support and various resources.

Finally, I would like to thank my family. Without their support, this thesis would not be possible.

Abstract

Biometrics has nowadays become the universal interest in the areas such as forensics and identification. Among the various biometrics information (e.g. fingerprint, face, palm, veins, iris, gait), “face” is the most popular one since face recognition only needs a camera which is available on most of the smart-phones. Due to the rapid development of face recognition technology (e.g. deep learning), face recognition has been applied to many applications, such as the “smile to pay” facial recognition system based on Alipay’s mobile phone app.

Despite the success of face recognition system, it is vulnerable to spoofing attack which can be conducted with a special medium. An attacker can easily bypass the face recognition system with a printed photo/replay video of an authenticated user. Due to the security issue raised by face spoofing, face anti-spoofing has recently emerged as an active topic with great significance for both academia and industry. Recently, numerous face spoofing detection schemes have been proposed based on either hand-crafted features or deep learning features. However, these methods are limited in their scope that the face images (videos) for training and the images (videos) for testing are all collected from the similar condition, which constrains the application of such face spoofing detection methods since the environments of face capturing can be diverse in practice. In light of this, we aim to address the face anti-spoofing problem by improving the generalization capability under practical face capturing conditions in this thesis.

The reason which prevents the current methods from being applied to the practical application is due to the large performance drop since the classifier obtained by training data cannot be well generalized to testing data. It is indeed that we cannot collect face data from all possible conditions and environments. However, it is possible to collect a limited number of data based on a specific scenario (e.g. door controlled access). We aim to build a more robust face spoofing detection classifier by leveraging the limited

number of face data collected by ourselves and the large amount of face data which are publicly available. Based on this motivation, we can improve the generalization capability with only limited face data in an application-specific domain by leveraging data from a richer — and related — domain, seeking to learn meaningful features, through neural network distilling, to distinguish presentation attack samples from genuine ones in a setup with limited training data. In particular, we first train a deep neural network based on the sufficient labeled data set in an attempt to “teach” a neural network for the application-specific domain. Subsequently, we form training sample pairs, from the two domains, and formulate a novel optimization function, dedicated to capture spoofing-specific information and train a discriminative deep neural network on the application-specific domain.

It is also possible to conduct face anti-spoofing in an “off-line” fashion. More specifically, we first collected unlabeled face data for verification purpose, and conduct face spoofing detection by leveraging these unlabeled testing data after a period of time. Such motivation is practical based on applications such as face registration. In light of this, we introduce an unsupervised domain adaptation face anti-spoofing scheme to address the problem that learns the classifier for the target domain based on training samples in a different source domain. In particular, an embedding function is first imposed based on source and target domain data, which maps the data to a new space where the distribution similarity can be measured. Subsequently, the Maximum Mean Discrepancy between the latent features in source and target domains is minimized such that a more generalized classifier can be learned. State-of-the-art representations including both hand-crafted and deep neural network learned features are further adopted into the framework to quest the capability of them in domain adaptation.

Finally, we aim to address the problem that no testing face data information is available during the training phase, which is important for practical application. More specifically, we propose a novel framework leveraging the advantages of the representational ability of deep learning and domain generalization for face spoofing detection. In particular, the generalized deep feature representation is achieved by taking both spatial and temporal information into consideration, and a novel network architecture tailored for the spatial-temporal input is proposed. The network is first initialized by training

with augmented facial samples based on cross-entropy loss and further enhanced with a specifically designed generalization loss, which coherently serves as the regularization term to train the network in an end-to-end fashion. The training samples from different domains can seamlessly work together for learning the generalized feature representation by manipulating their feature distribution distances.

Contents

Acknowledgments	i
Abstract	iii
List of Figures	x
List of Tables	xiv
List of Abbreviations	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Literature Review	3
1.2.1 Motion Analysis based Methods	3
1.2.2 Texture Analysis based Methods	5
1.2.3 Sensor based Methods	7
1.2.4 Deep Learning based Methods	8
1.2.5 Fusion of Different Methods	8
1.3 Objectives	9
1.4 Organization of the Thesis	11

2	Deep Neural Network Distilling for Face Spoofing Detection	12
2.1	Introduction	13
2.2	Proposed Method	13
2.2.1	Distilling from Teacher Network	15
2.2.2	Neural Network Training	17
2.3	Experiments and Results	18
2.3.1	Evaluation Protocols	18
2.3.2	Comparisons with Baseline Methods	22
2.3.3	Exploring Different Types of Distilling Information	25
2.3.4	Exploring the Number of Subjects in the Application-Specific Domain	26
2.4	Summary	26
3	Unsupervised Domain Adaptation for Face Spoofing Detection	29
3.1	Introduction	30
3.2	Proposed Method	32
3.2.1	Maximum Mean Discrepancy Minimization	32
3.2.2	Outlier Removal	33
3.2.3	Unsupervised Domain Adaptation for face anti-spoofing	36
3.3	Feature Analysis	40
3.3.1	Hand-crafted Features	40
3.3.2	Deep Learning based Feature	44
3.4	Experiments and Discussions	45
3.4.1	Databases	45

3.4.2	Experimental Settings	48
3.4.3	Intra-Database Experimental Results	49
3.4.4	Cross-Database Experimental Results	50
3.4.5	Discussions	64
3.5	Summary	65
4	Learning Generalized Deep Feature Representation for Face Spoofing De-	
	tection	66
4.1	Introduction	67
4.2	Proposed Method	68
4.2.1	3D Convolutional Neural Network	68
4.2.2	Data Augmentation	71
4.2.3	Model Generalization	73
4.3	Experimental Results	75
4.3.1	Databases	75
4.3.2	Evaluation Protocol	77
4.3.3	Experimental Setup	78
4.3.4	Experimental Results	79
4.3.5	Computational Time Analysis	88
4.3.6	Discussions	89
4.4	Summary	90
5	Conclusions and Future Works	91
5.1	Conclusions	91
5.2	Future Works	94

5.2.1	Face Spoofing Detection based on Learning	95
5.2.2	Face Spoofing Detection based on Sensor Information	96
Author's Publications		98
Bibliography		100

List of Figures

1.1	An example of face verification system with (a): the genuine face input and (b): the spoofed face input.	2
1.2	Examples of VR attack and “special glasses” attack.	4
1.3	Summary of related works based on face spoofing detection.	5
2.1	The architecture of the proposed cross-domain face spoofing detection. We first train a teacher model by using available data from a richer data domain. Then a student model is trained by leveraging data from both the rich data domain and labeled data from application-specific domain (often with limited training samples).	14
2.2	Examples of USSA database (from [1]). First and second rows: real faces. Third row: spoofed faces captured by using front facing camera. Last row: spoofed faces capture by using front facing cameras. The spoofing mediums from the left to the right are: MacBook, Nexus 5, Nvidia Shield Tablet, and printed photo.	19
2.3	Examples of Idiap REPLAY-ATTACK database (from [2]). First row: controlled scenario, second row: adverse scenario. From the left to the right: real faces, mobile attack, printed attack 1, and printed attack 2. . .	19

2.4	Examples of CASIA database (from [2]). First row: low quality, second row: normal quality, third row: high quality. From the left to the right: real faces, warped photo attack, cut photo attack and video replay attack.	20
2.5	Examples of MSU database (from [3]). First row: Google camera, second row: MacBook camera. From the left to the right: real faces, iPad replay attack, iPhone replay attack and printed attack.	21
2.6	Equal Error Rate comparison by varying the number of subjects.	27
3.1	2D visualization of CASIA database and Idiap REPLAY-ATTACK database with different features. The figure is better to be viewed in color format.	31
3.2	Examples of outliers in CASIA and Idiap REPLAY-ATTACK databases. Left: color saturation. Right: face detection failure.	34
3.3	The framework of face spoofing detection with unsupervised domain adaptation (DA). After feature extraction from facial image, we first adopt a feature embedding by using Principle Component Analysis (PCA). Subsequently, we learn a mapping function to align the eigenspaces between source domain data and target domain data. Finally, a classifier is trained based on representer theory [4].	40
3.4	Comparison between a genuine face and spoofed face.	42
3.5	Examples of Rose-Youtu Liveness Database. From top to bottom: face images in genuine, cropped mask, full mask, upper mask, paper print and video replay versions. (For paper print attack, both warped paper and still paper attacks are considered.) From left to right: face images captured by iPhone 5S, Hasee mobile phone, Huawei mobile phone, iPad and ZTE mobile phone.	47

3.6	Outlier samples selected by the proposed method. The first row shows the genuine samples and the second row shows the spoofed samples. The samples are collected from CASIA, Idiap REPLAY-ATTACK, MSU and Rose-Youtu databases (from left to right).	54
3.7	HTER performance (with SA method) with the variation of target domain data number. (a) train: CASIA, test: Idiap, (b) train: CASIA, test: MSU, (c) train: Idiap, test: CASIA, (d) (train: Idiap, test: MSU), (e) train: MSU, test: CASIA, (f) train: MSU, test: Idiap	59
3.8	HTER performance (with KSA method) with the variation of the number of genuine target domain data. (a) train: CASIA, test: Idiap, (b) train: CASIA, test: MSU, (c) train: Idiap, test: CASIA, (d) (train: Idiap, test: MSU), (e) train: MSU, test: CASIA, (f) train: MSU, test: Idiap	62
4.1	The pipeline of the proposed scheme for face spoofing detection. The final objective function is determined by both classification loss and generalization loss. The output from FC2 layer is employed as latent discriminative feature for classification. The volume conv model contains the volume convolution layer, BatchNormalization, LeakyReLU, Max-Pooling and Dropout layer. The second fully connected layer (FC2) is used for latent discriminative feature extraction.	69
4.2	Architecture visualization of 3D-CNN model.	71
4.3	Illustrations of data based on spatial augmentation.	71
4.4	Illustrations of data based on gamma correction based augmentation. (a) Original Face; (b) Face with the gamma correction ratio 1.0/2.2; (c) Face with the gamma correction ratio 2.2/1.0.	73
4.5	Genuine and spoofed face samples for visualization.	82

4.6	Spatial visualization of genuine and spoofed face samples based on 128 feature maps.	83
4.7	Temporal visualization of genuine and spoofed face samples based on frequency spectrum. The horizontal axis ranges in $[-\pi, \pi]$	84
4.8	Convergence visualization for Rose-Youtu liveness database.	88

List of Tables

2.1	Equal Error Rate (%) performance for face anti-spoofing.	23
2.2	Equal Error Rate (%) performance for face anti-spoofing.	23
2.3	Equal Error Rate (%) on different CNN feature extractions and knowl- edge Distillation.	25
3.1	Results of intra-database performance	49
3.2	Performance (HTER) of cross-database with Wavelet statistical feature .	51
3.3	Performance (HTER) of cross-database with CoALBP (HSV) feature .	52
3.4	Performance (HTER) of cross-database with CoALBP (YCbCr) feature	52
3.5	Performance (HTER) of cross-database with LPQ (HSV) feature	52
3.6	Performance (HTER) of cross-database with LPQ (YCbCr) feature . . .	53
3.7	Performance (HTER) of cross-database with deep learning based feature	53
3.8	Performance improvement with CoALBP feature in HSV space.	54
3.9	Percentage of outlier samples for different databases	57
3.10	Performance (HTER) comparisons between OR+DA and DA+OR (CoALBP HSV).	58
3.11	Performance (HTER) comparisons between OR+DA and DA+OR (Deep Learning based Feature).	58
3.12	HTER Results (%) by Kernel Subspace Alignment with CoALBP (HSV)	61
3.13	HTER Results (%) by Kernel Subspace Alignment with Deep Learning based Feature	61

3.14	Performance comparisons between feature synthesis [5] and our methods. (CoALBP HSV)	63
3.15	Performance comparisons between feature synthesis [5] and our method. (Deep Learning based Feature)	63
3.16	Performance (HTER) of cross-database with concatenated feature . . .	64
4.1	The Proposed 3D CNN Architecture	70
4.2	Cross-Camera Experimental Protocol.	76
4.3	Performance Comparisons by different 3D CNN structures with Rose-Youtu Liveness Database. The results are measured by Equal Error Rate (EER).	80
4.4	Performance comparisons with the state-of-the-art methods.	81
4.5	HTER performance (%) for cross-camera experiments (the protocols are defined in Table 4.2).	86
4.6	Computational time (seconds) analysis based on a 360-frames video with different methods	89

List of Abbreviations

Adam	Adaptive Moment Estimation
CASIA	Institute of Automation, Chinese Academy of Science
CNN	Convolutional Neural Network
CoALBP	Co-occurrence of Adjacency Local Binary Pattern
DA	Domain Adaptation
DG	Domain Generalization
DOG	Difference of Gaussian
EER	Equal Error Rate
FC	Fully Connected
HSV	Hue, Saturation, Value
HTER	Half Total Error Rate
IDA	Image Distortion Analysis
IQA	Image Quality Assessment
LBP	Local Binary Pattern
LPQ	Local Phase Quantization
MMD	Maximum Mean Discrepancy
MsLBP	Multi-Scale Local Binary Pattern
MSU	Michigan State University
NUAA	Nanjing University of Aeronautics and Astronautics
RELU	Rectified Linear Unit

RGB	Red, Green, Blue
RKHS	Reproducing Kernel Hilbert Space
SIFT	Scale-invariant feature transform
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
USSA	Unconstrained Smartphone Spoof Attack
VR	Virtual Reality
YCbCr	Luminance, Blue Chroma Components, Red Chroma Components

Chapter 1

Introduction

1.1 Motivation

With the ubiquity of smartphones and fast proliferation of various mobile applications, user verification through biometrics, such as face, fingerprint and iris, has become important and popular with a wide range of applications including phone unlocking and online payment, e.g. Apple Pay, Alipay [6–9]. In the past several decades, we have witnessed the great advances of face verification techniques [10, 11, ?, 11]. However, such verification is associated with an apparent security risk [12, 13], i.e. their vulnerability to spoofing attacks. A spoofing attack here refers to the fraudulent behavior that one masquerades as another one by falsifying data to thereby gain an illegitimate advantage. Among the different biometric modalities, spoofing a face is notably the easiest given the ease of acquiring an individuals face image on Internet and social networks such as Facebook and LinkedIn. It is also reported that spoofing attacks have become the main hurdle for companies to market user identity management solutions based on face verification technologies. This triggers the need for innovating on efficient and reliable detection to filter off spoofed face images before they are fed to the face verification engine. Fig. 1.1 shows the pipeline of a face verification system and how an attacker

conducts face spoofing process to bypass face verification system (with printed photo attack).

Figure 1.1: An example of face verification system with (a): the genuine face input and (b): the spoofed face input.

Face spoofing consists of diverse attack modes. Some common attacking modes include printed photo attack [14], replay attack [15] and 3D facial mask attack [16]. Printed photo attack typically requires a face image to be reproduced with high-quality printing before it is displayed in front of a camera for user verification. Replay attack requires displaying a face image or a short face video using a digital display device, e.g. a tablet, for user verification. 3D facial mask attack requires a 3D face mask model to be displayed. Among the three modes, replay attack and printed photo attack are relatively easy to launch because that they do not require the construction of a complex 3D face model. An attacker can simply search for the target users photo or video on Internet and display it through a digital display or printing to launch the attack.

Besides the attack modes mentioned above, more sophisticated attacks are introduced recently, which further bring severe security concern to face verification system. In [17], a Virtual-Reality (VR) based attack was introduced to bypass face verification system based on mobile phone. More recently, motivated by the white-box attack of deep learning model, an attacker can bypass the face verification system based on deep learning by wearing “special glasses” [18–21]. However, designing such VR attack and “special glasses” can be even more expensive and one reason is that they require the attacker to have some knowledge about the principle of face verification system [18]. In this thesis, we only focus on printed photo attack and replay video attack which are more common and easier to launch compared with other attack modes.

1.2 Literature Review

There has been an increasing interest on the security issue of face recognition [22–27]. Here we summarize the related works based on motion analysis, texture analysis, sensor information and deep learning (as shown in Fig. 1.3). Moreover, we also review the literatures which fused multiple face spoofing detection algorithms into an ensemble manner.

1.2.1 Motion Analysis based Methods

Motion based methods aim at extracting liveness information that can distinguish the genuine face from the spoofed one through eye blinking, lips movement, and head rotation [28]. In [29], the planar object movements were detected as cues for translation, in-plane rotation, panning and out-of-plane rotation. In [30][31], the subtle movements of different facial parts were extracted as important features under the assumption that the genuine and spoofed faces can be distinguished by the movement cues. Furthermore, in [32–35], the background motion information was also utilized to detect face

(a) VR attack (courtesy from [17])

(b) “Special glasses” attack (courtesy from [19])

Figure 1.2: Examples of VR attack and “special glasses” attack.

Figure 1.3: Summary of related works based on face spoofing detection.

spoofing. Though motion-based methods have achieved satisfactory performance for face spoofing detection, there are several disadvantages which limit the application of motion analysis based methods. First, user cooperation (e.g. eye blinking) is required for supplying the liveliness information for authentication. Secondly, extracting optical flow [36] which can be used for motion analysis is to some extent time consuming, which limits their application for practical use on resource-constrained mobile platforms. Lastly, the motion analysis based methods cannot be applied in the face verification systems where only a single face image of a client is available. In this case, we have no motion information available.

1.2.2 Texture Analysis based Methods

The idea of facial texture and distortion analysis originates from the assumption that the spoofed medium is likely to lack high-frequency information, due to the face me-

dia reproduction process. By analyzing the texture artifacts left behind during an attack, we can extract useful information such that the genuine and spoofed faces can be properly distinguished. In [37], a texture analysis method based on two dimensional Fourier spectrum is conducted. In [38], Tan *et al.* proposed a total-variation based decomposition method and extracted the different-of-Gaussian (DoG) information on the high-frequency part. The final model is learned in a bilinear sparse low-rank regression manner. Texture features designed for object detection/recognition tasks have also been proved to be effective for face spoofing detection. In [39], multi-scale Local Binary Pattern (LBP) with Support Vector Machine (SVM) classifier was proposed, achieving superior performance on NUAA [38] and Idiap REPLAY-ATTACK databases [40]. The multi-scale LBP feature was further extended to facial component based method followed by fisher vector [41], such that more discriminative information can be extracted. Other texture features, such as Scale-invariant feature transform (SIFT) and Speed-up Robust feature (SURF) [42], can also be applied to the face anti-spoofing task. As the high-frequency information can also be discarded in the temporal domain, the texture features based on 2D plane can be extended to 3D plane [43][44]. Moreover, by jointly exploring color and texture information, the face anti-spoofing performance can be largely improved [2, 45, 46]. More recently, a dynamic texture face spoofing was proposed [47] by considering volume local binary count patterns. The researchers further showed that by incorporating flashlight, the texture pattern can be detected more readily [48]. Another stream of feature design is based on image quality methods. In [49], the author analyzed the quality in the perspective of physics-based liveness artifacts. In [50], 25 quality assessment based metrics were employed as the discriminative features for face spoofing detection. In [51], the authors extended this method in a regression manner to tackle the problem whereby samples were taken from multiple camera models. Distortion based feature analysis was also introduced for face spoofing detection task [3]. In [3], four different features (specular reflection, blurriness, chromatic mo-

ment and color diversity) were utilized as discriminative information for face spoofing detection. The specular reflection component was firstly extracted by the method in [52], and then the statistical features based on specular component percentage, mean value of specular pixels and the variance were computed. For the blurriness feature, the no-reference blur score based on [53] and [54] were utilized. Regarding the chromatic moment and color diversity, the statistical features based on HSV and color quantization were adopted in [55]. However, both texture-based and distortion-based features are likely to be overfitted to one particular setup, which may limit their application for practical scenarios when confronting diverse image/video capturing conditions.

1.2.3 Sensor based Methods

In addition to motion analysis and texture analysis methods, additional sensors can also be leveraged for face spoofing detection. One example is to leverage the infrared (IR) sensor which can help to reduce the influence of diverse illumination [12][56][57]. Another example is to utilize a specific LED and photodiodes with difference wavelengths [58][59] which can measure the reflectance information as the discriminative feature for face spoofing detection. We can also leverage other sensors which were already proved to be effective for face recognition task, including facial vein detection sensor [60], 3D face acquisition sensor [61–63] and thermal imaging sensor [64][65]. For example, for thermal imaging sensor, some methods have been proposed based on the motivation that only genuine faces have thermal information [66] with a face database captured by the camera with thermal imaging sensor [67]. 3D information [16] can also be leveraged to counter the spoofing attack with the flat surface. With such additional information, specific hand-crafted features can be designed for the face spoofing detection purpose.

1.2.4 Deep Learning based Methods

Deep learning based methods have also been proved to be effective for biometric spoofing detection tasks. Yang *et al.* [68] first proposed to use Convolutional Neural Network (CNN) for face spoofing detection. Some other works [69–72] proposed to modify the network architecture which can further improve the detection accuracy. In [73], CNN has been proved to be effective for face, fingerprint and iris spoofing detection. Nogueira *et al.* [74] further showed that a pre-trained CNN model based on ImageNet [75] can be transferred to spoofing detection without any fine-tuning process. Besides simply employing CNN for the face spoofing detection problem, Xu *et al.* [76] proposed to capture temporal information by combining CNN with Long Short Term Memory (LSTM) network. Additional information (e.g. eye blinking) can also be considered as auxiliary information by associating it with deep learning [77], which further improves the face spoofing detection performance. More recently, Atoum *et al.* [78] proposed depth-based CNN for face spoofing detection to extract depth information based on RGB face images. Wu *et al.* [79] proposed a pre-processing algorithm based on consecutive face video frames and further extracted discriminative feature with CNN. However, due to the lack of large-scale training data based on face anti-spoofing problem, these methods are lack of generalization capability.

1.2.5 Fusion of Different Methods

As mentioned above, different approaches were proposed to tackle the face spoofing detection problem from different perspectives. Therefore, it is likely that we may not be able to achieve good performance by simply applying a single technique for face spoofing detection. Taking motion based analysis as an example, though motion based methods can extract liveness information which can indeed distinguish the real faces and spoofed faces based on printed paper attack, such methods may fail if the system is attacked by video replay medium where the liveness information exists. One research

direction to counter this problem is to fuse different techniques together [80–82]. In [80] and [81], the motion and texture information were combined together which were able to achieve an impressive performance in IJCB 2011 face spoofing detection competition [83]. In [82], different liveness cues were combined together which also showed good generalization ability for face spoofing detection problem. In [84], the motion and texture features were further analyzed based on different types of attack conditions.

1.3 Objectives

Though both hand-crafted feature and deep learning based methods have achieved desired performance in intra-database scenario, there is significant performance drop for cross-database face spoofing detection. Recently, there are some works aiming to explore this problem from different perspectives. In [3], the authors proposed to employ distortion based features which can be less influenced by the diverse facial appearance. In [2], the authors observed that transforming a face image to a new color space can improve the face spoofing detection performance. Considering that face spoofing detection is eventually an application-driven problem, in this thesis, besides the controlled condition with sufficient training data which most existing works focused on, we consider to improve the generalization capability by considering three practical application, 1) limited labeled training data available based on controlled condition, 2) unlabeled testing data available based on uncontrolled condition, 3) no information available based on uncontrolled condition. As such, the challenging aspects of the training data in target domain have been thoroughly taken into account. We summarize the contribution below,

- (i) On the one hand, collecting genuine and spoofed face samples from all possible environments and conditions is indeed impossible. On the other hand, even based on a certain condition, collecting data can also be time-consuming since we need to consider the diversity of facial appearance and attacking types. In this

scenario, we consider to apply face spoofing detection based on an application-specific application where limited face samples with all possible attacking types are available. We formulate this problem as training a more generalized classifier by leveraging the available face samples from an application-specific domain as well as the genuine and spoofed face data collected from the public database. In particular, we propose a knowledge distilling based framework to train a more robust convolutional neural network where better results can be obtained compared with other baseline methods.

- (ii) Off-line face spoofing detection can also be applied in some other cases. We provide one example regarding off-line face spoofing detection based on industry requirement. Face information nowadays is required for online bank registration. Such information can be further used for face verification when logging in the online bank system. However, an attacker may register an account with a spoofed face based on illegal purposes such as voice phishing and money laundering. In this case, we have no prior information regarding the real face of the client and also the face capturing condition. We formulate this problem by storing a bunch of face data for verification and further leveraging such “unlabeled” face data to train a more generalized classifier. In particular, we leverage the advantage of unsupervised domain adaptation and show that by learning a suitable mapping function from source to the target domain, we can achieve around 20% improvement compared with the results without domain adaptation.
- (iii) Finally, we are interested in the problem whether we can directly train a classifier that can be better generalized to testing face data. Such scenario is the most common one which needs to be addressed in the field of face spoofing detection. Unlike the existing works which only focus on training a classifier based on the face samples collected from a certain condition and environment, we assume that

we can have diverse face data from multiple domains (in terms of conditions and environments). We then formulate the problem as whether we can extract feature which can be shareable across domains, thus a more generalized classifier can be trained based on the feature. We take both spatial and temporal information into consideration by learning a 3D based convolutional neural network with a novel regularization term. The results show that we can achieve much better generalization capability compared with other baseline methods.

1.4 Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, we discuss the problem based on the scenario that limited “labeled” face data are available during training. A distilling based network training is proposed to “teach” a neural network for the application-specific domain to learn a more generalized network based on limited genuine and spoofed face samples in the application-specific domain and public-available face samples from another domain.

In Chapter 3, we discuss the problem based on the scenario that the “unlabeled” face samples for verification are available during the training phase. We introduce an unsupervised domain adaptation framework by mapping training and testing data in a latent domain thus a more robust classifier can be obtained.

In Chapter 4, we discuss the problem of training a network only based on “labeled” training data. We propose a novel network architecture which can capture information in both spatial and temporal data regarding face videos by leveraging the advantage of domain generalization such that the generalization capability of the trained network can be better improved.

Chapter 5 summarizes the concluding remarks on this thesis and recommends several future research directions.

Chapter 2

Deep Neural Network Distilling for Face Spoofing Detection

Face spoofing detection is drawing more and more attention from the scientific and industry communities nowadays due to the empowering face verification for authentication requirement. Common methods based on either hand-crafted features or deep learning features can achieve desired performance based on a specific environment when having sufficient training samples in hand. However, one challenging aspect in this problem when developing detection solutions is the difficulty of collecting enough and representative attack samples from an application-specific environment. In this vein, we tackle the problem how to train a robust model with limited genuine and spoofed face data in an application-specific domain. In particular, we propose to leverage data from a richer and related domain (which is also based on face spoofing detection but obtained from a different conditions) to learn meaningful features, through neural network distilling, to distinguish presentation attack samples from genuine ones in a setup with limited training data. In particular, we first train a deep neural network based on the sufficient labeled data set in an attempt to “teach” a neural network for the application-specific domain. Subsequently, we form training sample pairs from the two domains, and for-

ulate a novel optimization function dedicated to capture spoofing-specific information and train a discriminative deep neural network on the application domain. Extensive experiments validate the effectiveness of the proposed scheme in cross-domain face spoofing detection setups.

2.1 Introduction

In this chapter, we aim at answering the question of how to train a network solution with limited face samples in a particular environment. In face spoofing detection, it is practical to collect sufficient fully-labeled training data samples captured under one invariant environment (e.g., limited types and modes of attacks). However, it is not feasible to acquire large volumes of data for any application-specific domain with unconstrained conditions, especially for the spoofing samples (Scheirer et al. [85] recently hinted at this difficulty when considering fingerprint spoofing detection). Such unbalanced training samples may easily lead to the over-fitting problem when training Convolutional Neural Networks (CNN) and falsely give the impression of solving the problem.

To fully exploit the available data to train a robust deep learning model for any application-specific domain (frequently with limited training samples), we propose a deep neural network distillation process to the face spoofing detection problem. In particular, we bring to bear the notion of teacher/student networks. We start by leveraging a “teacher” network, trained with sufficient labeled face samples from one domain (e.g., a public dataset), and then distill its learned knowledge to a “student” network specialized in the domain of interest, for which we have limited training samples.

2.2 Proposed Method

Given training data from one invariant domain with the number of samples N_R , a teacher model can be pre-trained for deep face representations with the aim of spoofing detec-

Figure 2.1: The architecture of the proposed cross-domain face spoofing detection. We first train a teacher model by using available data from a richer data domain. Then a student model is trained by leveraging data from both the rich data domain and labeled data from application-specific domain (often with limited training samples).

tion. The subsequent task is to train a student model by leveraging the teacher model as well as the available data from both domains. Assuming the number of samples in the application-specific domain is N_A , in most of the cases $N_R \gg N_A$. To enrich the training samples, we augment the training data in pairs extracted from the two different domains. As such, the number of training samples is increased to $N_R \times N_A$. To fully exploit the intrinsic properties of the application-specific scenario, as well as possible commonalities between the two domains, the student model is learned based on the training pairs and the teacher model in a multi-tier fashion. First, the distribution of learning representation between two domains is minimized by means of the maximum mean discrepancy (MMD) optimization process. In addition, the paired information is facilitated to guide the network training of the student network by the similarity embedding with CNN features. Finally, to exploit intrinsic application-specific domain characteristics, as well as to achieve a good balance between knowledge distilling and domain-specific training, the cross-entropy loss is further optimized based on the application-specific labeled data to fine-tune the CNN model. Fig. 2.1 illustrates this process.

2.2.1 Distilling from Teacher Network

The first step for distilling knowledge from a teacher network to a student network consists of selecting the actual network architecture to be used. In our case, we opted to select the winning CNN architecture of the ImageNet ILSVRC 2012 [86] competition. Essentially speaking, large volumes of data with great diversity is required to train a CNN model. As such, the CNN model based on rich data can be first trained and then subsequently employed to guide the application-specific domain network.

Regarding the application-specific domain CNN training, two critical issues arise. First, we need to properly map the data onto a feature space where the distribution of the application-specific domain is similar to the one of the rich-data domain (distilling source). In other words, we need to connect samples from both domains in such a way we minimize their differences. In addition, we need to find the best possible way of extracting meaningful spoofing artifacts from the available data that are not (or at least less) influenced by non-useful, “noise”, spoofing information (e.g., facial appearance, illumination changes) with the given CNN architecture.

Network distilling was first proposed in [87], which trains a specialist model with a well-trained generalist model. To utilize the knowledge of a teacher network, the probability distribution generated from the teacher model is considered to form a more informative target loss on the student model. Considering the distribution $\mathbf{q}_i$ generated by the data from the application-specific domain and its corresponding distribution $\mathbf{p}_i$ from the teacher model, the student model is distilled by minimizing the Kullback-Leibler divergence $KL(\mathbf{p}_i||\mathbf{q}_i)$. In this work, we transfer the idea of distilling neural network to address the problems of face spoofing detection. Our scheme is different from [87] in two main aspects. First, instead of distilling the knowledge based on the probability distribution, we distill knowledge based on the fully-connected layer, as only two nodes are available in the probability layer. Second, we consider two types of knowledge in our framework instead of the probability distribution based on the MMD

distance [88] and similarity embedding.

In particular, we adopt AlexNet [86] for both teacher and student network. We aim at training a model $\mathcal{A}$ with only a few face image data samples $I^a = [\mathbf{I}_1^a, \mathbf{I}_2^a, \dots, \mathbf{I}_{N_A}^a]$ and the corresponding label to distinguish between genuine data and spoofed data is denoted as $l(\mathbf{I}_i^a), i = 1, \dots, N_A$ in the application-specific domain. To this end, the rich-data domain information is facilitated, where the model $\mathcal{R}$ is trained by a large volume of labeled data captured in a certain condition $I^r = [\mathbf{I}_1^r, \mathbf{I}_2^r, \dots, \mathbf{I}_{N_R}^r]$ with label $l(\mathbf{I}_i^r), i = 1, \dots, N_R$. Moreover, the feature vector produced by CNN model is denoted as $\phi_R(\mathbf{I}_i^r)$ and $\phi_A(\mathbf{I}_j^a)$ for the i -th face image from rich-data domain and the j -th face image from the application-specific domain, respectively.

2.2.1.1 Minimizing Maximum Mean Discrepancy

Maximum Mean Discrepancy (MMD) [88] is defined as the criterion to measure the similarity of two probability distributions. Such criterion is widely adopted in the domain-adaptation field [89, 90]. To leverage the rich-data domain to train model $\mathcal{A}$, the MMD between the distributions of rich-data domain and application-specific data domains is minimized. We transfer the idea of MMD distance minimization to our problem by linking the application-specific data domain and the rich data domain

$$\mathcal{L}_{MMD}(I^r, I^a) = \left\| \frac{1}{N_R} \sum_{i=1}^{N_R} \phi_R(\mathbf{I}_i^r) - \frac{1}{N_A} \sum_{j=1}^{N_A} \phi_A(\mathbf{I}_j^a) \right\|^2 \quad (2.1)$$

By minimizing the loss function above, we can train the model $\mathcal{A}$, which shares a similar feature distribution as that from the $\mathcal{R}$. We follow [91] to optimize the MMD loss.

2.2.1.2 Similarity Embedding of Paired Samples

Directly training a student model based on labeled application-specific domain data might not be feasible due to the limitation/lack of training samples. Therefore, we

augment the data in a pairwise manner by leveraging the available rich-data domain. The paired samples further enable us to exploit the commonalities of the rich-data and application-specific domains with the label information. In particular, although I^r and I^a are captured under different conditions, the spoofing trail and artifacts existing on the facial medium are supposed to share some common peculiarities. To extract such information, we train $\mathcal{A}$ by distilling the information from the well-trained model $\mathcal{R}$. Specifically, $\phi_R(\mathbf{I}_i^r)$ and $\phi_A(\mathbf{I}_j^a)$ are pushed closer if $\mathbf{I}_i^r$ and $\mathbf{I}_j^a$ share identical labels. By contrast, the distance between $\phi_R(\mathbf{I}_i^r)$ and $\phi_A(\mathbf{I}_j^a)$ is maximized if different labels are observed. As such, the similarity embedding can be formulated as,

$$\mathcal{L}_{sim}(I^r, I^a) = \begin{cases} \sum_{i,j} (1 - \frac{\langle \phi_R(\mathbf{I}_i^r), \phi_A(\mathbf{I}_j^a) \rangle}{\|\phi_R(\mathbf{I}_i^r)\| \|\phi_A(\mathbf{I}_j^a)\|}), & l(\mathbf{I}_i^r) = l(\mathbf{I}_j^a) \\ \sum_{i,j} \max(0, \frac{\langle \phi_R(\mathbf{I}_i^r), \phi_A(\mathbf{I}_j^a) \rangle}{\|\phi_R(\mathbf{I}_i^r)\| \|\phi_A(\mathbf{I}_j^a)\|}), & l(\mathbf{I}_i^r) \neq l(\mathbf{I}_j^a) \end{cases} \quad (2.2)$$

where $\langle \cdot \rangle$ refers to the inner product of vectors, and $l(\cdot)$ denotes the label with respect to a given sample.

2.2.2 Neural Network Training

In addition to the information distilled from the model $\mathcal{A}$, the labeled data I^a also enable us to fine-tune the CNN model with cross-entropy loss for better adaptation to the application-specific domain. The final training process incorporates the MMD, the paired sample similarity embedding, and the cross-entropy loss $\mathcal{L}(I^a, l(I^a))$ [92] for optimization. The optimization objective is then formulated as,

$$\min_{\phi_A(\cdot)} \mathcal{L}(I^a, l(I^a)) + \lambda \mathcal{L}_{MMD}(I^r, I^a) + \mu \mathcal{L}_{sim}(I^r, I^a) \quad (2.3)$$

where λ and μ are regularization parameters, and the cross-entropy loss is defined as

$$\mathcal{L}(I^a, l(I^a)) = - \sum_{i=1}^{N_A} (l(I_i^a) \log(p_i) + (1 - l(I_i^a)) \log(1 - p_i)) \quad (2.4)$$

p_i is the output probability score from the student network.

2.3 Experiments and Results

2.3.1 Evaluation Protocols

To validate the proposed scheme, we utilize the Unconstrained Smartphone Spoof Attack database (USSA) [1] which contains 1000 subjects in total, and three other public available databases, including CASIA Face-Anti-Spoofing Database (CASIA) [93], Idiap Replay-Attack Database (Idiap) [40] and MSU Mobile Face Spoof Database (MSU) [3]. Details of the adopted databases are given below.

The USSA database [1] consists of more than 9000 images from 1000 subjects collected from the internet. Two types of cameras, Front-facing camera (1280×960) and Rear-facing camera (3264×2448) in the Google Nexus 5 are considered to capture the face images. Four different spoof mediums are used for face replay, which include Apple MacBook, Nexus 5, Nvidia Shield Tablet and Printed photos.

The Idiap database consists of 1200 face videos with 50 different subjects in total. The videos were captured by only the front-facing camera of a Macbook with 320×480 pixels in size. Two environments are considered when taking the videos. One is the controlled environment with uniform background and illumination condition. The other is more complex with natural lighting and reflection in the background. For the spoofing medium, iPad 1 (with the size 1024×768 pixels), iPhone 3GS (with the size 480×320 pixels) and A4 printed paper are considered for presentation attacks.

The CASIA database has 600 face videos in total from 50 subjects. Compared with

Figure 2.2: Examples of USSA database (from [1]). First and second rows: real faces. Third row: spoofed faces captured by using front facing camera. Last row: spoofed faces capture by using front facing cameras. The spoofing mediums from the left to the right are: MacBook, Nexus 5, Nvidia Shield Tablet, and printed photo.

Figure 2.3: Examples of Idiap REPLAY-ATTACK database (from [2]). First row: controlled scenario, second row: adverse scenario. From the left to the right: real faces, mobile attack, printed attack 1, and printed attack 2.

Figure 2.4: Examples of CASIA database (from [2]). First row: low quality, second row: normal quality, third row: high quality. From the left to the right: real faces, warped photo attack, cut photo attack and video replay attack.

the Idiap database, the acquisition camera models are more diverse. The quality levels of capturing devices range from low resolution, medium resolution (two different USB cameras with the size of 480×640 pixels) and high resolution (Sony NEX-5 camera with the size 1280×720 pixels). The CASIA database has diverse attack types, including warping, cutting and video-replay attacks. Though the camera models and attacking types are more diverse, compared with Idiap REPLAY-ATTACK, the capturing background and the ethnicity of subjects (all Chinese) are limited.

The MSU database has 280 videos with 35 subjects. Both Laptop camera (with the size 640×480 pixels) and Android phone camera (with the size 720×480 pixels) are considered for face sample collection. The videos were taken under various illumination conditions with different human ethnicities. Two different spoofing attacks, printed photo and replay video attacks, were considered in MSU database.

We adopted the classic Viola-Jones algorithm [94] for face detection and we further resize the detected face image into 224×224 and the input of the network. For network training, we employed the Stochastic Gradient Descent (SGD) to train model $\mathcal{A}$ by

Figure 2.5: Examples of MSU database (from [3]). First row: Google camera, second row: MacBook camera. From the left to the right: real faces, iPad replay attack, iPhone replay attack and printed attack.

feeding the input face images into the model in a batch fashion. We experimentally set the learning rate as 10^{-4} , the weight decay rate as 0.0005, the momentum rate as 0.9 and the learning rate decay as 10^{-7} for training both teacher and student networks. The regularization parameter for the loss function is set as $\lambda = 0.001$ and $\mu = 1$.

For the performance evaluation, we set up a new experimental protocol, where the training samples include a limited number of application-specific samples and rich-domain face samples from another database as the auxiliary data to help training. Moreover, the rest of the samples in the application-specific domain database are used for testing. This protocol also simulates the scenario when it is impractical to collect samples from many different identities. As such, for the labeled data in the application-specific domain, a limited number of samples are collected. Since we focus on the problem of face spoofing detection with limited face samples, the Equal Error Rate (EER) is employed to measure the detection performance. For performance comparison, in the application-specific domain, we randomly select ten subject as our labeled data when adopt USSA database as application-specific domain, and one subject when considering

CASIA, Idiap and MSU as application-specific domain. The FC7 layer from AlexNet is considered for knowledge distilling in our framework. The process is then repeatedly performed for five times and the average results are reported. Moreover, the last fully-connected layer (FC7) with 4096 dimensions is employed as the final feature, and the Support Vector Machine (SVM) [95] with linear kernel is finally adopted to train the classifier for face anti-spoofing detection at the end.

2.3.2 Comparisons with Baseline Methods

Five baseline methods are considered for performance comparisons. These methods can be divided based on four types of features adopted, which have been proved to be effective in dealing with different artifacts incurred by face spoofing such as texture patterns, color distortions, and geometric distortion:

- **Multi-scale Local Binary Pattern:** The multi-scale local binary pattern (MsLBP) [39] was proposed to extract texture patterns appearing on the spoofed medium. In particular, uniform LBP features were extracted on the gray-scale face image, and they were concatenated as a single feature vector with 479 dimensions at the end.
- **Color Texture:** proposed in [2], this method considers both texture patterns and color distortions, which have been shown to be effective for presentation attack detection. The features proposed in [2] (CoALBP+LPQ in HSV and YCbCr color spaces) are concatenated as a single feature vector for comparison.
- **Image Distortion Analysis:** Image distortion related feature [3] deals with the artifacts including geometric distortion (e.g. specular reflection appeared on the spoofing medium), blurring distortion and color distortion. We follow [3] to extract the 121-dimension feature vector for comparison.

- **Convolutional Neural Network baseline:** Due to the powerful discriminative ability of deep learning, the AlexNet with only cross-entropy loss is considered as one baseline method.
- **Convolutional Neural Network with Fine-tuning:** We further consider, as a baseline, the use of transfer learning from one deep neural network to another. In particular, we first train a network with cross-entropy loss by using data from the rich-data domain, followed by fine-tuning the network with labeled data from the application-specific domain.

Table 2.1: Equal Error Rate (%) performance for face anti-spoofing.

Train/Test Data	USSA	USSA	USSA	USSA
Auxiliary Data	*	CASIA	Idiap	MSU
MsLBP [39]	38.5	43.7	45.3	37.0
Color Texture [2]	22.4	27.9	23.3	20.7
Image Distortion Analysis [3]	25.5	48.9	40.7	65.1
CNN baseline	29.3	26.3	31.1	29.1
CNN Fine-tuning	*	22.7	24.5	24.6
Proposed Method	*	15.6	17.8	15.1

[†] Train/Test Data refer to the samples from application specific domain. Auxiliary Data refer to the samples from rich data domain. We only report the results with MsLBP, Color Texture, IQA and CNN baseline if no auxiliary data are used since we can not conduct fine-tuning and distilling without auxiliary data.

Table 2.2: Equal Error Rate (%) performance for face anti-spoofing.

Train/Test Data	CASIA	Idiap	MSU	CASIA	Idiap	MSU
Auxiliary Data	—	—	—	Idiap	MSU	CASIA
MsLBP [39]	34.4	39.7	45.9	62.1	36.5	51.7
Color Texture [2]	19.3	20.6	22.9	30.5	21.6	14.6
Image Distortion Analysis [3]	43.7	13.9	23.4	54.9	55.9	41.4
CNN baseline	19.1	26.6	39.5	49.9	19.8	40.4
CNN Fine-tuning	*	*	*	32.8	16.0	28.2
Proposed Method	*	*	*	16.8	12.5	13.3

[†] Train/Test Data refer to the samples from application specific domain. Auxiliary Data refer to the samples from rich data domain. We only report the results with MsLBP, Color Texture, IQA and CNN baseline if no auxiliary data are used since we can not conduct fine-tuning and distilling without auxiliary data.

For each baseline method, we evaluate the performance in two scenarios, including

the training with application-specific domain only and with both application-specific and rich data domains. As such, in the second strategy, the auxiliary data are included in the training process. We first consider the scenario by considering USSA database as application-specific domain. This database is closer to our motivation since collecting training data with more than 1000 subjects is indeed daunting. The results are shown in Table.2.1. Based on the results, performance degradations can be observed in most of the cases when using a limited number of data for training compared with the results reported in previous works [2, 3, 39]. Such observation is reasonable since both hand-crafted and data-driven features are likely to over-fit to person-specific information in this setup, even when considering an intra-database setting.

When auxiliary data from the rich-data domain are used, we notice an even larger performance drop in some cases, which is due to the over-fitting caused by the more challenging cross-domain setting, as the classifier trained on one domain often does not generalize well to another one. On the other hand, as data-driven features are transferred from one general domain to another (more specific) domain, deep learning based methods also suffer from such over-fitting problem. We notice the performance of deep learning with weights initialized by ImageNet is marginally lower than that of hand-crafted features. However, we also notice that, when the model weights are initialized by training on a large amount of face samples, we can achieve better performance. Such observation indicates the importance of model weight initialization for a specific task.

Turning our attention now to the effects of network distilling allied with MMD and similarity embedding into the deep learning training (our proposed method in this work), we can observe the performance improves significantly. This is due to possible commonalities across the used domains that are leveraged in the new connected training process, such that the learned network is effective for face spoofing in the application-specific domain.

To further demonstrate the robustness of our proposed framework, we consider CA-

SIA, Idiap and MSU database as application-specific domain. The results are shown in Table. 2.2. We observe that the results are consistent with the finding in Table. 2.1, which shows that training a classifier with limited samples can lead to overfitting problem and our proposed knowledge distilling framework can generally outperform all the other baseline methods.

Table 2.3: Equal Error Rate (%) on different CNN feature extractions and knowledge Distillation.

Train/Test Data	CASIA		Idiap		MSU	
Auxiliary Data	Idiap	MSU	CASIA	MSU	CASIA	Idiap
Network Distilling FC6 (MMD)	33.1	29.4	36.3	37.5	30.0	42.8
Network Distilling FC6 (Similarity Embedding)	30.5	24.5	22.6	27.7	25.8	42.4
Network Distilling FC6 (MMD + Similarity Embedding)	28.9	23.7	22.2	25.9	21.1	39.1
Network Distilling FC7 (MMD)	26.6	17.8	28.5	23.5	25.2	21.7
Network Distilling FC7 (Similarity Embedding)	19.3	14.9	16.1	16.4	18.2	21.7
Network Distilling FC7 (MMD + Similarity Embedding)	16.8	12.5	13.3	13.5	11.1	19.2

2.3.3 Exploring Different Types of Distilling Information

In addition to the previous investigations, we also analyze the features generated by the deep neural network more appropriate for knowledge distilling in face spoofing detection. In particular, we contrast the FC6 and FC7 layers in AlexNet. In the FC6 case, the feature vectors $\phi_s(\cdot)$ and $\phi_t(\cdot)$ are extracted as the output of FC6 layer. To further evaluate the domain knowledge in the distilling process, we modify the final objective function with the regularization terms containing only either MMD minimization or similarity embedding. We adopt CASIA, Idiap and MSU database as application-specific domain for this ablation study. Table 2.3 shows the results.

The results using the FC6 layer is worse than those of FC7. Such observation is reasonable as the information is transferred from general to specific in the CNN model. Deeper layers may contain more specific (fine-grained) information that is desirable for spoofing detection. We also observe that the similarity embedding method is effective as it directly utilizes label information compared with the MMD approach in isolation. Finally, the best performance can be obtained with both MMD and similarity embedding

methods in the regularization function, implying that the MMD and similarity embedding can be treated as complementary terms.

2.3.4 Exploring the Number of Subjects in the Application-Specific Domain

Finally, we analyze the training capabilities by using a different number of subjects in the training stage. To this end, we further train the network by considering M subjects with label information in USSA database, where $M = \{10, 20, 50\}$. Fig. 2.6 depicts the results of these experiments. As we can observe, the overall detection performance is improved by increasing the number of training subjects, which is reasonable since we can train a classifier which is less influenced by facial appearance. We also observe that the deep learning method can also generally outperform the hand-crafted feature based methods. Moreover, our proposed distilling framework can still achieve significant improvement with a large margin, which shows the robustness of our proposed knowledge distilling algorithm.

2.4 Summary

In this chapter, we propose a distilling neural network based scheme for cross-domain face spoofing detection. Given one source/initial domain with rich labeled samples and an application-specific domain, with a limited amount of labeled samples, the network in the application-specific domain is specifically trained with the supervision of the rich-data domain trained network. In particular, training samples from the two domains are augmented in pairs, and an optimization problem based on the MMD distance and similarity embedding is formulated to train the student network for the application-specific domain.

Compared with other biometric spoofing detection techniques, which simply train

(a) CASIA as Rich-Data Domain

(b) Idiap as Rich-Data Domain

(c) MSU as Rich-Data Domain

Figure 2.6: Equal Error Rate comparison by varying the number of subjects.

a deep model from scratch or via standard transfer learning, our solution can effectively bridge two different domains with the distilling framework, which shows to be more effective compared with directly training a face spoofing detection network from both scratch and transfer learning, specially under the small sample size problem in the application-specific domain. The proposed method is specially tailored for application domains with limited training data. If large amounts of training data are available, then the traditional fine-tuning process catches-up and becomes competitive when compared with the distilling modeling we investigate in this article.

Chapter 3

Unsupervised Domain Adaptation for Face Spoofing Detection

In Chapter 2, we proposed to leverage the labeled rich-data domain and the limited labeled data in the application-specific domain for face spoofing detection. However, due to the unlimited variations of the conditions (illumination, facial appearance, camera quality, etc.) in face acquisition, directly obtaining labeled training data from a specific-domain can be difficult which greatly hinders the application of existing face spoofing detection methods in real application scenarios, as given the testing samples it is always inconvenient to generate labeled training samples captured in similar conditions. In this section, we propose an unsupervised domain adaptation framework for face anti-spoofing to bridge the gap between the generic and domain adapted classifiers. In particular, given the labeled samples in source domain and unlabeled samples in target domain, it is practical for us to train a domain specific classifier based on inherent properties of these data.

3.1 Introduction

In this section, the cross-domain face anti-spoofing technique is introduced to deal with the scenario that face samples for verification may not be taken by similar camera models or under similar illumination conditions to the training face samples. In general, the performance of the cross-domain face spoofing detection can directly reflect the generalization capability of the anti-spoofing algorithm. In particular, we focus on solving this problem from the perspective of machine learning. Inspired by the success of unsupervised domain adaptation in many applications [96][97], we employ this technique to improve the generalization ability of face spoofing detection. The earlier work addressing the domain adaptation problem on face spoofing detection was reported in [5], where features of spoofed faces were synthesized by domain adaptation under the assumption that samples from different subjects can be formulated with a linear transformation. However, this method was conducted in a supervised manner by assuming the fixed linear transformations, which may not always hold in practice as face samples for verification can be captured by different camera models.

We assume that the source domain data (training samples) and target domain data (testing samples) preserve a certain statistical distribution. Fig. 3.1 visualizes the features¹ of face samples from different domains. We can observe that the features extracted from one particular domain tend to form a compact distribution, implying that they are taken under similar conditions. As such, unsupervised domain adaptation (DA) can be imposed to improve the generalization ability of face spoofing detection by minimizing the Maximum Mean Discrepancy (MMD) with which we adapt the pre-trained classifier from the source domain to target domain [88][99][100]. It is also worth mentioning that the proposed strategy is applicable in real-world scenarios, as it is feasible to collect unlabeled samples for face verification and anti-spoofing purpose via a variety of web services with photo information.

¹We employ t-SNE [98] for feature visualization.

(a) Sample visualization with CoALBP feature

(b) Sample visualization with LPQ feature

Figure 3.1: 2D visualization of CASIA database and Idiap REPLAY-ATTACK database with different features. The figure is better to be viewed in color format.

3.2 Proposed Method

Our proposed framework leverages the advantage of unsupervised domain adaptation. We provide the details of our algorithm below.

3.2.1 Maximum Mean Discrepancy Minimization

Maximum Mean Discrepancy (MMD) is defined as a distance metric for comparing two probability distributions (a.k.a. two-sample test) [88]. Typically, two distributions are identical if and only if the MMD distance equals to zero.

Given the facial feature samples $\mathbf{X}_s = [\mathbf{x}_{s1}; \mathbf{x}_{s2}; \dots; \mathbf{x}_{sn_1}]$ which are used for training as source domain data and unlabeled face samples $\mathbf{X}_t = [\mathbf{x}_{t1}; \mathbf{x}_{t2}; \dots; \mathbf{x}_{tn_2}]$ for verification as target domain data, MMD function which measures the distance between two probability distributions can be defined as follows,

$$D(\mathbf{X}_s, \mathbf{X}_t) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(\mathbf{x}_{si}) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(\mathbf{x}_{ti}) \right\|^2 \quad (3.1)$$

where n_1 is the number of samples in source domain (for training), n_2 is the number of samples in target domain (for testing) and ϕ refers to the embedding function which maps the data from feature space to the Reproducing Kernel Hilbert Space (RKHS) where the distance between two probability distributions can be properly measured [88].

The MMD function can also be reformulated in matrix form by considering a kernel representation of source domain data and target domain data, which is given as

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{ss} & \mathbf{K}_{st} \\ \mathbf{K}_{ts} & \mathbf{K}_{tt} \end{pmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2)} \quad (3.2)$$

where $\mathbf{K}_{ss} = [\phi(\mathbf{x}_{si})\phi(\mathbf{x}_{sj})^T]$, $\mathbf{K}_{tt} = [\phi(\mathbf{x}_{ti})\phi(\mathbf{x}_{tj})^T]$ and $\mathbf{K}_{st} = [\phi(\mathbf{x}_{si})\phi(\mathbf{x}_{tj})^T]$ refer to source domain kernel, target domain kernel and cross domain kernel respectively. The

MMD function in matrix form can be represented as

$$D(\mathbf{X}_s, \mathbf{X}_t) = \text{trace}(\mathbf{KL}) \quad (3.3)$$

We denote L_{ij} as the element of matrix $\mathbf{L}$ in the i th row and j th column, where $L_{ij} = \frac{1}{n_1^2}$ if $i \leq n_1, j \leq n_1$, $L_{ij} = \frac{1}{n_2^2}$ if $n_1 < i \leq n_1 + n_2, n_1 < j \leq n_1 + n_2$, otherwise, $L_{ij} = -\frac{1}{n_1 n_2}$.

3.2.2 Outlier Removal

Directly adopting domain adaptation (DA) in anti-spoofing may not achieve the desired performance because both training and testing face samples are contaminated with outlier samples. Specifically, the outlier samples can be caused by the failure of face detection, over saturation of the medium, unexpected face image blur, etc. There are two main drawbacks of involving the outlier samples in source domain data. First, training classifier can be sensitive to such outlier samples, since face spoofing detection relies on the distortion cues and the unexpected distortion from genuine samples can deteriorate the robustness of classifier. Second, the outliers can lead to a larger distance between two distributions and prevent us from learning robust transformation matrix to adapt all the source domain data to target domain [101]. Fig. 3.2 provides two examples of outliers in spoofing detection. The left one is the face sample which is over-saturated. The right one shows the frame where the failure of face detection happens with Viola-Jones algorithm. We can also observe from Fig. 3.1 that such outliers have a negative impact on compact distribution formation.

From this perspective, only the most informative data should be selected. In [102], the informative data selection strategy was applied in the cross-language text categorization task with a similar motivation. Here we focus on the facial images and remove the outlier samples from the source domain by minimizing the MMD distance before con-

Figure 3.2: Examples of outliers in CASIA and Idiap REPLAY-ATTACK databases. Left: color saturation. Right: face detection failure.

ducting domain adaptation (DA). We transfer the idea kernel mean matching (KMM) [100] which was proposed to tackle covariate shift problem to outlier removal.

In particular, let us denote the number of inlier samples as n'_1 , where $n'_1 < n_1$. To remove the outliers, we impose a binary correspondence matrix $\mathbf{P} \in \{0, 1\}^{n'_1 \times n_1}$, which satisfies

$$\begin{aligned} \sum_{j=1}^{n_1} P_{ij} &= 1, \quad \forall i = 1, \dots, n'_1 \\ \sum_{i=1}^{n'_1} P_{ij} &\leq 1, \quad \forall j = 1, \dots, n_1 \end{aligned}$$

where P_{ij} is the (i, j) th element in $\mathbf{P}$. In other words, each row of the matrix $\mathbf{P}$ indicates one and only one inlier sample from the source domain. We also introduce a vector $\mathbf{p} = \mathbf{P}^T \mathbf{1}_{n'_1}$, where $\mathbf{1}_{n'_1} \in \mathbb{R}^{n'_1}$ denotes the vectors with all ones. Since each column of $\mathbf{P} \in \{0, 1\}^{n'_1 \times n_1}$ contains at most one non-zero element, we have $\mathbf{p} \in \{0, 1\}^{n_1}$ and $\sum_{i=1}^{n_1} p_i = n'_1$. In other words, $p_i = 1$ indicates the i th sample from $\mathbf{p}$ is selected as an inlier training sample, and $p_i = 0$ indicates an outlier sample. The kernel matrix defined

on all the samples after removing the outliers is then represented as

$$\mathbf{K}^* = \begin{pmatrix} \mathbf{PK}_{ss}\mathbf{P}^T & \mathbf{PK}_{st} \\ \mathbf{K}_{ts}\mathbf{P}^T & \mathbf{K}_{tt} \end{pmatrix} \quad (3.4)$$

where $\mathbf{K}^* \in \mathbb{R}^{(n'_1+n_2) \times (n'_1+n_2)}$.

To maximize the similarity of the distribution of the selected source domain samples and target domain samples, we choose $\mathbf{P}$ such that the MMD function below is minimized,

$$\min_{\mathbf{P}} \quad trace(\mathbf{K}^*\mathbf{L}^*) \quad (3.5)$$

where $\mathbf{L}^*$ is defined as $L_{ij}^* = \frac{1}{(n'_1)^2}$ if $i \leq n'_1, j \leq n'_1$, $L_{ij}^* = \frac{1}{n'_1 n_2}$ if $n'_1 < i \leq n'_1 + n_2, n'_1 < j \leq n'_1 + n_2$, otherwise, $L_{ij}^* = -\frac{1}{n'_1 n_2}$.

Then, the objective function of minimizing $\mathbf{P}$ can be rewritten as

$$\begin{aligned} \min_{\mathbf{P}} \quad & trace\left(\frac{1}{(n'_1)^2} \mathbf{PK}_{ss}\mathbf{P}^T \mathbf{1}_{n'_1} \mathbf{1}_{n'_1}^T - \frac{1}{n'_1 n_2} \mathbf{PK}_{st} \mathbf{1}_{n_2} \mathbf{1}_{n'_1}^T\right) \\ & + trace\left(\frac{1}{n'_1 n_2} \mathbf{K}_{ts} \mathbf{1}_{n'_1} \mathbf{1}_{n_2}^T - \frac{1}{n_2^2} \mathbf{K}_{tt} \mathbf{1}_{n_2} \mathbf{1}_{n_2}^T\right) \end{aligned} \quad (3.6)$$

where $\mathbf{1}_{n'_1} \in \mathbb{R}^{n'_1}$ and $\mathbf{1}_{n_2} \in \mathbb{R}^{n_2}$ denote the vectors with all ones. The above objective function can be further simplified given as,

$$\min_{\mathbf{P}} \quad trace\left(\frac{1}{(n'_1)^2} \mathbf{PK}_{ss}\mathbf{P}^T \mathbf{1}_{n'_1} \mathbf{1}_{n'_1}^T - \frac{2}{n'_1 n_2} \mathbf{PK}_{st} \mathbf{1}_{n_2} \mathbf{1}_{n'_1}^T\right) \quad (3.7)$$

By observing that $trace\left(\frac{1}{(n'_1)^2} \mathbf{PK}_{ss}\mathbf{P}^T \mathbf{1}_{n'_1} \mathbf{1}_{n'_1}^T\right) = trace\left(\frac{1}{(n'_1)^2} \mathbf{1}_{n'_1}^T \mathbf{PK}_{ss}\mathbf{P}^T \mathbf{1}_{n'_1}\right)$, the above objective function can be further rewritten as,

$$\min_{\mathbf{P}} \quad \frac{1}{(n'_1)^2} \mathbf{p}^T \mathbf{K}_{ss} \mathbf{p} - \frac{2}{n'_1 n_2} \mathbf{p}^T \mathbf{K}_{st} \mathbf{1}_{n_2} \quad (3.8)$$

This objective function is difficult to solve due to the binary constraints on $\mathbf{p}$. To

get an efficient solution, we relax $\mathbf{p} \in \mathbb{R}^{n_1}$ which satisfies $0 \leq p_i \leq 1$ and $\sum_{i=1}^{n_1} p_i = n'_1$. Therefore, p_i can be treated as a weight for the i th sample, and if p_i is smaller than a threshold, the i th sample can be treated as an outlier. The above problem is essentially a quadratic programming problem, which can be efficiently solved [103]. After obtaining the vector $\mathbf{p}$, a fixed number of training samples with lowest values are treated as outliers. In our work, for each database with different features, we first conduct cross-validations by using libSVM [95] and get an accuracy (acc%). Then, $(1-\text{acc})\%$ of the total samples with the smallest weight values are removed as outliers. To our best knowledge, it is generally nontrivial to determine the threshold for outlier samples, and our simple approach works quite well.

3.2.3 Unsupervised Domain Adaptation for face anti-spoofing

The main objective of domain adaptation (DA) for face spoofing detection is to model the distributions of face samples and learn a mapping function which can align the distributions from the source domain data to the target domain data. This allows us to transfer the prior knowledge of the source domain to target domain for anti-spoofing purpose. Considering this, we introduce the following two principles to further analyze this problem.

- The performance of face spoofing detection depends heavily on the distribution of facial appearances, illumination conditions and camera quality of given face samples.
- The extracted features which account for facial appearances, illumination conditions and camera quality can be approximated in a low-dimensional linear subspace.

The first observation is based on the recent studies of face spoofing detection on various databases. It can be observed that the performance of face spoofing detection by

using the same features can be diverse, implying that the facial appearance, illumination condition, and adopted camera models have a significant impact on the detection accuracy. Moreover, straightforwardly imposing features for the cross-database scenario cannot achieve satisfied performance, the reason of which is mainly due to the mismatch of face capturing conditions.

The second principle has been analyzed in many face recognition tasks. By projecting facial features on eigenvectors, we can extract more robust facial information on a low-dimension manifold [104, 105]. On the other hand, the diverse reflection on the facial surface caused by different illumination conditions can also be modeled as low-dimensional linear subspace which can further improve face recognition performance [106]. Furthermore, both of the camera models and recapturing process are associated with image quality. We also observe that the influence of camera models can be dominant [51]. Therefore, it is reasonable to assume that the diverse of camera models can be formulated in low-dimension subspace.

Based on the above principles, we formulate the mapping function ϕ based on Principle Component Analysis (PCA) which is given as

$$\begin{aligned}\phi : \hat{\mathbf{X}}_s &\Rightarrow \mathbf{U}_s, \quad \mathbf{X}_t \Rightarrow \mathbf{U}_t \\ \mathbf{U}_s &= \arg \max_{\mathbf{U}^T \mathbf{U} = \mathbf{I}_d} \{\mathbf{U}^T \hat{\mathbf{X}}_s^T \hat{\mathbf{X}}_s \mathbf{U}\} \\ \mathbf{U}_t &= \arg \max_{\mathbf{U}^T \mathbf{U} = \mathbf{I}_d} \{\mathbf{U}^T \mathbf{X}_t^T \mathbf{X}_t \mathbf{U}\}\end{aligned}\tag{3.9}$$

where $\mathbf{U}_s$ and $\mathbf{U}_t$ are the eigenspaces of source domain features after outlier removal $\hat{\mathbf{X}}_s$ and target domain features $\mathbf{X}_t$ respectively. $\mathbf{I}_d$ is the identity matrix with size $d \times d$.

Then the MMD function based on subspace of extracted features can be formulated as

$$D(\hat{\mathbf{X}}_s, \mathbf{X}_t) = \|\mathbf{U}_s - \mathbf{U}_t\|_F^2\tag{3.10}$$

Our goal is to learn a mapping function $\mathbf{M}$ from $\mathbf{U}_s$ to $\mathbf{U}_t$ which minimizes the M-

MD function defined in (3.10). Therefore, we employ the subspace alignment (SA) algorithm [107] to minimize (3.10) by learning a transformation matrix from the subspace of source domain to the subspace of target domain. After obtaining the transformation matrix, the source domain data are mapped to another space based on transformation matrix and the classifier is trained via transformed source domain data. The procedures of subspace alignment is summarized as follows,

- Normalizing the source and target data by Z-Score [108].
- Applying Principle Component Analysis (PCA) to derive d eigenvectors corresponding to the top d largest eigenvalues (following the theory proposed in [107]), which are used as bases of source and target subspaces as $\mathbf{U}_s$ and $\mathbf{U}_t$.
- Learning the mapping function $\mathbf{M}$ by

$$\mathbf{M}^* = \operatorname{argmin}_{\mathbf{M}} \|\mathbf{U}_s \mathbf{M} - \mathbf{U}_t\|_F^2 \quad (3.11)$$

To extract the non-linear information induced by the subspace, we also extend the PCA to kernel PCA case by using a Gaussian kernel. We refer the kernel based method as kernel Subspace Alignment (KSA). The only difference between Subspace Alignment and kernel Subspace Alignment is that the eigenvectors in the second step are obtained by kernel PCA on $\hat{\mathbf{X}}_s$ and $\mathbf{X}_t$.

We seek for the representer theory [4] to train a binary classifier which can be represented as

$$y = \sum_{i=1}^{n'_1} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) \quad (3.12)$$

where n'_1 is the number of training samples after outlier removal, $\mathbf{x}_i$ is the training samples from $\hat{\mathbf{X}}_s$, $\mathbf{x}$ is the sample for testing from $\mathbf{X}_t$, y is the predicted label and α_i is

the dual variable which can be solved by

$$\max_{\alpha_i} \sum_{i=1}^{n_1'} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n_1'} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) y_j \alpha_j \quad (3.13)$$

where y_i is the label of $\mathbf{x}_i$. We then introduce the kernel representation induced by the mapping function of unsupervised domain adaptation.

The optimal $\mathbf{M}^*$ of SA is obtained as $\mathbf{M}^* = \mathbf{U}_s^T \mathbf{U}_t$. Recall that the feature representation after PCA is given as $\hat{\mathbf{X}}_s \mathbf{U}_s$ and $\mathbf{X}_t \mathbf{U}_t$. Therefore, after obtaining $\mathbf{M}^*$, we can compute training kernel and testing kernel by

$$\begin{aligned} \mathbf{K}_{ss} &= \hat{\mathbf{X}}_s \mathbf{U}_s \mathbf{U}_s^T \mathbf{U}_t \mathbf{U}_t^T \mathbf{U}_s \mathbf{U}_s^T \hat{\mathbf{X}}_s^T \\ \mathbf{K}_{st} &= \hat{\mathbf{X}}_s \mathbf{U}_s \mathbf{U}_s^T \mathbf{U}_t \mathbf{U}_t^T \mathbf{X}_t^T \end{aligned} \quad (3.14)$$

$\mathbf{K}_{ss}$ is applied in (3.13) to train a binary classifier and $\mathbf{K}_{st}$ is applied for score prediction. The whole process of our algorithm is summarized in Algorithm 1 (the pipeline is shown in Fig. 3.3).

Algorithm 1: Unsupervised Domain Adaptation for Face Spoofing Detection

Input : Source domain face features $\mathbf{X}_s$, target domain face features $\mathbf{X}_t$, source domain label $\mathbf{y}_s$

Output: Classifier coefficients α_i

- 1 Compute the kernel representation $\mathbf{K}$ based on source domain data and target domain data;
 - 2 Optimize objective function (3.8) to obtain $\mathbf{p}$, then conduct outlier removal to obtain $\hat{\mathbf{X}}_s$ based on $\mathbf{p}$;
 - 3 Learn the mapping function $\mathbf{M}^*$ based on SA or KSA (3.11), where the dimension of PCA is chosen by following [107] with deviation vale as 10^5 and confidence as 0.1;
 - 4 Compute the kernel representation $\mathbf{K}_{ss}$ and $\mathbf{K}_{st}$ based on (3.14);
 - 5 Train a kernel based classifier to get α_i based on (3.13).
-

Figure 3.3: The framework of face spoofing detection with unsupervised domain adaptation (DA). After feature extraction from facial image, we first adopt a feature embedding by using Principle Component Analysis (PCA). Subsequently, we learn a mapping function to align the eigenspaces between source domain data and target domain data. Finally, a classifier is trained based on representer theory [4].

3.3 Feature Analysis

The state-of-the-art methods which specifically focus on developing meaningful features did not leverage the unlabeled testing samples during training. Therefore, in this chapter, we do not intend to compare with the results based on features and instead employ the state-of-the-art features for this specific task. In this section, we introduce and analyze the state-of-the-art image-based features for unsupervised domain adaptation based face anti-spoofing. In particular, to ensure sufficient statistical power, both hand-crafted and convolutional neural network (CNN) based features are employed, which are all widely adopted in research community so far.

3.3.1 Hand-crafted Features

Following the previous work on face spoofing detection [2, 3] and multimedia recapturing analysis [109], we regard the face anti-spoofing as a special image recapturing de-

tection problem based on the observation that face spoofing is ultimately a multimedia recapturing process (an example of genuine and spoofed faces are shown in Fig. 3.4). As such, three types of distortions are generally considered in the selection of hand-crafted features.

(i) Loss of details

When a scene is captured by a physical camera device, a certain level of detail loss is introduced into the digital image. One reason originates from sharpness reduction which is caused by aberration, lens aperture distortion, color filter array demosaicing and resizing. By recapturing the printed image on a paper or displayed image on a screen, there will be a dramatic increase in the degree of loss.

(ii) Color distortion

The imperfection of color production originates from the imperfect camera color filter and a limited gamut of the display medium. When an image is recaptured from a display medium, there will be a significant loss of color details in the recaptured one compared with the originally captured version. The color of recaptured images tends to be distorted with different brightness, saturation [110] and contrast [111] compared to the original image.

(iii) Additive texture patterns

During image recapturing process, the patterns caused by the superposition of two grid structures in fine scale are also inevitable due to many reasons, e.g. low resolution of printing and more pattern artifacts of displaying devices, regular texture patterns appearing on the printed papers and screens, and unique polarity inverse driving pattern. Therefore, the additive texture patterns can also provide us useful information for spoofing detection.

Figure 3.4: Comparison between a genuine face and spoofed face.

The pioneer work regarding face spoofing detection [37] used Fourier Spectrum, which can be viewed as the measurement of detail since the degree of loss will influence the energy of high frequency component in Fourier Spectrum. Cao *et al.* [109] imposed multi-scale Discrete Wavelet Transform (DWT) for loss-of-the-detail analysis. We adopt the statistical moments from [109] as part of feature based on the loss-of-the-detail artifact where the mean and the standard deviation of the absolute subband coefficients of DWT are computed. More specifically, consider the wavelet coefficients $\{y_{Q,1}, y_{Q,2}, \dots, y_{Q,N_Q}\}$ at the Q th subband, where N_Q is the number of coefficients in Q th subband. The mean m_Q and deviation σ_Q of the absolute coefficients are adopted as features

$$m_Q = \frac{1}{N_Q} \sum_{i=1}^{N_Q} |y_{Q,i}| \quad (3.15)$$

$$\sigma_Q = \sqrt{\frac{1}{N_Q} \sum_{i=1}^{N_Q} (|y_{Q,i}| - m_Q)^2} \quad (3.16)$$

The loss-of-the-detail artifacts are also highly correlated with image quality [112] where the spectrum coefficients are modeled in Gaussian General Density (GGD). In particular, it is defined as

$$f(x; \mu, \sigma, \gamma) = \left(\frac{\gamma}{2\sigma\Gamma(\frac{1}{\gamma})\beta} \right) e^{-\left(\frac{|x-\mu|}{\sigma\beta}\right)^\gamma} \quad (3.17)$$

where μ is the mean, σ is the standard deviation, γ is the shape parameter and β is the scale parameter defined as

$$\beta = \sqrt{\frac{\Gamma(\frac{1}{\gamma})}{\Gamma(\frac{3}{\gamma})}} \quad (3.18)$$

where $\Gamma(\cdot)$ denotes the gamma function.

By modeling the subband coefficients with GGD, we are more interested in the shape parameter γ which controls the distribution shape. In particular, a small γ value corresponds to fewer variations of wavelet subband coefficients and vice versa. Therefore, γ can be viewed as an indicator of the amount of detail loss in the facial image. We combine the moment m_Q , σ_Q and the γ value as our feature based on the loss-of-the-detail artifact. More specifically, we conduct a 3-level DWT on R, G, and B planes respectively and extract the wavelet features.

Hand-crafted texture features (e.g. LBP) are also proved to be effective for face spoofing detection [113]. One recent work [2] conducted analyses on various types of texture features (LBP, CoALBP [114], LPQ [115], BSIF [116] and SID [117]) and found that CoALBP and LPQ can achieve the state-of-the-art performance on these public databases. Moreover, the methods are extended for color information extraction in [2] since color distortion is also an important cue for face spoofing detection. In particular, color space conversion was conducted and the texture features were extracted from three different color channels regarding the color space. The authors also showed that color space conversion can improve the cross-database detection capability. In this chapter, we adopt the texture features CoALBP and LPQ as the descriptor to analyze the domain adaptation performance under the cross-database scenario.

CoALBP (Co-occurrence of Adjacent LBP) aims at richer and longer range pixel dependency [114] than traditional LBP [113]. CoALBP is an extension of LBP which extracts multiple (we use K as an example) co-occurrence dependencies of LBP pattern

L. Such co-occurrence dependencies can be represented as

$$h_k(i, j) = H(\mathbf{L}(x) - i, \mathbf{L}(x + \delta_k) - j), \quad k = 1, 2, \dots, K \quad (3.19)$$

where x is the location of LBP block $\mathbf{L}$ and h_k is the k th co-occurrences (where H is the co-occurrence operation defined in [114]) of couples of LBPs separated spatially by the δ_k . We then concatenate the vectors to obtain the final feature h . We follow the recent work [2] to extract CoALBP feature with the radius $R = \{1, 2, 4\}$ and the corresponding direction distance $B = \{2, 4, 8\}$.

LPQ (Local Phase Quantization) was proposed for texture classification when blurring appeared [115]. Given an image $\mathbf{I}$, the patch surrounding the pixel $\mathbf{x}$ is analyzed in the frequency domain Short-Time Fourier Transform (STFT),

$$\hat{\mathbf{I}}_{\mathbf{x}}(\mathbf{u}) = \sum_{\mathbf{y}} \mathbf{I}(\mathbf{y}) w(\mathbf{y} - \mathbf{x}) e^{-j2\pi \mathbf{u} \cdot \mathbf{y}} \quad (3.20)$$

where $\mathbf{x}$ and $\mathbf{y}$ are image spatial coordinates, $\mathbf{u}$ indicates spatial frequencies coordinate (in LPQ, only $\mathbf{u}_0 = (a, 0)$, $\mathbf{u}_1 = (0, a)$, $\mathbf{u}_2 = (a, a)$ and $\mathbf{u}_3 = (-a, a)$ are considered, $a \ll 1$), $w(\cdot)$ is a compact window which enforces transform locality and $\hat{\mathbf{I}}_{\mathbf{x}}(\mathbf{u})$ is the output of STFT. The basic LPQ component is defined as the $\mathbf{L}_{\mathbf{x}}(\mathbf{u}_i)$ which is the phase of STFT coefficients and can be quantized with a 2-bit uniform quantizer. The final LPQ feature is encoded as a 256-dimensional feature vector in classification. We follow [2] to extract LPQ feature by setting the parameters as $a = \frac{1}{7}$ and the neighboring block size as 7.

3.3.2 Deep Learning based Feature

We train a face spoofing detection CNN based on the architecture AlexNet [86] with both genuine and spoofed face images. In AlexNet, there are five convolutional layers

(with pooling layer and rectifier) and three fully-connected layers. The final layer is used for classification. Pre-trained AlexNet with ImageNet database is proven to be effective for fingerprint spoofing detection [74]. The recent work [118] on deep learning showed that the hidden layers transit from general to specific, implying that the shallow layers contain more general information (e.g. edge) while the deep layers tend to be customized towards a specific task. It is worth mentioning that the ImageNet ILSVRC competition is based on object recognition related tasks while the face spoofing detection mainly focuses on distortion cues which are different from object recognition. Therefore, we fix the five convolutional layers as the weights pre-trained by ImageNet which are more related to general tasks. For fully-connected layers, we change the final layer from 1000 nodes (in AlexNet) to 2 nodes since we only have 2 classes (genuine, spoofed) for classification. We then randomly initialize the weight of fully connected layer as Gaussian distribution $N(0, 0.001)$. During training, we back-propagate the gradient through the whole network to update the parameters. Stochastic Gradient Descent (SGD) is employed for training. We set the momentum to 0.9, learning rate to 0.001, weight decay to 0.0005 and learning rate decay to 10^{-7} . We use Caffe framework to train the network.

3.4 Experiments and Discussions

3.4.1 Databases

For face anti-spoofing, the commonly used public databases are Idiap REPLAY-ATTACK [40], CASIA Face AntiSpoofing [93] and MSU mobile face spoofing database [3].

These three databases have been widely adopted for spoofing detection evaluation, and yet there exist some limitations. For REPLAY-ATTACK database, although it consists of various illumination conditions, the camera model and attacking types are limited. For CASIA database, the face videos are captured by three different camera models

with different quality levels and attacking types. However, the illumination condition is fixed. MSU database takes both illumination condition and camera model variations into consideration. However, compared with Idiap REPLAY-ATTACK database and CASIA database, the number of face video and attacking types is limited. This further inspires us to build a more generalized face anti-spoofing database, targeting at compensating for the features missing from the current databases.

This inspires us to build a new face anti-spoofing database which contains mobile phones with both high, middle and low-quality levels. In particular, important factors that need to be taken into consideration include:

- The face samples contained in the database should be captured in real-world scenarios and the spoofing medium will be able to bypass the face verification system without anti-spoofing detection.
- Face videos captured with a large variety of cameras under different quality levels and illumination settings are preferred.
- Different spoofing attacking types are better to be considered to improve the robustness of anti-spoofing.

We introduce a new and more comprehensive face anti-spoofing database, Rose-Youtu Liveness Database, which covers a large variety of illumination conditions, camera models, and attack types. The Rose-Youtu Liveness Database (Rose-Youtu) consists of more than 4000 videos with 25 subjects in total (with 20 subjects publicly available). The scale is large in contrast to CASIA's 600 videos, Idiap's 1200 videos, and MSU's 280 videos. For each subject, there are 150-200 video clips with the average duration around 10 seconds. Five mobile phones were used to collect the database: (a) Hasee smart-phone (with the resolution 640×480), (b) Huawei Smart-phone (with the resolution 640×480), (c) iPad 4 (with the resolution 640×480), (d) iPhone 5s (with the resolution 1280×720) and (e) ZTE smart-phone (with the resolution 1280×720). All

Figure 3.5: Examples of Rose-Youtu Liveness Database. From top to bottom: face images in genuine, cropped mask, full mask, upper mask, paper print and video replay versions. (For paper print attack, both warped paper and still paper attacks are considered.) From left to right: face images captured by iPhone 5S, Hasee mobile phone, Huawei mobile phone, iPad and ZTE mobile phone.

face videos are captured by a front-facing camera. The standoff distance between face and camera is about 30 – 50 cm.

We consider three spoofing attack types including printed paper attack, video replay attack, and masking attack. For printed paper attack, face image with still printed paper and quivering printed paper (A4 size) are used. For video replay attack, we display a face video on Lenovo LCD screen (with the resolution 4096×2160) and Mac screen (with the resolution 2560×1600). For masking attack, masks with and without cropping are considered. Moreover, the face videos are captured with different backgrounds which guarantee the face videos are coupled with different illumination conditions. To keep consistent with the genuine face video, the standoff distance between spoofing medium and camera is also about 30 – 50 cm. Some examples of our database are shown in Fig. 3.5. We divide the Rose-Youtu Database into training and testing subsets. Videos belonging to the first 10 indexed subjects are used for training and the others are for testing.

3.4.2 Experimental Settings

In this chapter, we adopt Viola-Jones algorithm [94] for face detection. For texture feature (CoALBP, LPQ) extraction, we follow [2] to resize the face region into 64×64 . For deep learning based feature extraction, we normalize the facial region to be 227×227 . The original facial image is used for wavelet feature since image resizing may introduce unexpected detail loss. For cross-database scenario, we conduct domain adaptation between training and testing samples before classification. The parameters C for SVM classifier with the linear kernel is determined based on cross-validation. The PCA dimension of domain adaptation is determined by the theory in [107]. In particular, we set the deviation value equaling to 10^5 and confidence equaling to 0.1 as suggested in [107].

As mentioned before, in this chapter, the state-of-the-art features are employed in

the face anti-spoofing with domain adaptation. As such, the comparisons of the experimental results are conducted for each adopted feature separately, which can better reflect the performance of the proposed scheme. In the experiment, we thus evaluate the performance by considering both hand-crafted and deep learning based features to show the effectiveness of our proposed method. We first show the performance of intra-database. The evaluation protocols of Idiap REPLAY-ATTACK, CASIA and MSU databases are consistent with the prior works [3, 40, 93]. In particular, Half Total Error Rate (HTER) is employed for Idiap REPLAY-ATTACK database, and Equal Error Rate is used for CASIA and MSU databases. For our new developed Rose-Youtu database, we use the first 10 indexed subjects for training and the remaining subjects for testing. The results are shown in Table 3.1, which demonstrate the effectiveness of our adopted features. We can observe that for CASIA and Idiap REPLAY-ATTACK databases, the hand-crafted feature CoALBP achieves the best performance while deep learning based feature performs better in MSU and Rose-Youtu databases.

Table 3.1: Results of intra-database performance

	CASIA	Idiap	MSU	Rose-Youtu	Rose-Youtu*
Wavelet	10.9%	9.9%	12.8%	26.6%	25.7%
CoALBP (HSV)	5.5%	3.7%	9.8%	16.4%	16.6%
CoALBP (YCbCr)	10.0%	1.4%	8.1%	17.1%	17.5%
LPQ (HSV)	7.4%	7.9%	12.2%	30.4%	28.9%
LPQ (YCbCr)	16.2%	6.3%	7.4%	27.6%	25.0%
Deep Learning based Feature	7.6%	2.1%	5.8%	8.0%	8.0%

[†] Half Total Error Rate (HTER) is employed for Idiap REPLAY-ATTACK database, and Equal Error Rate is used for CASIA, MSU and Rose-Youtu databases. * refers to the results obtained based on public released database.

3.4.3 Intra-Database Experimental Results

First, we can observe that among the features adopted, wavelet features turn out to achieve worse performance compared with other features. Such results are reasonable

since the motivation of using wavelet feature is to extract the loss-of-the-detail information, which is very likely to be influenced by many factors (e.g. the quality of the camera and attack mediums, the distance between person and camera, etc). This may deteriorate the discriminative capability of wavelet feature. Deep learning based features are obtained in a data-driven manner. Therefore, it is also reasonable that deep learning based features can achieve relatively better performance among all the features. For CoALBP feature, we notice that it has achieved the best performance on some of the databases which are less diverse. However, the performance drops rapidly on the database with diverse content (e.g. Rose-Youtu database). Another interesting observation is that when using the same feature in different color spaces, the performance can be different. We conjecture the reason that different color spaces are designed for different purposes, such that the extracted features can also be encoded with different discriminative information.

Furthermore, we evaluate the performance of the cross-database face spoofing detection by imposing unsupervised domain adaptation with various features. To make fair comparisons with recent works [68][2], HTER is used for cross-database scenario evaluation. We follow the protocol defined in [68] which divides the training database into five folds. In particular, one of them is used as the development set for threshold τ determination, and the others are used for training. The final HTER performance is obtained by averaging the results. Considering that the outlier removal method originates from kernel mean matching algorithm (KMM) [100] which was proposed for distribution alignment, we consider the KMM method as one of our baselines.

3.4.4 Cross-Database Experimental Results

Three public databases (CASIA, Idiap REPLAY-ATTACK, and MSU) and our newly developed Rose-Youtu liveness database (with 25 subjects) are used for cross-database spoofing detection evaluation. To facilitate the comparison, one database (composed of both training and testing folds together) is used for training and another is used for test-

Table 3.2: Performance (HTER) of cross-database with Wavelet statistical feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	49.9%	49.2%	47.7%	48.6%	49.1%	50.0%
KMM	50.0%	43.7%	51.1%	45.2%	46.2%	49.9%
SA	39.6%	36.9%	35.2%	33.1%	48.4%	37.9%
KSA	37.5%	22.1%	36.7%	35.8%	42.4%	45.6%
SA[§]	36.8%	28.5%	34.3%	31.3%	41.4%	35.2%
KSA[§]	33.1%	19.1%	32.1%	33.9%	41.2%	35.1%

[†] “C”, “I” and “M” denote CASIA, Idiap REPLAY-ATTACK and MSU database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

ing. Thus we have 12 scenarios in total. Various feature representations are considered for evaluation. However, deep learning based feature is only considered when using Rose-Youtu database for either training or testing since we observe that only the deep learning based feature can achieve desired performance on Rose-Youtu database. For unsupervised domain adaptation, we report the results with and without outlier removal. We use “w/o DA” to represent without domain adaptation and “[§]” to denote the domain adaptation with outlier removal. We also use “C”, “I”, “M” and “Y” to denote CASIA, Idiap REPLAY-ATTACK, MSU and Rose-Youtu database respectively. The experimental results for cross-database face spoofing detection are shown from Table 3.2 to Table 3.7.

Based on the results, we can observe that cross-database performance highly depends on the databases we use. For example, when using CASIA and MSU for training and testing, we can achieve around 25% HTER with CoALBP in HSV color space, around 15% HTER with CoALBP in YCbCr color space and deep learning based feature. Moreover, deep learning based feature also achieves around 30% HTER under several other scenarios. However, we can only obtain close to 50% for other cases, which correspond to the random-guess scenario. We conjecture the reason may lie in that the face videos from these two databases were captured under dissimilar illumina-

Table 3.3: Performance (HTER) of cross-database with CoALBP (HSV) feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	50.3%	24.9%	50.0%	50.0%	50.0%	50.0%
KMM	50.0%	22.4%	54.3%	60.0%	34.9%	51.9%
SA	40.2%	22.9%	37.5%	31.5%	40.9%	35.4%
KSA	37.5%	21.6%	41.2%	32.9%	37.3%	39.0%
SA[§]	33.4%	21.7%	33.2%	29.2%	37.7%	30.6%
KSA[§]	35.1%	20.9%	39.8%	29.0%	34.2%	36.9%

[†] “C”, “I” and “M” denote CASIA, Idiap REPLAY-ATTACK and MSU database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

Table 3.4: Performance (HTER) of cross-database with CoALBP (YCbCr) feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	50.0%	15.1%	50.1%	50.0%	44.8%	50.0%
KMM	50.0%	15.0%	50.0%	50.0%	42.8%	51.0%
SA	46.1%	15.4%	39.3%	41.7%	35.5%	51.3%
KSA	37.5%	21.2%	41.6%	33.1%	37.3%	42.9%
SA[§]	45.3%	14.9%	34.5%	40.5%	34.9%	47.9%
KSA[§]	35.1%	20.9%	39.7%	29.0%	34.2%	36.9%

[†] “C”, “I” and “M” denote CASIA, Idiap REPLAY-ATTACK and MSU database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

Table 3.5: Performance (HTER) of cross-database with LPQ (HSV) feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	45.5%	54.9%	43.7%	53.5%	58.7%	59.1%
KMM	44.1%	52.8%	41.0%	57.5%	49.5%	50.7%
SA	37.5%	28.0%	41.4%	29.5%	50.0%	35.9%
KSA	38.3%	41.0%	42.7%	37.5%	42.2%	39.4%
SA[§]	33.4%	21.2%	39.1%	24.9%	42.8%	33.2%
KSA[§]	36.4%	39.7%	39.0%	35.7%	39.8%	37.8%

[†] “C”, “I” and “M” denote CASIA, Idiap REPLAY-ATTACK and MSU database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

Table 3.6: Performance (HTER) of cross-database with LPQ (YCbCr) feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	43.9%	44.3%	49.9%	46.2%	46.8%	50.0%
KMM	48.4%	49.4%	50.0%	36.6%	48.5%	49.7%
SA	47.9%	25.2%	39.4%	30.2%	36.5%	35.8%
KSA	43.1%	25.3%	44.6%	38.1%	46.0%	46.5%
SA[§]	40.7%	16.9%	33.1%	27.8%	33.3%	31.8%
KSA[§]	41.2%	16.3%	42.0%	32.5%	43.5%	41.6%

[†] “C”, “I” and “M” denote CASIA, Idiap REPLAY-ATTACK and MSU database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

Table 3.7: Performance (HTER) of cross-database with deep learning based feature

	$C \rightarrow I$	$C \rightarrow M$	$C \rightarrow Y$	$I \rightarrow C$	$I \rightarrow M$	$I \rightarrow Y$	$M \rightarrow C$	$M \rightarrow I$	$M \rightarrow Y$	$Y \rightarrow C$	$Y \rightarrow I$	$Y \rightarrow M$
w/o DA	45.8%	15.6%	46.8%	34.4%	68.6%	48.0%	50.1%	49.9%	31.0%	32.6%	43.6%	28.4%
KMM	44.9%	14.1%	46.7%	25.9%	49.7%	47.9%	32.1%	51.2%	31.3%	31.6%	43.6%	27.8%
SA	41.4%	14.0%	32.7%	36.5%	35.4%	43.2%	13.4%	34.9%	30.3%	35.0%	38.5%	29.0%
KSA	43.1%	16.9%	34.1%	22.0%	39.1%	40.4%	19.8%	37.2%	30.8%	33.9%	42.0%	29.2%
SA[§]	39.2%	14.3%	31.6%	26.3%	33.2%	42.8%	10.1%	33.3%	30.0%	30.7%	36.2%	24.9%
KSA[§]	39.3%	15.1%	33.9%	12.3%	33.3%	40.1%	9.1%	34.9%	30.4%	30.1%	38.8%	26.1%

[†] “C”, “I”, “M” and “Y” denote CASIA, Idiap REPLAY-ATTACK, MSU and Rose-Youtu database respectively, $A \rightarrow B$ refers to using database A for training and B for testing, “[§]” refers to the domain adaptation technique with outlier removal.

tion conditions. Moreover, the influences of illumination appear to be more dominant in face spoofing detection task compared with facial appearance and camera quality, since we cannot achieve satisfied performance based on the cross-database evaluation performances between Idiap REPLAY-ATTACK with these features.

3.4.4.1 Outlier Analysis

We then revisit the characteristics of samples which are likely to be categorized as outliers. Basically, outliers are jointly determined by the source and target domains. When CASIA database is used for source domain, the blurry and low-contrast samples are treated as outliers since the samples from other databases have relatively better quality and contrast. For Idiap Replay-Attack database, we can see that outliers samples can

contain glasses reflection, which is reasonable since we notice that the reflection from other databases is not as strong as Idiap Replay-Attack. For MSU database, we observe that the illumination plays an important role. (For other databases there are not many strong illumination samples.) For Rose-Youtu database, we notice that all the subjects have outlier samples. We further analyze the content of the database and find the outliers may come from the unexpected motion (due to camera moving), the distance between camera and client, and blurring. We list several outlier samples in Fig. 3.6.

Figure 3.6: Outlier samples selected by the proposed method. The first row shows the genuine samples and the second row shows the spoofed samples. The samples are collected from CASIA, Idiap REPLAY-ATTACK, MSU and Rose-Youtu databases (from left to right).

Table 3.8: Performance improvement with CoALBP feature in HSV space.

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	0.0	0.0	0.0	0.0	0.0	0.0
SA	16.9	3.2	16.8	20.8	12.3	19.4
KSA	15.2	4.0	10.2	21.0	15.8	13.1

3.4.4.2 Benefits of Features and Domain Adaptation

Based on the results from the cross-database scenario, we can find that our proposed unsupervised domain adaptation scheme can significantly improve the performance of cross-database face spoofing detection, which indicates the effectiveness of domain adaptation scheme. We also show the results in Table 3.8 by comparing the performance improvement at each domain adaptation scenario with CoALBP feature in HSV color space as an example. We notice that the results are variant among different scenarios. This observation shows that the powerful features across domain and the unsupervised domain adaptation technique jointly improve the spoofing detection performance. This further provides useful evidence that our framework is effective in terms of the generalization ability of face spoofing detection. However, it is also worth mentioning that the bottleneck of domain adaptation exists when domain adaptation only achieves little performance improvement. One example can be found by using CASIA for training and MSU for testing. We can observe that a relatively good cross-database performance with CoALBP and deep learning based features can be achieved, which implies that the feature distributions of these two domains could be similar. Therefore, domain adaptation may not drive further performance improvement in this scenario. Moreover, we can observe the asymmetric performance by reversing the database for training and testing. For example, significant performance improvement can be achieved from MSU to CASIA with CoALBP feature and deep learning based feature. However, a little performance gain is observed for the scenario of CASIA to MSU. We conjecture two possible reasons regarding this issue.

- The performance can be highly relevant with the employed databases. Since CASIA database is considered to be more diverse compared with MSU due to the variations of camera models and attacking types, it is reasonable that training on CASIA can generalize better compared with training on MSU. Therefore, we observe a large performance gap between training on CASIA and training on MSU.

After conducting domain adaptation, we observe that such performance gap drops significantly, and the results of training on CASIA and testing on MSU are much closer to the results training on MSU and testing on CASIA (compared with the original 15.6% and 50.1%), which demonstrate the effectiveness of domain adaptation that aligns two different distributions closer. We can also observe that the performance is only improved by 0.5%, by training on CASIA and testing on MSU. The reason for this phenomenon may lie in that the baseline model has already achieved a relatively good performance.

- On the other hand, although subspace alignment learns a transformation (mapping) from one domain to another, the classifiers trained by source domain are different and this can also lead to different performances. Moreover, we also observe that although the improvement gaps are different for different databases, our framework can generally improve the accuracy based on different types of features and databases.

We also notice that domain adaptation may not achieve significant performance improvements in some other cases (e.g. training on MSU and testing on Rose-Youtu). In essence, domain adaptation aims to solve the classification problem when the training samples and testing samples have different distributions. However, as indicated in [119], domain adaptation cannot fully address the problem when there is a dramatic deviation for the distribution of target domain from source domain encoded by a specific feature. For the Rose-Youtu database, the face samples are captured in an uncontrolled condition by various camera models under diverse illumination conditions. As such, we conjecture that the distribution of the samples from Rose-Youtu database is quite different from other databases due to the uncontrolled capturing conditions of Rose-Youtu database. Such large gap cannot be compensated by domain adaptation either due to the inherent limitation of the domain adaptation [119]. Therefore, we notice that by using Rose-Youtu database for both training and testing with domain adaptation, the

performance does not have significant improvement for most of the cases.

Another interesting observation is that outlier removal can further boost the domain adaptation performances with at least 2 – 3%. This is reasonable since removing the outliers can provide us reliable samples for conducting domain adaptation and learning the powerful prediction model, as shown in the feature visualization in Fig. 3.1. The asymmetric performance can also be observed with/without outlier removal when considering domain adaptation, as little improvement can be achieved when training on CASIA and testing on MSU while large improvement can be obtained when using MSU for training and CASIA for testing. In addition to the reasons aforementioned, another explanation for this phenomenon lies in that the percentages of outliers can also be different by using different types of features even when we conduct outlier removal on the same database. The average percentages of removal outlier samples are shown in Table 3.9. Therefore, by using different types of features, the numbers of outlier samples are different. This may result in the asymmetric performances as well. In addition, since in some scenarios, the model has already achieved relatively good performance, both domain adaptation and outlier removal will only achieve limited improvement.

Table 3.9: Percentage of outlier samples for different databases

	C	I	M	Y
Multi-scale Wavelet	6.5%	6.2%	11.5%	—
CoALBP (YCbCr)	6.5%	3.8%	6.4%	—
CoALBP (HSV)	6.3%	8.8%	12.1%	—
LPQ (YCbCr)	9.8%	10.2%	6.7%	—
LPQ (HSV)	10.6%	11.6%	10.0%	—
Deep Learning	4.4%	2.4%	3.5%	6.7%

Moreover, we swap the process of domain adaptation and outlier removal to analyze the influences of outlier removal process on classifier training. The results of CoALBP feature in HSV space and deep learning based feature by using Subspace Alignment are listed in Tables 3.10 and 3.11. It is observed that outlier removal process is effective for domain adaptation. Moreover, outlier removal can also help with training a better

classifier by comparing the results of only domain adaptation with the results of first performing domain adaptation followed by outlier removal.

Table 3.10: Performance (HTER) comparisons between OR+DA and DA+OR (CoALBP HSV).

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	50.3%	24.9%	50.0%	50.0%	50.0%	50.0%
only DA	40.2%	22.9%	37.5%	31.5%	40.9%	35.4%
OR+DA	33.4%	21.7%	33.2%	29.2%	37.7%	30.6%
DA+OR	37.5%	22.8%	37.6%	29.1%	34.0%	35.2%

Table 3.11: Performance (HTER) comparisons between OR+DA and DA+OR (Deep Learning based Feature).

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	45.8%	15.6%	34.4%	68.6%	50.1%	49.9%
only DA	41.4%	14.0%	36.5%	35.4%	13.4%	34.9%
OR+DA	39.2%	14.3%	26.3%	33.2%	10.1%	33.3%
DA+OR	41.0%	13.9%	33.7%	34.9%	13.4%	33.3%

3.4.4.3 Influences of Different Number of Samples

We further analyze the influences of the number of data from the target domain on the final performance with SA method. Specifically, we conduct an experiment with CoALBP feature in HSV space by using different percentages of samples from the target domain. The results are shown in Fig. 3.7, where the performance with different percentage of target domain samples (1%, 5%, 10%, 20%, 50%, 100%) are demonstrated. It is also worth mentioning that the results are obtained by averaging the results for five times. We can see that the more target domain samples we can get, the lower HTER can be achieved.

Figure 3.7: HTER performance (with SA method) with the variation of target domain data number. (a) train: CASIA, test: Idiap, (b) train: CASIA, test: MSU, (c) train: Idiap, test: CASIA, (d) (train: Idiap, test: MSU), (e) train: MSU, test: CASIA, (f) train: MSU, test: Idiap

3.4.4.4 Unbalanced Domain Adaptation

In practice, it may not be convenient to collect spoofed face data. To investigate the scenario that only genuine samples are available for domain adaptation in the target domain, we conduct experiments by using both hand-crafted feature (CoALBP feature on HSV space) and deep learning based feature by KSA method since we observe KSA can achieve better performance when only obtaining genuine data in hand. The results are shown in Tables 3.12 and 3.13. Based on the results, we find that the performance drop is marginal when only using genuine samples for domain adaptation for most of the cases. Moreover, in almost all cases, the domain adaptation with genuine samples perform much better compared with the method without domain adaptation. This can be explained by the reason that only genuine samples can still provide valuable cross-domain information (e.g. facial appearance, lighting, camera quality) for domain adaptation.

However, for some databases the performance drops a lot when only adopting only genuine samples compared with the results by adopting both genuine and spoofed samples. This may originate from the distinct characteristics of the databases. In some databases (e.g. Idiap and MSU), the illumination condition and motion information are more consistent for genuine and spoofed samples. As such, the distribution of the target domain will not change significantly after we remove the spoofed samples. However, it can be noticed that in some other databases (e.g. CASIA and Rose-Youtu) large motion variations and diverse illumination conditions exist, especially when comparing genuine and spoofed samples. In this case, the distribution between the source and target domain will be changed, and this may significantly influence the effectiveness of domain adaptation. Moreover, when the performance without domain adaptation can already reach a saturation level, domain adaptation may not help a lot no matter what data are finally used in target domain.

To analyze the influence by using the different number of genuine data, we also conduct an experiment with CoALBP feature in HSV space by using different percentage

Table 3.12: HTER Results (%) by Kernel Subspace Alignment with CoALBP (HSV)

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	50.3	24.9	50.0	50.0	50.0	50.0
Genuine and Fake	35.1	20.9	39.8	29.0	34.2	36.9
Only Genuine	42.5	26.5	46.3	33.4	36.4	45.6

Table 3.13: HTER Results (%) by Kernel Subspace Alignment with Deep Learning based Feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	45.8	15.6	34.4	68.6	50.1	49.9
Genuine and Fake	39.3	15.1	12.3	33.3	9.1	34.9
Only Genuine	44.2	16.1	28.0	37.5	13.6	40.2

(1%, 5%, 10%, 20%, 50%, 100%) of genuine samples from the target domain. The results are shown in Fig. 3.8. We can observe that the more target domain samples we can obtain, the lower HTER can be achieved. This trend is consistent with the results when using both genuine and spoofed data for domain adaptation. This further provides useful evidence on the effectiveness of the proposed scheme.

3.4.4.5 Comparison with Domain Adaptation Based Feature Synthesis

In [5], a domain adaptation based feature synthesis method was proposed to tackle the cross-domain face anti-spoofing problem. Considering that it is usually applicable to collect genuine samples in the target domain, we compare the performance of the domain adaptation approach in [5] with our proposed method. Specifically, we divide our target domain into two parts. One part has labeled genuine face samples which are used for spoofed feature synthesis and classifier training, and the other part has no label information which is used for testing. The Center Shift algorithm reported in [5] is employed to learn the mapping function based on source domain data. The results are shown in Tables 3.14 and 3.15.

As we can see from the results, such adaptation method cannot improve the performance of cross-domain face spoofing detection. This may be reasonable since our work

Figure 3.8: HTER performance (with KSA method) with the variation of the number of genuine target domain data. (a) train: CASIA, test: Idiap, (b) train: CASIA, test: MSU, (c) train: Idiap, test: CASIA, (d) (train: Idiap, test: MSU), (e) train: MSU, test: CASIA, (f) train: MSU, test: Idiap

Table 3.14: Performance comparisons between feature synthesis [5] and our methods. (CoALBP HSV)

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	50.3	24.9	50.0	50.0	50.0	50.0
[5]	51.2	28.9	51.3	49.1	40.2	52.2
Ours	33.4	21.7	33.2	29.2	37.7	30.6

Table 3.15: Performance comparisons between feature synthesis [5] and our method. (Deep Learning based Feature)

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$
w/o DA	45.8	15.6	34.4	68.6	50.1	49.9
[5]	49.2	18.1	39.6	36.7	49.6	49.6
Ours	39.2	14.3	26.3	33.2	10.1	33.3

is different from [5] in several ways as stated as follow.

- The motivation of [5] is that the spoofed samples of target domain are difficult to collect. Therefore, they proposed a feature synthesis method by utilizing the labeled data in the target domain. By contrast, our work is conducted in a totally unsupervised manner and no labeled information is available in the target domain.
- The work from [5] is based on prior knowledge of person identification and camera information. Therefore, a linear adaptation can be employed for feature synthesis. Their assumptions are reasonable based on the applications such as door access control, where person identity and camera information are available. However, we are dealing with uncontrolled cross-domain face spoofing detection where the face identity, camera model are totally different from the training data. Their assumptions no longer hold in such scenario.

3.4.4.6 Results by Using Concatenated Feature

As discussed in [2], concatenating different features usually leads to a better performance than using each individual one. Therefore, we follow [2] to concatenate the CoALBP and LPQ feature in HSV and YCbCr color space into a single feature and

Table 3.16: Performance (HTER) of cross-database with concatenated feature

	$C \rightarrow I$	$C \rightarrow M$	$I \rightarrow C$	$I \rightarrow M$	$M \rightarrow C$	$M \rightarrow I$	Average
w/o DA (reported in [2])	30.3	20.4	37.7	34.1	46.0	33.9	33.7
w/o DA (reproduced by ourselves)	29.6	18.5	39.9	29.1	44.7	37.0	33.1
SA	35.8	21.7	42.4	20.8	41.2	25.4	31.2
KSA	30.5	21.0	38.5	20.9	43.0	26.8	30.1
SA[§]	33.9	20.2	41.4	18.6	40.5	23.3	29.7
KSA[§]	27.4	20.3	36.0	18.6	40.1	24.0	27.7

conduct cross-database face spoofing detection experiments based on CASIA, Idiap REPLAY-ATTACK, and MSU databases. The results are reported in Table 3.16. We observe that all methods are generally improved by using the concatenated feature. Our proposed method can still achieve performance improvement in most of the cases, which again shows the effectiveness of our proposed domain adaptation scheme for cross-domain face spoofing detection using different types of features.

3.4.5 Discussions

Practically, deploying face anti-spoofing system in real application scenarios requires the algorithm to have a good generalization ability to deal with various acquisition conditions. Since the creations of these databases are totally independent, our experiments take advantage of this benefit and show that the proposed approach can significantly improve the performance in cross-database scenarios. This further demonstrates the strong generalization ability of the scheme. As the first attempt on unsupervised domain adaptation in face anti-spoofing, several limitations existing in our approach should be improved in the future.

Firstly, although state-of-the-art features are incorporated in the current framework, how to design and learn sophisticated features that can better fit the domain adaptation scheme should be further investigated. Secondly, current domain adaptation scheme requires sufficient data to transfer the knowledge from the source domain to the target domain, and in the future domain adaptation with zero-shot learning strategy will be

studied. As such, we can transfer the pre-trained knowledge even to a single face image. Finally, the complexity of the proposed scheme will be further reduced by speeding up the kernel method to meet the real-time requirement of face spoofing detection.

3.5 Summary

We proposed a novel framework utilizing advanced unsupervised domain adaptation algorithms for face anti-spoofing. The novelty of this framework lies in transferring the feature space of face samples from the labeled source domain to the unlabeled target domain, such that reliable model can be learned for spoofing detection. State-of-the-art hand-crafted and deep learning based features are incorporated into the domain adaptation framework and their classification accuracies are further evaluated. Extensive experiments have been conducted based on the available databases and our new database. The results show that we can achieve clearly improved generalization ability with an average of 20% improvement by domain adaptation as compared with the straightforward learning approach without domain adaptation.

Chapter 4

Learning Generalized Deep Feature Representation for Face Spoofing Detection

In the previous two chapters, we assume that the information related with testing face samples is more or less available during training stage. However, in practical scenario, such information can also be unknown. With numerous approaches proposed to deal with the artifacts within a single image which can be better generalized to testing face samples, there are still two important issues in face anti-spoofing. On the one hand, how to generalize well to the “unseen data” needs further investigation, as obtaining enough data with sufficient variability in the training process is not always practical. On the other hand, much less work has been dedicated to extracting information along the temporal direction, which can also provide valuable cues (liveness information, unexpected motion [31, 32], temporal aliasing, etc.). More importantly, learning spatial plus temporal features would become more difficult as more training data would be necessary and the lack of generalization could be even more pronounced. All these issues cast challenges on the generalization capability of robust feature representation. In view of this,

we focus on deep feature representation in a generalized way by exploiting the information from both spatial and temporal dimensions. In particular, 3D convolutional neural networks (3D CNN), which have been proved to be efficient for action recognition task [120], are employed to learn spoofing-specific information based on typical printed and replay video attacks. The solution incorporates 2D and 3D features related to the presentation attack problem, and learns not only spatial variations associated with attacks but also artifacts that take place over time. More specifically, we adopt the 3D CNN architecture with a data augmentation strategy for the spoofing detection task. To obtain a more robust and generalized 3D CNN model, the lack of generalization is dealt with by introducing a regularization mechanism, which focuses on improving classification accuracy during training as well as generalizing to unknown conditions by minimizing the feature distribution dissimilarity across domains. These capabilities allow us to make a further step regarding the detection of attacks under unknown or different conditions.

4.1 Introduction

Generally speaking, both spatial and temporal artifacts (e.g., unexpected texture patterns, color distortions and blurring [109, 121]) may occur during the face spoofing process. Regarding the texture pattern, such pattern appearing in spatial dimension is caused by the mismatch of the replay device resolution and the capturing device resolution, while in temporal domain it is derived from the divergence between flash frequency of display device (e.g., 120 Hz) and the video frequency (e.g., 25 Hz). The color distortion is due to the mismatch of color gamut between the display medium and the recapturing model. Besides the texture pattern and color distortion, the unexpected motion such as display device shaking along the temporal dimension can also be beneficial for spoofing detection. Instead of using the hand-crafted features in inferring the distinctive information, applying Convolutional Neural Network (CNN) to spoofing detection has shown

promising results for different spoofing setups. However, as the current adopted CNN models for spoofing detection are all based on 2D images trained in a label-guided manner [73, 74], there are two outstanding limitations:

- Due to the limitation of the 2D CNN structure, the temporal statistics encoded in contiguous frames are ignored.
- Directly applying the classification loss with label information can lead to overfitting problem to a certain database collection. In this scenario, the trained model cannot generalize well to the unseen data.

In view of these limitations, we develop a 3D CNN architecture such that discriminative information can be learned from both spatial and temporal dimensions. In particular, when training and testing samples are captured under similar environments, our model can achieve lower error rate compared with 2D CNN models as well as other handcrafted features used in prior art. More importantly, when training a CNN by considering face samples collected from different cameras under diverse illumination conditions, the extracted features across domains are expected to lie in a similar manifold such that a classifier trained with such features will have better generalization ability. In view of this, we also take advantage of domain generalization in network training by introducing a regularization term, which forces the learned features to share similar distributions. The pipeline of our proposed scheme is shown in Fig. 4.1.

4.2 Proposed Method

4.2.1 3D Convolutional Neural Network

In the 2D convolutional neural network, the convolution process is only applied on the 2D feature maps to compute the response in the spatial dimension, which has largely ignored the temporal information. In contrast with 2D CNN, the 3D CNN is conducted

Figure 4.1: The pipeline of the proposed scheme for face spoofing detection. The final objective function is determined by both classification loss and generalization loss. The output from FC2 layer is employed as latent discriminative feature for classification. The volume conv model contains the volume convolution layer, BatchNormalization, LeakyReLU, MaxPooling and Dropout layer. The second fully connected layer (FC2) is used for latent discriminative feature extraction.

by convolving an input cube, which is stacked by multiple contiguous frames with a 3D kernel (we visualize the architecture in Fig. 4.2). We refer to the 3D convolution kernel size in the l -th layer by $W_l \times H_l \times T_l$, where T_l denotes the temporal depth and $W_l \times H_l$ represents the spatial size of the kernel. As such, the temporal information can also be preserved in the feature map. By jointly considering the temporal information, we can achieve better feature learning capability for face spoofing detection. In particular, each convolution operation is performed followed by a non-linear activation function such as ReLU. Mathematically, such process can be formulated as

$$y_{d_2,l}^{ijk} = \sum_{d_1} \sum_{m=0}^{W_l-1} \sum_{n=0}^{H_l-1} \sum_{p=0}^{T_l-1} w_{d_1,d_2,l}^{mnp} x_{d_1,l-1}^{(i+m)(j+n)(k+p)} + b_{d_2,l} \quad (4.1)$$

$$x_{d_2,l}^{ijk} = \sigma(y_{d_2,l}^{ijk})$$

where $x_{d_1,l-1}^{ijk}$ is the value of a unit at position (i, j, k) in the d_1 -th feature map from the $(l-1)$ -th layer, $w_{d_1,d_2,l}^{mnp}$ is the value of the element at position (m, n, p) of the 3D

Table 4.1: The Proposed 3D CNN Architecture

Layer	Type/Module	Output Size	Filter/Pooling Size	Setting
1	3D Convolution	$128 \times 8 \times 128 \times 128$	$3 \times 3 \times 3$	
2	3D BatchNormalization			
3	LeakyReLU			Leaky Factor: 0.1
4	3D MaxPooling	$128 \times 8 \times 64 \times 64$	$1 \times 2 \times 2$	
5	3D Convolution	$128 \times 8 \times 64 \times 64$	$3 \times 3 \times 3$	
6	3D BatchNormalization			
7	LeakyReLU			Leaky Factor: 0.1
8	3D MaxPooling	$128 \times 8 \times 32 \times 32$	$1 \times 2 \times 2$	
9	3D Convolution	$128 \times 8 \times 32 \times 32$	$3 \times 3 \times 3$	
10	3D BatchNormalization			
11	LeakyReLU			Leaky Factor: 0.1
12	3D MaxPooling	$128 \times 8 \times 16 \times 16$	$1 \times 2 \times 2$	
13	3D Convolution	$128 \times 8 \times 16 \times 16$	$3 \times 3 \times 3$	
14	3D BatchNormalization			
15	LeakyReLU			Leaky Factor: 0.1
16	3D MaxPooling	$128 \times 4 \times 8 \times 8$	$2 \times 2 \times 2$	
17	3D Convolution	$128 \times 4 \times 8 \times 8$	$3 \times 3 \times 3$	
18	3D BatchNormalization			
19	LeakyReLU			Leaky Factor: 0.1
20	3D MaxPooling	$128 \times 2 \times 4 \times 4$	$2 \times 2 \times 2$	
21	Linear	1024		
22	BatchNormalization			
23	ReLU			
24	Dropout			Dropout Rate: 0.5
25	Linear	2		
26	LogSoftMax	2		

[†] For 4D Tensor, the dimension is denoted as “Feature Map $\times$ Time $\times$ Width $\times$ Height”. For 3D Tensor, the dimension is denoted as “Time $\times$ Width $\times$ Height”.

convolution kernel connected to the d_2 -th feature map in the l -th layer, $b_{d_2,l}$ is the bias term, and $\sigma(\cdot)$ denotes a non-linear activation layer. Subsequently, a 3D pooling layer is applied to reduce the resolution of feature maps and enhance the invariance of the input signals to distortions. According to the research in [122], smaller receptive fields of 3D convolution kernels with deeper architectures can yield better performance for video classification. Although our problem is different from [122], we found out that adopting a smaller receptive field leads to better results for face spoofing detection as well. Therefore, in the 3D CNN architecture, we only consider the spatial-temporal receptive field as $3 \times 3 \times 3$. The proposed 3D CNN model is detailed in Table 4.1. This architecture has five convolutional layers followed by the fully connected layer. The

study regarding the appropriate number of convolutional layers is presented in Section 4.3.4.

Figure 4.2: Architecture visualization of 3D-CNN model.

4.2.2 Data Augmentation

As it can be observed from Table 4.1, our proposed 3D CNN model has more than 4M parameters to be optimized. However, existing samples in public databases are not enough to train such model. Therefore, the underfitting problem can not be avoided due to the large number of parameters in the model and the sparsity of training samples. To address this issue, we propose a data augmentation method based on video cubes to increase the number of training data. It should be noted that traditional augmentation methods such as injecting additional noise may not be feasible for the spoofing detection problem, given that the distortion information plays a key role in face spoofing detection. Therefore, the strategy of augmenting the video cubes is developed concerning this task.

Figure 4.3: Illustrations of data based on spatial augmentation.

4.2.2.1 Spatial Augmentation

To mitigate the variation of background for face spoofing detection, face detection is usually conducted as a pre-processing step [40]. However, variations of background near face regions can even be beneficial to face spoofing detection when considering deep learning approaches, as spoofing artifacts can be from the background region or the bezel of spoofing medium. Therefore, we propose to shift the bounding box in four different directions (up, down, right and left) by $\alpha \cdot l$, where l is equal to the width/height of bounding box. The parameter α is a predefined scaling factor, which is empirically set to 0.2 in our work. We stop the spatial augmentation if the bounding box moves out of the image boundary. We show an example of spatial augmentation in Fig. 4.3.

4.2.2.2 Gamma Correction based Augmentation

To take the display medium diversity due to different types of capturing devices into consideration, we conduct a gamma correction based augmentation on each individual frame of a given video cube. Considering the face captured by a certain camera model with gamma value γ_1 , the gamma correction process to γ_2 can be represented as

$$I_{aug} = \lfloor ((I/255)^{\gamma_2/\gamma_1}) * 255 \rfloor \quad (4.2)$$

where I and I_{aug} are the original pixel and augmented pixel, respectively, in RGB space. ‘ $\lfloor \cdot \rfloor$ ’ denotes the round and truncation operations, where the output value is truncated into the range [0,255]. Since the camera performs linear correction ($\gamma = 1.0$) and exponential gamma correction (e.g. $\gamma = 2.2$) before display¹, we choose the ratio γ_2/γ_1 to be 1.0/2.2 and 2.2/1.0 for augmentation in our work. We show an example of gamma correction based augmentation in Fig. 4.4.

¹<http://www.cambridgeincolour.com/tutorials/gamma-correction.htm>

Figure 4.4: Illustrations of data based on gamma correction based augmentation. (a) Original Face; (b) Face with the gamma correction ratio 1.0/2.2; (c) Face with the gamma correction ratio 2.2/1.0.

4.2.3 Model Generalization

Although deep learning is powerful in learning representative information when training data are diverse, it may still suffer from performance degradation when test data are “unseen”, such as the test samples obtained from a different environment from the training data. Generally speaking, it is impossible to involve face samples captured by all types of cameras from every potential scenario. In view of this, we leverage the advantage of domain generalization [123] to solve this problem. More specifically, given face samples from a few different capturing conditions, by partitioning the face samples into different domains based on the capturing conditions, we aim at learning a robust representation across different domains for face spoofing detection by introducing the generalization loss as the regularization term. As such, the generalization capability of the network can be better enhanced.

Assume that there are face samples from L domains for training, which are denoted by $X = [X_1, X_2, \dots, X_L]$, with X_i representing the samples from the domain i . The total number of samples in X is $N_1 + N_2 + \dots + N_L$, where $N_1, N_2, \dots, N_L$ are the number of samples from each domain. Moreover, the input features of the f -th fully-connected layer of network is assumed to be $Y_f = [Y_{f,1}^\top, Y_{f,2}^\top, \dots, Y_{f,L}^\top]^\top$, $Y_f \in \mathbb{R}^{(N_1+N_2+\dots+N_L) \times D}$ where $Y_{f,i} \in \mathbb{R}^{N_i \times D}$ refers to the features of f -th fully connected layer from domain i . We further denote $Y_{f,i,k} \in \mathbb{R}^D$ as the feature of k -th sample from $Y_{f,i}$. To align the feature distributions from different domains, we adopt

the Maximum Mean Discrepancy (MMD) [88], a popular metric to measure the similarity between two distributions, to minimize the feature distribution divergence across domains. As such, given two distributions, they are identical if the MMD distance between them equals to zero. To learn the generalized feature representation, we aim at optimizing the network, which embeds the input samples X to $\mathbf{Y}_f$, such that the MMD distances among different domains can be minimized [88].

The MMD distance among multiple domains is given by,

$$d(\mathbf{Y}_f) = \frac{1}{L(L-1)} \sum_{i \neq j} \left\| \frac{1}{N_i} \sum_{k_1=1}^{N_i} \mathbf{Y}_{f,i,k_1} - \frac{1}{N_j} \sum_{k_2=1}^{N_j} \mathbf{Y}_{f,j,k_2} \right\|^2 \quad (4.3)$$

which can be further rewritten as,

$$d(\mathbf{Y}_f) = \text{trace}(\mathbf{K}_f \mathbf{Q}), \quad (4.4)$$

where $\mathbf{K}_f$ is the Gram matrix defined based on $\mathbf{Y}_f$, $\mathbf{K}_f = \mathbf{Y}_f \mathbf{Y}_f^T$, and $\mathbf{Q}$ is the coefficient matrix defined based on the samples from domain pairs given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{1,1} & \mathbf{Q}_{1,2} & \cdots & \mathbf{Q}_{1,L} \\ \mathbf{Q}_{2,1} & \mathbf{Q}_{2,2} & \cdots & \mathbf{Q}_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Q}_{L,1} & \mathbf{Q}_{L,2} & \cdots & \mathbf{Q}_{L,L} \end{bmatrix} \quad (4.5)$$

In particular, the matrix block $\mathbf{Q}_{i,j}$ from domains i and j is given by

$$\mathbf{Q}_{i,j} = \begin{cases} \frac{1}{LN_i N_j} \mathbf{1}_{N_i \times N_j} & \text{if } i = j \\ -\frac{1}{L(L-1)N_i N_j} \mathbf{1}_{N_i \times N_j} & \text{otherwise,} \end{cases} \quad (4.6)$$

where $\mathbf{1}_{N_i \times N_j}$ denotes the all-ones matrix with dimension $N_i \times N_j$. The gradient of the

generalization loss with respect to the network parameter Θ can be computed as

$$\frac{\partial d(\mathbf{Y}_f)}{\partial \Theta} = \frac{\partial d(\mathbf{Y}_f)}{\partial \mathbf{Y}_f} \frac{\partial \mathbf{Y}_f}{\partial \Theta} = 2\mathbf{Q}\mathbf{Y}_f \frac{\partial \mathbf{Y}_f}{\partial \Theta}, \quad (4.7)$$

where $\frac{\partial \mathbf{Y}_f}{\partial \Theta}$ can be obtained via back-propagation method [124].

To learn the generalized feature representation with our proposed 3D CNN network, we train the network from scratch on the given face samples collected from multiple domains with cross-entropy loss ($\mathcal{L}$) [92]. Moreover, the MMD distance among the domains is required to be minimized simultaneously. As such, the network parameters can be learned as

$$\Theta^* = \arg \min_{\Theta} \mathcal{L} + \lambda \mathcal{R}, \quad (4.8)$$

where Θ is the network parameters and $\mathcal{R}$ is represented by

$$\mathcal{R} = \sum_{f=1}^F \text{trace}(\mathbf{Y}_f \mathbf{Y}_f^\top \mathbf{Q}). \quad (4.9)$$

Here F is the number of fully connected layer in the network, which is set to be 2 in our work since we have two fully connected layers in our proposed network.

4.3 Experimental Results

4.3.1 Databases

We adopt four face spoofing detection databases, ① Idiap REPLAY-ATTACK [40], ② CASIA Face AntiSpoofing [93], ③ MSU mobile face spoofing database [3], and ④ Rose-Youtu Face Liveness Detection database [125] for the face anti-spoofing task.

Table 4.2: Cross-Camera Experimental Protocol.

Protocol	Training Camera Models	Testing Camera Model
1	Long-time-used USB camera New USB camera Macbook Sony NEX-5	Macbook Air
2	Long-time-used USB camera New USB camera Macbook Sony NEX-5	Google Nexus 5
3	Long-time-used USB camera New USB camera Macbook Air Google Nexus 5 Sony NEX-5	Macbook
4	Macbook Macbook Air Google Nexus 5	Long-time-used USB camera
5	Macbook Macbook Air Google Nexus 5	New USB camera
6	Macbook Macbook Air Google Nexus 5	Sony NEX-5
7	Macbook Macbook Air New USB camera	Google Nexus 5
8	Long-time-used USB camera New USB camera Macbook	Sony NEX-5
9	Macbook New USB camera Sony NEX-5	Long-time-used USB camera
10	Long-time-used USB camera Sony NEX-5 Macbook Air	Google Nexus 5

4.3.2 Evaluation Protocol

For Idiap Replay-Attack database, it is divided into three sub-folds, including training fold, development fold and testing fold. We report the Equal Error Rate (EER) on the development fold and use the threshold determined by EER on the development fold to obtain the Half Total Error Rate (HTER) on the testing fold. For CASIA, MSU and Rose-Youtu databases, a classifier is trained with the training fold and then EER rate is evaluated on the testing fold following the protocols defined in [3, 93, 125].

To further evaluate the generalization capability of our proposed method, we conduct experiments to evaluate the performance in the scenario of cross-camera based face spoofing detection. In particular, we first train a 3D CNN model by employing genuine and spoofed face samples captured by multiple camera models. Then we evaluate the performance by testing with another camera model which was not involved in the training phase. To conduct such cross-camera based experiments, we merge the CASIA, Idiap REPLAY-ATTACK and MSU databases together and re-arrange the training, validation and testing sets based on camera models. The samples for training and validation are from the same set of camera models while the samples for testing are from another camera model. Here, we have six camera models in hand including “Long-time-used USB camera”, “New USB camera”, “Macbook”, “Macbook Air”, “Google Nexus 5”, and “Sony NEX-5”. By considering the generalization capability where camera models are different between training samples and testing samples², we randomly create ten different cross-camera scenarios to evaluate the performance of our 3D CNN framework. The details of experimental protocols are listed in Table 4.2. During training, we randomly divide the training data as training fold and development fold. The average Half Total Error Rate is reported by repeating the process for five times.

²The medium of spoofing attack (e.g., paper printed attack and video replay attack) can also be taken into consideration in cross-domain face spoofing detection. However, compared with the diversity of camera models, the conditions of spoofing medium are easier to be controlled. Therefore, in this work, we only consider the influence of camera models.

4.3.3 Experimental Setup

For the image-based face anti-spoofing detection method, Viola-Jones face detection algorithm [94] was employed for face localization in each video frame. As stated in [40], localizing the face region can effectively mitigate the noise information induced by background. However, for our temporal-based method, directly employing face detection based on individual frames and concatenating each frame into a temporal cube is not practical, as this may destroy the temporal consistency. To extract reliable face region and preserve useful temporal information, we propose a “Max-Min” strategy to choose the largest face detection bounding box among the frames in a face video as the final face region. To be more specific, considering the upper-left point $(x_{1,i}, y_{1,i})$ and the bottom-right point $(x_{2,i}, y_{2,i})$ of the bounding box in the i -th frame of a given video with T frames ($i = 1, 2, \dots, T$), the final bounding box is located by $(x_{1,min}, y_{1,min})$ and $(x_{2,max}, y_{2,max})$, where $x_{1,min} = \min\{x_{1,1}, \dots, x_{1,T}\}$, $y_{1,min} = \min\{y_{1,1}, \dots, y_{1,T}\}$, $x_{2,max} = \max\{x_{2,1}, \dots, x_{2,T}\}$, and $y_{2,max} = \max\{y_{2,1}, \dots, y_{2,T}\}$. Frames without detectable face region are not considered. Based on our experiment, we found that it is a simple but effective way to crop face regions and maintain temporal information simultaneously. After obtaining the bounding box for each video, the face region is resized to 128×128 pixels. Basically, the temporal size can be determined based on the memory of GPU card. We set the temporal size to be 8 frames in our experiments.

For the learning process, we first initialize the parameters of CNN according to [126] and train the network only with cross-entropy loss. Then, the last convolutional layer as well as the fully-connected layer are fine-tuned with both cross-entropy loss and generalization loss. The idea behind such training strategy is that shallow layers are more likely to be generalized [118]. By fine-tuning the deeper layers, more discriminative information is expected to be extracted, which can be better generalized by minimizing MMD distance. Here, the domain is determined by the number of camera models for training. (The generalization regularization is omitted if there is only one camera model

for training.) The weight λ of the regularization term is set in the way that at the end of training, the classification loss and regularization term loss are approximately the same. Such setting is reasonable since the feature representation which has both discrimination and generalization ability can be learned. More specifically, the weight is selected in a range $\{0.001, 0.01, 0.1, 1, 10\}$. For the learning parameter setting, we experimentally set the two different momentum values as $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the learning rate is set to 0.001 for training the network from scratch and 0.0001 for fine-tuning. Moreover, the weight decay is set to 0.00005 and learning rate decay is set to 10^{-7} . The network is trained with the adaptive moment estimation (Adam) method [127] in a mini-batch manner with the size 8 during network initialization step and 100 for each domain during the fine-tuning step. It is observed that the objective function can converge after 50 epochs for the initialization step and 10 epochs for the fine-tuning step. The GPU card employed for our task is Tesla K40 and the framework is implemented by Torch library³. After training the network, we employ the second fully connected layer (with dimension 1024 elements) as our latent discriminative feature. The Support Vector Machine [95] with linear kernel is used for classifier training. Finally, the output scores of 3D cubes belonging to the same video is combined with the average operation to generate the final detection result.

4.3.4 Experimental Results

4.3.4.1 Analysis of the Network Structure

We first analyze the proposed 3D CNN structure by comparing with other 3D CNN structures with different number of 3D convolutional layer to analyze the relationship between the depth of network and the final performance. We employ the Rose-Youtu Liveness Database in this analysis due to its diversity. In particular, a 3D convolutional layer contains a 3D convolution model, a 3D BatchNormalization model, a LeakyReLU

³<http://torch.ch/>

model and a 3D Maxpooling model. The dimensions of the second fully-connected layer are all set to 1024 for the model with different number of convolutional layers. The results are listed in Table 4.3 in terms of Equal Error Rate (EER).

Table 4.3: Performance Comparisons by different 3D CNN structures with Rose-Youtu Liveness Database. The results are measured by Equal Error Rate (EER).

Structure	EER(%)
#Conv Layer=3	10.8
#Conv Layer=4	8.4
#Conv Layer=5	7.0
#Conv Layer=6	7.6

Based on the results, we observe that we can achieve better performance with the increase of network depth. Such observation is also consistent with other computer vision tasks [126, 128]. However, we also notice that the performance drops when we further increase the depth of network to be more than five Conv layers, which may originate from the overfitting of network.

4.3.4.2 Intra-Database Evaluation

We then evaluate our algorithm by assuming the training and testing face samples are all from the same camera models and capturing conditions, which is referred to intra dataset validation in the literature. The performance of the proposed algorithm is compared with the state-of-the-art algorithms on different databases.

We adopt both frame- and sequence-based methods as our baselines, including texture based ([2, 39, 43, 47, 121, 129, 130]), image quality assessment based ([50],[3]) as well as data-driven based ([68, 78, 131]) methods. The comparison results are shown in Table 4.4. Generally speaking, our proposed algorithm outperforms other baseline methods in most of the cases, which demonstrates the effectiveness of our method by considering learning both spatial and temporal information in a data-driven fashion. For Idiap REPLAY-ATTACK and MSU databases, we can observe that both hand-crafted features

Table 4.4: Performance comparisons with the state-of-the-art methods.

Method		Idiap		CASIA	MSU	Rose-YouTu
		EER (%)	HTER (%)	EER (%)	EER (%)	EER (%)
Sequence-based	de Freitas Pereira <i>et.al.</i> (2014) [43]	7.9	7.6	10.0	14.2*	25.0*
	Bharadwaj <i>et.al.</i> (2013) [129]	0.2	0.0	14.4	—	—
	Wen <i>et.al.</i> (2015) [3]	7.4	7.6*	26.5*	5.8	39.5*
	Tirunagari <i>et.al.</i> (2015) [130]	5.3	3.8	21.8	—	—
	Pinto <i>et.al.</i> (2015) [121]	—	2.8	14.0	—	—
	Boulkenafet <i>et.al.</i> (2016) [2]	0.0	3.5	3.2	3.5	9.6*
	Zhao <i>et.al.</i> (2017) [47]	1.7	0.8	6.5	—	—
	Gan <i>et.al.</i> (2017) [131]	0.2	0.0	6.4*	4.8*	11.3*
	Proposed Method	0.3	1.2	1.4	0.0	7.0
Frame-based	Galbally <i>et.al.</i> (2014) [50]	—	15.2	32.4	—	—
	Matta <i>et.al.</i> (2011) [39]	13.9	13.8	18.2	10.9	27.1*
	Yang <i>et.al.</i> (2014) [68]	6.1	2.1	7.6*	5.8*	8.0*
	Boulkenafet <i>et.al.</i> (2016) [2]	0.4	2.8	2.1	4.9	12.5*
	Atoum <i>et.al.</i> (2017) [78]	0.8	0.7	2.7	—	—

‘—’ represents that the results were not available. ‘*’ represents that the results were achieved with the implementation by ourselves.

(spatial and temporal based) and deep learning based methods can achieve satisfactory performance. The reason lies in that the face capturing conditions are relatively simple in Idiap REPLAY-ATTACK and MSU databases. In particular, only one camera model is adopted in Idiap REPLAY-ATTACK and two for MSU database, and moreover, there is almost no motion information in videos, which makes the extracted features less influenced by camera and unexpected movements. For the CASIA database, it is observed that the sequence-based methods are generally worse than frame-based methods, which is reasonable since CASIA database contains diverse motion such as “paper wrapping” and “eye blinking”. Such information can deteriorate the performance when considering temporal information. On the other hand, the quality-based method [50] cannot achieve satisfactory performance compared with texture-based methods (e.g. [2, 39]). This is due to the fact that the diversity of the camera models (Long-time-used USB camera, New USB camera and Sony NEX-5) can significantly influence the quality distortion

information from spoofing artifacts. Again, by applying our proposed framework, we can achieve good performances when camera models and motion information are diverse and complicated. Compared with CASIA database, Rose-Youtu database is even more diverse not only in terms of camera models and motion, but also the illumination conditions. For Rose-Youtu database, our method can also outperform the state-of-the-art methods [2, 68]. Compared with other databases, for which our proposed method can achieve very low error rate, the error rate based on Rose-Youtu with the 3D CNN framework is still relatively higher.

Figure 4.5: Genuine and spoofed face samples for visualization.

Besides objectively comparing the detection accuracy, we are also interested in the information extracted from the proposed 3D CNN network. To have a better understanding of our trained neural network, we feed both genuine and spoofed face sample as shown in Fig. 4.5 to the proposed network, the outputs of Conv1 layer in both spatial domain and temporal domain, which can be regarded as a set of 3D cubes (feature maps), are visualized. For the spatial domain, we average the output in terms of temporal domain and obtain 128 spatial feature maps with size 128×128 . Then, all elements in spatial feature maps are normalized to $[0,1]$. For the temporal domain, we compute the Discrete Fourier Transform based on the temporal domain by considering the direct current (DC) component of spatial information, which leads to 128 frequency spectra. Spatial and temporal information obtained from the Conv1 layer regarding genuine and spoofed face samples are visualized in Fig. 4.6 and 4.7, respectively.

(a) Spatial visualization of a genuine sample.

(b) Spatial visualization of a spoofed sample.

Figure 4.6: Spatial visualization of genuine and spoofed face samples based on 128 feature maps.

(a) Temporal visualization of genuine sample.

(b) Temporal visualization of spoofed sample.

Figure 4.7: Temporal visualization of genuine and spoofed face samples based on frequency spectrum. The horizontal axis ranges in $[-\pi, \pi]$

Regarding the visualization of spatial domain, it can be observed that the obtained spatial feature maps are similar to the output by employing high-pass filter, which demonstrates that the high frequency component can be used to distinguish between the genuine and spoofed face samples. Spatial feature maps of the genuine sample tend to be darker in some cases compared with the spoofing one. It indicates that the trained network preserves the lighting and color information which play important roles in face spoofing detection (e.g. unexpected reflection and color gamut distortion present in the spoofing medium).

Regarding the visualization of temporal domain, it can be observed that frequency spectra belonging to the spoofed sample tend to contain high-frequency components with a larger magnitude in some cases (marked with the red box). Such results indicate that spoofed samples are more likely to suffer from temporal aliasing caused by re-sampling in the recapturing process, which is the key feature for face spoofing detection in the temporal domain.

4.3.4.3 Cross-Camera-Model Evaluation

Furthermore, we analyze the performance when camera models and illumination conditions are different for training and testing, and the protocols are illustrated in Table 4.2. The methods adopting $LBP - TOP_{8,8,8,1,1,1}$ feature [43] and color texture [2] are employed as our texture-based baselines, image distortion statistics [3] is employed as the quality-based baseline and the proposed 3D CNN framework without domain generalization regularization is employed as another deep learning based baseline. Considering that we can apply domain generalization based on the hand-crafted feature as well, the baseline hand-crafted features are also enhanced with domain generalization as proposed in [132]. The parameters for domain generalization [132] are determined by cross-validation, as explained in [89, 132]. The average HTER results are compared in Table 4.5.

Table 4.5: HTER performance (%) for cross-camera experiments (the protocols are defined in Table 4.2).

Protocol	1	2	3	4	5	6	7	8	9	10	Average
de Freitas Pereira <i>et.al.</i> [43]	43.3	51.4	44.3	40.3	35.7	61.8	40.5	54.0	23.4	42.4	43.7
de Freitas Pereira <i>et.al.</i> [43] with DG [132]	42.7	43.6	40.5	36.4	46.8	51.3	46.7	46.1	20.1	37.2	41.1
Wen <i>et.al.</i> [3]	43.2	45.3	55.2	54.5	55.8	57.2	42.0	51.3	33.8	53.3	49.2
Wen <i>et.al.</i> [3] with DG [132]	39.5	49.6	50.0	48.4	47.0	44.2	44.2	35.9	24.9	37.2	42.1
Boulkenafet <i>et.al.</i> [2]	25.5	42.5	27.6	47.8	49.4	34.6	47.2	49.1	41.6	35.1	40.0
Boulkenafet <i>et.al.</i> [2] with DG [132]	22.8	42.3	35.4	39.7	45.5	44.7	39.1	27.8	21.4	33.0	35.2
3D CNN	28.1	46.7	37.0	53.6	41.3	49.0	44.9	34.4	15.9	31.4	38.2
Proposed Method	19.0	32.7	26.0	38.1	34.3	36.9	38.1	21.9	11.5	28.6	28.7

“DG” refers to domain generalization.

Firstly, we can observe that the LBP-TOP and the distortion-based features cannot achieve desired performance under cross-camera settings for most of the cases. Such observation indicates that the trained classifiers with these two features are overfitted to the training data. The performance can be improved to some extent by applying domain generalization, which is reasonable since domain generalization can encode discriminative information that is shared among different domains. Moreover, it is observed that the performance varies from 25% to 49% by employing color texture feature [2]. When the training camera models are diverse, color texture can achieve the relatively good performance. By adopting domain generalization based on color texture, we can observe that the performance can also be improved in some scenarios. However, since the parameters for training samples adopted by domain generalization may not be able to encode the variation scale for test data (e.g., in Protocol 3, the training samples are much more diverse compared with those in Protocol 6), there is a performance drop. Therefore, it is reasonable that simply applying domain generalization on hand-crafted features may not work. We also observe that simply employing 3D CNN framework without generalization will not achieve desired performance expect for some special cases (e.g. protocol 9) where similar temporal information can be shared between training and testing samples. This can be explained by the following two reasons.

- Although the data augmentation process is applied to enlarge the size of training samples, such augmentation process is still limited due to the capturing devices and illumination condition. When the testing data are captured from a totally different environment, we still have the overfitting problem by simply applying deep learning for face spoofing detection.
- Different domain may introduce different artifacts. Since deep learning is conducted in a data-driven manner, it is likely that the neural network model may learn spoofing information which only dominate in a specific domain of training data. Therefore, when applying the neural network to extract features on testing data acquired under different conditions, we may not be able to extract the discriminative information.

Finally, the 3D CNN model with generalization regularization term learned in an end-to-end manner can significantly improve the performance for all protocols, as the generalization term can force the network to learn generalized spoofing features, which are less influenced by camera models and illumination conditions. On the other hand, it is worth mentioning that the error rate is still relatively high when the training samples are not diverse enough (protocols 4-7 in which the motions are consistent for training samples while different types of motion such as paper wrapping exist in testing samples).

4.3.4.4 Empirical Analysis of the 3D CNN Convergence

An empirical analysis of the convergence of our proposed 3D CNN network is conducted regarding the classification loss as well as the generalization loss. As indicated in Table 4.1, we adopt several LeakyReLU and pooling modules, which make the network highly nonlinear. However, it is shown that we can still reach a local minima which leads to good performances. For a better understanding of the convergence, we visualize both the average classification loss and the generalization loss (MMD distance)

for each mini-batch by Rose-Youtu database in Fig. 4.8. As we can see, in the beginning, the classification loss is relatively smaller compared with the generalization loss. The reason is due to the initialization of 3D CNN network by only classification loss. However, after training for a few epochs, the classification loss becomes larger since the network is more fitted to the generalization term. Finally, after training the network for more than 5 epochs, the classification loss and the generalization loss are close to each other.

Figure 4.8: Convergence visualization for Rose-Youtu liveness database.

4.3.5 Computational Time Analysis

Table 4.6 summarizes the computational time of different methods based on a 360-frame video with the resolution 320×480 . For baseline methods, the algorithm is implemented by using Matlab on an Intel Core i5 CPU@3.2GHz machine. For our own method, we compute the time on GPU Tesla K40. As we can observe from Table 4.6, our method can achieve a competitive time consumption compared with other methods. Considering that we analyze the time consumption based on a 360-frame video, our proposed algorithm can process in a “real-time” fashion. It should be noted that leveraging GPU is not a disadvantage since such computational resources have become easier to obtain

with the development of hardware technology nowadays. Although the method in [43] can be computed faster, the error rate of [43] is much higher than our results. Our framework will be further optimized (e.g., network compression) to meet the computational requirement of face spoofing applications on mobile devices.

Table 4.6: Computational time (seconds) analysis based on a 360-frames video with different methods

Method	Time per video
de Freitas Pereira <i>et.al.</i> [43]	7.35
Wen <i>et.al.</i> [3]	43.2
Boulkenafet <i>et.al.</i> [2]	27.8
Proposed Method	10.4

4.3.6 Discussions

Learning a robust and generalized feature representation for face spoofing detection is a challenging task. Since the capturing of face samples is totally independent, it is difficult to collect a large database containing all possible camera models, illumination conditions and facial appearances. Our proposed framework takes advantage of both deep learning and domain generalization technique to significantly improve the performance of intra and cross-condition setups, including the case that both training and testing samples are taken from similar condition (intra), and the case that training and testing samples are not well aligned (cross). As the first attempt to tackle the face spoofing detection task with 3D CNN, there are still some limitations. Though our proposed 3D CNN can achieve good performance when training and testing samples are from similar conditions, how to design a more robust network which can be generalized better should be further investigated, as error rates for cross-condition evaluation are still higher than intra setup. Moreover, how to apply a more reasonable distance measure in order to generalize the network better will also be studied in the future.

4.4 Summary

In this chapter, we propose a 3D CNN framework to tackle the face spoofing detection problem. Compared with other deep learning based biometric spoofing detection method, the novelty of our work lies in twofold. First, we apply a 3D CNN network which takes both spatial and temporal information into consideration with a specifically designed data augmentation method for face spoofing detection, which shows better classification capability compared with simply training a 3D CNN model from scratch. Secondly, to further improve the generalization performance, we employ a generalization regularization by minimizing the Maximum Mean Discrepancy distance among different domains. Our framework can be efficiently trained in an end-to-end manner and the experimental results show better generalization ability compared with state-of-the-art methods.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

Biometrics offers a powerful and practical solution to authentication-required applications. Due to the breakthrough of biometrics authentication via deep learning and its better security capability compared with traditional authentication methods (e.g., password, secret question, token code), more and more attention has been attracted from both academia and industry nowadays. Typical biometric modalities include fingerprints, iris, face, voice prints and even typing patterns, among which “face” is the most popular one as it does not require any additional hardware infrastructure and almost all mobile phones are equipped with a front-facing camera. Despite the success of face recognition, it is still vulnerable to the presentation attacks due to the popularity of social media from which facial images are easy to acquire. For instance, a presentation attack can record the face information of a person by printing (printing attack), replaying on screen (replay attack) or even counterfeiting the face via 3D masking, which brings extremely challenging security issues. While numerous face spoofing techniques have been proposed, there are lack of generalization capability which prevent these methods from practical application.

In this thesis, we study to improve the generalization capability of face spoofing detection through different types of applications, which leads to different formulations of cross-database face spoofing detection problems. Specifically, the cross-database face spoofing detection problems we address are:

Network distilling based face spoofing detection

In Chapter 2, we propose a novel framework based on face spoofing detection by assuming that we can collect limited genuine and spoofed face samples in an application-specific scenario. The problem we aim to solve is that, although limited data can be obtained, such amount of data can not be generalized well to the application-specific scenario due to the diversity of facial appearance. Our idea is to leverage the large amount of public-available face spoofing database (which we refer to rich data domain) to help with training. In particular, we propose a knowledge distilling framework based on Convolutional Neural Network (CNN) (AlexNet [86]) by carefully designing two regularization terms as the distilling information based on different level. One is the distribution distilling which we propose to minimize the Maximum Mean Discrepancy (MMD) to reduce the distribution divergence between the rich data domain and application specific domain such that the data from two domains are expected to lie on similar manifold. We further propose to maximize the cosine similarity based on face sample pair with the same label information (otherwise we minimize the cosine similarity) such that the conditional distribution divergence can be reduced. To evaluate the effectiveness of our proposed knowledge distilling framework, we compare with both hand-crafted based methods [2][47][39][43] as well as CNN based method [68] by only considering limited labeled application-specific domain data and the labeled data from two domains. Based on the experimental results, our proposed framework can achieve significant improvement compared with other baseline methods. This is because the baseline algorithm can not smartly leverage the rich data domain which can cause overfitting problem during classifier training process. On the other hand, the baseline methods can also suf-

fer from underfitting problem due to limited number of data in the application-specific domain. Again, our proposed knowledge distilling based framework can successfully overcome these problems.

Unsupervised domain adaptation based face spoofing detection

When applying face spoofing detection in the mobile phone based application, it is not feasible to have the assumption that we can collect face data from an application-specific domain since the face capturing condition based on mobile phone can be extreme diverse due to a lot of factors (e.g. mobile phone setting, illumination, facial pose, etc.). However, it is reasonable to assume that we can collect sufficient unlabeled face data for verification purpose during training process. Motivated by the prior analysis of subspace representation of face and illumination, we propose an unsupervised domain adaptation framework based on subspace alignment for face spoofing detection in Chapter 3 by leveraging the face samples for verification to help train a more robust classifier. We conduct extensively experiments to demonstrate the effectiveness of our proposed algorithm. First, we analysis both state-of-the-art handcrafted feature as well as deep learning feature, we observe that by leveraging unlabeled face samples for testing, we can achieve 20% improvement in average in terms of half total error rate. Moreover, considering that it is more practical that the testing face samples are all genuine, we further conduct experiments by only leveraging genuine face samples for unsupervised domain adaptation, the experimental results also show great improvement with 5% on one of the state-of-the-art handcrafted feature (CoALBP) and with 10% on deep learning based feature. To further improvement the capability of domain adaptation, we propose an outlier removal strategy as a data pre-processing step. Based on the experimental results, we can achieve another 2-3% improvement based on the results with domain adaptation. Lastly, to show the universality of our proposed algorithm, we compare our algorithm with another person-specific domain adaptation based face spoofing detection algorithm [5], the results show that our algorithm can perform significantly better

than [5] which might be due to the strong assumption from [5] that we must know the person-specific and camera information for the testing face samples.

Domain generalization based face spoofing detection

In Chapter 4, we consider a more generalized application, can we directly train a generalized classifier by the labeled face data in hand? To tackle this problem, we consider a deep learning based framework by taking both spatial information as well as temporal information into consideration since face spoofing distortion can appear in face content itself and also in face video sequence. In view of this, we impose a 3D Convolutional Neural Network (3D CNN) with a novel regularization term which can extract both spatial and temporal discriminative information in a data-driven manner. Based on the results of intra-domain scenario (where training face samples and testing face samples are captured under similar condition) by publicly available database, our proposed 3D CNN can outperform other sequential based methods [43][2][47] with a clear margin. However, due to the limited number of face samples, we still face overfitting problem by directly training a 3D CNN network with classification loss. To reduce the overfitting problem, we further a novel regularization term based on domain generalization to extract deep features which can be sharable to different database. Therefore, we expect to extract spoofing-specific information by training the 3D CNN network with classification loss and domain generalization loss. Our proposed framework can achieve much better based on ten different cross-camera experimental setting compared with other baseline methods as well as the 3D CNN without domain generalization loss.

5.2 Future Works

Face spoofing detection is a challenging task especially for the cross-database scenario, where training samples and testing samples are captured from different environment. Therefore, a robust feature representation which can capture intrinsic property of spoof-

ing information and a sophisticated classifier are important to tackle this problem. For the past two years, I am focusing on the problem how to adapt the training data to the testing data therefore a more robust classifier can be obtained, how to leverage the source domain data information to help train a robust classifier when only few data are available in the target domain, how to train a generalized network by considering both spatial and temporal information. Although some improvements can be achieved, it is still far away to be applied to industry for the uncontrolled face spoofing detection requirement. I briefly summarize some future directions based on different perspectives.

5.2.1 Face Spoofing Detection based on Learning

- Hand-crafted features are powerful for face spoofing detection. However, the hand-crafted features are very easily to overfit to lighting condition, camera quality, even human facial appearance. Therefore, how to extract the feature which can be generalized well needs to be discussed. One possible artifact that is worth looking at is the aliasing pattern appeared on printed paper and digital screen since such artifact is very difficult to be avoided during spoofing process.
- Domain adaptation algorithms are proved to be useful for cross-domain face spoofing detection. However, the ability for domain adaptation depends on the number of samples we consider. In practice (especially for real time based face spoofing detection), we only have little (mostly only one single face image) in hand, domain adaptation can not be applied in this case. Therefore, zero-shot learning strategy [133–136] needs to be concerned to tackle this problem.
- Deep learning achieves extremely good performance on many computer vision tasks [86][126]. The performance of face spoofing detection by using deep learning has also been analyzed in our works. Based on the results, we notice that deep learning can outperform the traditional hand-crafted feature when the training da-

ta are diverse. However, since it is impossible to consider all the environments (camera, illumination, etc.), deep learning based method also suffers from performance drop in cross-domain scenario. In our former work, we focused on domain adaptation based on the fully connected layer. According to the very recent work on cross-domain classification, we can use the large amount of source domain data and few target domain data to generate more target domain data by Generative Adversarial Network (GAN) [137, 138]. By training a classifier based on the generated data (since it is difficult to acquire large amount of samples for both genuine and spoof face samples), a good performance can be achieved even compared with the domain adaptation in some scenarios. Therefore, it is insightful to see whether we can generate genuine and spoof face samples by using GAN network which can be further applied to train a more robust classifier.

5.2.2 Face Spoofing Detection based on Sensor Information

- The common medium for face spoofing attack contains “paper”, “video screen” and “mask”. Compared with “mask”, the “paper” and “video screen” attack are much cheaper which achieves satisfied trade-off between the attacking cost and attacking attempt rate. While the existed methods only consider using one camera to capture face image/video, it is possible to use more than one camera to capture face image/video which can help obtain additional depth information which benefit to avoid “paper” and “video screen” attack. In view of this, it is interesting to build a face spoofing database with the face samples obtained by more than one camera model and more sophisticated computer vision algorithm can be developed to analyze depth information as auxiliary information to counter face spoofing attack.
- As mentioned, one of the most challenging part for face spoofing detection is the “uncontrolled” illumination condition. To reduce the influence caused by illumi-

nation, we can leverage additional sensor (e.g. Infra-red (IR) sensor) which can filter out the visible light by keeping the thermal information which can benefit both face recognition and face spoofing detection. Although the IR sensor based face spoofing detection has been discussed in the previous work, the large-scale IR face database based on face spoofing detection is not publicly available and the IR sensor is also not diverse. The face spoofing detection based on IR sensor (for both intra- and inter-domain) is worth being investigated.

Author's Publications

1. **Haoliang Li**, Wen Li, Hong Cao, Shiqi Wang, Feiyue Huang and Alex C. Kot, "Un-supervised Domain Adaptation for Face Anti-Spoofing," IEEE Transactions on Information Forensics and Security (in press).
2. **Haoliang Li**, Peisong He, Shiqi Wang, Anderson Rocha, Xinghao Jiang and Alex C. Kot, "Learning Generalized Feature Representation for Face Anti-spoofing," IEEE Transactions on Information Forensics and Security (in press).
3. **Haoliang Li**, Shiqi Wang, Peisong He, Anderson Rocha and Alex C. Kot, "Deep Neural Network Distilling for Face Spoofing Detection," submitted to IEEE Transactions on Circuits and Systems for Video Technology.
4. **Haoliang Li**, Sinno J Pan, Shiqi Wang and Alex C. Kot, "Heterogeneous Domain Adaptation via Matrix Factorization in RKHS," submitted to IEEE Transactions on Neural Network and Learning Systems.
5. **Haoliang Li**, Wen Li, Sinno J Pan and Alex C. Kot, "Discovering and Incorporating Latent Target-Domains for Domain Adaptation," to be submitted to IEEE Transactions on Image Processing.
6. **Haoliang Li**, Sinno J Pan, Shiqi Wang and Alex C. Kot, "Domain Generalization with Adversarial Feature Learning," To appear in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

7. **Haoliang Li**, Shiqi Wang and Alex C. Kot, "Image Recapture Detection with Convolutional and Recurrent Neural Network", in Proceedings of Electronic Imaging, IS&T, 2017
8. **Haoliang Li**, Alex C. Kot and Leida Li, "Color Space Identification from Single Images", in Proceedings of IEEE International Symposium on Circuits and System (ISCAS), 2016
9. **Haoliang Li**, Shiqi Wang and Alex C. Kot, "Face Spoofing Detection with Image Quality Regression", in Proceedings of IEEE International Conference on Image Processing Theory, Tools and Application, 2016
10. Leida Li, Dong Wu, Jinjian Wu, **Haoliang Li**, Weisi Lin, Alex C. Kot, "Image Sharpness Assessment by Sparse Representation". IEEE Transactions on Multimedia, vol.18, no.6, pp.1085 - 1097, June 2016
11. Leida Li, Yu Zhou, Jinjian Wu, Weisi Lin and **Haoliang Li**, "GridSAR: Grid Strength and Regularity for Robust Evaluation of Blocking Artifacts in JPEG Images," Elsevier Journal of Visual Communication and Image Representation, vol.30, pp.153-163, July 2015.
12. Khosro Bahrami, Alex ChiChung Kot, Leida Li and **Haoliang Li**, "Blurred Image Splicing Localization by Exposing Blur Type Inconsistency", IEEE Transactions on Information Forensics and Security, vol.10, no.5, pp.999–1009, April 2015

Bibliography

- [1] K. Patel, H. Han, and A. Jain, “Secure Face Unlock: Spoof Detection on Smartphones,” *IEEE Trans. Information Forensic and Security*, vol. 11, no. 10, pp. 2268 – 2283, Oct 2016.
- [2] 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.
- [3] D. Wen, H. Han, and A. K. Jain, “Face spoof detection with image distortion analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 746–761, Apr 2015.
- [4] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 416–426.
- [5] J. Yang *et al.*, “Person-specific face antispoofing with subject domain adaptation,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 797–809, Apr 2015.
- [6] L. Alipay, “com co. 2017. alipay-makes life easy.”
- [7] M. E. Burge, “Apple pay, bitcoin, and consumers: The abcs of future public payments law,” *Hastings LJ*, vol. 67, p. 1493, 2015.

- [8] A. Pay, "Apple pay: Your wallet without the wallet.(2015)," *Retrieved August*, vol. 22, p. 2015, 2015.
- [9] B. Fung, "Cashless society: Is it the future?" in *Speech at Suomen Pankkis Going Cashless conference, Helsinki*, vol. 14, 2015.
- [10] W. W. Bledsoe, "The model method in facial recognition," *Panoramic Research Inc., Palo Alto, CA, Rep. PR1*, vol. 15, p. 47, 1966.
- [11] M. D. Kelly, "Visual identification of people by computer." STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE, Tech. Rep., 1970.
- [12] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Springer Science & Business Media, 2006, vol. 479.
- [13] C. Authur, "iphone 5s fingerprint sensor hacked by germany's chaos computer club," *Guardian News. Np*, vol. 23, 2013.
- [14] A. Anjos and S. Marcel, "Counter-measures to photo attacks in face recognition: a public database and a baseline," in *Biometrics (IJCB), 2011 international joint conference on*. IEEE, 2011, pp. 1–7.
- [15] K. Patel *et al.*, "Live face video vs. spoof face video: Use of moiré patterns to detect replay video attacks," in *Biometrics (ICB), 2015 International Conference on*. IEEE, 2015, pp. 98–105.
- [16] N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 1–6.
- [17] Y. Xu *et al.*, "Virtual u: Defeating face liveness detection by building virtual models from your public photos." in *USENIX security symposium*, 2016, pp. 497–512.

- [18] X. Chen *et al.*, “Targeted backdoor attacks on deep learning systems using data poisoning,” *arXiv preprint arXiv:1712.05526*, 2017.
- [19] M. Sharif *et al.*, “Adversarial generative nets: Neural network attacks on state-of-the-art face recognition,” *arXiv preprint arXiv:1801.00349*, 2017.
- [20] Y. Li *et al.*, “Understanding osn-based facial disclosure against face authentication systems,” in *Proceedings of the 9th ACM symposium on Information, computer and communications security*. ACM, 2014, pp. 413–424.
- [21] M. Sharif *et al.*, “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1528–1540.
- [22] P. Johnson *et al.*, “Biometric liveness detection: Framework and metrics,” in *International Biometric Performance Conference (IBPC)*, vol. 1, 2012.
- [23] S. Prabhakar, S. Pankanti, and A. K. Jain, “Biometric recognition: Security and privacy concerns,” *IEEE security & privacy*, vol. 99, no. 2, pp. 33–42, 2003.
- [24] R. M. Bolle, J. H. Connell, and N. K. Ratha, “Biometric perils and patches,” *Pattern Recognition*, vol. 35, no. 12, pp. 2727–2738, 2002.
- [25] L. Thalheim, J. Krissler, and P.-M. Ziegler, “Body check,” *translated by Robert W. Smith*, <http://www.heise.de/ct/english/02/11/114>, 2002.
- [26] J. Galbally *et al.*, “On the vulnerability of face verification systems to hill-climbing attacks,” *Pattern Recognition*, vol. 43, no. 3, pp. 1027–1038, 2010.
- [27] P. Mohanty, S. Sarkar, and R. Kasturi, “From scores to face templates: a model-based approach,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2065–2078, 2007.

- [28] K. Kollreider, H. Fronthaler, and J. Bigun, "Non-intrusive liveness detection by face images," *Image and Vision Computing*, vol. 27, no. 3, pp. 233–244, 2009.
- [29] W. Bao *et al.*, "A liveness detection method for face recognition based on optical flow field," in *IEEE International Conference on Image Analysis and Signal Processing*, 2009, pp. 233–236.
- [30] K. Kollreider, H. Fronthaler, and J. Bigun, "Evaluating liveness by face images and the structure tensor," in *IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 75–80.
- [31] G. Pan *et al.*, "Eyeblink-based anti-spoofing in face recognition from a generic webcam," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.
- [32] G. Pan, L. Sun, and Z. Wu, *Liveness detection for face recognition*. INTECH Open Access Publisher, 2008.
- [33] G. Pan *et al.*, "Monocular camera-based face liveness detection by combining eyeblink and scene context," *Telecommunication Systems*, vol. 47, no. 3-4, pp. 215–225, 2011.
- [34] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET Biometrics*, vol. 3, no. 3, pp. 147–158, Sep 2013.
- [35] J. Komulainen, A. Hadid, and M. Pietikainen, "Context based face anti-spoofing," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 1–8.
- [36] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

- [37] J. Li *et al.*, “Live face detection based on the analysis of fourier spectra,” in *Defense and Security*. International Society for Optics and Photonics, 2004, pp. 296–303.
- [38] X. Tan *et al.*, “Face liveness detection from a single image with sparse low rank bilinear discriminative model,” in *European Conference on Computer Vision*. Springer, 2010, pp. 504–517.
- [39] J. Määttä, A. Hadid, and M. Pietikainen, “Face spoofing detection from single images using micro-texture analysis,” in *IEEE International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–7.
- [40] I. Chingovska, A. Anjos, and S. Marcel, “On the effectiveness of local binary patterns in face anti-spoofing,” in *IEEE International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2012, pp. 1–7.
- [41] J. Yang *et al.*, “Face liveness detection with component dependent descriptor,” in *IEEE International Conference on Biometrics (ICB)*, 2013, pp. 1–6.
- [42] D. Gragnaniello *et al.*, “An investigation of local descriptors for biometric spoofing detection,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 849–863, Apr 2015.
- [43] T. de Freitas Pereira *et al.*, “Face liveness detection using dynamic texture,” *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, pp. 1–15, 2014.
- [44] 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, no. 11, pp. 2396–2407, Nov 2015.

- [45] Z. Boulkenafet, J. Komulainen, and A. Hadid, “Face anti-spoofing based on color texture analysis,” in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 2636–2640.
- [46] Z. Boulkenafet *et al.*, “Scale space texture analysis for face anti-spoofing,” in *IEEE International Conference on Biometrics (ICB)*, 2016, pp. 1–6.
- [47] X. Zhao, Y. Lin, and J. Heikkilä, “Dynamic texture recognition using volume local binary count patterns with an application to 2d face spoofing detection,” *IEEE Transactions on Multimedia*, 2017.
- [48] P. P. Chan *et al.*, “Face liveness detection using a flash against 2d spoofing attack,” *IEEE Transactions on Information Forensics and Security*, 2017.
- [49] J. Bai *et al.*, “Is physics-based liveness detection truly possible with a single image?” in *Circuits and systems (ISCAS), Proceedings of 2010 IEEE international symposium on*. IEEE, 2010, pp. 3425–3428.
- [50] J. Galbally, S. Marcel, and J. Fierrez, “Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition,” *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 710–724, Feb 2014.
- [51] H. Li, S. Wang, and A. C. Kot, “Face spoofing detection with image quality regression,” in *IEEE International Conference on Image Processing Theory, Tools and Application*, 2016, pp. 1–6.
- [52] R. T. Tan and K. Ikeuchi, “Separating reflection components of textured surfaces using a single image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 178–193, Feb 2005.

- [53] F. Crete *et al.*, “The blur effect: perception and estimation with a new no-reference perceptual blur metric,” in *Electronic Imaging*. International Society for Optics and Photonics, 2007, pp. 64 920I–64 920I.
- [54] P. Marziliano *et al.*, “A no-reference perceptual blur metric,” in *IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2002, pp. III–57.
- [55] Y. Chen *et al.*, “Automatic classification of photographs and graphics,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2006, pp. 973–976.
- [56] I. Pavlidis and P. Symosek, “The imaging issue in an automatic face/disguise detection system,” in *IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications*. IEEE, 2000, pp. 15–24.
- [57] X. Sun, L. Huang, and C. Liu, “Context based face spoofing detection using active near-infrared images,” in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 4262–4267.
- [58] Y. Kim *et al.*, “Masked fake face detection using radiance measurements,” *JOSA A*, vol. 26, no. 4, pp. 760–766, 2009.
- [59] Z. Zhang *et al.*, “Face liveness detection by learning multispectral reflectance distributions,” in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, 2011, pp. 436–441.
- [60] A. Seal *et al.*, “Automated thermal face recognition based on minutiae extraction,” *International Journal of Computational Intelligence Studies*, vol. 2, no. 2, pp. 133–156, 2013.
- [61] K. W. Bowyer, K. Chang, and P. Flynn, “A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition,” *Computer vision and image understanding*, vol. 101, no. 1, pp. 1–15, 2006.

- [62] X. Xie *et al.*, “One-snapshot face anti-spoofing using a light field camera,” in *Chinese Conference on Biometric Recognition*. Springer, 2017, pp. 108–117.
- [63] N. Kose and J.-L. Dugelay, “Reflectance analysis based countermeasure technique to detect face mask attacks,” in *Digital Signal Processing (DSP), 2013 18th International Conference on*. IEEE, 2013, pp. 1–6.
- [64] P. Buddharaju *et al.*, “Physiology-based face recognition in the thermal infrared spectrum,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 613–626, 2007.
- [65] G. Hermosilla *et al.*, “A comparative study of thermal face recognition methods in unconstrained environments,” *Pattern Recognition*, vol. 45, no. 7, pp. 2445–2459, 2012.
- [66] L. Sun, W. Huang, and M. Wu, “Tir/vis correlation for liveness detection in face recognition,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2011, pp. 114–121.
- [67] T. I. Dhamecha *et al.*, “Disguise detection and face recognition in visible and thermal spectrums,” in *Biometrics (ICB), 2013 International Conference on*. IEEE, 2013, pp. 1–8.
- [68] J. Yang, Z. Lei, and S. Z. Li, “Learn convolutional neural network for face anti-spoofing,” *arXiv preprint arXiv:1408.5601*, 2014.
- [69] N. N. Lakshminarayana *et al.*, “A discriminative spatio-temporal mapping of face for liveness detection,” in *Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.

- [70] X. Tu and Y. Fang, “Ultra-deep neural network for face anti-spoofing,” in *International Conference on Neural Information Processing*. Springer, 2017, pp. 686–695.
- [71] O. Lucena *et al.*, “Transfer learning using convolutional neural networks for face anti-spoofing,” in *International Conference Image Analysis and Recognition*. Springer, 2017, pp. 27–34.
- [72] L. Li *et al.*, “An original face anti-spoofing approach using partial convolutional neural network,” in *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*. IEEE, 2016, pp. 1–6.
- [73] D. Menotti *et al.*, “Deep representations for iris, face, and fingerprint spoofing detection,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, Apr 2015.
- [74] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, “Fingerprint liveness detection using convolutional neural networks,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1206–1213, Jun 2016.
- [75] J. Deng *et al.*, “Imagenet: A large-scale hierarchical image database,” in *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
- [76] Z. Xu, S. Li, and W. Deng, “Learning temporal features using lstm-cnn architecture for face anti-spoofing,” in *IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 141–145.
- [77] K. Patel, H. Han, and A. K. Jain, “Cross-database face antispoofing with robust feature representation,” in *Chinese Conference on Biometric Recognition*. Springer, 2016, pp. 611–619.

- [78] Y. Atoum *et al.*, “Face anti-spoofing using patch and depth-based cnns,” in *In Proceeding of International Joint Conference on Biometrics*, Denver, CO, October 2017.
- [79] L. Wu *et al.*, “Motion analysis based cross-database voting for face spoofing detection,” in *Chinese Conference on Biometric Recognition*. Springer, 2017, pp. 528–536.
- [80] W. R. Schwartz, A. Rocha, and H. P. Edrini, “Face spoofing detection through partial least squares and low-level descriptors,” in *IEEE International Joint Conference on Biometrics (IJCB)*, 2011, pp. 1–8.
- [81] R. Tronci *et al.*, “Fusion of multiple clues for photo-attack detection in face recognition systems,” in *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 2011, pp. 1–6.
- [82] J. Yan *et al.*, “Face liveness detection by exploring multiple scenic clues,” in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*. IEEE, 2012, pp. 188–193.
- [83] M. M. Chakka *et al.*, “Competition on counter measures to 2-d facial spoofing attacks,” in *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 2011, pp. 1–6.
- [84] J. Komulainen *et al.*, “Complementary countermeasures for detecting scenic face spoofing attacks,” in *Biometrics (ICB), 2013 International Conference on*. IEEE, 2013, pp. 1–7.
- [85] A. Rattani, W. J. Scheirer, and A. Ross, “Open set fingerprint spoof detection across novel fabrication materials,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2447–2460, 2015.

- [86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [87] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [88] A. Gretton *et al.*, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.
- [89] S. J. Pan *et al.*, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, Feb 2011.
- [90] B. Gong *et al.*, "Geodesic flow kernel for unsupervised domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2066–2073.
- [91] M. Long *et al.*, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*. JMLR. org, 2015, pp. 97–105.
- [92] P.-T. De Boer *et al.*, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.
- [93] Z. Zhang *et al.*, "A face antispoofing database with diverse attacks," in *IEEE International Conference on Biometrics (ICB)*, 2012, pp. 26–31.
- [94] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–511.

- [95] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [96] A. Bergamo and L. Torresani, “Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach,” in *Advances in Neural Information Processing Systems*, 2010, pp. 181–189.
- [97] W. Li *et al.*, “Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1134–1148, Jun 2014.
- [98] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [99] M. Long *et al.*, “Transfer learning with graph co-regularization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1805–1818, Jul 2014.
- [100] J. Huang *et al.*, “Correcting sample selection bias by unlabeled data,” in *Advances in Neural Information Processing Systems*, 2006, pp. 601–608.
- [101] R. Aljundi *et al.*, “Landmarks-based kernelized subspace alignment for unsupervised domain adaptation,” in *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [102] J. T. Zhou *et al.*, “Transfer learning for cross-language text categorization through active correspondences construction,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [103] N. Gould and P. L. Toint, “Preprocessing for quadratic programming,” *Mathematical Programming*, vol. 100, no. 1, pp. 95–132, 2004.

- [104] X. Jiang, "Asymmetric principal component and discriminant analyses for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 931–937, May 2009.
- [105] X. Jiang, B. Mandal, and A. C. Kot, "Eigenfeature regularization and extraction in face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 383–394, Jun 2008.
- [106] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, Feb 2003.
- [107] B. Fernando *et al.*, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013, pp. 2960–2967.
- [108] M. Rjam, "An introduction to mathematical statistics and its applications," 2000.
- [109] H. Cao and A. C. Kot, "Identification of recaptured photographs on lcd screens," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 1790–1793.
- [110] O. Bimber and D. Iwai, "Superimposing dynamic range," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 150.
- [111] S. Winkler, "Perceptual distortion metric for digital color video," in *Electronic Imaging*. International Society for Optics and Photonics, 1999, pp. 175–184.
- [112] K. Bahrami and A. C. Kot, "A fast approach for no-reference image sharpness assessment based on maximum local variation," *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 751–755, Jun 2014.

- [113] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul 2002.
- [114] R. Nosaka, Y. Ohkawa, and K. Fukui, “Feature extraction based on co-occurrence of adjacent local binary patterns,” in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2011, pp. 82–91.
- [115] V. Ojansivu and J. Heikkilä, “Blur insensitive texture classification using local phase quantization,” in *International conference on Image and Signal Processing*. Springer, 2008, pp. 236–243.
- [116] J. Kannala and E. Rahtu, “Bsif: Binarized statistical image features,” in *IEEE International Conference on Pattern Recognition (ICPR)*, 2012, pp. 1363–1366.
- [117] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [118] J. Yosinski *et al.*, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.
- [119] S. Ben-David *et al.*, “A theory of learning from different domains,” *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.
- [120] S. Ji *et al.*, “3d convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.
- [121] A. Pinto *et al.*, “Face spoofing detection through visual codebooks of spectral temporal cubes,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4726–4740, 2015.

- [122] D. Tran *et al.*, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.
- [123] H. Li *et al.*, “Domain generalization with adversarial feature learning,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [124] Y. LeCun *et al.*, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems*, 1990, pp. 396–404.
- [125] H. Li *et al.*, “Unsupervised domain adaptation for face anti-spoofing,” *IEEE Transactions on Information Forensics and Security* (in press).
- [126] K. He *et al.*, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
- [127] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [128] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [129] S. Bharadwaj *et al.*, “Computationally efficient face spoofing detection with motion magnification,” in *The IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 105–110.
- [130] S. Tirunagari *et al.*, “Detection of face spoofing using visual dynamics,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 762–777, Apr 2015.

- [131] J. Gan *et al.*, “3d convolutional neural network based on face anti-spoofing,” in *Multimedia and Image Processing (ICMIP), 2017 2nd International Conference on*. IEEE, 2017, pp. 1–5.
- [132] K. Muandet, D. Balduzzi, and B. Schölkopf, “Domain generalization via invariant feature representation,” in *International Conference on Machine Learning (ICML)*, 2013, pp. 10–18.
- [133] M. Rohrbach, M. Stark, and B. Schiele, “Evaluating knowledge transfer and zero-shot learning in a large-scale setting,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1641–1648.
- [134] R. Socher *et al.*, “Zero-shot learning through cross-modal transfer,” in *Advances in neural information processing systems*, 2013, pp. 935–943.
- [135] E. Kodirov *et al.*, “Unsupervised domain adaptation for zero-shot learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2452–2460.
- [136] Y. Fu *et al.*, “Transductive multi-view zero-shot learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.
- [137] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [138] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.